Applications of Machine Learning and Artificial Intelligence in Granular Mechanics

Project Completion Report

Sanskaar Srivastava

Sidhant Thalor

Sidharth Budania

C Zoliansangi

Rohan Nimesh

Harsh Nirmal

Department of Chemical Engineering

Indian Institute of Technology Kanpur

Contents

1 Introduction

Granular materials, comprising discrete solid particles that interact through contact forces, represent one of the most common forms of matter encountered in both industrial processes and natural systems. These materials span a vast range of scales and compositions—from geological formations and soils to food grains, pharmaceutical powders, and construction aggregates. With over 75% of raw material feedstock in industry consisting of granular matter, their economic impact is profound. Yet, despite their apparent simplicity, granular materials exhibit complex behaviors such as jamming, force chains, strain localization, pattern formation, and segregation, making them uniquely challenging to model. These emergent phenomena arise from the collective dynamics of particles obeying Newtonian mechanics but interacting nonlinearly through frictional and collisional contacts, leading to macroscopic behaviors that can mimic solids, liquids, or gases depending on external stimuli and packing density.

Traditional modeling techniques for granular materials primarily fall into three categories: continuum mechanics, kinetic theory, and discrete element methods (DEM). Continuum-based methods—including the finite element method (FEM), finite difference method (FDM), smoothed particle hydrodynamics (SPH), and material point method (MPM)—approximate granular media as continuous fields with effective constitutive relations. These methods, while computationally efficient and suitable for many macroscopic simulations, often fail to capture discrete particle interactions and transient mesoscale phenomena. On the other hand, DEM, as introduced by Cundall and Strack, explicitly resolves inter-particle contact forces and has become the benchmark for high-fidelity modeling of granular matter at the particle scale. However, the approach is computationally intensive, especially for large or dense systems involving nonspherical particles, where contact detection becomes a nontrivial optimization problem.

To address the shortcomings of classical approaches, machine learning (ML) has recently gained traction as a powerful alternative for modeling granular systems. ML techniques excel at identifying complex, nonlinear relationships from data without relying on explicit physical assumptions. With advancements in computational hardware and access to large-scale experimental and simulation datasets, ML models have demonstrated promising capabilities in capturing both particle-level interactions and bulk material responses. They offer the potential to serve as fast, accurate surrogate models, enabling real-time simulations and supporting inverse modeling tasks such as material property inference and process optimization.

Notably, Ladický et al. pioneered the application of supervised learning to particle-based fluid simulations, predicting accelerations with regression forests at interactive speeds. This foundation has since expanded to include diverse ML architectures such as Convolutional Neural Networks (CNNs), U-Nets, Graph Neural Networks (GNNs), and Physics-Informed Neural Networks (PINNs). These models have been employed for a variety of tasks, including constitutive law discovery, flow defect prediction, packing density optimization, and surrogate modeling for granular simulations. Furthermore, ML has been utilized to uncover latent structures in disordered systems and support transfer learning across material types, particle shapes, and loading conditions—challenges that are traditionally difficult to address with physics-based models.

Despite these advancements, integrating ML into granular material modeling presents its own challenges. Training accurate and generalizable models demands large, high-quality datasets and careful feature engineering or representation learning. Additionally, ensuring physical consistency, interpretability, and robustness of ML predictions remains an ongoing

research effort. Nevertheless, the synergy between data-driven methods and classical physics-based approaches offers a compelling direction for the future of granular material research.

In this review, we synthesize recent progress in the application of machine learning to granular material modeling. We examine developments across various modeling scales—from particle-level simulations to continuum approximations—and categorize the methodologies based on their roles in constitutive modeling, surrogate simulations, and real-time predictions. By highlighting key advances, limitations, and open questions, this work aims to provide a comprehensive foundation for future research at the intersection of granular physics and data-driven modeling.

2 Literature Review

2.1 Historical Approaches to Granular Materials

The scientific exploration of granular materials has a rich history, beginning with the foundational work of Charles-Augustin de Coulomb in 1776, who formulated early theories of soil shear strength, and Osborne Reynolds in 1885, whose discovery of dilatancy fundamentally altered our understanding of granular flow. These pioneering efforts laid the groundwork for classical soil mechanics and influenced continuum-based approaches, which model granular matter as continuous media governed by constitutive laws. These methods, often relying on theories such as Mohr-Coulomb and critical state frameworks, enabled the development of practical geotechnical models. However, continuum approaches struggled to account for the inherently discrete and heterogeneous nature of granular assemblies. They lacked the resolution to capture localized phenomena such as shear banding, force chains, and jamming, which arise from particle-scale interactions and spatial inhomogeneities.

A transformative milestone came with the introduction of the *Discrete Element Method (DEM)* by Cundall and Strack in 1979. This method represented a paradigm shift by discretizing the system into individual particles and explicitly modeling their interactions through contact mechanics and Newtonian dynamics. DEM enabled the simulation of complex behaviors such as inter-particle collisions, frictional sliding, and rolling resistance, providing unprecedented insights into particle kinematics and force propagation in granular materials. Over the decades, DEM has become the gold standard for microscale modeling of granular flows, agglomeration, segregation, and compaction. However, despite its accuracy and physical fidelity, DEM is computationally intensive, especially for large-scale or long-duration simulations, limiting its practical use in industrial or engineering-scale applications.

2.2 Early Integration of Machine Learning in Granular Materials

The integration of machine learning (ML) into materials science began in earnest in the early 2000s, predominantly focused on crystalline solids and molecular systems. Researchers used data-driven models to discover correlations between microstructural features and material properties such as conductivity, hardness, and failure strength. As computational resources and data availability improved, ML applications expanded toward amorphous systems and disordered matter, including granular materials.

The earliest applications of ML in the granular domain largely centered around *supervised learning* to predict soil properties from basic geotechnical parameters. For instance, artificial neural networks (ANNs) were employed to estimate the California Bearing Ratio (CBR) based on parameters such as optimum moisture content (OMC), maximum dry density (MDD), and Atterberg limits. These efforts demonstrated ML's ability to uncover complex nonlinear relationships among engineering variables without relying on explicit physical models. Although

limited in scope and often dependent on small datasets, these initial studies served as proof-of-concept that ML could provide valuable predictive tools in granular mechanics.

2.3 Recent Advances in Machine Learning for Granular Materials

In the last decade, particularly in the past five years, machine learning applications in granular materials have expanded significantly in both scope and sophistication. Wang and Feng (2024) categorized the major research directions into three principal domains: microscopic particle interactions, constitutive modeling of material behavior, and macroscopic engineering-scale simulations.

2.3.1 Microscopic Particle Interactions

At the microscale, ML has been utilized to augment and accelerate DEM simulations by learning surrogate models of contact mechanics. Traditional DEM contact laws, often empirical and computationally expensive, can be replaced or enhanced by data-driven models that learn from high-fidelity simulations or experimental data. These ML models can capture intricate dependencies such as rate effects, nonlocality, and history dependence in particle collisions, leading to more accurate and computationally efficient simulations.

Furthermore, ML has been applied to identify latent structural features within disordered granular packings. For example, researchers have used neural networks to detect "flow defects"—regions within jammed or glassy systems that are prone to rearrangement under shear. By learning from particle-level descriptors such as local packing density, coordination number, and free volume, ML models have shown the ability to forecast the onset of localized instabilities. In granular explosives, similar techniques have been employed to predict particles susceptible to shock heating, thereby aiding the design of safer materials.

2.3.2 Constitutive Behavior and Material Property Prediction

Another major thrust of research involves using ML to model constitutive relationships that describe stress-strain behavior in granular systems. Unlike classical models, which often assume idealized behavior and require empirical calibration, ML models can learn from raw simulation or experimental data to capture complex phenomena such as anisotropy, non-coaxiality, and path-dependent plasticity. Deep learning frameworks—including fully connected neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs)—have been successfully trained to represent multiaxial loading responses and evolving internal states of granular media.

In geotechnical engineering, ML algorithms have been deployed to predict foundational parameters like bearing capacity, slope stability, and settlement of shallow foundations. Techniques such as Random Forests (RF), Support Vector Machines (SVM), and CNNs have consistently outperformed traditional empirical formulas in terms of accuracy and generalizability. Studies have also demonstrated the use of dimensionality reduction (e.g., PCA, t-SNE) and feature selection techniques to enhance model interpretability and performance.

Moreover, optimization of granular packings using ML has gained traction. Researchers have applied generative models and regression-based approaches to discover novel particle shapes and configurations that maximize packing density or minimize void ratio. These inverse design frameworks exemplify how ML can go beyond prediction to actively guide material design.

2.3.3 Macroscopic Simulations and Engineering-Scale Applications

At the engineering scale, ML has been increasingly used to enhance simulation efficiency and enable real-time decision-making. One notable development is the coupling of ML with DEM

to *accelerate time integration*, allowing for larger time steps without sacrificing accuracy. This is achieved by learning surrogate force models or timestep-stability criteria from a corpus of smaller simulations.

In multi-scale modeling frameworks, ML plays a pivotal role in bridging micro-to-macro scales. For instance, neural networks trained on DEM data have been embedded into continuum solvers (e.g., FEM or MPM) as constitutive laws, thereby enabling hybrid simulations that combine accuracy with scalability. These surrogate models drastically reduce the computational cost of full-resolution DEM while preserving essential microscale features.

Additionally, ML classifiers have been applied to kinematic data from granular flows to sort and categorize materials based on properties such as particle stiffness, size distribution, or restitution coefficient. Remarkably, even simple neural networks have demonstrated the ability to infer hidden material properties purely from observable motion trajectories, revealing the latent structure-to-behavior mapping in granular systems.

3 Numerical and Theoretical Methods

3.1 Conventional Neural Networks in Machine Learning-Driven Modeling of Granular Materials

A deep understanding of various neural networks is essential for their effective application to specific tasks. This section introduces seven commonly used machine learning models in simulating granular materials, including the multi-layer perceptron (MLP), basic recurrent neural network (RNN) and its variants, such as long-short-term memory (LSTM) and gated recurrent unit (GRU), temporal convolutional neural networks (TCNN), convolutional neural networks (CNN), and graph neural networks (GNN). Based on their distinct characteristics, these models are categorized into three groups: 1) single-step-based networks, 2) multi-step-based or time-sequence networks, and 3) spatial information-based neural networks.

3.1.1 Single-Time Step Neural Networks: Multi-Layer Perceptron (MLP)

The Multi-Layer Perceptron (MLP) serves as a foundational architecture in single-time step neural networks, establishing direct mappings between instantaneous inputs and outputs. As illustrated in Figure 1, this network comprises three primary components: an input layer receiving feature vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, hidden layers containing trainable parameters, and an output layer generating predictions $\mathbf{O} = (\hat{y}_1, \ldots, \hat{y}_m)$. The architectural diagram demonstrates weight matrices $\mathbf{W}_1, \mathbf{W}_2$ and bias vectors $\mathbf{b}_1, \mathbf{b}_2$ connecting these layers.

The feedforward propagation mechanism operates through successive transformations. The hidden layer output $\mathbf{H}$ is computed using the activation function $f(\cdot)$ applied to the linear combination of inputs and parameters:

$$\mathbf{H} = f\left(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1\right) \tag{1}$$

Subsequently, the final output $\mathbf{O}$ emerges through another activation $g(\cdot)$:

$$\mathbf{O} = g\left(\mathbf{W}_2\mathbf{H} + \mathbf{b}_2\right) \tag{2}$$

Here, $\mathbf{W}_1 \in \mathbb{R}^{d_h \times d_x}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_o \times d_h}$ represent weight matrices transforming input dimensions d_x to hidden layer size d_h, then to output dimension d_o. The bias vectors $\mathbf{b}_1 \in \mathbb{R}^{d_h}$ and $\mathbf{b}_2 \in \mathbb{R}^{d_o}$ enable threshold adjustments, while activation functions (f for hidden layers, g for output) typically employ ReLU or sigmoid operations.

The learning process involves three iterative phases: forward propagation of input signals, error computation via loss function $\mathcal{L}(\mathbf{O}, \mathbf{y}_{\text{true}})$, and backpropagation through gradient

descent. Parameter updates follow:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L} \tag{3}$$

where $\theta = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2\}$ encompasses all trainable parameters and η denotes the learning rate controlling update magnitudes.

Despite its universal approximation capabilities and computational efficiency for static problems, the MLP exhibits notable limitations. The architecture inherently lacks temporal memory, requiring explicit history variables $\varphi^{(t)}$ for time-dependent scenarios:

$$\hat{\boldsymbol{\sigma}}^{(t)} = f^{\text{MLP}}\left(\boldsymbol{\varepsilon}^{(t)}, \varphi^{(t)}, \mathbf{W}, \mathbf{b}\right) \tag{4}$$

Additionally, deep MLP configurations risk gradient vanishing/explosion during backpropagation through numerous layers. These constraints motivate the development of specialized architectures for sequential data processing.

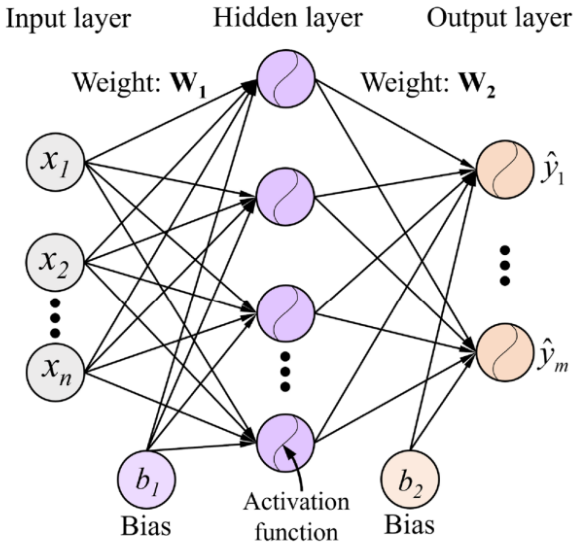

Figure 1: MLP architecture with one hidden layer, illustrating weight matrices ($\mathbf{W}_1, \mathbf{W}_2$) and bias vectors ($\mathbf{b}_1, \mathbf{b}_2$). Reproduced from original research.

3.1.2 Time-Sequence Neural Networks

Time-sequence neural networks excel at modeling history-dependent phenomena through temporal connections and memory mechanisms. These architectures process sequential data by maintaining internal states that encode historical information, making them particularly effective for granular material simulations involving temporal evolution.

A. Recurrent Neural Networks (RNN)

The basic Recurrent Neural Network (RNN) architecture replaces standard hidden layer nodes with recurrent cells featuring temporal connections. As shown in Figure 2, each RNN cell processes sequential inputs $\mathbf{x}^{(t)}$ while maintaining hidden states $\mathbf{H}^{(t)}$ that capture historical context through recursive transformations:

$$\mathbf{H}^{(t)} = \tanh\left(\mathbf{W}^{(t)}\mathbf{x}^{(t)} + \mathbf{U}^{(t)}\mathbf{H}^{(t-1)} + \mathbf{b}\right) \tag{5}$$

$$\mathbf{O}^{(t)} = g\left(\mathbf{W}_2\mathbf{H}^{(t)} + \mathbf{b}_2\right) \tag{6}$$

where $\mathbf{W}^{(t)} \in \mathbb{R}^{d_h \times d_x}$ and $\mathbf{U}^{(t)} \in \mathbb{R}^{d_h \times d_h}$ are time-dependent weight matrices, $\mathbf{b}$ represents bias vectors, and $g(\cdot)$ denotes the output activation function. The hidden state $\mathbf{H}^{(t)}$ serves as a compressed representation of historical information up to time t, with d_h indicating hidden state dimension and d_x the input feature size. While effective for short sequences, basic RNNs suffer from gradient vanishing/explosion issues in long-term dependencies due to repeated matrix multiplications during backpropagation through time.

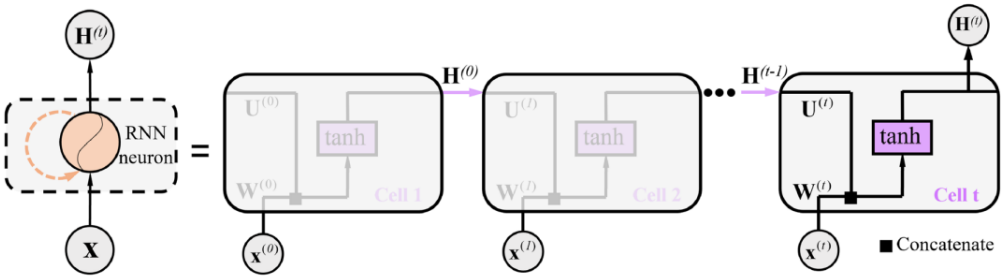

Figure 2: The recurrent neuron in the basic RNN

A.1. Long Short-Term Memory (LSTM)

The LSTM architecture introduces three gating mechanisms to regulate information flow. As shown in Figure 3, the LSTM cell contains:

$$\begin{aligned} \mathbf{F}_t &= \sigma\left(\mathbf{W}_F \mathbf{x}^{(t)} + \mathbf{U}_F \mathbf{H}^{(t-1)} + \mathbf{b}_F\right) \\ \mathbf{I}_t &= \sigma\left(\mathbf{W}_I \mathbf{x}^{(t)} + \mathbf{U}_I \mathbf{H}^{(t-1)} + \mathbf{b}_I\right) \\ \mathbf{O}_t &= \sigma\left(\mathbf{W}_O \mathbf{x}^{(t)} + \mathbf{U}_O \mathbf{H}^{(t-1)} + \mathbf{b}_O\right) \end{aligned} \tag{7}$$

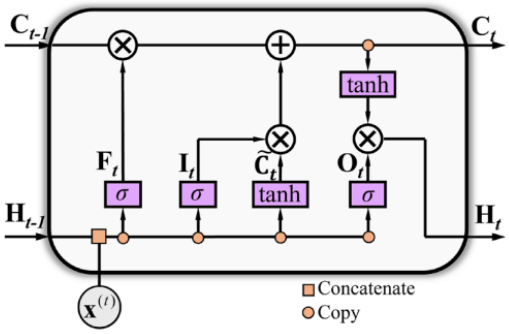

Figure 3: LSTM memory cell

where σ denotes the sigmoid activation function. The cell state update equations are:

$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tanh\left(\mathbf{W}_C \mathbf{x}^{(t)} + \mathbf{U}_C \mathbf{H}^{(t-1)} + \mathbf{b}_C\right) \tag{8}$$

$$\mathbf{H}^{(t)} = \mathbf{O}_t \odot \tanh(\mathbf{C}_t) \tag{9}$$

The forget gate $\mathbf{F}_t$ controls historical information retention, input gate $\mathbf{I}_t$ governs new information integration, and output gate $\mathbf{O}_t$ regulates hidden state exposure. The element-wise product operator $\odot$ enables selective information passage, with $\mathbf{C}_t \in \mathbb{R}^{d_h}$ representing the cell

state vector.

A.2. Gated Recurrent Unit (GRU)
The GRU simplifies LSTM architecture while maintaining comparable performance through two gating mechanisms:

$$\begin{aligned}\mathbf{Z}_t &= \sigma\left(\mathbf{W}_Z\mathbf{x}^{(t)} + \mathbf{U}_Z\mathbf{H}^{(t-1)} + \mathbf{b}_Z\right)\\ \mathbf{R}_t &= \sigma\left(\mathbf{W}_R\mathbf{x}^{(t)} + \mathbf{U}_R\mathbf{H}^{(t-1)} + \mathbf{b}_R\right)\end{aligned} \tag{10}$$

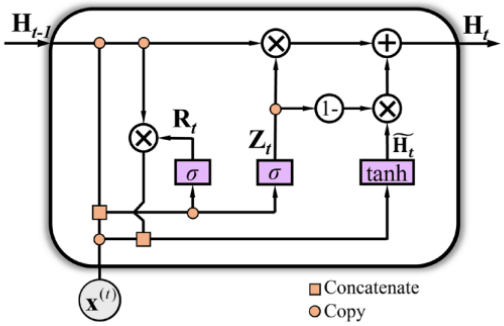

Figure 4: GRU cell architecture

The update mechanism combines historical and current information:

$$\mathbf{H}^{(t)} = \mathbf{Z}_t \odot \mathbf{H}^{(t-1)} + (1 - \mathbf{Z}_t) \odot \tanh\left(\mathbf{W}\mathbf{x}^{(t)} + \mathbf{U}(\mathbf{R}_t \odot \mathbf{H}^{(t-1)})\right) \tag{11}$$

where $\mathbf{Z}_t \in [0,1]^{d_h}$ is the update gate vector and $\mathbf{R}_t \in [0,1]^{d_h}$ the reset gate vector. This architecture reduces parameters by 25% compared to LSTM while maintaining similar temporal modeling capabilities.

B. Temporal Convolutional Neural Networks (TCNN)
Temporal Convolutional Neural Networks (TCNNs) provide an alternative architecture for sequence modeling that combines temporal awareness with computational efficiency. Unlike recurrent networks that use memory cells, TCNNs extract historical information through causal convolutions applied along the temporal dimension. As shown in Figure 5, the input data $\mathbf{X} \in \mathbb{R}^{T\times d_x}$ is structured as a 2D array where one dimension represents time steps and the other denotes feature depth (d_x).

The architecture employs causal convolution (CC) with zero-padding at the sequence forefront to maintain temporal alignment. For a kernel size K, the input is padded with $K-1$ zero lines, ensuring outputs $\mathbf{O}^{(t)}$ only depend on current and previous time steps:

$$\mathbf{O}^{(t)} = \sum_{k=0}^{K-1} \mathbf{W}_k \mathbf{X}^{(t-k)} \tag{12}$$

where $\mathbf{W}_k \in \mathbb{R}^{n_f \times d_x}$ are filter weights and n_f is the number of filters. Multiple filters operate in parallel during the CC procedure, each performing dot products along the temporal dimension with fixed stride s. The resulting output vectors are flattened and processed through subsequent layers.

A key innovation in TCNNs is the use of dilated convolutions to exponentially expand receptive fields without increasing parameters. For L layers with dilation factor $d = 2^{l-1}$ at layer l, the effective receptive field becomes:

$$\text{Receptive field size} = (K-1) \times d^{L-1} + 1 \tag{13}$$

This enables modeling of long-term dependencies while maintaining temporal causality. The architecture's filter independence allows full parallelization across GPU cores, significantly accelerating training compared to sequential RNN processing.

Advantages:

- Parameter count remains $\mathcal{O}(K \cdot n_f)$ regardless of sequence length
- Enables seamless transfer learning between models with identical filter configurations
- Avoids gradient vanishing/explosion through fixed convolutional paths

Limitations:

- Fixed context windows require architectural pre-definition of historical scope
- Stacked CC layers increase memory demands for parameter storage
- Lacks dynamic forgetting mechanisms for redundant historical information

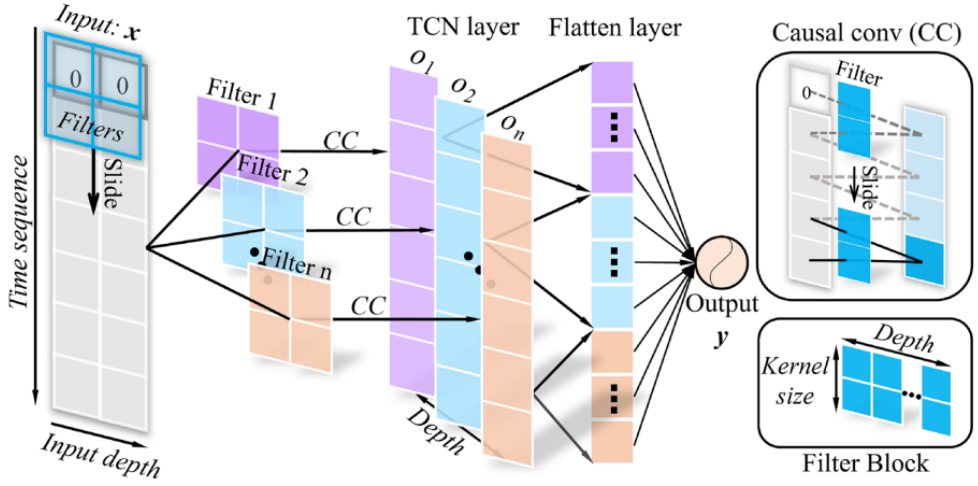

Figure 5: The feedforward process in the temporal convolution neural network

While TCNNs demonstrate superior parallelization capabilities and stable gradients, their fixed temporal context and memory-intensive deep architectures present challenges for applications requiring adaptive history lengths or resource-constrained environments. The balance between receptive field size and computational efficiency remains a key consideration in TCNN design.

3.1.3 Geometry Information-Based Neural Networks

The prediction of granular mechanical responses has traditionally been treated as time-sequential problems using vector-based inputs. However, recent advances in machine learning enable direct extraction of spatial features from geometric representations of particle assemblies. This subsection examines two prominent architectures for processing geometric information: Convolutional Neural Networks (CNNs) and Graph Neural Networks (GNNs).

A. Convolutional Neural Networks (CNN)

CNNs revolutionized image processing through their ability to extract hierarchical spatial patterns via learnable filters. For granular materials, CNNs enable property prediction by encoding particle assembly structures into pixel matrices. The feedforward process involves three-dimensional filters scanning input data across height, width, and channel dimensions:

$$\mathbf{A}_{out}(i,j,k) = \sum_{m=0}^{C_{in}-1} \sum_{p=0}^{K_h-1} \sum_{q=0}^{K_w-1} \mathbf{W}(p,q,m,k) \cdot \mathbf{A}_{in}(i+p, j+q, m) + \mathbf{b}(k) \tag{14}$$

where $\mathbf{A}_{in} \in \mathbb{R}^{H \times W \times C_{in}}$ is the input tensor, $\mathbf{W} \in \mathbb{R}^{K_h \times K_w \times C_{in} \times C_{out}}$ the filter bank, and $\mathbf{b} \in \mathbb{R}^{C_{out}}$ the bias vector. Zero-padding maintains spatial dimensions during convolution:

$$\mathbf{A}_{padded} = \text{ZeroPad}(\mathbf{A}_{in}, \lfloor K_h/2 \rfloor, \lfloor K_w/2 \rfloor) \tag{15}$$

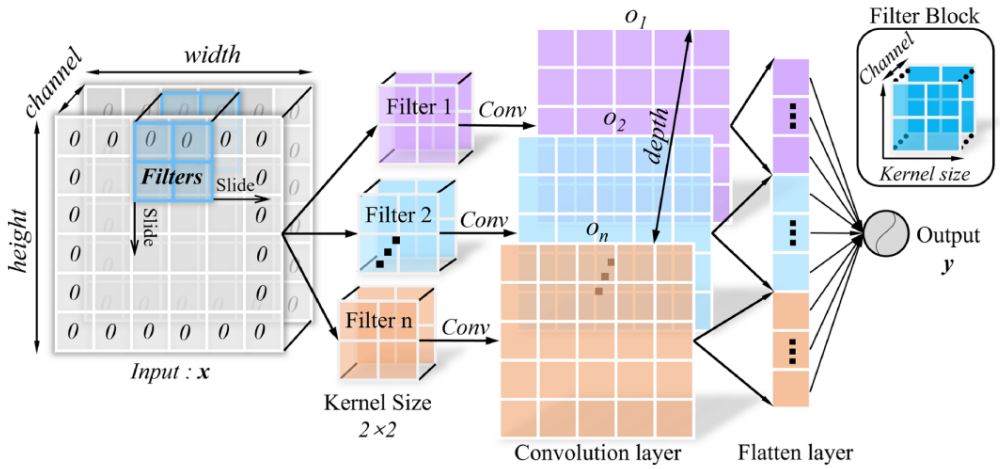

Figure 6: The feedforward process of the convolutional neural network

Key advantages of CNN include:

- Translation invariance through weight sharing
- Hierarchical feature learning via stacked convolution layers
- Efficient parameterization compared to fully-connected networks

However, CNNs exhibit limitations when applied to granular systems:

- Requirement for regular grid input structure
- Sensitivity to occlusions and noise in particle images
- High computational cost for 3D volumetric data
- Need for large labeled datasets of particle configurations

Graph Neural Networks (GNNs)

GNNs operate directly on graph-structured data, making them ideal for modeling irregular particle assemblies. Given a graph $G = (V, E)$ with nodes $v_i \in V$ representing particles and edges $e_{ij} \in E$ encoding interactions, the message passing framework updates node features through neighborhood aggregation:

$$\mathbf{h}_i^{(k)} = \phi\left(\mathbf{h}_i^{(k-1)}, \bigoplus_{j\in\mathcal{N}(i)} \psi(\mathbf{h}_i^{(k-1)}, \mathbf{h}_j^{(k-1)}, \mathbf{e}_{ij})\right) \tag{16}$$

where ϕ and ψ are learnable functions, $\bigoplus$ denotes permutation-invariant aggregation (mean/sum/max), and $\mathbf{e}_{ij}$ contains edge attributes like particle separation vectors.

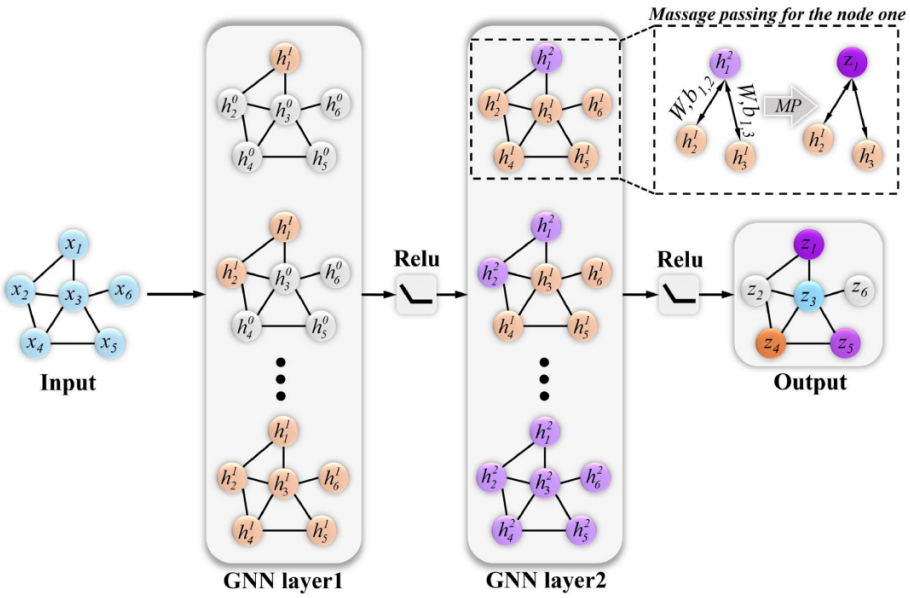

Figure 7: Architecture of Graph Neural Networks

The edge update mechanism for particle systems can be formulated as:

$$\mathbf{m}_{ij}^{(k)} = \text{MLP}\left(\mathbf{h}_i^{(k-1)} \parallel \mathbf{h}_j^{(k-1)} \parallel \|\mathbf{r}_{ij}\| \parallel \mathbf{f}_{ij}\right) \tag{17}$$

where $\mathbf{r}_{ij}$ is relative position and $\mathbf{f}_{ij}$ contains force/displacement history.
GNNs offer unique advantages for granular modeling:

- Native handling of variable particle counts and neighborhoods
- Explicit encoding of interaction topology through edges
- Permutation invariance to particle ordering
- Ability to incorporate physical constraints via edge features

Current challenges of Graph Neural Networks include:

- Quadratic memory scaling with particle count
- Difficulty handling dynamic graph topologies
- Limited theoretical understanding of stability
- Sensitivity to edge feature noise and sparsity

Neural Networks	Advantages	Disadvantages
MLP	Simple architecture; High training efficiency	Requires artificially added history variables in time-sequence problems; Gradient vanishing or explosion; Sensitive to noise and irregularities
RNN	No extra history variables are needed in sequential prediction; Strong non-linear mapping capability	Complex network architecture; Many training parameters; Gradient vanishing or explosion; Weak ability to record long-history information; Weak parallelism
LSTM	No extra history variables are needed in sequential prediction; Strong non-linear mapping capability; Eliminates gradient vanishing or explosion with gate structure; Strong ability to capture long-term dependencies	Increased structural complexity compared to RNN; More training parameters than RNN; Weak parallelism
GRU	Similar to LSTM but with simpler gate structures	Similar limitations to LSTM
TCNN	No extra history variables are needed in sequential prediction; Strong non-linear mapping capability; Excellent parallelism; Good portability of trained parameters; Suitable for longer history information	Numerous training parameters; Weak ability to filter out redundant history information
CNN	Strong high-dimensional mapping capability; Excellent parallelism; Good portability of trained parameters; Good visualization; Location invariance of input; Strong ability to extract rich spatial features	Numerous training parameters; Not suitable for non-grid input data; Weaker resistance to noise; Requires large amounts of labeled data
GNN	Suitable for non-grid input data; Strong ability to extract rich spatial features and reflect relationships between adjacent objects	Numerous training parameters; Vulnerability to graph structure modifications (e.g., adding/removing nodes/edges); Less resistance to noise

Table 1: Features of seven typically used neural networks in ML-aided granular materials simulation. This table summarizes their advantages and disadvantages.

3.1.4 Network Comparison and Selection Guidelines

Advantages and Limitations of all Typical Neural Networks is mentioned in Table 1. Selection criteria should consider: 1) Input data structure (grid, sequence, graph), 2) Required temporal context length, 3) Computational resource constraints, and 4) Need for interpretability. Hybrid architectures combining temporal and spatial processing often yield optimal performance for multiscale granular phenomena, with TCNNs preferred for fixed-context parallel processing and GNNs for irregular particle interactions.

3.2 Machine Learning Models Based on Microscopic Grain Information

The Discrete Element Method (DEM) has become widely used in the study of granular mechanics for simulating the mechanical behavior of granular materials and addressing various engineering problems such as landslides, shear deformation of soil and sand, tunnel surface instability, and fluidized beds. Although DEM can effectively represent the discrete characteristics of granular materials at the grain scale, it typically involves high computational costs. The rise of deep learning techniques offers an opportunity to reduce this computational burden by combining machine learning (ML) models with conventional particle-based (microscopic) methods. This section presents a summary of notable studies in this domain, beginning with a brief overview of the basic DEM framework.

3.2.1 Overview of the Discrete Element Method

Unlike continuum-based approaches, DEM models granular materials as collections of distinct particles. The macroscopic behavior of such systems arises from the interactions between particles over time, making DEM especially suitable for analyzing problems involving significant deformations.

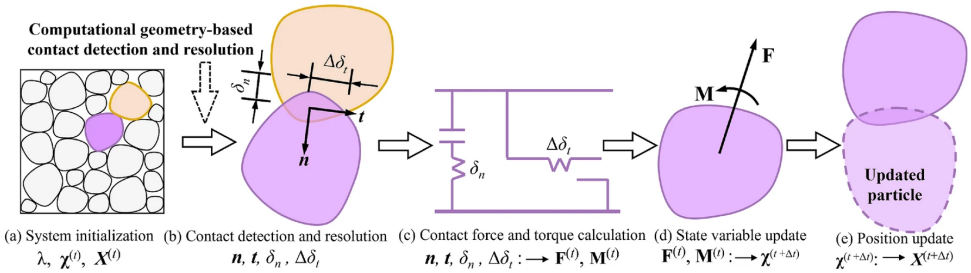

Figure 8: Overview of machine learning integration with DEM in granular mechanics.

Figure 8 outlines the core computational steps involved in the Discrete Element Method (DEM), divided into the following phases:

(a) **System Initialization:** This phase involves two main tasks: assigning physical properties (such as mass and friction coefficient) to the particles, and setting up geometric characteristics like shape, size, and position. Initial values for state variables velocity, angular velocity, acceleration, and angular acceleration are also provided to each particle.

(b) **Contact Detection and Resolution:** At this stage, particle pairs that come into contact are identified using collision detection techniques. Contact-related features such as normal and tangential vectors, and inter-particle overlaps, are then computed for each pair using appropriate resolution algorithms.

(c) **Force and Torque Computation:** Using the previously calculated contact features and a contact model, the resultant contact force and torque acting on each particle are de-

termined. These values are essential for computing both translational and rotational motion.

(d) **State Variable Update:** The acceleration and angular acceleration of each particle are updated based on Newton's second law, incorporating the newly computed forces and torques. This leads to the refreshed values of the velocity and angular velocity.

(e) **Position Update:** Finally, the particle positions are updated using the current state variables. After this, the system advances to the next iteration.

3.2.2 The ML-Aided Discrete Element Modeling

In conventional DEM simulations, the stages involving contact detection and resolution are typically the most computationally demanding. To address this, machine learning (ML) techniques offer a promising solution by significantly improving computational efficiency. Integrating ML-based models into the DEM workflow, especially for contact detection and resolution, has the potential to greatly speed up the overall calculation process.

In ML-integrated DEM systems, contact detection and resolution are handled using classification and regression neural networks. The primary difference between these networks lies in their output layers. The classification network outputs a binary value (0 or 1), indicating whether two particles are in contact—one treated as the object grain and the other as the cue particle. Meanwhile, the regression network predicts specific contact features such as contact points, normal vectors, and overlap magnitudes between particles, based on their geometric attributes.

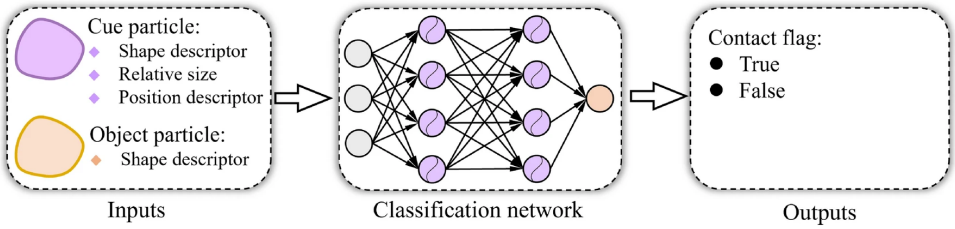

Figure 9: Classification neural network for contact detection.

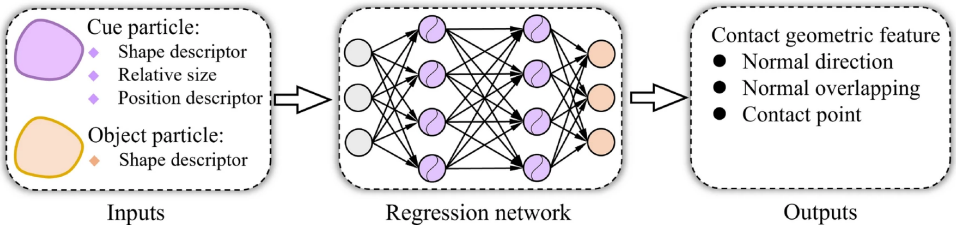

Figure 10: Regression neural network for contact resolution.

Lai et al. investigated the extraction of contact status and contact features of particles with elliptical and irregular shapes using a classification and a regression neural network. These two networks shared identical input parameters, which included grain shape characteristics and cue particle parameters like size, shape, and position. The resulting networks were then incorporated into the DEM framework to simulate various scenarios such as random packing, oedometer compression, angle of repose, and compaction of irregular-shaped particles. Hwang et al. implemented one classification and one regression neural network to estimate the contact condition (either detached or intersected) and geometric contact features between two identical irregular particles. These features included the average contact point, average

normal vector of intersecting vertices, and penetration depth. In their training process, labeled contact data were generated using the deepest point method, which calculates the relative orientation and position of the two particles. These labeled inputs were then used to train the machine learning models.

3.2.3 The ML–Based Grain–Level Kinematic Features Simulations

The mechanical behavior of granular materials is significantly affected by the evolution of their microstructure, which is influenced by particle-level kinematic features. Many researchers have thus focused on developing ML models that can predict particle motion laws without relying on empirical or analytical contact force models.

Using ML techniques, studies have investigated both the collision behavior of sparse particles and the deformation behavior of dense granular media. For example, in the work of Katerina et al., a convolutional neural network (CNN) was used to forecast the velocity of a target object by inputting the system's current and prior image data. Another approach by Wu et al. employed a novel object-centric prediction strategy to improve the translational invariance of the predicted physical laws. Their method integrated physical insights into the ML model to predict object dynamics at future time steps, using data such as shape, position, and friction coefficient from prior steps.

Battaglia et al. incorporated physical states from the previous step into a graph-based interaction network capable of reasoning about object collisions in complex environments. Additionally, Chang et al. proposed a Neural Physics Engine (NPE), which used the past velocities and positions of a target object and its surroundings as input to forecast its motion in the next time step.

On the other hand, machine learning models trained using data generated from DEM simulations have been used to accelerate the computation process by substituting DEM with ML-based approximations. For instance, Ummenhofer et al. and Liu et al. developed a CNN that predicts inter-particle collision laws using continuous filters, allowing them to replace complex particle-boundary interaction calculations with learned representations.

To accelerate the simulation of grain flows, Li et al. designed a graph neural network (GNN) that learns the microstructural state of grains obtained from DEM simulations. This information is then used to predict the force chain structure in uniaxial compression scenarios. Similarly, Cheng et al. utilized contact and particle kinematics from a DEM simulator to train a GNN capable of estimating contact forces in granular assemblies under compression.

Bapst et al. proposed a GNN that only uses the initial particle positions to represent the long-term system evolution in shear flow. In another work, Mayr et al. introduced a Boundary-Graph Neural Network (BGNN) to handle particle interactions under complex boundary conditions. Their method inserts virtual nodes to represent boundary regions and triangle elements to model boundary orientation, allowing successful simulations of granular flow in rotating drums, hoppers, and mixers.

Kumar et al. argued that the messages passed within the GNN encode latent information about the interaction laws in DEM simulations. They introduced a sparse message representation where messages are learned as a linear combination of force vectors. To ensure minimal message complexity, the model uses L1 regularization, guiding the GNN to encode forces in a compact vector space. As a result, they recovered the law governing the fundamental linear spring interaction between DEM particles:

$$F_n = k_n \cdot \text{abs}(\Delta\omega - r_i - r_j)$$

These kinematics-based models focus on using past particle position and velocity data to update and accelerate the current particle state in simulations. This approach avoids explicitly computing contact forces or tracking boundaries, significantly improving computational performance. However, current research suggests that most models are trained with DEM data

using particles of regular shapes—such as circular or spherical ones—thereby overlooking the role of particle shape irregularities in grain trajectory evolution.

3.2.4 Summary

This section presents a concise overview of the developments in grain information-based machine learning (ML) models, highlighting two major directions. The first focuses on constructing ML-based contact models that incorporate contact states and geometric features into particle-based numerical simulations—a topic that has not received as much attention in existing research. The second line of work applies suitable ML architectures, such as convolutional neural networks (CNNs) and graph neural networks (GNNs), to predict grain-level kinematic behaviors in both sparse and dense systems at the microscopic scale.

Both of these approaches offer promising potential to significantly lower the computational cost associated with traditional particle- or grain-based numerical methods. A detailed summary of the advantages and limitations of these ML-driven microscopic grain information models is presented in Table 2.

Table 2: Summary of advantages and disadvantages of ML-aided microscopic modelling

ML-aided microscopic modelling	Advantages	Disadvantages
ML-based contact models	(1) High computational efficiency compared to traditional contact models; (2) Strong ability to account for the influence of particle shape; (3) User friendly	(1) The training data normally incorporates contact assumptions; (2) Limited by the completeness of the training data
ML-aided grain-level kinematic feature simulation	(1) Lower computational complexity; (2) High computational efficiency	(1) Error accumulation problem; (2) Ignoring the influence of particle rotation; (3) Weaker suitability to the system where particles are added or removed

Compared to conventional contact models, ML-based contact models provide a significant advantage by directly predicting contact features and interactions between particle pairs using their positional information. This bypasses the traditional contact detection and resolution procedures, greatly improving the computational efficiency of DEM simulations. Moreover, ML models are capable of capturing particle shape influences when sufficient grain-level geometric information is available, allowing generalization across particles of varying shapes.

This methodology enhances prior contact detection techniques that simplified particles as 2D circles or 3D spheres to reduce computational load, often at the cost of ignoring rotational resistance. Although more precise contact detection methods exist, their application is limited by the complexity of computational geometry, making them less accessible to users without such expertise (e.g., Zhang et al., Gao et al.).

In grain-level kinematic feature simulation, geometric neural networks can effectively predict the acceleration of individual grains from their current positions, eliminating the iterative acceleration calculations used in traditional DEM models. This leads to a substantial improvement in simulation efficiency.

Despite these advantages, there are notable limitations. ML-based contact models are constrained by the completeness and quality of training data, particularly the contact geometry. Since contact states depend on relative positions and particle shapes, generating a robust dataset requires simulating a wide variety of contact scenarios, which is resource-intensive.

Additionally, data generation often relies on theoretical contact models, wherein features such as overlap distance or contact norm direction are empirically defined and vary across models. This inconsistency hampers the scalability and development of generalized ML-based contact models.

When using GNNs or CNNs to simulate the evolution of particle systems, one prominent issue is the accumulation of prediction errors. This is because each update depends on the particle positions from the prior time step, causing errors to propagate. Furthermore, current ML-based kinematic models often ignore critical physical attributes such as particle rotation and angular velocity, which are vital in granular material behavior. Additionally, graph-based networks like GNNs assume a fixed number of nodes, making them less suitable for dynamic systems where particles can be added or removed during simulations.

To overcome these challenges, the following research directions are proposed:

1. **Enhancing training data for ML-based contact models:** The main barrier to building a general-purpose ML-based contact model lies in the completeness and quality of the training data. Future work should focus on developing advanced methods to extract contact features that do not rely heavily on empirical simplifications. Improvements can be made by reducing assumptions during data preparation and by generating more comprehensive and diverse datasets to improve generalization across a broader set of contact scenarios.

2. **Mitigating error accumulation in particle rollouts:** Long-term simulations of particle systems using geometry-based ML networks often suffer from cumulative prediction errors. Research should consider integrating error correction techniques, such as active learning strategies, that allow the model to adapt predictions dynamically and limit error propagation during simulations.

3. **Incorporating rotational dynamics into ML models:** Most current grain-level ML models focus only on position and translational velocity, omitting rotational aspects like angular velocity. Future work could involve modifying CNN or GNN architectures to account for these additional physical degrees of freedom, enabling more accurate and realistic simulations of particle behavior.

3.3 The ML-Based Constitutive Models of Granular Materials

The development of the ML-based constitutive model of granular materials can be traced back to the 1990s [**Sidarta, 1998**; **Zhao, 1999**; **Ghaboussi, 1998**], and the development of the ML-based constitutive model of granular materials is undeniably one of the most prominent subjects in ML-assisted numerical methods. The construction of ML surrogate stress–strain models for grain media depends on numerous factors, such as the hyperparameters, optimization algorithm, loss function, etc., used in neural networks [**Jin, 2021**]. However, the two fundamental factors that govern the performance of the ML-based constitutive model are the data resource and the feature selection of input–output corresponding to the used networks.

3.3.1 The Data Sources

Table 3 offers a summary of partial works on ML-based constitutive models of granular materials over the past decades. There are mainly two types of data resources used when developing the ML constitutive models of granular materials: one is the experimental data, and the other is synthetic data.

Material	Reference	Experiment	Loading	Drained	Data source	Neural network
Sand	Sidarta_1998	Triaxial	M	D	Experiment	MLP
Sand	Ghaboussi_1998	Triaxial	M	D+U	Experiment	MLP
Soil	Zhu_1998	Triaxial	M	D	Experiment	MLP
Sand	Penumadu_1999	Triaxial	M	D	Experiment	MLP
Coarse sand	Romo_2001	Triaxial	M	/	Experiment	RNN
Lateritic gravel	Habibagahi_2003	Triaxial	M	D	Experiment	MLP
Sand	Shahin_2006	Triaxial	M	D	Experiment	MLP
Sand	Hashash_2008	Triaxial	M	D	Experiment	MLP
Soil	He_2009	Triaxial	M	U	Experiment	MLP
Lateritic gravel	Johari_2011	Triaxial	M	D	Experiment	MLP
Soil	Lv_2011	Triaxial	M	U	Experiment	MLP
Sand	Sezer_2011	Direct shear	M	/	Experiment	MLP
Rockfill	Araei_2014	Triaxial	M	/	Experiment	MLP
Sand	Rashidian_2014	Triaxial	M	D	Experiment	MLP
Sand	Kohestani_2016	Triaxial	M	D	Experiment	MLP
–	Basheer IA (2000)	/	C	/	Synthetic data, and experiment	MLP
–	Liu D (2018)	Simple shear	C	/	Synthetic data (DEM)	LSTM
–	Sun W (2019)	Tension-shear	C	/	Synthetic data (DEM)	GRU
–	Zhou A, Xu YS (2019)	/	M	/	Synthetic data (MCC)	LSTM
–	Ye GL (2020)	Triaxial	C	D+U	Synthetic data (EM)	LSTM
–	Zhao T (2021)	Triaxial	C	/	Synthetic data (DEM)	GRU
–	Feng Y (2022)	Triaxial	C	/	Synthetic data (DEM)	TCNN
–	Zhou W (2022)	Triaxial	M	/	Synthetic data (DEM)	LSTM

Table 3: Summary of neural network models developed for granular materials

3.3.2 The Experimental Data

The experimental data, which directly reflects the stress–strain response of granular materials, implicitly encapsulates the most authentic constitutive laws without any assumption, and thus the ML models developed from the experimental data can reveal the most essential mechanical features of granular materials. As listed in Table 3, the mechanical response of different granular materials, such as soil, sand, clay, ballast, and rockfill, has been investigated by different neural networks. Most ML models focus on the mechanical behaviour of granular materials under the drained and undrained triaxial test, and the remaining research is dedicated to developing ML surrogate models which can represent the mechanical features of granular media under direct shearing [**Sezer, 2011**], simple shearing [**Y**; **Liu, 2018**], and tension-shear [**Sun, 2019**].

While the experimental data can provide reliable inputs for neural networks to extract underlying principles governing the behaviour of materials, the limitations of the experimental data should also be taken into consideration. The training of neural networks normally requires a sufficient amount of data samples, making a purely experimental data-driven approach expensive. Additionally, restricted by the experimental facility, most experimental data used for training ML models are generated under specific shearing or triaxial compression conditions, covering only a partial range of stress–strain space and material types. Thus, the robustness of trained ML models is limited.

3.3.3 The Synthetic Data

Compared to experimental data, synthetic data can be a cost-effective alternative to develop ML-based constitutive models for granular materials, as it is not bound by experimental constraints and can span a wider range of stress–strain space. As demonstrated in Table 3, the synthetic data generally can be acquired in two approaches. The first is the phenomenological constitutive models, such as the critical state-based models **[Zhou, 2009**; **Lojander, 2011]** and the deviatoric hardening model **[Wang, 2020**; **Pietruszczak, 1985]**, and the other is the particle-based numerical techniques, such as the discrete element method (DEM).

Given these advantages, it is possible to generate extensive amounts of synthetic data encompassing various materials and more extensive stress–strain space to establish more robust machine learning models. Ma et al. **[Zhou, 2022]** provide an example of this, where one ML model capable of simulating the stress–strain response of granular materials with different particle size distributions (PSDs) and initial void ratio (e_0) under random loading paths was obtained through DEM-generated data. In addition, the development of the synthetic data-based ML model can also provide prior knowledge for constructing experiment data-based machine learning models. In **[Basheer, 2002]**, several mapping methods based on synthetic data were compared before selecting the true sequential dynamic mapping method for simulating the cyclic behaviour of soils with experimental data.

While ML models derived from synthetic data can capture the fundamental mechanical characteristics of granular materials under specific loading paths, they are limited in uncovering deeper constitutive laws, since synthetic data are generated under some assumptions (e.g., the homogenization in theoretical models) and simplification (e.g., the simplified shape of grain in DEM), which results in the loss of some intrinsic physical information of granular materials. However, there is no doubt that synthetic data could be a cost-effective supplement to experimental data in the development of ML-based constitutive models.

4 Simulation of the Brazil Nut Effect in a Granular Medium

4.1 Overview

The Brazil Nut Effect (BNE) refers to the phenomenon where a larger particle in a vibrated granular medium tends to rise to the top, contrary to expectations of size-based percolation. In this simulation, we model the BNE in MATLAB using a two-dimensional discrete particle system subject to vertical tapping. The goal is to observe the upward motion of a larger particle ("Brazil nut") and analyze local packing dynamics.

4.2 Simulation Parameters and Initialization

The system is defined as a 2D container of width $W = 1$ units and height $H = 1$ units, containing $N = 100$ particles. Each particle is treated as a circle with radius r_i, where most particles have radius $r_s = 0.04$ and a single Brazil nut has radius $r_b = 0.15$. Particles are initialized with randomized horizontal positions and increasing vertical positions to avoid overlap. The Brazil nut is placed near the base to simulate realistic initial conditions.

4.3 Tapping and External Forcing

The system is tapped using a Dirac-delta-like vertical impulse and a sinusoidal impulse applied periodically.

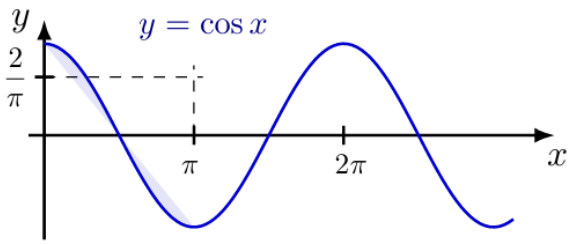

Figure 11: Cosine Wave

4.3.1 Sinusoidal Tapping

For a base tap modeled as a sinusoidal vibration, which is expressed as

$$a_{\text{base}}(t) = -(2\pi f)^2 A \cos(2\pi f t), \tag{18}$$

the vertical acceleration is:

$$a_{\text{base}}(t) = A \cos(2\pi f t), \tag{19}$$

where f is the tapping frequency and A is the amplitude. The net acceleration for a particle i in contact with the floor becomes:

$$a_i = a_{\text{base}} - g, \tag{20}$$

where $g = 9.81\,\text{m/s}^2$ is gravitational acceleration. For other particles, $a_i = -g$.

Sinusoidal tapping offers a smooth and continuous mode of energy input, closely replicating the realistic vibrations encountered in experimental and industrial granular systems. This method allows for precise tuning of vibration parameters such as amplitude and frequency, enabling controlled and repeatable investigations into particle dynamics. Unlike impulsive or discrete taps, sinusoidal forcing minimizes shock-induced disturbances, promoting gradual particle rearrangement and enhancing the visibility of convection and segregation phenomena.

Furthermore, its harmonic nature makes it more amenable to theoretical and analytical modeling, facilitating deeper insights into the mechanics of granular media under periodic excitation.

4.4 Dirac Delta Tapping in Granular Systems

Dirac delta tapping represents an idealized form of tapping, where the base of a container imparts an instantaneous and infinitely short impulse to the granular assembly. Mathematically, this is modeled using the Dirac delta function $\delta(t - t_0)$, which applies a finite impulse at time t_0 but with zero duration:

$$a(t) = a \cdot \delta(t - t_0)$$

where a is the magnitude of the upward acceleration imparted. This abstraction allows for simplified modeling of the effects of a tap without explicitly modeling the base motion over time.

In simulations, Dirac delta tapping is often approximated by imparting an instantaneous velocity or acceleration to the particles or the container base at discrete time intervals. The use of such impulsive excitation helps isolate the influence of external perturbations from continuous vibratory modes, making it particularly useful in Discrete Element Method (DEM) studies or molecular dynamics simulations.

Compared to sinusoidal or harmonic vibrations, Dirac delta tapping offers a clean and discrete perturbation mechanism, aiding in the quantification of particle rearrangements, contact

force evolution, and structural relaxation in a controlled manner. Although idealized, it provides valuable insight into the fundamental dynamics of driven granular systems.

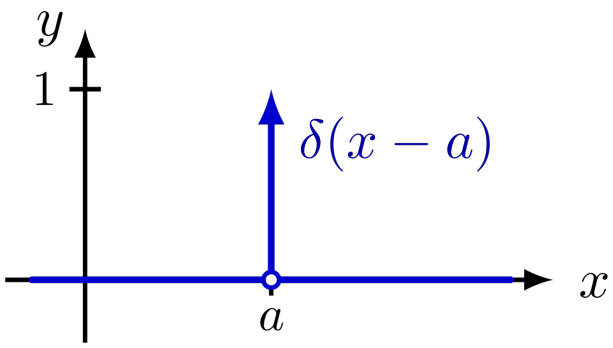

Figure 12: Dirac Delta Function

4.5 Layered Contact Detection using Breadth-First Search (BFS)

To accurately simulate the propagation of vibration through granular media, we implement a layer-based contact detection strategy using the Breadth-First Search (BFS) algorithm. This method assigns a layer index to each particle based on its neighbouring contacts with particles below it, enabling realistic modeling of force transmission from the vibrating base.

4.5.1 Objective

Let the particle positions be $\{(x_i, y_i)\}_{i=1}^{N}$ and radii be $\{r_i\}_{i=1}^{N}$. Given a dynamic floor height $y_{\text{base}}(t)$ due to vibration, the goal is to compute a `layer_index` array such that:

$$\texttt{layer\_index}[i] = \min \text{ number of contact steps from floor to particle } i.$$

4.5.2 Mathematical Formulation

We define the contact graph $G = (V, E)$, where each vertex v_i represents a particle. An edge $(v_i, v_j) \in E$ exists if:

$$\|\vec{r}_i - \vec{r}_j\| \leq r_i + r_j + \epsilon,$$

where $\vec{r}_i = (x_i, y_i)$ is the position vector and ϵ is a small numerical tolerance(of order 10^{-4},just to evade the division by zero error).

We assign layer values as follows:

$$\ell(v_i) = \begin{cases} 0 & \text{if } y_i - r_i \leq y_{\text{base}} + \epsilon, \\ k & \text{if } v_i \text{ is in contact with some } v_j \text{ where } \ell(v_j) = k - 1, \\ -1 & \text{otherwise.} \end{cases}$$

4.6 Algorithm Description

1. **Initialization:**
   - Set $\texttt{layer\_index}[i] = -1$ for all particles i.
   - Identify all particles that are in contact with the floor:

$$\text{if } y_i - r_i \leq y_{\text{base}} + \epsilon, \quad \text{then assign } \texttt{layer\_index}[i] = 0.$$

2. **BFS Traversal:**

- Initialize queue $\mathcal{Q}$ with all particles in Layer 0.
- For each layer l:
  - For each particle $i \in \mathcal{Q}$:
    - Check all unassigned particles j.
    - If $\|\vec{r}_i - \vec{r}_j\| \leq r_i + r_j + \epsilon$, then:

$$\texttt{layer\_index}[j] = l + 1.$$

    - Enqueue j for the next iteration.
- Repeat until no new contacts are found.

4.6.1 Pseudocode (MATLAB)

```
function layer_index = compute_layered_contacts(x, y, radii,base_y)
    num_particles = length(x);
    epsilon = 1e-4;

    % Initialize all particles as unassigned
    layer_index = -1 * ones(num_particles, 1);

    %Assigning Layer 0 (touching the floor)
    current_layer = 0;
    layer_particles = find(y - radii <= base_y+epsilon);
    layer_index(layer_particles) = current_layer;

    % Iteratively building layers
    while ~isempty(layer_particles)
        next_layer_particles = [];

        for i = layer_particles'
            for j = 1:num_particles
                if layer_index(j) == -1
                    dx = x(i) - x(j);
                    dy = y(i) - y(j);
                    dist = sqrt(dx.^2 + dy.^2);
                    if dist <= radii(i) + radii(j) + epsilon
                        layer_index(j) = current_layer + 1;
                        next_layer_particles(end+1) = j;
                    end
                end
            end
        end

        current_layer = current_layer + 1;
        layer_particles = next_layer_particles;
    end
end
```

Listing 1: Layered Contact Detection via BFS

4.6.2 Physical Relevance

This layered propagation scheme is crucial to model how vibration energy from the base travels upward. Only particles directly touching the base experience the full acceleration input; higher layers move due to momentum transferred through contacts. This mimics real-world energy dissipation in dense granular media and enables emergent behaviors such as convection and segregation.

4.6.3 Numerical Relevance & Debugging

Our initial approach was to create a mesh of particles in a random order and check for the interaction between the i^{th} and j^{th} particle. The problem arises when the particles with $j > i$ have $l(j) < l(i)$ especially on the base. Causing the force transmission update to the i^{th} particle not possible since it has already been processed.

Hence, it was necessary to establish a contact-based graph model which would update the force transmissions and collision model testing accurate by recursively moving up one or two layers in the connected component.

The disconnected particles can be treated free-falling, modelling their physics can be done solely on the basis of whether they are in contact with the base or not.

4.7 Motion and Collisions

Particle motion is computed using:

$$v_{y,i}^{t+1} = v_{y,i}^{t} + a_i \cdot \Delta t, \tag{21}$$

$$y_i^{t+1} = y_i^{t} + v_{y,i}^{t+1} \cdot \Delta t, \tag{22}$$

with horizontal motion and wall collisions handled similarly. Pairwise collisions are resolved using conservation of momentum and impulse-based corrections:

$$J = \frac{2m_1 m_2}{m_1 + m_2}(\vec{v}_1 - \vec{v}_2) \cdot \hat{n}, \tag{23}$$

where $\hat{n}$ is the unit normal vector and m_i is the particle mass (proportional to r_i^2).

Note:- Instead of considering the original mass of a particle only the r_i^2 term is taken into account, since the system is considered 2-Dimensional and the density of both type of particles are same.

4.8 Packing Fraction Estimation

We estimate local packing fraction ϕ in the bottom 25% of the container using:

$$\phi = \frac{1}{A_{\text{region}}} \sum_{i \in \text{region}} A_i^{\text{overlap}}, \tag{24}$$

where A_i^{overlap} is the area of particle i that lies within the region and $A_{\text{region}} = W \cdot H_{\text{region}}$. For particles partially intersecting the region, circular segment geometry is used to compute the overlapping area:

$$A_{\text{segment}} = r^2 \cos^{-1}\left(\frac{h}{r}\right) - h\sqrt{r^2 - h^2}, \tag{25}$$

where h is the vertical distance from the particle center to the region boundary.

4.9 Results and Observations

Two key observables are tracked over time:

1. The vertical position $y(t)$ of the Brazil nut.
2. The bottom-region packing fraction $\phi(t)$.

Results show that with periodic tapping, the Brazil nut rises gradually through the granular bed due to percolation and convection-like mechanisms. Simultaneously, packing fraction fluctuates due to compression and dilation caused by vibration.

4.9.1 Sinusoidal Tapping

As mentioned above, a sinusoidal wave:-

$$y_{\text{base}}(t) = A\cos(2\pi f t), \tag{26}$$

The parameters used in the simulation our $A = 0.03$,$f = 7$ resulting in the equation:-

$$y_{\text{base}}(t) = 0.03\cos(14\pi t), \tag{27}$$

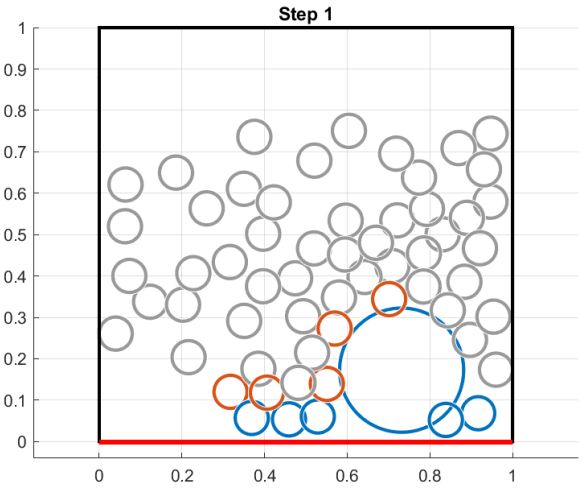

Figure 13: At $t = 0s$

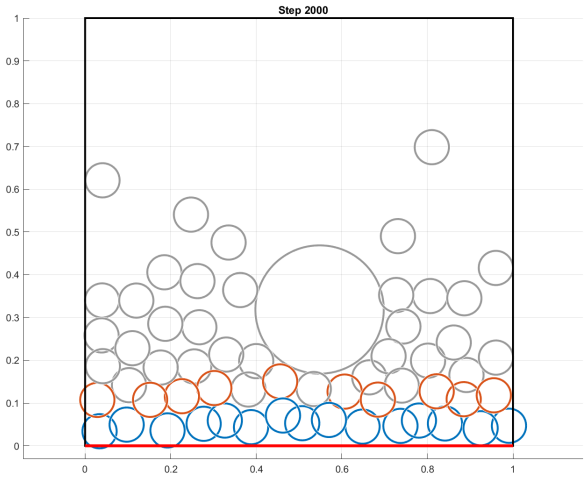

Figure 14: At $t = 30s$

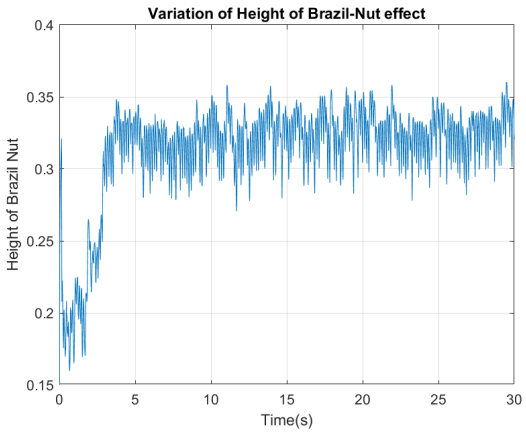

Figure 15: Height Variation of Brazil Nut

As Evident from the height variation curve for the brazil nut effect it is approximately a decaying exponential curve. Which is a practical approximation, since when the brazil nut rises above leaving behind the smaller particles eventually it will rise to a maximum value where there aren't anymore smaller particles above it.

4.9.2 Dirac Delta Tapping

As mentioned above the Dirac Delta Tapping:-

$$a(t) = a \cdot \delta(t - t_0)$$

In our simulation we have defined that parameters as $a = 250$ & $t_0 = 20s$. The tapping repeats at an interval of $t_0 = 20s$. Producing the following results:-

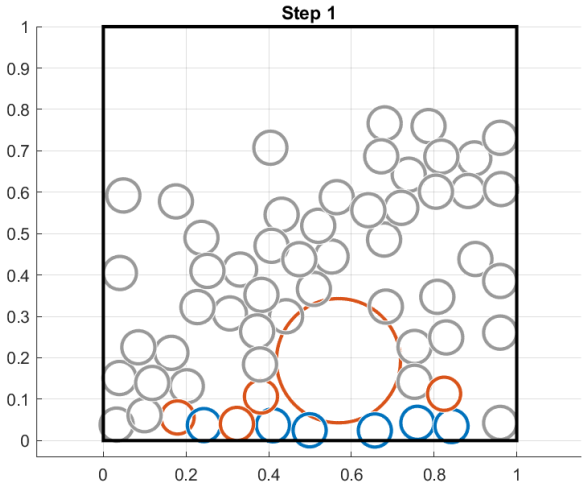

Figure 16: At $t = 0s$

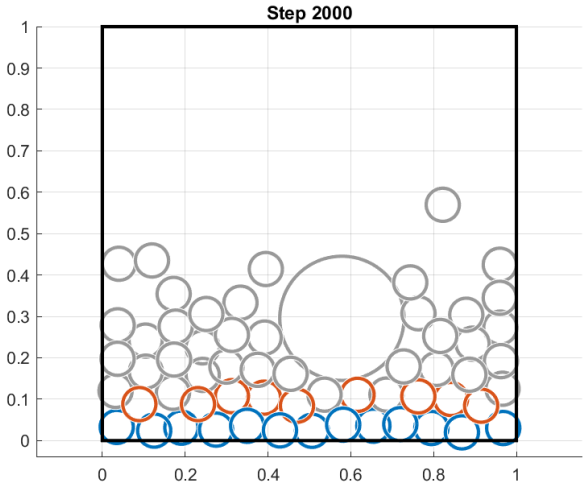

Figure 17: At $t = 30s$

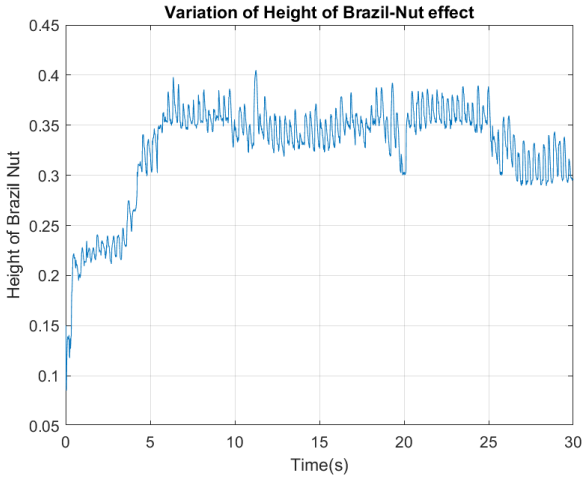

Figure 18: Height Variation of Brazil nut

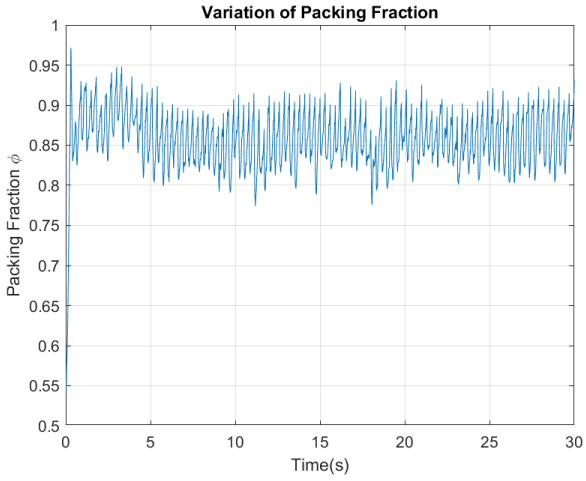

Figure 19: Packing Fraction of Brazil Nut

4.10 Why Does Packing Fraction result in Non-realistic Values?

The system though optimised to the greates possible extent we could, in constraints to the computation power we have, couldn't be optimised to a further extent where the particles states are calculated simultaenously. Using the graph based modelling for the simulation, we were able to reduce the overlaps to a great extent. Since the particles still overlap to some extent there whole volume gets added into the packing fraction equation which results in higher values than RCP(=0.64). Though the behaviour of an exponential decaying graph still remains eminent.

4.11 Importance of Dimensionless Acceleration Parameter Γ and Time Step Δt

In the context of vibrated granular systems, the dimensionless acceleration parameter Γ and the time step Δt are two critical quantities that govern the accuracy, stability, and physical realism of the simulation.

Dimensionless Acceleration Parameter Γ

The dimensionless acceleration parameter Γ is defined as:

$$\Gamma = \frac{(2\pi f)^2 A}{g},$$

where f is the frequency of vibration, A is the amplitude, and g is the gravitational acceleration. Physically, Γ represents the ratio of maximum acceleration of the vibrating base to the gravitational acceleration. When $\Gamma < 1$, the base acceleration is insufficient to lift the particles, and the granular bed remains static or slightly rearranged. However, when $\Gamma > 1$, the system enters a fluidized regime where particles lose contact with the base, allowing for phenomena such as convection, segregation, and the Brazil Nut Effect to emerge. Thus, Γ serves as a critical control parameter dictating the dynamical regime of the granular system. Our system operates at Γ=5.9157.

Time Step Δt

The choice of the time step Δt is fundamental to ensuring both numerical stability and physical fidelity. A sufficiently small Δt ensures accurate resolution of inter-particle collisions, gravitational settling, and vibration-induced motions. If Δt is too large, fast dynamics—especially during contact interactions—may be missed or lead to nonphysical behavior such as particles interpenetrating or excessive energy accumulation. In simulations involving sinusoidal tapping, the time step should ideally resolve the period of oscillation $T = 1/f$ with at least 20–50 steps per cycle to capture the temporal dynamics effectively. Typically, Δt is chosen such that it satisfies:

$$\Delta t \ll \frac{1}{f}, \quad \text{and} \quad \Delta t \ll \sqrt{\frac{r}{g}},$$

where r is the characteristic particle radius. Maintaining this constraint ensures that both gravity and vibration-driven dynamics are properly captured.

Together, Γ and Δt control the excitation strength and temporal resolution of the simulation, and improper tuning of either can result in qualitatively incorrect behaviors. Hence, careful calibration of these parameters is essential for accurate modeling of vibrated granular systems.

In our system:-

$$\Delta t = 0.015 \ll \frac{1}{f} = 0.1429$$

Hence the Δt chosen is about 10 times lesser than the frequency scale, further increasing the reliability of the system.

4.12 Conclusion

This simulation provides a physically informed and visually interpretable demonstration of the Brazil Nut Effect. By layering contact logic, realistic tapping, and packing fraction estimation, it bridges numerical simulation and granular physics theory. The use of contact layers and region-wise packing analysis can be extended to study jamming, segregation, and convection in granular systems.

5 Discussion

While recent advances in machine learning (ML) have propelled the modeling of granular materials, several critical limitations and oversights persist in the literature across neural network architecture selection, microscopic simulations, and constitutive modeling.

Despite the impressive predictive capacity of neural networks (NNs), previous studies have often focused on benchmarking popular architectures without rigorously addressing their suitability to domain-specific challenges in granular mechanics. For instance, while multi-layer perceptrons (MLPs) are praised for their simplicity, their inability to capture historical dependencies is often glossed over. Conversely, time-sequence models such as LSTMs and GRUs are commended for handling loading history, but existing evaluations rarely test them under non-ideal or irregular loading conditions where their performance deteriorates. The literature lacks comparative studies under such realistic scenarios, and most models are only evaluated on well-structured datasets with uniform time steps.

Additionally, spatially aware architectures like CNNs and GNNs are introduced primarily from a feature-extraction standpoint. However, their vulnerability to noise, data incompleteness, and graph-structure perturbations are seldom quantified or systematically evaluated. GNNs, in particular, are assumed to be highly suitable for granular materials, but the dynamic restructuring of graphs during particle interactions—especially in systems where particles are added or removed—is still an unresolved issue.

ML-based microscopic modeling, particularly when applied to DEM acceleration, has mostly focused on improving computational efficiency. However, these studies often overlook fundamental physical fidelity. Many ML models trained to replace contact detection or force calculation rely on data generated from simplified particle shapes (spheres or circles), ignoring the profound impact of non-sphericity and angularity on real granular behavior. While the inclusion of shape parameters is occasionally attempted, such models are rarely validated against diverse and complex particle geometries found in practical applications.

Moreover, critiques of contact modeling literature reveal an overreliance on regression-based outputs that may not satisfy physical consistency. For instance, ML-generated contact forces may violate energy conservation or momentum transfer laws, yet the studies seldom assess such constraints. Also, most models fail to capture rotational interactions and three-dimensional torque effects, reducing the applicability of these ML-enhanced DEM methods for simulating realistic granular systems with rotational freedom.

Another limitation is the lack of generalization across granular systems. Existing models are often trained on a specific scenario (e.g., angle of repose or triaxial compression) and exhibit poor transferability to unseen systems. Few studies investigate model robustness or domain adaptation, which limits practical deployment.

ML-based constitutive modeling, while a promising alternative to empirical models, has its own challenges. First, the dual use of experimental and synthetic data for training introduces a significant inconsistency in quality, yet many studies assume interchangeability. Experimental data often suffer from noise and sparsity, while synthetic data (from DEM or theoretical models) may embed assumptions that obscure physical insight. There is a lack of methodological rigor in reconciling these two data types.

Furthermore, the training strategies employed across studies are fragmented. While time-sequence networks effectively handle cyclic loading paths, their complexity and computa-

tional overhead are often ignored. MLPs, despite being easier to train, show error accumulation, especially in cyclic or multiaxial loading cases. However, rather than addressing this fundamental drawback, many studies resort to ad hoc history variable engineering, which limits generalizability and increases reliance on problem-specific heuristics.

Critically, the conclusions made by many previous works about the superiority of time-sequence models (e.g., GRUs) are based on idealized datasets with fixed strain intervals and do not reflect the variability encountered in experimental settings. Studies often neglect to analyze how neural networks behave when fed inputs with inconsistent sampling or missing data—conditions that are commonplace in engineering practice.

Lastly, there is insufficient effort to embed physical constraints or priors into ML models, leading to predictions that may fit training data well but diverge from known mechanical behavior under extrapolation. The long-standing gap between data-driven modeling and physical interpretability remains largely unaddressed.

Overall Critique

In summary, while prior studies have made valuable contributions in introducing and testing ML techniques for granular materials, many have prioritized performance metrics like mean absolute error over physical consistency, generalizability, and interpretability. There is a need to move beyond demonstration-based research toward rigorous, reproducible, and physics-aware ML modeling frameworks. Future work should bridge the gap between data-driven and physics-based approaches by embedding domain knowledge, validating across heterogeneous data types, and benchmarking under real-world complexities.

6 Conclusion and Future work

Recent developments in machine learning (ML) have shown considerable promise in the field of granular material modeling, enabling more flexible, data-driven alternatives to traditional numerical methods. ML techniques have been explored at various scales, ranging from particle-level modeling to macroscopic stress–strain behavior, with the use of diverse neural network architectures to address different aspects of the problem.

One major area of progress lies in the application of different neural network models to granular material simulations. Neural networks such as MLPs, RNNs, GRUs, CNNs, and GNNs have each demonstrated distinct advantages depending on the structure of the data and the nature of the problem. MLPs are favored for their simplicity and efficiency but struggle with temporal dynamics unless aided by engineered history inputs. On the other hand, RNN-based models, including GRUs and LSTMs, can inherently capture time-dependent behaviors but are more complex to train and exhibit sensitivity to variations in input data intervals. CNNs and GNNs offer powerful spatial feature extraction capabilities, especially useful for encoding particle configurations or topology. However, these models are often limited by large data requirements, noise sensitivity, and high training costs. Ultimately, no universal architecture exists, and network selection remains context-dependent.

At the microscopic level, ML has been successfully employed to accelerate simulations that traditionally rely on computationally intensive techniques like the discrete element method (DEM). ML-based surrogate models have replaced components such as contact detection and force resolution, greatly improving efficiency. Furthermore, kinematic prediction models using CNNs and GNNs have enabled direct estimation of particle accelerations, bypassing iterative physical calculations. Despite these advantages, challenges remain—particularly in the generalization of these models across different granular systems and particle shapes.

Most existing studies rely on data from regular or spherical particle systems, limiting their applicability to more complex or realistic scenarios. Additionally, the exclusion of rotational motion and torque in current models undermines their completeness in fully characterizing granular behavior.

On the macroscopic scale, machine learning has enabled the development of constitutive models that can predict stress–strain responses under various loading conditions. These models are typically trained using either experimental data or synthetic data generated via simulations. Synthetic data offers broader stress–strain coverage and ease of generation, whereas experimental data is more representative but costly and limited in quantity. ML-based constitutive models trained on these datasets have been able to capture non-linear, history-dependent behaviors with reasonable accuracy. Time-sequence neural networks have been particularly effective in modeling cyclic loading due to their memory capabilities. However, these models are often sensitive to irregular loading intervals and may not generalize well. Meanwhile, simpler models like MLPs require the addition of history variables to manage state-dependent responses, which introduces a dependency on the choice and design of these variables. Additionally, many studies emphasize predictive accuracy without evaluating the physical consistency or robustness of the models under extrapolated or unseen conditions.

Collectively, while ML offers innovative tools to enhance the modeling of granular materials, several gaps remain in ensuring that these models are interpretable, transferable, and physically sound. Future efforts must focus on bridging these gaps by developing frameworks that are not only data-driven but also informed by fundamental physical principles.

To advance the state-of-the-art, the following directions can be taken:

1. **Hybrid and Physics-Informed Architectures**: Develop hybrid models that combine the strengths of RNNs, GNNs, and CNNs to capture both temporal and spatial dependencies. Incorporate physics-informed neural networks (PINNs) to embed domain knowledge directly into the model training process.
2. **Synthetic and Active Data Generation**: Enhance ML model generalization through broader, more diverse synthetic datasets. Active learning frameworks could be utilized to iteratively refine datasets for underrepresented contact conditions and particle shapes.
3. **Error Correction in Time-Series Models**: Introduce recursive or online learning mechanisms to counter error accumulation in rollout-based predictions. Real-time feedback or filtering techniques could correct trajectory deviations dynamically.
4. **Inclusion of Rotational and Complex Particle Dynamics**: Extend ML models to incorporate rotational degrees of freedom such as torque and angular velocity. This could involve adapting CNN or GNN architectures to handle tensorial or quaternion inputs.
5. **Transfer and Meta-Learning for Constitutive Models**: Apply transfer learning to leverage synthetic pretraining followed by fine-tuning on limited experimental data. Explore meta-learning for faster adaptation across different granular systems and loading scenarios.
6. **Robustness to Irregular Loading**: Improve time-sequence models by enhancing resilience to variable strain intervals using interpolation, attention mechanisms, or data augmentation.
7. **Benchmark Frameworks**: Develop standardized datasets, test protocols, and evaluation metrics to enable fair benchmarking and accelerate comparative research in ML-aided granular mechanics.

7 Contributions of Team Members

1. **C Zoliansangi (220303):**
    - **Synthetic Data and GRU Analysis**: Examined the use of synthetic data from DEM for training GRU-based models in constitutive behavior prediction. Studied how GRUs capture long-term dependencies and compared their performance with other time-sequence models.
    - **Finite Element Method (FEM)**: Analyzed the role of FEM in continuum modeling of granular media. Investigated how FEM solvers can integrate ML-based stress models and the challenges associated with such hybrid frameworks.
    - **Kinematic Feature Representation**: Explored how particle-level kinematic features—like velocity, displacement, and contact configuration—serve as inputs to ML models such as CNNs and GNNs. Evaluated methods for preprocessing and encoding these features.
    - **Neural Network Architecture Comparison**: Reviewed multiple ML architectures (MLP, RNN, GRU, CNN, GNN) used in granular modeling. Compared them based on their suitability for different data types, learning stability, and predictive performance.
    - References Read:-[192,115,31,32,36]
2. **Harsh Nirmal (220431):**
    - **Brazil Nut Effect and Time Step Analysis**: Studied the simulation framework of the Brazil Nut Effect in vibrated granular media. Emphasized the importance of choosing appropriate time steps and the dimensionless acceleration parameter Γ to ensure simulation accuracy and stability.
    - **LSTM and RNN Models**: Conducted a literature review on LSTM and RNN architectures for time-dependent modeling of granular systems. Analyzed their respective capabilities in capturing loading history and cyclic behaviors, along with known limitations in training dynamics.
    - **Random Forest Applications**: Evaluated the usage of Random Forest models for regression and classification of granular behavior, including material property prediction. Emphasized model interpretability and performance comparison with neural networks.
    - **Material Point Method (MPM)**: Studied MPM as an alternative to FEM for simulating large-deformation granular flows. Compared its suitability for integration with ML-based constitutive laws and discussed its advantages in particle-continuum bridging.
    - References Read:-[33,34,99,102,192]
3. **Rohan Nimesh (220907):**
    - **Experimental Data Analysis**: Studied the role of experimental datasets in training ML-based constitutive models of granular materials. Critically evaluated literature focusing on triaxial and shear test data used in stress–strain modeling and discussed the limitations and sparsity issues of such datasets.
    - **Packing Fraction Estimation in Vibrated Systems**: Investigated methods for estimating local packing fraction in 2D granular simulations. Analyzed segment-based overlap models and their implications for tracking densification during Brazil Nut Effect simulations.

- **Finite Difference Method (FDM)**: Explored the use of FDM for simulating time evolution in granular dynamics. Studied how position and velocity updates were used for modeling particle motion under sinusoidal and Dirac delta tapping conditions.
- **Synthetic Data in ML Modeling**: Reviewed approaches for generating and utilizing synthetic data from DEM and theoretical models. Compared synthetic data's flexibility against experimental data and highlighted its use in ML model generalization and training diversity.
- References Read:-[115,128,160,161,163]

4. Sanskaar Srivastava (220967):

- Studied **Graph Neural Networks (GNNs)**, exploring their capacity to model complex particle interactions within granular assemblies. Investigated GNN architectures for encoding geometric and physical features, and applied them for predicting contact forces, kinematics, and emergent behaviors like force chains.
- Studied and applied the **Finite Element Method (FEM)** in the context of continuum modeling of granular media. Focused on understanding FEM's strengths and limitations in capturing macroscopic stress-strain behavior and explored its coupling with microscale data through machine learning-based surrogate models.
- Developed a full simulation of the **Brazil Nut Effect (BNE)** using a customized algorithm that incorporates contact layering, sinusoidal tapping, and inter-particle repulsion. Designed the tapping mechanism to emulate realistic vibration conditions, tracked the Brazil nut trajectory, and computed dynamic packing fractions.
- Researched and studied a hybrid **ML-integrated DEM framework** to reduce computational load by replacing conventional contact resolution steps with neural network-based surrogates.
- Designed the **BNE Algorithm**, a layer-by-layer update mechanism using BFS traversal for particle interaction propagation. This allowed accurate modeling of how energy transfer occurs through granular beds during external tapping.
- References Read:-[11, 40, 44, 45, 97, 105, 123, 141]

5. Sidhant Thalor (221055):

- Studied and analyzed various neural network models including **Multi-Layer Perceptrons (MLPs), Convolutional Neural Networks (CNNs), Graph Neural Networks (GNNs), Recurrent Neural Networks (RNNs), and Temporal Convolutional Neural Networks (TCNNs)** with respect to their applicability in modeling granular dynamics and sequential data in material simulations.
- Explored and compared **Principal Component Analysis (PCA)** and **t-Distributed Stochastic Neighbor Embedding (t-SNE)** for visualizing high-dimensional simulation data and identifying patterns in particle behavior under varying loading conditions.
- Reviewed the role of classical machine learning algorithms such as **Support Vector Machine (SVM)** in classifying granular systems based on features like compaction state, coordination number, or flow behavior.
- Studied the principles of **Material Point Method (MPM)** and its suitability for capturing large deformation and continuum-scale phenomena in granular assemblies, especially in scenarios where traditional mesh-based methods struggle considering it as an alternative to FEM.

- Conducted a comparative literature-based evaluation of deep learning models in granular mechanics. **Assessed strengths and limitations of each architecture** in terms of accuracy, generalization, computational cost, and interpretability.
- Reviewed recent advancements in hybrid modeling approaches that integrate machine learning into **Discrete Element Method (DEM) simulations** to improve efficiency and predictive capabilities across scales.
- References Read:- [101, 145, 69, 157, 11, 175, 148, 97, 179, 156, 153, 93]

6. **Sidharth Budania (221057):**

- Studied ML-based grain-level kinematic simulations using CNNs and GNNs to predict particle motion without explicit force modeling in granular systems.
- Explored the use of experimental data in constitutive modeling, focusing on stress–strain behavior and addressing limitations due to data sparsity.
- Analyzed the role of the dimensionless acceleration parameter (Γ) in inducing fluidization and segregation phenomena in vibrated beds.
- Reviewed continuum modeling methods such as SPH and FDM, emphasizing their limitations in capturing discrete particle interactions.
- Investigated DEM–ML integration strategies to improve efficiency in contact modeling and supported interpretation of Brazil Nut Effect simulation results.
- Reference Read:-[47,155,93,143,145,99,44,62,23]

Additional Contributions

In addition to the individual contributions listed above, all team members made significant contributions to the following aspects of the project:

- **Literature Review**: All members participated in the comprehensive analysis and synthesis of existing research, ensuring a thorough understanding of the state-of-the-art in the field.
- **Discussion and Drawbacks**: The group collaboratively discussed the findings, identifying limitations and potential drawbacks of the reviewed works to provide a balanced perspective.
- **Conclusion**: The team collectively formulated the concluding remarks, summarizing the key insights and implications derived from the study.
- **Future Work**: All members contributed to outlining potential directions for future research, proposing innovative ideas to address current gaps.

Overall, each member contributed equally to the success of the experiment.

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

[2] Alipour, M., & Lashkari, A. (2018). Sand instability under constant shear drained stress path. *International Journal of Solids and Structures*, *150*, 66–82.

[3] Amroune, M. (2021). Machine learning techniques applied to online voltage stability assessment: A review. *Archives of Computational Methods in Engineering*, *28*, 273–287.

[4] Andrade, J. E., Avila, C., Hall, S. A., Lenoir, N., & Viggiani, G. (2011). Multiscale modeling and characterization of granular matter: From grain kinematics to continuum mechanics. *Journal of the Mechanics and Physics of Solids*, *59*(2), 237–250.

[5] Andrade, J. E., & Tu, X. (2009). Multiscale framework for behavior prediction in granular media. *Mechanics of Materials*, *41*(6), 652–669.

[6] Aoyama, Y., Haeri, A., & Theodorou, E. A. (2023). Optimal control of granular material. *arXiv preprint arXiv:2302.03231*.

[7] Araei, A. A. (2014). Artificial neural networks for modeling drained monotonic behavior of rockfill materials. *International Journal of Geomechanics*, *14*(3), 04014005.

[8] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., & Baskurt, A. (2011). Sequential deep learning for human action recognition. In *Human Behavior Understanding: Second International Workshop, HBU 2011, Amsterdam, The Netherlands, November 16, 2011. Proceedings* (pp. 29–39). Springer.

[9] Bagheri, G., Bonadonna, C., Manzella, I., & Vonlanthen, P. (2015). On the characterization of size and shape of irregular particles. *Powder Technology*, *270*, 141–153.

[10] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[11] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

[12] Bandara, S., & Soga, K. (2015). Coupling of soil deformation and pore fluid flow using material point method. *Computers and Geotechnics*, *63*, 199–214.

[13] Banimahd, M., Yasrobi, S., & Woodward, P. K. (2005). Artificial neural network for stress–strain behavior of sandy soils: Knowledge-based verification. *Computers and Geotechnics*, *32*(6), 377–386.

[14] Bapst, V., Keck, T., Grabska-Barwińska, A., Donner, C., Cubuk, E. D., Schoenholz, S. S., Obika, A., Nelson, A. W., Back, T., Hassabis, D., et al. (2020). Unveiling the predictive power of static structure in glassy systems. *Nature Physics*, *16*(4), 448–454.

[15] Bardenhagen, S., Brackbill, J., & Sulsky, D. (2000). The material-point method for granular materials. *Computer Methods in Applied Mechanics and Engineering*, *187*(3–4), 529–541.

[16] Basheer, I. (2002). Stress–strain behavior of geomaterials in loading reversal simulated by time-delay neural networks. *Journal of Materials in Civil Engineering*, *14*(3), 270–273.

[17] Basheer, I. A. (2000). Selection of methodology for neural network modeling of constitutive hystereses behavior of soils. *Computer-Aided Civil and Infrastructure Engineering*, *15*(6), 445–463.

[18] Battaglia, P., Pascanu, R., Lai, M., & Jimenez Rezende, D. (2016). Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems* (Vol. 29, pp. 4509–4517).

[19] Belytschko, T., Lu, Y. Y., & Gu, L. (1994). Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, *37*(2), 229–256.

[20] Beuth, L., Benz, T., Vermeer, P. A., & Wieckowski, Z. (2008). Large deformation analysis using a quasi-static material point method. *Journal of Theoretical and Applied Mechanics*, *38*(1–2), 45–60.

[21] Beuth, L., Wieckowski, Z., & Vermeer, P. (2011). Solution of quasi-static large-strain problems by the material point method. *International Journal for Numerical and Analytical Methods in Geomechanics*, *35*(13), 1451–1465.

[22] Bowman, E. T., Soga, K., & Drummond, W. (2001). Particle shape characterisation using Fourier descriptor analysis. *Géotechnique*, *51*(6), 545–554.

[23] Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32.

[24] Brinkgreve, R. B. (2005). Selection of soil models and parameters for geotechnical engineering application. In *Proceedings of GeoFrontiers 2005, Austin, Texas* (pp. 69–98).

[25] Bui, H. H., Fukagawa, R., Sako, K., & Ohno, S. (2008). Lagrangian meshfree particles method (SPH) for large deformation and failure flows of geomaterial using elastic–plastic soil constitutive model. *International Journal for Numerical and Analytical Methods in Geomechanics*, *32*(12), 1537–1570.

[26] Bui, H. H., Sako, K., & Fukagawa, R. (2007). Numerical simulation of soil–water interaction using smoothed particle hydrodynamics (SPH) method. *Journal of Terramechanics*, *44*(5), 339–346.

[27] Bui, H. H., Sako, K., Fukagawa, R., & Wells, J. (2008). SPH-based numerical simulations for large deformation of geomaterial considering soil–structure interaction. In *Proceedings of the 12th International Conference of International Association for Computer Methods and Advances in Geomechanics (IACMAG)* (pp. 570–578).

[28] Chang, C. S., & Liao, C. L. (1994). Estimates of elastic modulus for media of randomly packed granules. *Applied Mechanics Reviews*, *47*(1S), S197–S206.

[29] Chang, M. B., Ullman, T., Torralba, A., & Tenenbaum, J. B. (2016). A compositional object-based approach to learning physical dynamics. *arXiv preprint arXiv:1612.00341*.

[30] Chen, X., Wang, L. G., Meng, F., & Luo, Z. H. (2021). Physics-informed deep learning for modelling particle aggregation and breakage processes. *Chemical Engineering Journal*, *426*, 131220.

[31] Cheng, Z., & Wang, J. (2022). Estimation of contact forces of granular materials under uniaxial compression based on a machine learning model. *Granular Matter*, *24*(1), 1–14.

[32] Cho, K., Van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. *arXiv preprint arXiv:1409.1259*.

[33] Choi, Y., & Kumar, K. (2024). Graph neural network-based surrogate model for granular flows. *Computers and Geotechnics*, *166*, 106015.

[34] Choi, Y., & Kumar, K. (2024). Inverse analysis of granular flows using differentiable graph neural network simulator. *Computers and Geotechnics*, *171*, 106374.

[35] Ciregan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3642–3649). IEEE.

[36] Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence* (pp. 1237–1242).

[37] Cleary, P. W., & Sawley, M. L. (2002). DEM modelling of industrial granular flows: 3D case studies and the effect of particle shape on hopper discharge. *Applied Mathematical Modelling*, *26*(2), 89–111.

[38] Cundall, P. A. (1971). A computer model for simulating progressive, large-scale movement in blocky rock system. In *Proceedings of the International Symposium on Rock Mechanics* (pp. 129–136).

[39] Cundall, P. A. (1974). Rational design of tunnel supports: A computer model for rock mass behaviour using interactive graphics for the input and output of geometrical data. *Technical Report*.

[40] Cundall, P. A., & Strack, O. D. L. (1979). A discrete numerical model for granular assemblies. *Géotechnique*, *29*(1), 47–65.

[41] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, *2*(4), 303–314.

[42] Das, S. K., & Das, A. (2019). Influence of quasi-static loading rates on crushable granular materials: A DEM analysis. *Powder Technology*, *344*, 393–403.

[43] Deen, N., Annaland, M. V. S., Van der Hoef, M. A., & Kuipers, J. (2007). Review of discrete particle modeling of fluidized beds. *Chemical Engineering Science*, *62*(1–2), 28–44.

[44] De Josselin de Jong, G., & Verruijt, A. (1969). Photoelastic experiments on granular materials. In *Proceedings of the International Conference on Soil Mechanics and Foundation Engineering*.

[45] Deresiewicz (1958).

[46] Duncan, J. M., & Chang, C. Y. (1970). Nonlinear analysis of stress and strain in soils. *Journal of the Soil Mechanics and Foundations Division*, *96*(5), 1629–1653.

[47] Durve, M. Tracking droplets in soft granular flows with deep learning techniques.

[48] Eggersmann, R., Kirchdoerfer, T., Reese, S., Stainier, L., & Ortiz, M. (2019). Model-free data-driven inelasticity. *Computer Methods in Applied Mechanics and Engineering*, *350*, 81–99.

[49] Eghbalian, M., Pouragha, M., & Wan, R. (2023). A physics-informed deep neural network for surrogate modeling in classical elastoplasticity. *Computers and Geotechnics*, *159*, 105472.

[50] Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.

[51] Feng, Y. (2021). An energy-conserving contact theory for discrete element modelling of arbitrarily shaped particles: Basic framework and general contact model. *Computer Methods in Applied Mechanics and Engineering*, *373*, 113454.

[52] Feng, Y. (2021). A generic energy-conserving discrete element modeling strategy for concave particles represented by surface triangular meshes. *International Journal for Numerical Methods in Engineering*, *122*(10), 2581–2597.

[53] Feng, Y. (2023). Thirty years of developments in contact modelling of non-spherical particles in DEM: A selective review. *Acta Mechanica Sinica*, *39*, 722343.

[54] Feng, Y., & Gao, W. (2021). On the strain energy distribution of two elastic solids under smooth contact. *Powder Technology*, *389*, 376–382.

[55] Feng, Y., Han, K., & Owen, D. (2017). A generic contact detection framework for cylindrical particles in discrete element modelling. *Computer Methods in Applied Mechanics and Engineering*, *315*, 632–651.

[56] Fern, J., Rohe, A., Soga, K., & Alonso, E. (2019). *The Material Point Method for Geotechnical Engineering: A Practical Guide*. CRC Press.

[57] Fragkiadaki, K., Agrawal, P., Levine, S., & Malik, J. (2015). Learning visual predictive models of physics for playing billiards. *arXiv preprint arXiv:1511.07404*.

[58] Fu, P., Walton, O. R., & Harvey, J. T. (2012). Polyarc discrete element for efficiently simulating arbitrarily shaped 2D particles. *International Journal for Numerical Methods in Engineering*, *89*(5), 599–617.

[59] Fu, Q., Hashash, Y. M., Jung, S., & Ghaboussi, J. (2007). Integration of laboratory testing and constitutive modeling of soils. *Computers and Geotechnics*, *34*(5), 330–345.

[60] Fuhg, J. N., Marino, M., & Bouklas, N. (2022). Local approximate Gaussian process regression for data-driven constitutive models: Development and comparison with neural networks. *Computer Methods in Applied Mechanics and Engineering*, *388*, 114217.

[61] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4), 193–202.

[62] Galván, J. R., Durve, M., & Scheel, M. (2024). Classification and grain size prediction of granular materials using audio-based neural networks.

[63] Ghaboussi, J., Garrett, J., Jr., & Wu, X. (1991). Knowledge-based modeling of material behavior with neural networks. *Journal of Engineering Mechanics*, *117*(1), 132–153.

[64] Ghaboussi, J., Pecknold, D. A., Zhang, M., & Haj-Ali, R. M. (1998). Autoprogressive training of neural network constitutive models. *International Journal for Numerical Methods in Engineering*, *42*(1), 105–126.

[65] Ghaboussi, J., & Sidarta, D. E. (1998). New nested adaptive neural networks (NANN) for constitutive modeling. *Computers and Geotechnics*, *22*(1), 29–52.

[66] Ghavamian, F., & Simone, A. (2019). Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network. *Computer Methods in Applied Mechanics and Engineering*, *357*, 112594.

[67] Gingold, R. A., & Monaghan, J. J. (1977). Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, *181*(3), 375–389.

[68] Goswami, S., Bora, A., Yu, Y., & Karniadakis, G. E. (2023). Physics-informed deep neural operator networks. In *Machine Learning in Modeling and Simulation: Methods and Applications* (pp. 219–254). Springer.

[69] Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

[70] Guan, Q., Yang, Z., Guo, N., & Hu, Z. (2023). Finite element geotechnical analysis incorporating deep learning-based soil model. *Computers and Geotechnics*, *154*, 105120.

[71] Guan, S., Qu, T., Feng, Y., Ma, G., & Zhou, W. (2023). A machine learning-based multiscale computational framework for granular materials. *Acta Geotechnica*, *18*(4), 1699–1720.

[72] Guan, S., Zhang, X., Ranftl, S., & Qu, T. (2023). A neural network-based material cell for elastoplasticity and its performance in FE analyses of boundary value problems. *International Journal of Plasticity*, *171*, 103811.

[73] Guo, N., & Zhao, J. (2016). 3D multiscale modeling of strain localization in granular media. *Computers and Geotechnics*, *80*, 360–372.

[74] Habibagahi, G., & Bamdad, A. (2003). A neural network framework for mechanical behavior of unsaturated soils. *Canadian Geotechnical Journal*, *40*(3), 684–693.

[75] Harlow, F. H. (1964). The particle-in-cell computing method for fluid dynamics. *Methods in Computational Physics*, *3*, 319–343.

[76] Hashash, Y., Jung, S., & Ghaboussi, J. (2004). Numerical implementation of a neural network based material model in finite element analysis. *International Journal for Numerical Methods in Engineering*, *59*(7), 989–1005.

[77] He, S., & Li, J. (2009). Modeling nonlinear elastic behavior of reinforced soil using artificial neural networks. *Applied Soft Computing*, *9*(3), 954–961.

[78] He, S., Li, Y., Feng, Y., Ho, S., Ravanbakhsh, S., Chen, W., & Poczos, B. (2019). Learning to predict the cosmological structure formation. *Proceedings of the National Academy of Sciences*, *116*(28), 13825–13832.

[79] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

[80] Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

[81] Hu, X., Zhang, Y., Guo, L., Wang, J., Cai, Y., Fu, H., & Cai, Y. (2018). Cyclic behavior of saturated soft clay under stress path with bidirectional shear stresses. *Soil Dynamics and Earthquake Engineering*, *104*, 319–328.

[82] Huang, D., Fuhg, J. N., Weißenfels, C., & Wriggers, P. (2020). A machine learning based plasticity model using proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering*, *365*, 113008.

[83] Hwang, S., Pan, J., Sunny, A. A., & Fan, L. S. (2022). A machine learning-based particle–particle collision model for non-spherical particles with arbitrary shape. *Chemical Engineering Science*, *251*, 117439.

[84] Ingebrand, T., Thorpe, A. J., Goswami, S., Kumar, K., & Topcu, U. (2024). Basis-to-basis operator learning using function encoders. *arXiv preprint arXiv:2410.00171*.

[85] Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 221–231.

[86] Johari, A., Javadi, A., & Habibagahi, G. (2011). Modelling the mechanical behaviour of unsaturated soils using a genetic algorithm-based neural network. *Computers and Geotechnics*, *38*(1), 2–13.

[87] Johnson, K. L. (1987). *Contact Mechanics*. Cambridge University Press.

[88] Jung, S., & Ghaboussi, J. (2006). Neural network constitutive model for rate-dependent materials. *Computers & Structures*, *84*(15–16), 955–963.

[89] Karapiperis, K., Stainier, L., Ortiz, M., & Andrade, J. E. (2021). Data-driven multiscale modeling in mechanics. *Journal of the Mechanics and Physics of Solids*, *147*, 104239.

[90] Kohestani, V., & Hassanlourad, M. (2016). Modeling the mechanical behavior of carbonate sands using artificial neural networks and support vector machines. *International Journal of Geomechanics*, *16*(1), 04015038.

[91] Kruyt, N. P., & Rothenburg, L. (1998). Statistical theories for the elastic moduli of two-dimensional assemblies of granular materials. *International Journal of Engineering Science*, *36*(10), 1127–1142.

[92] Kumar, K., & Choi, Y. (2023). Accelerating particle and fluid simulations with differentiable graph networks for solving forward and inverse problems. In *Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis* (pp. 60–65).

[93] Ladický, L., Jeong, S., Solenthaler, B., Pollefeys, M., & Gross, M. (2015). Data-driven fluid simulations using regression forests. *ACM Transactions on Graphics (TOG)*, *34*(6), 199.

[94] Lai, Z., & Chen, Q. (2017). Characterization and discrete element simulation of grading and shape-dependent behavior of JSC-1A Martian regolith simulant. *Granular Matter*, *19*(4), 69.

[95] Lai, Z., & Chen, Q. (2019). Reconstructing granular particles from X-ray computed tomography using the TWS machine learning tool and the level set method. *Acta Geotechnica*, *14*(1), 1–18.

[96] Lai, Z., Chen, Q., & Huang, L. (2020). Fourier series-based discrete element method for computational mechanics of irregular-shaped particles. *Computer Methods in Applied Mechanics and Engineering*, *362*, 112873.

[97] Lai, Z., Chen, Q., & Huang, L. (2022). Machine-learning-enabled discrete element method: Contact detection and resolution of irregular-shaped particles. *International Journal for Numerical and Analytical Methods in Geomechanics*, *46*(1), 113–140.

[98] Le, B., Yvonnet, J., & He, Q. C. (2015). Computational homogenization of nonlinear elastic materials using neural networks. *International Journal for Numerical Methods in Engineering*, *104*(12), 1061–1084.

[99] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551.

[100] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

[101] Lee, S., Ha, J., Zokhirova, M., Moon, H., & Lee, J. (2018). Background information of deep learning for structural engineering. *Archives of Computational Methods in Engineering*, *25*(1), 121–129.

[102] Li, H., Yu, H., Cao, N., Tian, H., & Cheng, S. (2021). Applications of artificial intelligence in oil and gas development. *Archives of Computational Methods in Engineering*, *28*(3), 937–949.

[103] Li, Z., Chow, J. K., & Wang, Y. H. (2017). Applying the artificial neural network to predict the soil responses in the DEM simulation. In *IOP Conference Series: Materials Science and Engineering* (Vol. 216, p. 012040). IOP Publishing.

[104] Li, Z., Huang, D. Z., Liu, B., & Anandkumar, A. (2023). Fourier neural operator with learned deformations for PDEs on general geometries. *Journal of Machine Learning Research*, *24*(1), 1–26.

[105] Li, Z., Li, X., Zhang, H., Huang, D., & Zhang, L. (2023). The prediction of contact force networks in granular materials based on graph neural networks. *The Journal of Chemical Physics*, *158*(5).

[106] Liang, W., & Zhao, J. (2019). Coupled MPM/DEM multiscale modelling geotechnical problems involving large deformation. In *16th Asian Regional Conference on Soil Mechanics and Geotechnical Engineering*.

[107] Liang, W., & Zhao, J. (2019). Multiscale modeling of large deformation in geomechanics. *International Journal for Numerical and Analytical Methods in Geomechanics*, *43*(5), 1080–1114.

[108] Lim, K. W., & Andrade, J. E. (2013). Granular element method for computational particle mechanics. *Computer Methods in Applied Mechanics and Engineering*, *241–244*, 262–274.

[109] Lim, K. W., & Andrade, J. E. (2014). Granular element method for three-dimensional discrete element calculations. *International Journal for Numerical and Analytical Methods in Geomechanics*, *38*(2), 167–188.

[110] Liu, L., & Cai, W. (2021). Multiscale DeepONet for nonlinear operators in oscillatory function spaces for building seismic wave responses. *arXiv preprint arXiv:2111.04860*.

[111] Liu, M., & Liu, G. (2010). Smoothed particle hydrodynamics (SPH): An overview and recent developments. *Archives of Computational Methods in Engineering*, *17*(1), 25–76.

[112] Liu, Z., Su, L., Zhang, C., Iqbal, J., Hu, B., & Dong, Z. (2020). Investigation of the dynamic process of the Xinmo landslide using the discrete element method. *Computers and Geotechnics*, *123*, 103561.

[113] Logarzo, H. J., Capuano, G., & Rimoli, J. J. (2021). Smart constitutive laws: Inelastic homogenization through machine learning. *Computer Methods in Applied Mechanics and Engineering*, *373*, 113482.

[114] Lu, L., Gao, X., Dietiker, J. F., Shahnam, M., & Rogers, W. A. (2021). Machine learning accelerated discrete element modeling of granular flows. *Chemical Engineering Science*, *245*, 116832.

[115] Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, *3*(3), 218–229.

[116] Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, *82*, 1013–1024.

[117] Lv, Y., Nie, L., & Xu, K. (2011). Study of the neural network constitutive models for turfy soil with different decomposition degree. In *2011 Second International Conference on Mechanic Automation and Control Engineering* (pp. 6111–6114). IEEE.

[118] Ma, G., Guan, S., Wang, Q., Feng, Y., & Zhou, W. (2022). A predictive deep learning framework for path-dependent mechanical behavior of granular materials. *Acta Geotechnica*, *17*(8), 3463–3478.

[119] Ma, X., & Zhang, D. Z. (2006). Statistics of particle interactions in dense granular material under uniaxial compression. *Journal of the Mechanics and Physics of Solids*, *54*(7), 1426–1448.

[120] Mandl, L., Goswami, S., Lambers, L., & Ricken, T. (2024). Separable DeepONet: Breaking the curse of dimensionality in physics-informed machine learning. *arXiv preprint arXiv:2407.15887*.

[121] Mašín, D. (2005). A hypoplastic constitutive model for clays. *International Journal for Numerical and Analytical Methods in Geomechanics*, *29*(4), 311–336.

[122] Mayr, A., Lehner, S., Mayrhofer, A., Kloss, C., Hochreiter, S., & Brandstetter, J. (2023). Boundary graph neural networks for 3D simulations. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 37, pp. 9099–9107).

[123] Micheli, A. (2009). Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, *20*(3), 498–511.

[124] Nitka, M., Combe, G., Dascalu, C., & Desrues, J. (2011). Two-scale modeling of granular materials: A DEM-FEM approach. *Granular Matter*, *13*(3), 277–281.

[125] Oñate, E., Idelsohn, S. R., Del Pin, F., & Aubry, R. (2004). The particle finite element method—An overview. *International Journal of Computational Methods*, *1*(2), 267–307.

[126] Pande, G., Pietruszczak, S., & Wang, M. (2020). Role of gradation curve in description of mechanical behavior of unsaturated soils. *International Journal of Geomechanics*, *20*(6), 04019159.

[127] Penumadu, D., & Zhao, R. (1999). Triaxial compression behavior of sand and gravel using artificial neural networks (ANN). *Computers and Geotechnics*, *24*(3), 207–230.

[128] Petalas, A. L., Dafalias, Y. F., & Papadimitriou, A. G. (2020). SANISAND-F: Sand constitutive model with evolving fabric anisotropy. *International Journal of Solids and Structures*, *188–189*, 12–31.

[129] Peters, J. F., Hopkins, M. A., Kala, R., & Wahl, R. E. (2009). A poly-ellipsoid particle for non-spherical discrete element method. *Engineering Computations*, *26*(6), 645–657.

[130] Poorooshasb, H. B., & Pietruszczak, S. (1985). On yielding and flow of sand; A generalized two-surface model. *Computers and Geotechnics*, *1*(1), 33–58.

[131] Qu, T., Di, S., Feng, Y., Wang, M., & Zhao, T. (2021). Towards data-driven constitutive modelling for granular materials via micromechanics-informed deep learning. *International Journal of Plasticity*, *144*, 103046.

[132] Qu, T., Feng, Y., & Wang, M. (2021). An adaptive granular representative volume element model with an evolutionary periodic boundary for hierarchical multiscale analysis. *International Journal for Numerical Methods in Engineering*, *122*(10), 2239–2253.

[133] Qu, T., Guan, S., Feng, Y., Ma, G., Zhou, W., & 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.

[134] Qu, T., Zhao, J., Guan, S., & Feng, Y. (2023). Data-driven multiscale modelling of granular materials via knowledge transfer and sharing. *International Journal of Plasticity*, *171*, 103786.

[135] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707.

[136] Rangel, R. L., Franci, A., Oñate, E., & Gimenez, J. M. (2024). Multiscale data-driven modeling of the thermomechanical behavior of granular media with thermal expansion effects. *Computers and Geotechnics*, *176*, 106789.

[137] Rashidian, V., & Hassanlourad, M. (2014). Application of an artificial neural network for modeling the mechanical behavior of carbonate soils. *International Journal of Geomechanics CHI*, *14*(1), 142–150.

[138] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[139] Reynolds, D. A. (2009). Gaussian Mixture Models. In *Encyclopedia of Biometrics*. Springer.

[140] Roberts, N., Khodak, M., Dao, T., Li, L., Ré, C., & Talwalkar, A. (2021). Learning operations for neural PDE solvers. In *Proceedings of ICLR SimDL Workshop*.

[141] Rodriguez-Ortiz (1974) & Serrano & Rodriguez-Ortiz (1973).

[142] Romo, M. P., García, S. R., Mendoza, M. J., & Taboada-Urtuzuástegui, V. (2001). Recurrent and constructive-algorithm networks for sand behavior modeling. *International Journal of Geomechanics*, *1*(4), 371–387.

[143] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.

[144] Roscoe, K., & Burland, J. B. (1968). On the generalized stress–strain behavior of wet clay. In J. Heyman & F. Leckie (Eds.), *Engineering Plasticity* (pp. 535–609). Cambridge University Press.

[145] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

[146] Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., & Battaglia, P. (2020). Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning* (pp. 8459–8468). PMLR.

[147] Sanvitale, N., Mehdizadeh, S. A., & Bowman, E. T. (2021). Image segmentation of photoelastic disks using U-Net deep learning. *Computers and Geotechnics*, *133*, 104027.

[148] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80.

[149] Sezer, A. (2011). Prediction of shear development in clean sands by use of particle shape information and artificial neural networks. *Expert Systems with Applications*, *38*(5), 5603–5613.

[150] Shahin, M. A., & Indraratna, B. (2006). Modeling the mechanical behavior of railway ballast using artificial neural networks. *Canadian Geotechnical Journal*, *43*(11), 1144–1152.

[151] Sidarta, D., & Ghaboussi, J. (1998). Constitutive modeling of geomaterials from non-uniform material tests. *Computers and Geotechnics*, *22*(1), 53–71.

[152] Sołowski, W., & Sloan, S. (2015). Evaluation of material point method for use in geotechnics. *International Journal for Numerical and Analytical Methods in Geomechanics*, *39*(7), 685–701.

[153] Spengler, M. (1999). Fast neural network emulation and control of physics-based models. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, Orlando, Florida* (pp. 9–20).

[154] Stefanos, D., & Gyan, P. (2015). On neural network constitutive models for geomaterials. *Journal of Civil Engineering Research*, *5*(5), 106–113.

[155] Strack, O., & Cundall, P. A. (1978). The distinct element method as a tool for research in granular media. *University of Minnesota, Technical Report*.

[156] Sulsky, D., Zhou, S. J., & Schreyer, H. L. (1995). Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, *87*(1–2), 236–252.

[157] Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (pp. 1017–1024).

[158] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems* (Vol. 27, pp. 3104–3112).

[159] Tavarez, F. A., & Plesha, M. E. (2007). Discrete element method for modelling solid and particulate materials. *International Journal for Numerical Methods in Engineering*, *70*(4), 379–404.

[160] Thakur, M. M., & Penumadu, D. (2020). Triaxial compression in sands using FDEM and micro-X-ray computed tomography. *Computers and Geotechnics*, *124*, 103638.

[161] Ti, K. S., Huat, B., Noorzaei, J., Jaafar, M. S., & Sew, G. S. (2009). A review of basic soil constitutive models for geotechnical application. *Electronic Journal of Geotechnical Engineering*, *14*, 1–18.

[162] Tian, Y., & Yao, Y. P. (2017). Modelling the non-coaxiality of soils from the view of cross-anisotropy. *Computers and Geotechnics*, *86*, 219–229.

[163] Ueda, K., & Iai, S. (2019). Constitutive modeling of inherent anisotropy in a strain space multiple mechanism model for granular materials. *International Journal for Numerical and Analytical Methods in Geomechanics*, *43*(4), 708–737.

[164] Ummenhofer, B., Prantl, L., Thuerey, N., & Koltun, V. (2019). Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations*.

[165] Vlassis, N. N., & Sun, W. (2021). Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening. *Computer Methods in Applied Mechanics and Engineering*, *377*, 113695.

[166] Voyiadjis, G. Z., Alsaleh, M. I., & Alshibli, K. A. (2005). Evolving internal length scales in plastic strain localization for granular materials. *International Journal of Plasticity*, *21*(10), 2000–2024.

[167] Wang, J., & Chan, D. (2014). Frictional contact algorithms in SPH for the simulation of soil–structure interaction. *International Journal for Numerical and Analytical Methods in Geomechanics*, *38*(7), 747–770.

[168] Wang, K., & Sun, W. (2019). Meta-modeling game for deriving theory-consistent, microstructure-based traction–separation laws via deep reinforcement learning. *Computer Methods in Applied Mechanics and Engineering*, *346*, 216–241.

[169] Wang, L., Cai, Y., & Liu, D. (2018). Multiscale reliability-based topology optimization methodology for truss-like microstructures with unknown-but-bounded uncertainties. *Computer Methods in Applied Mechanics and Engineering*, *339*, 358–388.

[170] Wang, M., Feng, Y., Guan, S., & Qu, T. (2024). 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*. *doi:10.1016/j.jrmge.2024.02.003*.

[171] Wang, M., Qu, T., Guan, S., Zhao, T., Liu, B., & Feng, Y. (2022). Data-driven strain–stress modelling of granular materials via temporal convolution neural network. *Computers and Geotechnics*, *152*, 105049.

[172] Wang, M., & Zhang, D. Z. (2021). Deformation accommodating periodic computational domain for a uniform velocity gradient. *Computer Methods in Applied Mechanics and Engineering*, *374*, 113607.

[173] Wang, X., Yin, Z. Y., Su, D., Xiong, H., & Feng, Y. (2021). A novel arcs-based discrete element modeling of arbitrary convex and concave 2D particles. *Computer Methods in Applied Mechanics and Engineering*, *386*, 114071.

[174] Wang, Z., Liu, K., Li, J., Zhu, Y., & Zhang, Y. (2019). Various frameworks and libraries of machine learning and deep learning: A survey. *Archives of Computational Methods in Engineering*, *26*(1), 1–24.

[175] Weng, J. J., Ahuja, N., & Huang, T. S. (1993). Learning recognition and segmentation of 3D objects from 2D images. In *1993 (4th) International Conference on Computer Vision* (pp. 121–128). IEEE.

[176] Werbos, P. J. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, *78*(10), 1550–1560.

[177] Wiewel, S., Becher, M., & Thuerey, N. (2019). Latent space physics: Towards learning the temporal evolution of fluid flow. *Computer Graphics Forum*, *38*(2), 71–82.

[178] Wieckowski, Z. (2004). The material point method in large strain engineering problems. *Computer Methods in Applied Mechanics and Engineering*, *193*(39–41), 4417–4438.

[179] Williams, J. R., & O'Connor, R. (1999). Discrete element simulation and the contact problem. *Archives of Computational Methods in Engineering*, *6*(4), 279–304.

[180] Williams, J. R., & Pentland, A. P. (1992). Superquadrics and modal dynamics for discrete elements in interactive design. *Engineering Computations*, *9*(2), 115–127.

[181] Wriggers, P. (2008). *Nonlinear Finite Element Methods*. Springer.

[182] Wu, J., Yildirim, I., Lim, J. J., Freeman, B., & Tenenbaum, J. (2015). Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Advances in Neural Information Processing Systems* (Vol. 28).

[183] Wu, L., Cui, P., Pei, J., Zhao, L., & Guo, X. (2022). Graph neural networks: Foundation, frontiers and applications. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 4840–4841).

[184] Wu, W., Bauer, E., & Kolymbas, D. (1996). Hypoplastic constitutive model with critical state for granular materials. *Mechanics of Materials*, *23*(1), 45–69.

[185] Xu, K., Huang, D. Z., & Darve, E. (2021). Learning constitutive relations using symmetric positive definite neural networks. *Journal of Computational Physics*, *428*, 110072.

[186] Yao, Y., & Sun, D. (2004). A critical state model for sands dependent on stress and density. *International Journal for Numerical and Analytical Methods in Geomechanics*, *28*(4), 323–337.

[187] Yao, Y. P., Hou, W., & Zhou, A. N. (2009). UH model: Three-dimensional unified hardening model for overconsolidated clays. *Géotechnique*, *59*(5), 451–469.

[188] Yimsiri, S., & Soga, K. (2000). Micromechanics-based stress–strain behaviour of soils at small strains. *Géotechnique*, *50*(5), 559–571.

[189] Yin, Z. Y., & Jin, Y. F. (2019). *Practice of Optimisation Theory in Geotechnical Engineering*. Springer.

[190] Yin, Z. Y., Karstunen, M., Chang, C. S., Koskinen, M., & Lojander, M. (2011). Modeling time-dependent behavior of soft sensitive clay. *Journal of Geotechnical and Geoenvironmental Engineering*, *137*(11), 1103–1113.

[191] Yin, Z. Y., Wang, P., & Zhang, F. (2020). Effect of particle shape on the progressive failure of shield tunnel face in granular soils by coupled FDM-DEM method. *Tunnelling and Underground Space Technology*, *100*, 103394.

[192] You, Z. (2003). Development of a micromechanical modeling approach to predict asphalt mixture stiffness using the discrete element method. *University of Illinois at Urbana-Champaign, PhD dissertation*.

[193] Zhang, D. Z., Ma, X., & Giguere, P. T. (2011). Material point method enhanced by modified gradient of shape function. *Journal of Computational Physics*, *230*(16), 6379–6398.

[194] Zhang, D. Z., & Rauenzahn, R. M. (2000). Stress relaxation in dense and slow granular flows. *Journal of Rheology*, *44*(5), 1019–1041.

[195] Zhang, N., Shen, S. L., Zhou, A., & Xu, Y. S. (2019). Investigation on performance of neural networks using quadratic relative error cost function. *IEEE Access*, *7*, 106642–106652.

[196] Zhang, P., Yang, Y., & Yin, Z. Y. (2021). BiLSTM-based soil–structure interface modeling. *International Journal of Geomechanics*, *21*(7), 04021096.

[197] Zhang, P., Yin, Z. Y., Jin, Y. F. (2021). State-of-the-art review of machine learning applications in constitutive modeling of soils. *Archives of Computational Methods in Engineering*, *28*(5), 3661–3686.

[198] Zhang, P., Yin, Z. Y., Jin, Y. F., & Liu, X. F. (2021). Modelling the mechanical behaviour of soils using machine learning algorithms with explicit formulations. *Acta Geotechnica*, *16*(1), 1–20.

[199] Zhang, P., Yin, Z. Y., Jin, Y. F., & Ye, G. L. (2020). An AI-based model for describing cyclic characteristics of granular materials. *International Journal for Numerical and Analytical Methods in Geomechanics*, *44*(9), 1315–1335.

[200] Zhang, S., Lan, P., Li, H. C., Tong, C. X., & Sheng, D. (2022). Physics-informed neural networks for consolidation of soils. *Engineering with Computers*, *39*(4), 2845–2865.