Archives of Computational Methods in Engineering (2026) 33:789–821
https://doi.org/10.1007/s11831-025-10322-8

REVIEW

Towards Scientific Machine Learning for Granular Material Simulations: Challenges and Opportunities

Marc Fransen[1] · Andreas Fürst[2] · Deepak Tunuguntla[3] · Daniel N. Wilke[4,15] · Benedikt Alkin[2,16] · Daniel Barreto[5] · Johannes Brandstetter[2,16] · Miguel Angel Cabrera[6] · Xinyan Fan[6] · Mengwu Guo[7] · Bram Kieskamp[8] · Krishna Kumar[9] · John Morrissey[10] · Jonathan Nuttall[1] · Jin Ooi[10] · Luisa Orozco[11] · Stefanos-Aldo Papanicolopulos[10] · Tongming Qu[12] · Dingena Schott[6] · Takayuki Shuku[13] · WaiChing Sun[14] · Thomas Weinhart[8] · Dongwei Ye[8] · Hongyang Cheng[8]

Received: 2 April 2025 / Accepted: 23 July 2025 / Published online: 20 August 2025

Abstract

Micro-scale mechanisms, such as inter-particle and particle-fluid interactions, govern the behaviour of granular systems. While particle-scale simulations provide detailed insights into these interactions, their computational cost is often prohibitive. At a recent Lorentz Center Workshop on "Machine Learning for Discrete Granular Media", researchers explored how machine learning approaches can aid the development of constitutive laws and efficient data-driven surrogates for granular materials while also addressing uncertainty quantification. Attended by researchers from both the granular materials (GM) and machine learning (ML) communities, the workshop brought the ML community up to date with GM challenges. This position paper emerged from the workshop discussions. In this position paper, we define granular materials and identify seven key challenges that characterise their distinctive behaviour across various scales and regimes–ranging from gas-like to fluid-like and solid-like. Addressing these challenges is essential for developing robust and efficient models for the digital twinning of granular systems in various industrial applications. To showcase the potential of ML to the GM community, we present classical and emerging machine/deep learning techniques that have been, or could be, applied to granular materials. We reviewed sequence-based learning models for path-dependent constitutive behaviour, followed by encoder-decoder type models for representing high-dimensional data in reduced spaces. We then explore graph neural networks and recent advances in neural operator learning. The latter captures the emerging field evolution of interacting particles via efficient latent space representation. Lastly, we discuss model-order reduction and probabilistic learning techniques for high-dimensional parameterised systems, both of which are crucial for quantifying and incorporating uncertainties arising from physics-based and data-driven models. We present a typical workflow aimed at unifying data structures and modelling pipelines and guiding readers through the selection, training, and deployment of ML surrogates for granular material simulations. Finally, we illustrate the workflow's practical use with two representative examples, focusing on granular materials in solid-like and fluid-like regimes.

1 Introduction

Granular materials (GMs), ranging from beach sand to raw materials such as iron ore, are integral to various industrial processes. They play an essential role across many engineering disciplines, including geotechnical [1], coastal [2], hydraulic engineering [3], pharmaceutical [3], additive manufacturing [4–6], agriculture [7, 8], bulk handling [9–11] and robotics [12, 13].

Although simple in appearance, as illustrated in Fig. 1, GMs exhibit solid-, fluid-, and gas-like behaviour [14], making it one of the most complex materials to handle. A first-principles approach for simulating GM behaviour involves modelling the interactions between individual grains. However, this is computationally expensive. Alternatively, continuum approaches can be adopted, where constitutive laws are required to describe the material's bulk behaviour that

Marc Fransen, Andreas Fürst, Deepak Tunuguntla and Daniel N. Wilke have equal contributions as co-first authors. Hongyang Cheng is the corresponding author. All other co-authors are listed alphabetically.

Extended author information available on the last page of the article

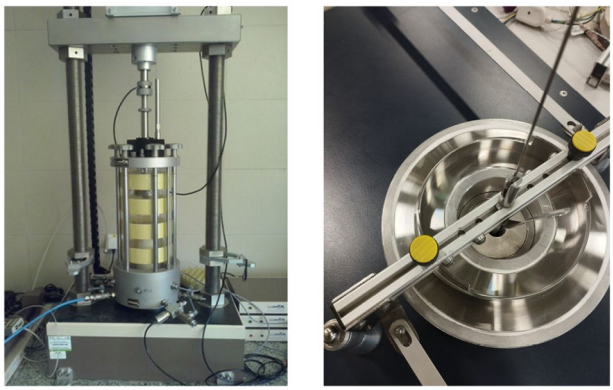
Fig. 1 Examples of granular materials of various sizes and shapes

emerges from grain-scale interactions by relating stress and strain and/or their rates under specific conditions.

For certain solid-like behaviour, existing theories from computational solid mechanics, e.g., Mohr-Coulomb, Drucker-Prager and more advanced models incorporating micro-structural information like fabric tensors [15, 16] can be utilised. Similarly, for fluid-like GM processes, such as dense granular flows, theories from computational fluid mechanics [17] provide useful frameworks. However, these models are typically crafted by "domain experts" within specialised fields and are often limited to specific conditions. As a result, they often fail to fully capture granular behaviour across the broad range of regimes and conditions encountered in practice.

It is important to recognise that these conventional approaches were developed by combining theoretical frameworks with sparse experimental data, typically obtained under highly controlled laboratory settings (e.g. Fig. 2). While they perform adequately within these limited domains, extending them beyond their calibration range becomes increasingly challenging. Unlike fluid and solid mechanics, granular mechanics lack a unified continuum theory, necessitating expensive, phenomenon-specific, multi-scale approaches to develop effective continuum models.

Current research interests towards modelling granular materials Current research focuses on (1) developing a unified theory capable of capturing GM phenomena across solid-like, fluid-like and gas-like regimes [14, 18]; (2) upscaling particle-level methods to handle the size and



(a) **(b)**

Fig. 2 Examples of laboratory devices for granular materials in **a** triaxial compression and **b** continuous ring-shear conditions

complexity of industrial-scale problems; and (3) incorporating real-world variability, including boundary condition uncertainties, measurement noise and the inherent stochastic nature of granular systems [19, 20]. From (1)–(3), a central question emerges: can we derive or learn unified constitutive theories and governing equation solvers for granular materials directly from particle-scale information, resulting in models that are not only more efficient but also provide deeper insights than contemporary, phenomenological approaches?

To address this, researchers increasingly rely on grain-scale numerical simulations, primarily employing the

Discrete Element Method (DEM), in conjunction with micro-scale observations, e.g., using the micro-Computed Tomography [21]. The discrete element method [22] (DEM) remains the most widely used technique for modelling granular systems and has demonstrated success across multiple fields, including process engineering [23], geotechnical engineering [24], bulk handling [25, 26], mining [27], chemical engineering [28] and additive manufacturing [6, 29].

From physics-based solvers to data-driven alternatives DEM simulations are information-rich, capturing particle positions, velocities and orientations, and interaction forces with other particles and geometries. This data can be subsequently mapped to macroscopic stress and strain fields, ultimately enabling the construction of closed continuum-scale models. While effective, this multi-step approach remains highly phenomenon- and regime-specific, relying heavily on specialised domain knowledge and ad hoc upscaling [30] or multi-scale coupling [31, 32] algorithms. Recall that, unlike solids and fluids, granular materials do not have a unified theory. As a result, the entire *experiments–micro–macro–continuum–model* pipeline becomes an iterative and costly endeavour. Addressing this challenge raises a new set of research questions:

1. Can data-driven approaches expedite the multi-step process of realising phenomenon-specific granular theories?
2. Can data-driven alternatives fully replace contemporary physics-based models by overcoming their inherent limitations?
3. Can data-driven methods help realise the unified theory that the GM community has long strived for?
4. More broadly, how can data-driven methods help utilise the potential of available GM data to complement our current approaches to modelling discrete media?

Over the last two decades, machine learning techniques have evolved into powerful alternatives for computational mechanics, enabling researchers to process vast amounts of data and extract deeper insights from detailed simulations (see the recent review [33]). As a result, ML has enabled researchers to perform tasks like constitutive model development [34] and surrogate modelling [35, 36], model identification [37] and design optimisation [26]. Deep neural network-based surrogates have recently attracted attention as a more universal methodology [38, 39], transforming computational fluid dynamics [40–44], weather forecasting [45–51], and molecular modelling [52–55] or protein folding predictions [56]. Beyond the ever-urgent topic of computational efficiency, neural surrogates – models that approximate the underlying physics by learning patterns and relationships directly from data – have the potential to generalise across various phenomena and characteristics, such as boundary conditions or coefficients [57–59].

One characteristic of classical physics-based numerical methods is their required conditions for numerically solving the equations [60]. Although these conditions are mitigated in these deep "neural surrogates", they still exist when looking at deep neural network-based techniques, where different deep learning architectures are prevalent across applications. This prevalence is exemplified when contrasting deep learning approaches with particle- and grid-based simulations.

Graph neural networks (GNNs) [61, 62] are a natural choice for surrogate modelling of particle-based dynamics in DEM simulations. Predicted node accelerations are often integrated numerically to advance particle dynamics in a hybrid ML-numerical fashion [36, 63, 64]. Many recent deep learning-based approaches for granular materials adopt GNN-based simulators [65–67]. For continuum models, typically solved on structured or unstructured grids, Fourier neural operator (FNO)-based [42], convolutional neural network (CNN)-based [43, 68], or Transformer-based [44, 69, 70] architectures could be suitable alternatives – assuming fixed grids for simplicity reasons. Most recently, NeuralDEM [71] has demonstrated the ability to model large-scale coupled DEM-computational fluid dynamics (CFD) simulations using transformer-based multi-branch neural operators.

Despite this promise, building large-scale neural surrogates or particle *foundation* models introduces the following concerns:

- Training such models demands extensive, high-quality datasets. Yet, the opaque nature of granular materials limits the availability and quality of experimental data, making physics-based models, such as discrete element methods (DEM) or continuum simulations, indispensable as sources of reliable training data. Moreover, industrial processes involve diverse granular phenomena, geometries, and material parameters, requiring unified training strategies to capture the underlying physics effectively.
- Interpretability and explainability remain major concerns. Domain experts need to extract and understand the learned physics to trust and apply these models. However, ML models often function as black boxes, lacking guarantees on their predictive behaviour.
- Model training is computationally intensive, sometimes comparable to GPU-parallelised particle simulations.

Additionally, as with physics-based models that require calibration, ML models suffer from performance drift and must be regularly updated as new data becomes available, thereby requiring efficient training strategies such as active and transfer learning[1].

- A final challenge concerns defining what constitutes a "good" model. Beyond accuracy and efficiency, models must maintain consistent performance under uncertainty, which is crucial when probing parameter spaces for calibration, uncertainty quantification, and design optimisation, among other digital twining[2] (DT) tasks.

Nevertheless, the confluence of computational granular mechanics and machine learning follows a similar path—much like the development of classical theories—by integrating prior knowledge with patterns learnt from complex, high-dimensional data. This implies that machine learning surrogates should be seen as complementary to, rather than replacements for, physics-based modelling advances.

2 Aim and Objectives

Contemporary physics-based approaches towards modelling GMs and systems face several challenges in providing a unified continuum theory for GMs. In this paper, we present seven key challenges that must be addressed to advance the field of granular modelling in Sec. 3.3 and explore how state-of-the-art machine learning methodologies could provide potential solutions to these challenges in Sec. 4. We aim to briefly analyse each challenge and identify corresponding ML models and algorithms that might offer viable solutions. Furthermore, we present two illustrative examples in Sec. 5 to demonstrate potential workflows and integration of numerical models and their ML surrogates. Here, the focus lies on data structures and the unification of workflows, which can be used in engineering and research. This paper is not a comprehensive review but reflects our collective viewpoint on the future direction of this emerging field. All the discussions took place during and after the Lorentz Center Workshop, "Machine Learning for Discrete Granular Media", from 29 April to 3 May 2024.

We address two primary audiences: (1) experts in computational mechanics, physics, and engineering working on GMs and (2) ML researchers interested in developing data-driven surrogates for modelling granular dynamics and related phenomena. The specific objectives of this paper are:

- To identify and articulate the key challenges faced by the GM modelling community.
- To highlight recent ML developments capable of capturing both local and global spatio-temporal responses of GMs, including their associated uncertainties.
- To act as a source of follow-up research directions, effectively establishing a collaborative workflow between computational granular mechanics and ML communities.

3 Challenges in Granular Material Simulations

3.1 Micro, Meso and Macro Scales of Granular Systems

The behaviour of granular systems is dominated by discrete micro-scale mechanisms, e.g. the interaction of particles through forces at contacts. DEM allows explicit modelling of these mechanisms across all regimes of GMs. However, the computational cost is often prohibitive, especially when considering micro-scale complexities such as particle shape variations, wide particle size distributions, or complex interaction physics, as illustrated in Fig. 1.

A common technique to reduce computational costs is *Coarse-graining* (CG) where meso-particles—computational particles representing aggregates of real particles—are used within DEM simulations. While CG lowers computational demands, it introduces new challenges, particularly in identifying equivalent mesoscopic properties to ensure that the *coarse-grained* system faithfully represents the original microstructure (Fig. 3).

Modelling industrial-scale granular systems is difficult due to the vast amount of particles involved. Continuum methods, such as the finite element method (FEM), material point method (MPM) or smooth particle hydrodynamics (SPH), are often used to reduce computational costs by avoiding explicit micro-scale modelling. These methods represent granular materials through continuum fields and partial differential equations, but this comes at the expense of phenomenological constitutive models, which may not fully capture discrete granular effects. Fig. 4 shows a micro-macro relationship between a representative volume element and a phenomenological representation of GMs, which has been sought for decades within the GM community.

Multi-scale methods attempt to bridge this gap by embedding discrete micro-scale simulations within macro-scale

[1] Active learning: a strategy to identify the most informative data points to query or label, typically in areas of high uncertainty. Transfer learning refers to reusing a pre-trained model (or parts of it) on a new but related task. A common approach involves freezing some layers of the network and fine-tuning others or adding new layers to adapt to the new data or task.

[2] Digital twins are defined as virtual representations of physical objects or systems [72] that are continuously updated from real-time data to facilitate decision-making

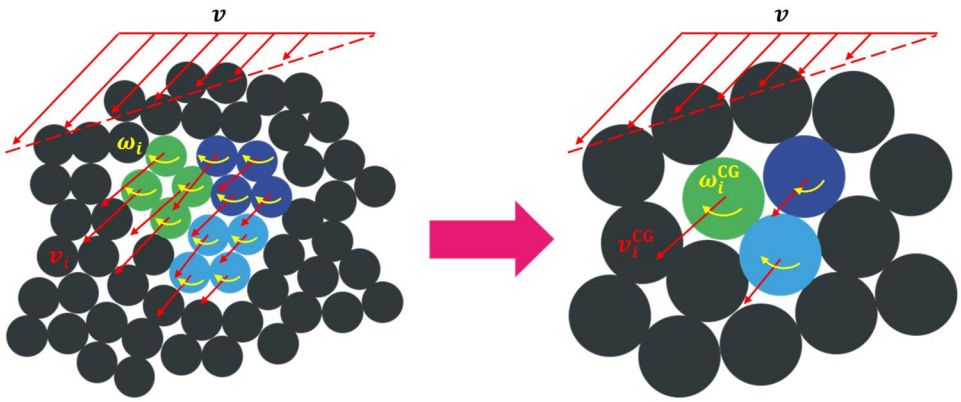


Fig. 3 Schematic of the velocity and rotation vectors on an "original" and a "coarse-grained" particle representation, subject to a continuous displacement field $\mathbf{v}$. Note that the translational and rotational degrees of freedom of each particle lead to $\mathbf{v}_i - \mathbf{v}(\mathbf{x}_i)$ or $\mathbf{v}_i^{\mathrm{CG}} - \mathbf{v}(\mathbf{x}_i) \neq 0$, accounting for the stochastic behaviour of granular systems. The position vector of particle i is denoted by $\mathbf{x}_i$

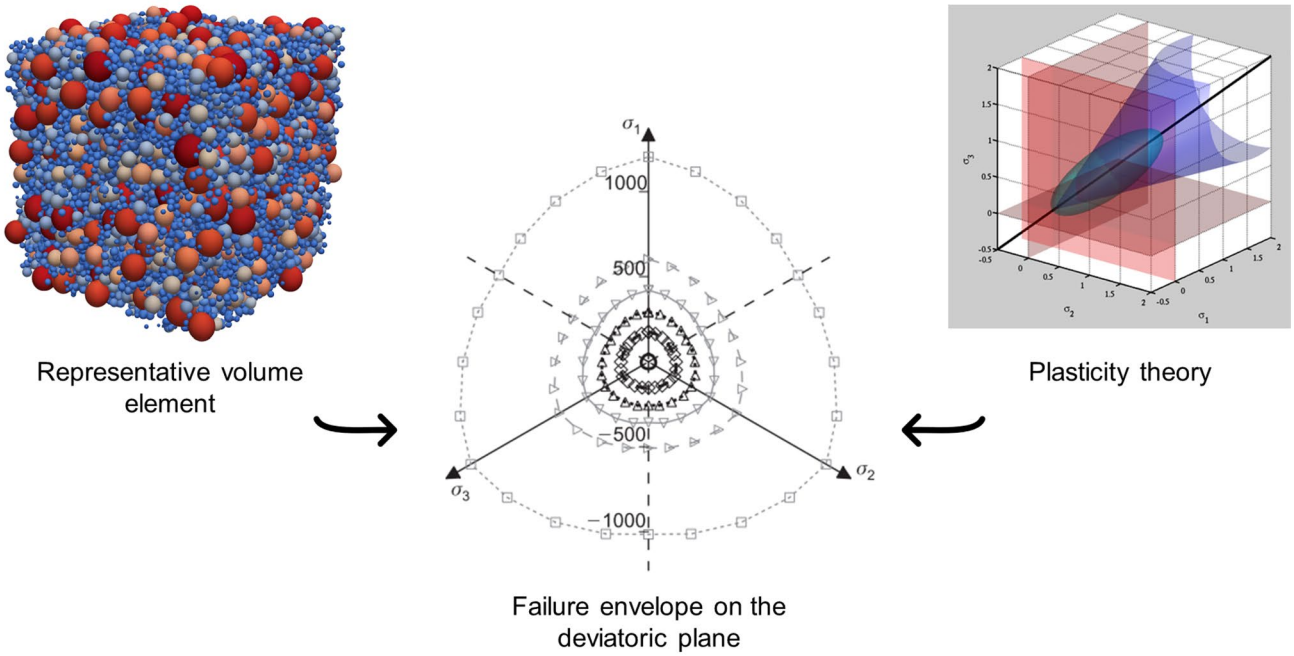


Fig. 4 Micro-macro relationship between a representative volume element (DEM) and a phenomenological representation (constitutive law) of granular materials. Figure adapted from [74]

continuum models, for instance, through replacing constitutive laws or parts of the domain with DEM simulations, see Fig. 5 a and b, respectively. Although such multi-scale methods improve physical realism, the embedded micro-scale simulations still remain the bottleneck to improve the computational efficiency. When applying the methods above to multi-physical problems, different length and time scales are encountered, e.g. when the fluid between particles has to be modelled because the length scales corresponding to micro, meso and macro levels for each physical process often differ. For example, the length and time scales for heat transfer or pore fluid flow differ from plastic mechanical deformation where strain typically localizes in the granular materials.

3.2 Physical Regimes of Granular Materials

On top of the challenges of determining appropriate length and time scales—critical for describing the relevant physics—the physical regimes of GMs often vary, resulting from evolving internal length and time scales due to external loads. This causes GMs to exhibit quasi-static (solid-like), steady-state flow (fluid-like) and dynamic (transitional) responses, as shown in Fig. 6. In the following, we briefly introduce these three physical regimes and their characteristic features, including kinetics, initial and boundary conditions, and typical geometries.

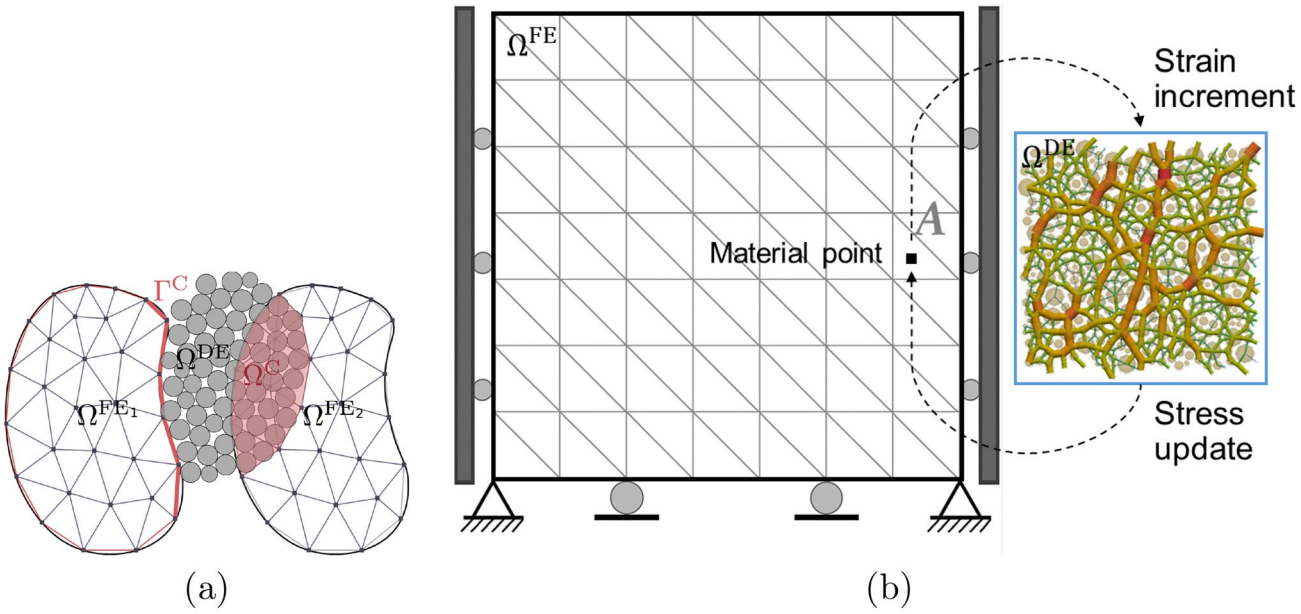


Fig. 5 Multi-scale modelling for granular materials: **a** concurrent methods where the finite element and the discrete element domains Ω^{FE} and Ω^{DE} are coupled either on an interacting surface or an overlapping volume and **b** hierarchical methods where the material's constitutive behaviour is predicted from DEM representative volume elements in Ω^{DE} to update the deformation field in the FEM domain Ω^{FE}. Figure adapted from [32, 75]

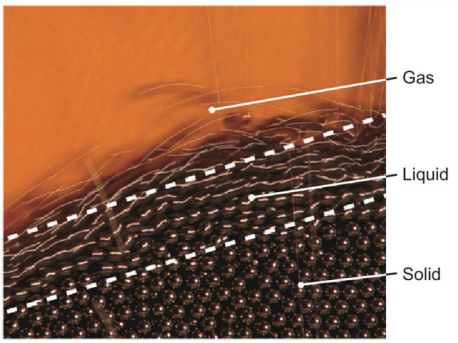


Fig. 6 Three physical regimes of granular materials. Reprinted with permission from [14]

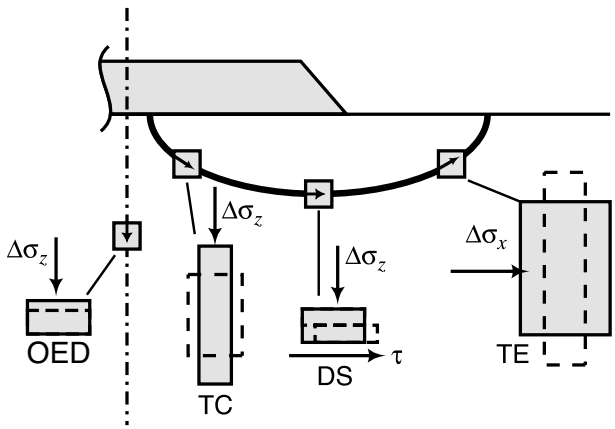


Fig. 7 Typical examples of quasi-static processes in granular materials (soil) underneath an embankment. The deformation field of a granular sample (dashed line) is controlled by the acting stress field (i.e., $\Delta\sigma_i$, τ), often captured by soil element tests such as the oedometer (OED), triaxial compression (TC), triaxial extension (TE) and direct shear (DS) tests. Figure adapted from [76]

3.2.1 Quasi-Static Behaviour of Granular Solids

Quasi-static processes are characterised by slow loading conditions, so it can be assumed that the process is in equilibrium at a given instant in time. Under these conditions, any inertial effects present in other transient and steady states are neglected. Typical examples include oedometer, triaxial and direct shear tests (see Fig. 7), which are essential for deriving material parameters later used in geotechnical design.

In a quasi-static state, loads (or stresses) may be changed to estimate (or determine) the sample deformation (or strain), or vice-versa and boundary conditions are set to reflect the process under consideration. For example, soil under an embankment may experience different boundary and stress conditions. Directly underneath the centre of the embankment, soils experience one-dimensional compression, as in an oedometer test (OED), where a cylindrical soil sample is compressed vertically under constant stress (incrementally applied) whilst deformation in other directions

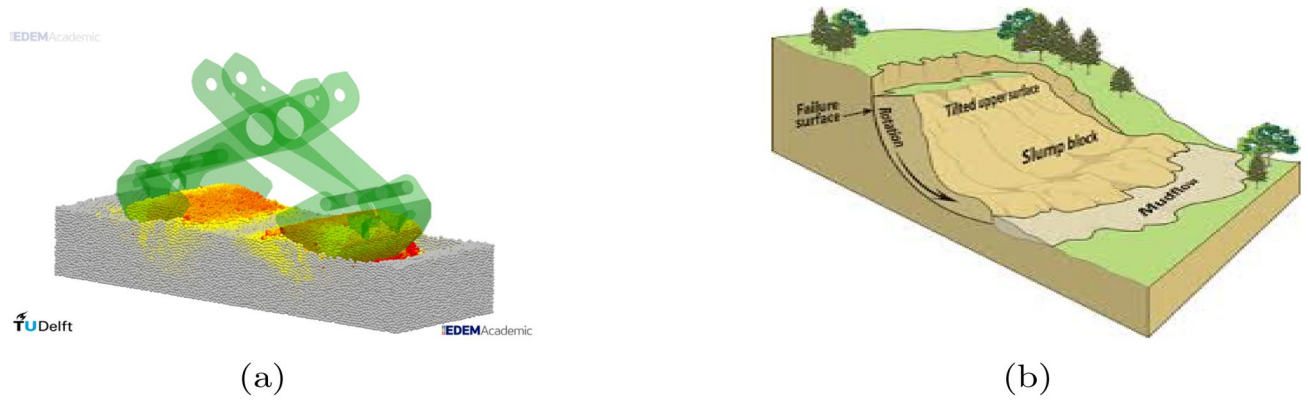


Fig. 8 Examples of dynamic granular flow processes: **a** grabbing and **b** landslide. Reprinted with permission from [9]

remains constant. Different locations along the failure plane may experience triaxial compression (TC), triaxial extension (TE) or direct shear (DS) where the combination of stresses and boundary conditions differ.

3.2.2 Steady-State Flow of Granular Fluids

Steady-state flow refers to conditions where the macroscopic properties of a granular material remain constant over time despite continuous internal movement and rearrangement of individual particles. In steady-state, properties such as shear rate, velocity profiles, and packing density reach equilibrium even though particles are still in motion and interacting. Examples of steady-state flow in granular materials include rotating drums, silos during discharge, and fluidised beds. In these systems, a stable flow is maintained under constant loading conditions. Modelling steady-state flows typically involves DEM to simulate particle interactions and, when fluids are present, a coupled DEM-CFD approach to capture particle-fluid dynamics. Such steady-state conditions are key to understanding the flowability of GMs in industrial applications like bulk material handling, where a consistent flow is essential for the efficiency of the machines and manufacturing processes.

3.2.3 Transient Regimes Between Solid-like and Fluid-like

Dynamic granular flows are non-steady, and their evolution strongly depends on timescales. Grabbing processes and landslides are representative examples (Fig. 8). These processes often involve abrupt transitions between quasi-static and flowing states.

In grabbing processes, the non-steady nature arises from the rapid manipulation or displacement of GMs. Here, timescales are crucial as they dictate the response of the material to external forces and vice versa, meaning that the grab response depends on packing density and interactions between material and equipment. Abrupt events such as dike failures and landslides are examples of dynamic granular flows at a catastrophic scale, characterised by sudden and often unpredictable shifts in particle distributions and motion due to stress accumulation, geological instabilities, or external triggers. Understanding the relevant timescales is essential for effective risk assessment, hazard mitigation, and disaster management.

From a modelling perspective, the challenges that grabbing processes (cf. Fig. 8a) pose is accurate representation of granular material-machine interactions, including the effects of real particle sizes and shapes, leading to large or even unfeasible computational expenses. Additionally, GMs exhibit varying degrees of cohesion which may fundamentally alter the rheological behaviour of the flows [77].

Similarly, modelling landslide failures (Fig. 8b) requires simulating the initiation and progression of soil, rock, and debris movement down slopes. Challenges include determining material properties, identifying failure triggers, and accurately capturing the evolution of instabilities across multiple spatial and temporal scales, often requiring CFD-DEM approaches [78]. Such models have versatile applications, from predicting failure initiation to delineating impact zones, runout areas, and flow velocities, which are crucial to assessing risks and designing mitigation measures.

3.3 Main Challenges in Granular Material Simulations

The peculiar behaviour of granular materials across scales and states, as described above, gives rise to several key challenges: closing knowledge gaps (Challenges 2, 4–6), overcoming computational bottlenecks (Challenges 1–3, 6), and enabling digital twins (Challenges 1, 6, 7).

1. **Prohibitive computational costs for large-scale DEM:** Modelling granular systems using DEM becomes computationally prohibitive as real-world problems become large. Problems with hundreds of millions of particles can be targeted with high-performance computing techniques. However, this comes at a significant computational resource cost and requires efficient parallelisation methods across CPU or GPU cores. Large-scale problems, however, can contain particles far exceeding this amount (e.g. granular flows during dike collapse or landslide and sediment transport in rivers), exceeding the bounds of current computational capabilities.
2. **Identifying effective coarse-grained representation:** For design and optimisation tasks, faster models are needed. Ensuring that the models effectively capture relevant phenomena at a larger length and time scale without excessive computational cost is essential. This necessitates the identification of coarse-grained properties and structures for meso-particles, allowing for simplified yet effective modelling within the DEM modelling framework.
3. **Expensive and repetitive DEM in multi-scale frameworks:** Multi-scale approaches require many micro-scale simulations that often undergo similar loading conditions within certain regions of the macroscopic problem. Although the difference is subtle, these simulations collectively contribute to the global phenomena like shear localisation and jamming/unjamming. Moreover, particle-scale characteristics, such as particle shape, surface roughness, and crushability, add extra complexity and computational cost to these micro-scale simulations.
4. **Separation and evolution of length and time scales:** Model resolution is crucial, as macro-scale phenomena emerge from micro-scale interactions. Understanding the extent to which micro-scale details are needed involves elucidating the relationship between the macroscopic characteristics of granular materials, including anisotropy, dilatancy and stress-path dependency, and particle-scale and microstructural characteristics.
5. **Lack of a unified continuum theory across all GM regimes:** Sec. 3.2 shows that many industrial applications involve moving structures and GMs transitioning from quasi-static to free-flowing. Describing GMs' transient behaviour is an active area of research; DEM remains the fundamental tool for obtaining relevant data to discover unified continuum theories for GMs in gas, fluid, and solid-like regimes.
6. **Coupling with multi-physics and external structures:** Many problems require integrating multi-physics solvers and handling different material boundaries and domains. These include coupling DEM with CFD or SPH, addressing the interaction between grains and pore fluids, and accommodating changing geometry and loading conditions, which are often uncertain and application-dependent. This capability is essential for accurate system simulations with important multi-physical and grain-structure interactions.
7. **Model calibration, validation, uncertainty quantification and other DT tasks:** Real-world observation data for granular materials are not only limited to the macro scale but are also often spatially sparse and noisy because of the stochastic nature of GM microstructure. This scarcity makes calibration and validation of GM models particularly challenging. Furthermore, the diverse responses of different GMs, calcareous and siliceous sands or various rock types, to external loading conditions complicate the development of generalised models. Quantifying and constraining parameter uncertainties across scales is therefore essential for enabling descriptive, predictive, and prescriptive GM digital twins.

4 Machine Learning Methods for Computational Mechanics and Engineering Problems

GM experts have traditionally focused on inferring physical laws from experiments and simulations to develop phenomenological models, often relying on deep domain knowledge and years of experience. In contrast, ML methods aim to learn features and patterns directly from data, uncovering hidden structures through statistical inference and optimisation [79].

ML encompasses a wide range of techniques for extracting knowledge from heterogeneous datasets. Classical methods in computational mechanics include supervised learning approaches such as support vector machines and Gaussian processes for regression and classification, as well as unsupervised learning techniques like principal component analysis and manifold learning for dimensionality reduction and clustering. Probabilistic methods, including Bayesian inference and stochastic process models, are particularly valuable for explicitly representing uncertainty. While classical ML focuses on interpretable, structured models, modern deep learning extends these capabilities to high-dimensional, often unstructured data via neural networks.

This section introduces key ML methods relevant to GM simulations, structured according to their functional role in surrogate modelling and digital twinning. We begin in Sects. 4.1 and 4.2 with sequence-based learning and data-driven constitutive modelling, which address the path-dependent behaviour of GMs and support interpretable surrogate construction. Sect. 4.3 introduces the encode-decode philosophy, where physical states are projected into latent

spaces to enable compact representations, exemplified by Sects. 4.3.1 and 4.3.2. Section 4.4 presents neural operator learning approaches that approximate mappings between function spaces, relevant for physics-based simulations such as CFD and/or DEM. In Sec. 4.5, we discuss data-driven reduced order modelling for high-dimensional parametrised systems. Finally, Sect. 4.6 returns to probabilistic learning and Bayesian inference, essential for quantifying uncertainty in surrogate models and enabling robust digital twins. Finally, in Sects. 4.7 and 4.8, we offer a more integrative discussion of how ML methods relate to individual challenges and outline a concept workflow towards deploying ML surrogate models in GM modelling and digital twinning.

4.1 Sequence-Based Learning for Path-Dependent Constitutive Behaviours

In engineering, particularly within materials science, system responses often exhibit path dependence; their output is dictated not just by the current state but by the entire loading history. This path-dependent behaviour, particularly evident in the elastic-plastic response of granular media, can be effectively framed as a time series problem. To accurately model such phenomena using AI, sequence-aware architectures like Long Short-Term Memory (LSTM) networks, Transformers, and Neural Ordinary Differential Equations (Neural ODEs) are crucial, given their inherent ability to capture temporal evolution and retain memory of past states [80]. Indeed, deep learning models such as recurrent neural networks (RNNs) are a natural choice for these problems, leveraging their capacity to maintain a "memory" of previous inputs and thus capture sequential and temporal dependencies in time-varying data. Previous successful applications of time-series machine learning models in elastoplasticity modelling of granular media include LSTMs [81, 82], Gated Recurrent Units (GRUs) [83, 84], and Temporal Convolutional Networks (TCNs) [85].

Effective generalisation of these models further demands diverse, physically representative datasets that cover monotonic, cyclic, and random loading paths [86]. Beyond direct sequence input, an alternative approach involves encoding the loading history through internal variables, allowing for the use of non-sequential machine learning models like Multi-Layer Perceptrons (MLPs) to predict path-dependent stress responses [87]. Recent studies employing MLPs have incorporated internal variables such as accumulated absolute strain [88] and Frobenius norm-based metrics [89].

Hybrid approaches, which integrate data-driven models with physics-based constraints or models, also significantly enhance accuracy and extrapolation in novel scenarios [90]. For example, uncertainty quantification can be performed for each AI inference. If the uncertainty exceeds a threshold, indicating that the AI model was not trained on such loading paths, then the computational task is switched to high-fidelity physics-based computations. The newly generated data from these "unseen" (or out-of-distribution[3]) paths can then be incrementally added to the training dataset, gradually improving the model's coverage of loading paths over time. Material frame indifference requires constitutive laws to be invariant under observer changes. For isotropic materials, this can be achieved using invariants. For anisotropic cases, enforcement can be done via data augmentation [91], loss constraints [92], spectral methods [93], or equivariant networks [94].

Moreover, techniques such as path encoding via latent spaces or graph representations, coupled with uncertainty estimates, are vital for understanding complex path effects and ensuring reliability [86]. However, it is crucial to acknowledge that for critical industrial applications, balancing the interpretability [34, 95], robustness [96], and speed [97] of these black-box neural network models still present significant challenges. Ultimately, this multifaceted strategy, combining advanced architectures, robust data strategies, and careful consideration of model limitations, is indispensable for deploying reliable AI in practical engineering applications.

While sequence-based models can capture history-dependent material behaviours, they often lack interpretability and physical grounding. This motivates the next step: learning constitutive laws directly from data, as explored in the following subsection.

4.2 Interpretable Data-Driven Constitutive Law Discovery

For highly complex path-dependent systems, such as granular materials, it might be feasible to build an application-specific dataset for certain BVPs. However, the expectation of enough training data, whether from physical experiments or computer simulations, to cover all possible scenarios for any models is unrealistic. As such, theory and model construction are an underdetermined inference problem [98]. To ensure robustness, adversarial attacks and knowledge graphs can be helpful tools. In adversarial attacks, one may train an artificial intelligence agent to design experiments that expose the weakness of a material model. This can be done in a deep reinforcement learning framework where the adversarial agent is rewarded by designing loading paths that maximise discrepancy [96]. The construction of knowledge graphs, on the other hand, can enhance the robustness of the models while also improving interoperability. Instead of mapping strain history to stress directly, knowledge graph

[3] Out-of-distribution data in ML refers to data points that differ significantly from the data that the model was trained on.

models require the determination of relationships among state variables and represent it with directed graphs [83, 95, 99]. A direct graph is a two-tuple $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ with node set $\mathbb{V}$ and directed edge set $\mathbb{E}$. In this case, relationships among state variables, such as void ratio, coordination numbers, and other measures, can be mathematically represented by connecting nodes with directed edges. This directed edge can represent causal relations [95] or the order of predictions that optimizes the prediction accuracy [83, 100]. While the knowledge graph does not necessarily lead to an expression simply enough for a human user to comprehend, the connectivity of the graphs does offer a set of falisiable hypotheses that can be proved or disproved by third parties.

Another popular approach to improving the credibility of machine learning models is to enhance their interpretability [101], such that the learned models achieve sufficient descriptive accuracy and relevance, which gives confidence in the accurate predictions. Murdoch et al. [101] defines two categories of tools to improve interpretability, i.e., model-based and posthoc interpretiability. In the first case, one may introduce modularity design on the machine learning models, providing insight into the inner workings of the models. For instance, Vlassis and Sun [102] uses neural networks to parametrize components of elastoplasticity models. Vlassis et al. [103] introduces semi-supervised learning to relate topological information of polycrystals to elastic stored energy. Bahmani et al. [34] leverages the neural additive method to enforce modularity by forcing the material models to be a linear combination of features of each strain component or strain invariants.

Post hoc analysis can also enhance the interpretability of machine learning models after they have been trained. For instance, Bahmani et al. [34] replaces the learned feature of the neural additive model with symbolic equations through symbolic regression [104]. Phan et al. [97] improve the expressivity of the neural additive model by introducing projection steps such that the feature space may reproduce polynomials of arbitrary orders and dimensions.

Ultimately, the training data determines the performance of a supervised ML model. Recent advancements in this area include applying active learning or deep reinforcement learning to quantify predictive uncertainty and prioritise the selection of informative strain paths, which can maximise the predictive performance of a model [100, 105, 106]. Additionally, transfer learning is proposed to facilitate cross-paradigm knowledge sharing. By fully using low-fidelity yet cost-effective data (e.g. numerical integration data from phenomenological models), the reliance on high-fidelity and expensive data, such as physical or virtual experimental data, can be minimized [35]. A similar concept of multi-fidelity can be found in the works of [107, 108], which adopted tailored multi-fidelity networks for their respective purposes.

If experimental data is not sufficient to train a predictive model, data obtained from simulations of synthesis granular assemblies may provide the necessary information to build a closure. Generating a synthesis of granular assemblies consistent with the observed data from micro-CT or SEM images can be done by generative AI, such as generative adversarial networks (GANs) or denoising diffusion probablistic models (DDPM), as shown in [109, 110]. Based on a "material datasets" derived either from experiments or simulations, the data-driven computing paradigm can employed to find the stress–strain pairs that satisfy the conservation equations at the macroscopic scale [111, 112], without the need of formulating and implementing a constitutive law. Interested readers are referred to [113] for further details.

4.3 Encode-Decode Style Models

Beyond learning explicit constitutive forms, ML can also uncover compact internal representations of physical systems in the space and time domains. This section introduces the encode-decode philosophy, where latent variables serve to bring high-dimensional inputs into feature embeddings.

As briefly stated in the introduction, recent advances in deep neural network-based learning show promise in addressing the previously described challenges in granular material modelling. Before exploring these developments in more detail, it is useful to first introduce the concept of *latent space* in deep learning.

Paraphrasing IBM Think[114], a latent space is a compact representation of data that retains only the essential aspects, capturing its underlying patterns and structure. The process of mapping data into this space is referred to as *encoding*. By doing so, encoding expresses complex data efficiently and meaningfully, enhancing the ability of machine learning models to understand and manipulate it while reducing computational requirements.

Typically, encoding data to a latent space involves some dimensionality reduction. That is, compressing high-dimensional data to a lower-dimensional space and, as a result, omitting irrelevant or redundant information. For example, the famous MNIST dataset [115] contains tens of thousands of 28x28 grayscale images of handwritten digits, each represented as a 784-dimensional vector where each dimension corresponds to a pixel value between 0 (black) and 1 (white). The vectors would be 2352-dimensional for colour images, with three values for each pixel (RGB). However, most of the image is an empty background, so reducing the vector to only the relevant dimensions (the latent space) can enhance a model's ability to process the images efficiently and accurately.

A well-known neural network architecture for encoding, compressing, and reducing data to a latent space is the autoencoder, as shown in Fig. 9a. A self-supervised model

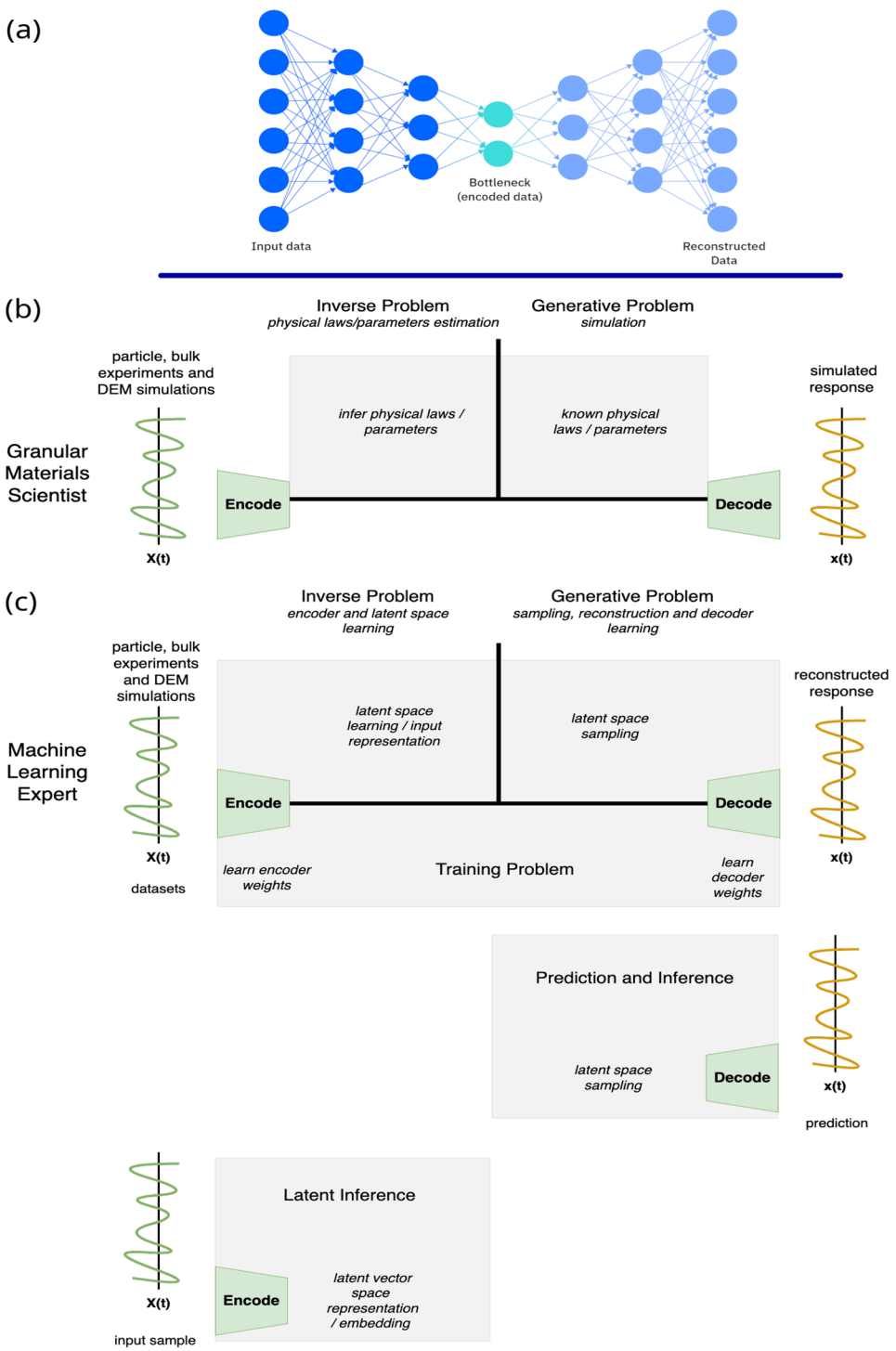


Fig. 9 A dual encode-decode perspective of granular material modelling and machine learning. **a** illustrates a simple autoencoder architecture, highlighting the encode-decode philosophy in deep learning. **b** presents the granular material scientist's perspective, focusing on model parameter estimation from particle and bulk measurements, DEM simulations, $X(t)$, and generating simulated responses using known parameters. **c** the ML expert's perspective is showcased, emphasising the training problem through model weights and latent space learning from datasets, with response reconstruction through latent sampling and decoding. It further illustrates the prediction and inference problems, where encoded input samples are decoded to predict and understand the response $x(t)$, alongside latent inference that encodes input samples into a latent parameter space for interpretation

aims to compress input data through dimensionality reduction and reconstruct the original data from this compressed form. In a typical autoencoder, the encoder consists of layers with progressively fewer nodes, compressing the input as it passes through each layer. The decoder then uses the compressed latent vector to reconstruct the original input. Hereby, we extract the most important features of the data and effectively learn the latent space of the input. Modelling a latent space is integral to several state-of-the-art deep learning algorithms in generative computer vision, language modelling, etc.

Therefore, given sufficient experimental and numerical data, both GM and ML communities are confronted with the same fundamental challenge: how to represent

(encode) and infer hidden laws or models from data into abstract mathematical representations. The data must contain meaningful instances of the relevant physics to be effectively learned. Once this knowledge is encoded/captured, it can be generalised or decoded to predict unseen scenarios. Fig. 9 illustrates this dual encoding-decoding perspective for both GM and ML experts.

4.3.1 Generative Latent Variable Models for Interpretable Representations

Generative latent variable models (LVMs) are mathematical frameworks used in computational mechanics to represent high-dimensional data using lower-dimensional latent spaces. These models establish mappings between latent variables and observable data, facilitating dimensionality reduction, data generation and analysis of mechanical systems. In computational mechanics, several categories of generative models have gained prominence.

Variational autoencoders (VAEs) are a category that defines probabilistic encoders and decoders, mapping between data and latent spaces. VAEs have found applications in dimensionality reduction for fluid dynamics and materials science [116]. Another significant category is GANs, which consist of generator and discriminator networks trained adversarially. GANs have been successfully applied to generate microstructures, flow fields and material property distributions [117]. Flow-based Models represent a third category, employing sequences of invertible transformations between latent and data spaces. These models have proven useful in molecular dynamics and turbulence modelling [118].

Creating interpretable latent spaces is a key focus in computational mechanics, with several approaches being developed. Disentanglement techniques [119], such as β-VAE, aim to learn latent variables corresponding to independent variation factors in the data [120]. Physics-informed Latent Spaces take a different approach, incorporating physical constraints or prior knowledge into latent space representations [121]. Manifold Learning methods explicitly model the underlying data manifold, including geometric deep learning approaches [122]. The importance of interpretable encoding has been demonstrated using statistical learning techniques to encode complex discrete element simulation data into more manageable and insightful representations [123]. Considering time as an untangled latent dimension known as temporal-preserving latent variable models, [124] highlights the importance of utilising some manifold coordinates rather than solely relying on data space indicators like reconstruction errors to enhance interpretable latent spaces.

4.3.2 Graph Neural Network Surrogates for Granular Systems

Deep learning surrogates have emerged as powerful frameworks for replicating complex physical systems, offering a computationally efficient alternative to traditional simulations [125]. These models directly approximate the underlying physics by learning patterns and relationships from data obtained by simulations. Once trained, surrogate models can provide predictions at significantly lower computational costs, circumventing the high expense associated with traditional solvers. Graph neural networks (GNNs) [61, 62] provide an effective framework for modelling the dynamics of granular materials and particle systems. By representing granular assemblies as graphs, with nodes corresponding to particles and edges encoding their interactions, GNNs can learn the underlying interaction laws governing particle motion in a permutation-invariant manner.

One instance of applying GNNs for granular flow modelling is the Graph Neural Network-based Simulator (GNS) [63]. GNS takes the current state of the granular domain (particle positions, velocities, boundaries) as input and predicts the acceleration of each particle, which is then numerically integrated to model the time evolution of the particles. Trained on high-fidelity simulations, GNS demonstrates accurate predictions while being orders of magnitude faster than traditional numerical methods. Applications of GNS for granular systems involve granular column collapse [65], granular flows with complex boundaries [36] and solving inverse problems [66, 67, 126].

In recent years, geometric deep learning [127] on graphs has investigated many directions to incorporate knowledge on domains, signals, and transformation groups into network designs. For example, equivariance has emerged as a powerful inductive bias for GNNs when applied to dynamical systems [128–130]. Such equivariant GNNs leverage geometric information of nodes while maintaining equivariance w.r.t. to rotations, translations, and reflections, the fundamental symmetries that many physics laws are based upon. However, a major drawback of GNNs is that their computational complexity grows with the number of particles as the number of nodes in the graph grows accordingly. Thus, computational complexity becomes infeasible for an increasing number of particles [44, 131]. There are many approaches for scaling GNNs to larger granular systems, including evaluating subgraphs and combining the solutions [132], limiting the receptive field of the neural network and applying domain decomposition [131, 133] and using a hierarchy of coarser graphs [134–136]. Another approach is to reduce the problem's dimensionality, as demonstrated in [137], which uses PCA on the data before training a GNS.

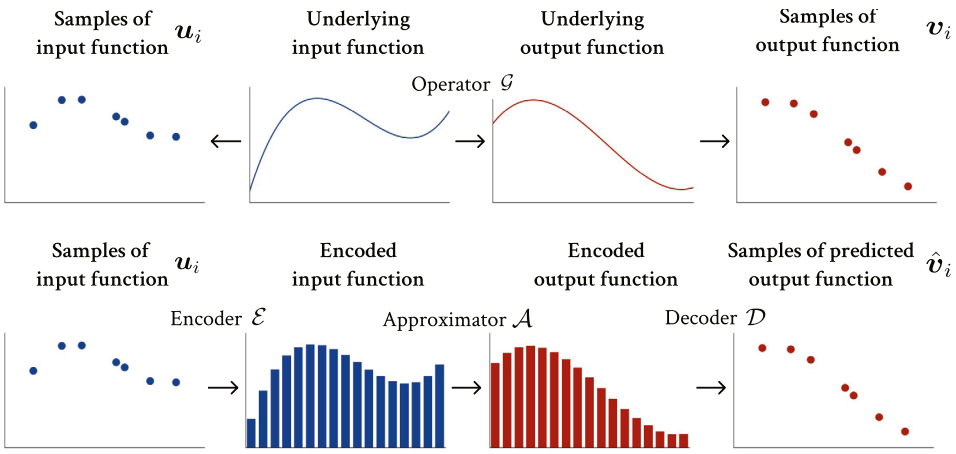


Fig. 10 Illustration of neural operator learning. (Top row) The ground truth operator $\mathcal{G}$ maps an input function to an output function. However, we only have access to samples of these functions – for example, particle velocities at their respective positions in a DEM simulation. (Bottom row) The neural operator $\hat{\mathcal{G}}$ approximates the ground truth operator $\mathcal{G}$ using three maps, an encoder $\mathcal{E}$, an approximator $\mathcal{A}$, and a decoder $\mathcal{D}$. The approximation of $\mathcal{G}$ is ideally independent of the number of sampled input points and can approximate the output function for an arbitrary number of points. (Reprint with permission from [71])

4.4 Neural Operator Learning for Mapping Between Function Spaces

Whereas the encode–decode approaches rely on latent embeddings or structured connectivity, neural operator learning [42, 44, 138–140] represents a formulation that offers architecture designs for learning mappings between function spaces, making them well-suited for approximating solutions of partial differential equations. Neural operators have been formulated for graph neural network layers [139], Fourier neural operator layers [42], convolution neural layers [68], or transformer layers [44, 70]. Following [140], we assume $\mathcal{U}, \mathcal{V}$ to be Banach spaces of functions on compact domains $\mathcal{X} \subset \mathbb{R}^{d_x}$ or $\mathcal{Y} \subset \mathbb{R}^{d_y}$, mapping into $\mathbb{R}^{d_u}$ or $\mathbb{R}^{d_v}$, respectively. Operator learning is defined to approximate a ground truth operator $\mathcal{G} : \mathcal{U} \to \mathcal{V}$ via a neural network $\hat{\mathcal{G}} : \mathcal{U} \to \mathcal{V}$. To train a neural operator $\hat{\mathcal{G}}$, a widely used approach is to construct a dataset of discrete data pairs $(\boldsymbol{u}_{i,j}, \boldsymbol{v}_{i,j'})$, which correspond to $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ evaluated at spatial locations $j = 1, \ldots, K$ and $j' = 1, \ldots, K'$, where K and K' can, but most not be equal. This is shown in the top part of Figure 10. On this dataset, $\hat{\mathcal{G}}$ is trained to map $\boldsymbol{u}_{i,j}$ to $\boldsymbol{v}_{i,j'}$ via supervised learning, as sketched in the bottom part of Figure 10. $\hat{\mathcal{G}}$ is composed of three maps [44, 141]: $\hat{\mathcal{G}} := \mathcal{D} \circ \mathcal{A} \circ \mathcal{E}$, the encoder $\mathcal{E}$, the approximator $\mathcal{A}$, and the decoder $\mathcal{D}$. First, the encoder $\mathcal{E}$ transforms the discrete function samples $\boldsymbol{u}_{i,j}$ to a latent representation of the input function. The approximator $\mathcal{A}$ then maps the latent representation of the input function to a representation of the output function. Lastly, the decoder evaluates the output function at spatial locations j'. The neural network $\hat{\mathcal{G}}$ is then trained via gradient descent, using the gradient of, e.g., a mean squared error loss in the discretized space $\mathcal{L}_i = \frac{1}{K'} \sum_{j'} \|\hat{\boldsymbol{v}}_{i,j'} - \boldsymbol{v}_{i,j'}\|_2^2$, where $\|_2$ is the Euclidean norm.

An important property of a neural operator is discretisation convergence, which ensures that the solutions produced by the neural operator remain invariant with respect to the number of input samples K and the number of output function samples K'. To ensure discretisation convergence, commonly used architectures for the encoder $\mathcal{E}$ include graph neural networks (GNNs) [139, 142], transformers [69, 70, 143, 144], or a hybrid approach combining both, as demonstrated in [44]. Once the encoded input function has been sufficiently sampled for accurate representation, any further increase in the number of input samples K should not affect it. This aligns with growing evidence that neural networks can effectively capture physical phenomena while requiring significantly less fine-grained discretisation compared to traditional numerical methods [41]. For the decoder $\mathcal{D}$, recent work [145] has shown that formulating the decoder as a conditional neural field allows for independent point-wise evaluation, thereby ensuring discretisation convergence.

Most state-of-the-art neural operator approaches are primarily designed for geometrically simple domains with regular grids, leaving neural operator formulations for particle-based dynamics relatively under explored. Recently, Universal Physics Transformers (UPT) [44] introduced a flexible and scalable neural operator designed to work with irregular grids and/or particle systems. Operating without grid- or particle-based latent structures enables flexibility and scalability across meshes and particles. The diverse applicability and efficacy of UPTs have been demonstrated in both mesh-based fluid simulations and Lagrangian-based fluid dynamics. Building

upon the UPT framework, [71] proposes a multi-branch neural operator capable of processing multi-physics quantities. Their approach demonstrates strong performance in hopper simulations with 250k particles and fluidised bed reactors comprising 500k particles and 160k CFD cells, achieving faithful modelling over trajectories of up to 28 seconds, equivalent to 2800 machine learning time steps.

4.5 Data-Driven Reduced-Order Modelling

While neural operator learning rely on large-scale deep neural networks, reduced-order modelling (ROM) [146–148], alternatively termed model order reduction, offer efficient strategy for accelerating numerical simulations of parametrised problems in computational mechanics and hence to enable multi-query, real-time computations. Generally speaking, ROM techniques reduce problem complexity by intelligently representing high-dimensional dynamical systems in low-dimensional latent spaces with controlled accuracy, thereby substantially improving computational efficiency. A ROM scheme typically features an offline/online decomposition, in which a reduced model is constructed offline from a collection of full-order solution data covering a time-parameter domain of interest. Such a low-dimensional system is evaluated online for new time instances and parameter configurations. The parameters of ROM are not limited to the physical features of the system, e.g. material properties or boundary/initial conditions [147, 149–151], but also the geometric parameters that describe the configuration of the computational domain [147, 152–154].

Consider a high-dimensional system $\dot{\mathbf{u}} = f(\mathbf{u};\boldsymbol{\mu})$ for the full-order state vector $\mathbf{u} \in \mathbb{R}^N$ characterised by d parameters $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^d$ varying in a parameter space $\mathcal{D}$, the offline construction of an r-dimensional reduced-order model ($r \ll N$) mainly entails two steps. The first step is to learn a time/parameter-independent encoding-decoding structure [155, 156] for dimensionality reduction (or manifold learning) [79, 157]. In particular, such a structure is written as follows

$$\mathbf{u} \approx \mathcal{F}_d \circ \mathcal{F}_e(\mathbf{u}), \tag{1}$$

in which $\mathcal{F}_e : \mathbb{R}^N \to \mathbb{R}^r$ is an encoder that maps to the reduced space and $\mathcal{F}_d : \mathbb{R}^r \to \mathbb{R}^N$ is a decoder recovering back to the full space. A classical treatment for this is the proper orthogonal decomposition [147, 158, 159] that takes advantage of linear principal component analysis [157]. More recently, deep autoencoders are becoming popular for the discovery of low-dimensional latent spaces [149, 155, 160, 161] and they are powerful tools for compressing high-dimensional solution data that represent complex mechanical behaviours in granular modelling.

The second step aims to represent reduced-order dynamics in the low-dimensional latent space. Towards this end, a

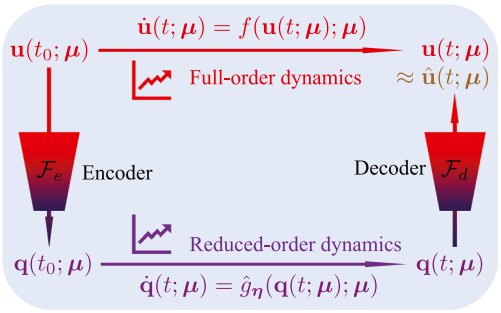


Fig. 11 A conceptual diagram for data-driven reduced-order modelling

'projection' of the full-order system onto the reduced space can be written as

$$\dot{\mathbf{q}} = [\nabla \mathcal{F}_d(\mathbf{q})]^{\dagger} f(\mathcal{F}_d(\mathbf{q});\boldsymbol{\mu}). \tag{2}$$

However, the assembly of such a reduced system is inconvenient in granular simulations, primarily because its formulation requires intrusive access to the full-order code that often involves complex, nonlinear constitutive laws. To overcome this difficulty, a straightforward strategy is to construct generic surrogate models directly for the reduced-state solution $\mathbf{q}(t;\boldsymbol{\mu})$ via, e.g. neural networks [150, 160, 162], polynomial response surfaces [163], radial basis function interpolation [164], or Gaussian processes [165–168]. An example of this reduced-state solution is illustrated in Fig. 11. Such a 'black-box' treatment can conveniently handle the interpolation of parameter dependency within the time domain covered by training data. Still, there is no guarantee of generalisation performance in predicting for future states. Alternatively, one can approximate the right-hand side of the reduced system with certain model parametrisation $\hat{g}_{\boldsymbol{\eta}}$, i.e.,

$$\dot{\mathbf{q}} = \hat{g}_{\boldsymbol{\eta}}(\mathbf{q};\boldsymbol{\mu}), \tag{3}$$

and pose the data-driven learning of reduced-order equations as the parameter estimation for $\boldsymbol{\eta}$ — an inverse problem [169]. When prior knowledge about the granular dynamics and/or their numerical implementation is incorporated into the model parametrisation $\hat{g}_{\boldsymbol{\eta}}$, improved temporal predictiveness can be seen when the learned low-dimensional system is evaluated beyond training coverage. The parameter estimation in reduced-order dynamics can be accomplished with various strategies, including data-driven operator inference [170–172], sparse identification of latent dynamics [155, 173, 174], neural ordinary differential equations [175], surrogate modelling of Runge-Kutta schemes [176] and recurrent learning for reduced-state time series [177, 178].

4.6 Probabilistic Framework for Uncertainty Quantification

With surrogate models introduced, we now turn to the challenge of quantifying their uncertainty. Probabilistic learning frameworks, particularly Bayesian methods, provide tools for uncertainty estimation and risk-aware inference. In computational science and engineering, we frequently explore how these uncertainties propagate from observational data and characterising input features typically defined within mathematical/numerical models to the predictions produced by these models. This is particularly challenging for granular materials due to their evolving microstructure, representative scales and physical regimes (see Sect. 3.3). Such complexities restrict observations to a single scale or regime at any given time.

Adding uncertainties to the state $\mathbf{u}$ and parameters $\boldsymbol{\mu}$ of a suitable mathematical/numerical model $\mathbb{F}$ describing the physical process results in the following state-parameter space model,

$$\begin{aligned} \mathbf{u}_t &= \mathbb{F}(\mathbf{u}_{t-1}, \boldsymbol{\mu}_t) + \boldsymbol{\nu}_t, \\ \mathbf{v}_t &= \mathbb{H}(\mathbf{u}_t) + \boldsymbol{\omega}_t. \end{aligned} \tag{4}$$

where $\boldsymbol{\mu}_t$, $\mathbf{u}_t$ and $\mathbf{v}_t$ are random variables describing uncertainties of the model parameters, states and observations, respectively. $\mathbb{H}$ represents a mapping between hidden state and observable vectors, and $\boldsymbol{\nu}_t$ and $\boldsymbol{\omega}_t$ are the modelling and measurement uncertainties, respectively, often defined to be zero-mean normal distributions. (4) describes the time evolution of the state of the model $\mathbf{u}_t$ and the state of the observables $\mathbf{v}_t$. The state/parameter estimations for $\mathbf{u}_t$ and $\boldsymbol{\mu}_t$ are often performed using Bayesian inference.

Bayes' rule provides a general framework for estimating the hidden state of a dynamical system from partial observations using a predictive model of the system dynamics $\mathbb{F}$. Bayesian statistics offers a methodology for quantifying the impact of various uncertainty sources and enables a probabilistic integration of prior knowledge about physics and mechanics into the inference process. The state, often augmented by the system's parameters as shown in Eq. (4), changes in time according to an iterative process of predicting and updating. Within the context of Bayesian uncertainty quantification [179], the probability distribution of the system's state is updated whenever new observations become available, a process also known as "data assimilation". In the context of material uncertainties, the system dynamics are predicted by a material model, and the observations are limited experimental data. The goal is to estimate the probability distribution of the model's parameters from the experimental data using simulations of material responses [180].

Probabilistic and Bayesian learning are powerful frameworks that provide a structured way to integrate model prediction and sparse, real-world observation data. They are critical to risk assessment, uncertainty quantification, and optimisation of GM processes and design. However, because GMs show strong nonlinearity and discontinuity, the estimation of the uncertainties are usually achieved via extensive sampling of the model solutions, e.g. via Markov-Chain Monte Carlo or other variations of the Monte Carlo sampling schemes [181, 182]. The purpose of using machine learning for these problems is two-fold: (1) to reduce the computational cost of the model evaluations via surrogate modelling (e.g. Gaussian process regression and responses surfaces) and (2) to learn the underlying conditional probability distribution via cluster algorithms [79] and making use of it to improve the efficiency of the sampling scheme. These data-driven surrogates can be integrated within the Bayesian framework in conjunction with physics-based models, as schematized in Fig. 12 and implemented in the open-source Bayesian uncertainty quantification software package GrainLearning [19].

In particular, Gaussian processes (GP) emulation is a powerful tool of probabilistic supervised learning for surrogate modelling with uncertainty quantification (UQ) [157, 183]. It is a generally applicable regression method in approximating maps between characterising input features and output quantities of interest (QoI) [151, 165, 184, 185]. GP regression is a non-parametric Bayesian approach that considers a GP prior over regression functions and then estimates the posterior predictions by conditioning on the observational data. The posterior mean function can be viewed as an approximation in a reproducing kernel Hilbert space [183], which enables effective treatment of nonlinearities. Additionally, the predictive distribution facilitates the function approximation with quantifiable uncertainties, hence certifying the data-driven model's credibility. Knowledge about governing physics can be incorporated into GP surrogate models by leveraging the property that Gaussianity can be preserved through linear operations. This allows to integrate differential equation constraints into the regression problem and has been applied to forward and inverse problems governed by linear equations [186, 187] and extended to nonlinear problems [188, 189].

4.7 Identifying and Classifying Machine Learning Solutions for Granular Material Challenges

Given the strong prospect for ML in modelling granular materials, Sects. 4.1–4.6, below we attempt to present possible ML-based solutions for the earlier stated challenges, Sect. 3.3, as follows.

1. **Prohibitive computational cost for large-scale problems with DEM:** A typical DEM computation cycle

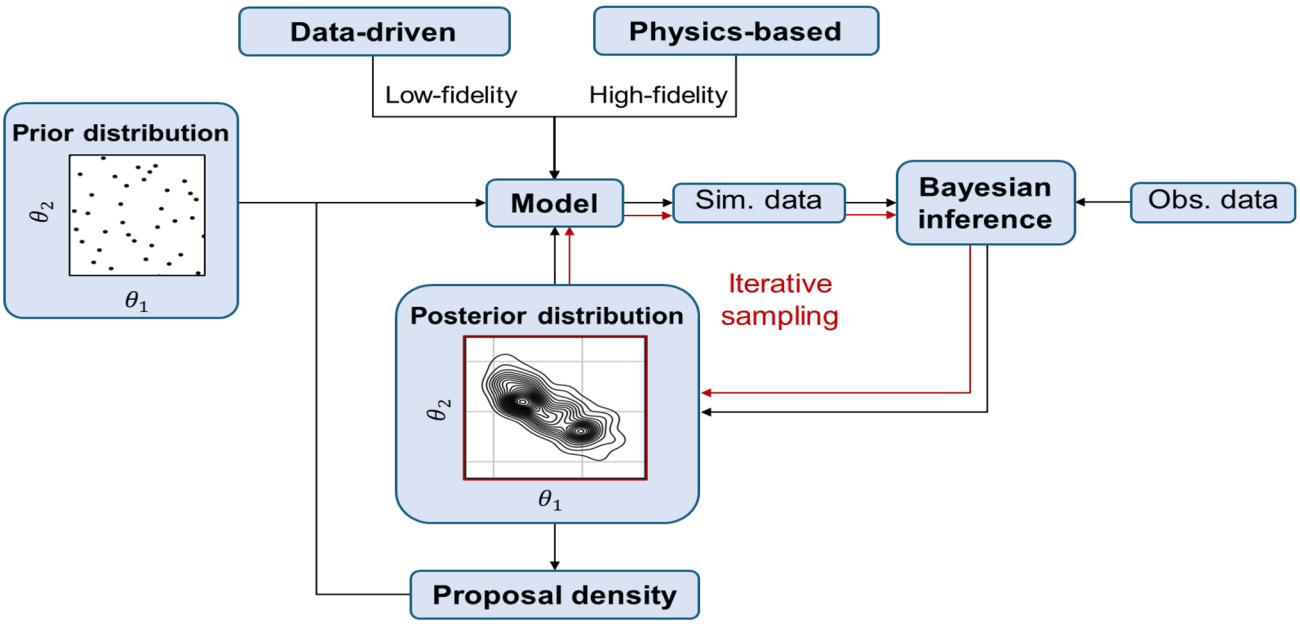


Fig. 12 Integration of data-driven and physics-based models for iterative Bayesian estimation of parameter ($\boldsymbol{\theta}$) uncertainty [19]

involves: (1) contact detection, (2) interaction force/torque calculation, and (3) particle dynamics updates. These computations are expensive due to the interactions' nonlinear and discontinuous nature. Machine learning offers a potential solution by creating surrogate models that learn a lower-dimensional representation of particle dynamics. Neighbour-aware methods, such as convolutional neural networks (CNNs) and GNNs, can learn local connectivity between particles and boundaries [190]. Moreover, integrating these approaches within DEM may lead to hybrid models has a better trade off between accuracy and speed. For example, CNNs and GNNs struggle with capturing dynamic contact changes, which requires careful validation against full DEM models to quantify accuracy. Other data-driven approaches include GNN-based neural surrogates [63, 65] and neural operators [44, 71], which learn end-to-end mappings from DEM data, bypassing the need for hand-crafted features, but may require substantial datasets to achieve generalisation.

2. **Identification of coarse-grained particles and structures for meso-scale representation:** Coarse-graining reduces computational complexity by representing clusters of particles or microstructures at a higher level, akin to molecular dynamics. Techniques like graph coarsening with GNNs can preserve essential mechanical properties by intelligently aggregating nodes while maintaining the connectivity patterns [191, 192]. However, achieving a balance between simplification and mechanical fidelity would still be challenging. Benchmarks must be in place to assess whether the mechanical properties like stiffness, friction and viscosity are maintained across various coarse-grained representations.
3. **Repetitive, still expensive micro-scale simulations within a multi-scale modelling framework:** In multi-scale modelling, the repetitive nature of micro-scale simulations can be a significant bottleneck. To address this, simple RNNs and MPLs have been utilised to represent the macroscopic, time-varying response of granular materials obtained from DEM or FEM simulations [185, 193, 194]. However, because these models ignore grain-scale interactions, their utility for out-of-distribution data is limited. One promising direction is the integration of GNN surrogates within a hierarchical multi-scale framework. This allows capturing the intricate micromechanics leading to plasticity while overcoming GNN's scalability issue [195]. Additionally, clustering and classification algorithms [196] can also be employed to identify regions of the materials that undergo similar load conditions, significantly reducing the number of representative volume element (RVE) problems to be solved.
4. **Separation and evolution of length and time scales:** The transition of granular material behaviour between solid-, fluid-, and gas-like is rooted in the evolving length and time scale, as their microstructure responds dynamically to external loads. Representation learning and other pattern recognition methods can potentially help extract relevant scales from data [197–199]. Notably, recent advances in the "data-driven mechanics" par-

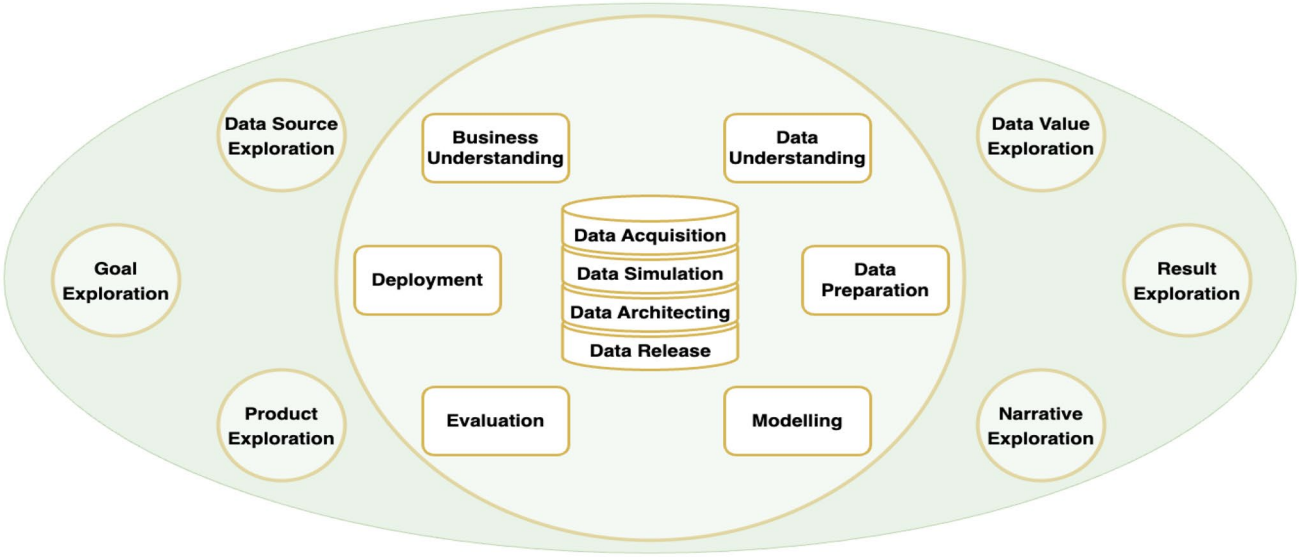


Fig. 13 Illustrates a variety of components of a data science trajectory (DST) [207] map, essential to any data science project

adigm [111, 112, 200] show an alternatively approach where the internal length no longer need to be determined to define certain constitutive laws but are learned directly from the so-called "material datasets", identified from experiments or micromechanical simulations.

5. **Lack of a unified continuum theory across all GM regimes:** At the meso scale, ML methods can be used to discover constitutive laws for granular materials [34, 102, 201], but these models are limited to a solid mechanics framework. The lack of continuum theories for GMs in the fluid-like regime remains the major hurdle against applying ML-based inference or optimisation to dicover unified continuum theories. Data-driven neural operators such as UPT [44] and geometry-informed neural operators (GINO) [202] are promising alternatives that bypass fixed theoretical assumptions by directly learning the macroscopic fields from data.
6. **Coupling to multi-physics and external rigid/deformable structures:** In many industrial applications, granular materials interact with other physical systems, making the data highly heterogeneous. Data-driven model order reduction or multi-branch neural operator learning are promising techniques for building cheap surrogate models for fluid-particle or fluid-structure coupled systems [71, 203–205]. A major technical challenge is to ensure consistency across these data belonging to different physics with their own time and spatial scales. Attention mechanisms in neural networks could be particularly helpful in focusing on the most relevant interactions in specific physical domains.
7. **Model calibration, validation, uncertainty quantification and other DT tasks:** In DT, especially when using ML-based surrogates, it is crucial to incorporate error or uncertainty estimates to ensure that the quality of inference matches that of the full-order model [168]. This is particularly important for GMs, where real-world data are not only sparse but also noisy. Probabilistic and Bayesian learning frameworks are therefore essential for assimilating such imperfect data into GM models. To maintain computational feasibility in high-dimensional settings, advanced techniques such as sparse Gaussian Processes or variational inference can be employed [206]. Lastly, because DT may often handle "edge cases"–rare or critical operating conditions–active learning strategies can be used to selectively acquire the most informative data that reduce model uncertainty and improve generalisation [180].

4.8 Proposed Machine Learning Workflow for Granular Material Simulations (GranML)

The previous subsection shows that a matchmaking between ML methods and GM challenges should lead to an extraction and utilisation of knowledge (either universal or domain/application-specific), driven by the underlying research questions. The two main reasons motivating this extensive academic endeavour are (i) a dramatic increase in the availability and the amounts of data and (ii) a boom in the variety of machine learning techniques. Given ML's potential, we also see new methodologies being proposed to extract this knowledge systematically [207]. Figure 13 illustrates a data science trajectory (DST) map, which is an extension of CRISP-DM [208] – a *de facto* standard for data mining and knowledge discovery projects. Based on

the definitions of each component of the data science trajectory (DST) (Fig. 4 in [207]), Sect. 5 showcases concept machine learning workflows for a few example applications. We use the word *concept* because of the multiple possible arrangements of the DST components.

Our concept workflow primarily consists of three stages, namely **data simulation, understanding, and preparation**, **modelling**, and **evaluation**. Each stage is critical to the overall model development and deployment process. To assist readers, particularly those with an engineering background, we briefly describe the key components of this workflow as follows.

- The simulation workflow begins with **data preparation and understanding** using physics-based granular material models. This step typically involves selecting appropriate spatial and temporal resolutions, preprocessing the data to obtain quantities of interest, and conducting sensitivity analyses across the sampling domain (e.g., time, parameters, boundary conditions) to generate datasets.
- The **modelling** stage involves selecting and training suitable ML models (as discussed in Sec. 4), including hyper-parameter tuning and retraining as new data becomes available. Techniques such as transfer learning and active learning can improve efficiency. Typically, datasets are split into training, testing, and validation subsets to ensure good generalisation to unseen data.
- In the **evaluation** stage, trained models are scored and assessed, leading to the **deployment** phase. Deployment involves integrating ML models with real-world sensing data to perform tasks such as uncertainty quantification, optimisation, and data assimilation. The significant speed advantage of ML surrogates over physics-based simulations makes gradient- and sampling-based techniques practically viable within this workflow.

This workflow ultimately supports the digital twinning of granular systems. A complete digital twin in granular systems operates through two parallel tracks, one focused on **modelling** and the other on **sensing**. The sensing track estimates hard-to-measure system variables, such as local forces, flow fields, or packing densities, by combining sparse measurements with inference techniques. This dual role of sensing involves both **digitising the physical world** through direct measurement of observable variables and applying **virtual sensing** to infer unmeasured or unobservable states, effectively extending the observable space. The modelling track integrates these direct and inferred quantities to construct a digital replica that evolves alongside the physical system. Crucially, a digital twin supports a **bi-directional data flow**. Not only does sensor data update the model in real time, but the model also feeds back insights, predictions, and control recommendations to the physical system. **Descriptive** digital twins enable accurate state reconstruction; **predictive** digital twins facilitate forecasting of behaviours such as clogging or segregation; and **prescriptive** digital twins empower simulation-driven decisions to adapt operations, optimise performance, or mitigate failure risks.

5 Demonstrations of the GranML Workflow for Granular Material Simulations

The main challenges in granular material simulations, summarised in Sect. 3.3, are conventionally tackled through advanced numerical methods and theories. By integrating numerical simulations with ML techniques, hybrid approaches are developed (see Sect. 4.7). These are exemplified by the following cases aimed to address at least Challenges 1, 2, 5, and 7 of Sect. 3.3: the quasi-static compression and shearing of sand in the solid-like regime (Sect. 5.1) and the transient behaviour of granular column collapse in the fluid-like regime (Sect. 5.2). The section concludes with a set of recommendations to facilitate the use of ML in granular material simulations.

5.1 Modelling Quasi-Static, Solid-like Behaviour of Granular Materials

5.1.1 Compression and Shearing of Densely Packed Grains

As discussed in Sect. 3.2.1, GMs at rest or moving slowly can often be considered in a quasi-static regime. Common engineering examples include powders, soils, and rocks compaction, provided the external loads are not rapidly applied. In such cases, the RVE scale is of interest, where individual grains can be simulated as rigid bodies interacting via contact forces, Coulomb friction, cohesion, and other mechanisms.

A typical scenario is triaxial compression (see Fig. 14). Here, a granular assembly comprising several thousand particles (often with varying shapes and sizes) is compressed to a confining pressure of a few kPa, mirroring subsurface conditions. Despite many studies still employing spherical particles for simplicity, advanced algorithms also accommodate non-spherical shapes [209] and more sophisticated contact laws [210–213].

These quasi-static simulations capture numerous snapshots of a system's microstructural evolution (e.g., particle rearrangements, creation and loss of contacts). The resulting data enable researchers to derive macroscopic properties such as bulk anisotropy, dilatancy, and the evolution of macroscopic Quantities of Interest (QoIs), including the stress tensor $\boldsymbol{\sigma}$ (especially their invariants including pressure p and deviatoric stress q) and excess pore pressure u, homogenized over the RVE domain. At the RVE scale, the primary

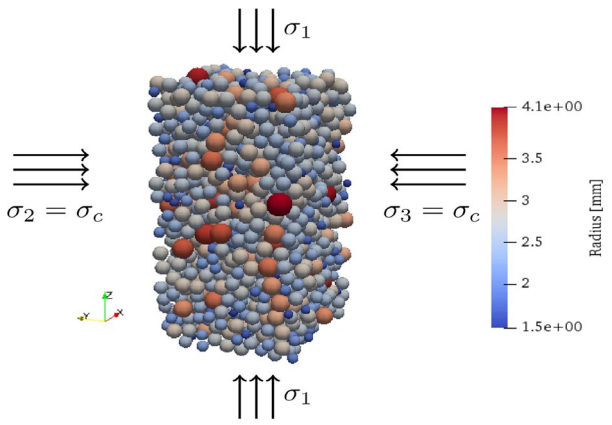


Fig. 14 Illustration of a typical DEM simulation of triaxial compression. Reprinted with permission from [37]

objective is often to understand the material's constitutive behaviour, namely the relationship between these QoIs and strain, e.g., ϵ_z. This task naturally leads to sequence learning and can be effectively addressed using relatively simple recurrent neural networks (RNNs), as demonstrated in early work of Wang and Sun [214].

5.1.2 LSTM-Based Surrogate Modelling of Stress–Strain Response

This section presents a concrete example of the machine learning workflow using long short-term memory (LSTM) networks to predict the stress-strain behaviour of GMs based on simulation data generated from DEM [215]. The schematic in Fig. 15 illustrates how the generic framework from Sect. 4.8 can be applied in a solid-like regime. A detailed tutorial can be found in the GrainLearning documentation [19]. Typically, the mean squared error between the LSTM and the DEM model is within 10^{-3}. See Figs. 18-23 in [214].

Data simulation, understanding and preparation: The Discrete Element Method (DEM) is used to simulate the compression of an RVE filled with grains to a certain porosity. DEM simulations of triaxial compressions are performed using the open-source DEM package YADE to generate training data and identify Quantities of Interest (QoI) sequences of stress and strain measures. These simulations vary in contact parameters and are saved at regular time intervals. The primary QoIs include mesoscopic material behaviour homogenised over the RVE, including sequences of stress, strain, and anisotropy measures, under various conditions (e.g., drained and undrained). A training dataset is then created, covering the evolution of these QoIs over the chosen loading paths for a wide range of contact parameters [180] and microstructures [216].

Modelling (LSTM model training and hyperparameter tuning): The training phase involves transforming the data from HDF5 format into TensorFlow datasets suitable for LSTM models. Key transformations include merging arrays from different HDF5 groups, standardising the data and splitting it into training, validation, and test sets. An LSTM model does not require an encoder and decoder. However, in this particular example, a sliding window technique [19] is used to divide the sequences into smaller lengths (see Fig. 15). Hyperparameter optimisation (including the sliding window size) uses the Weights and Biases (wandb) platform. This platform enables comprehensive tracking of training metrics, model configurations, and system performance across various runs. Users create configuration files to define the parameters and methods for optimisation, enabling efficient exploration of

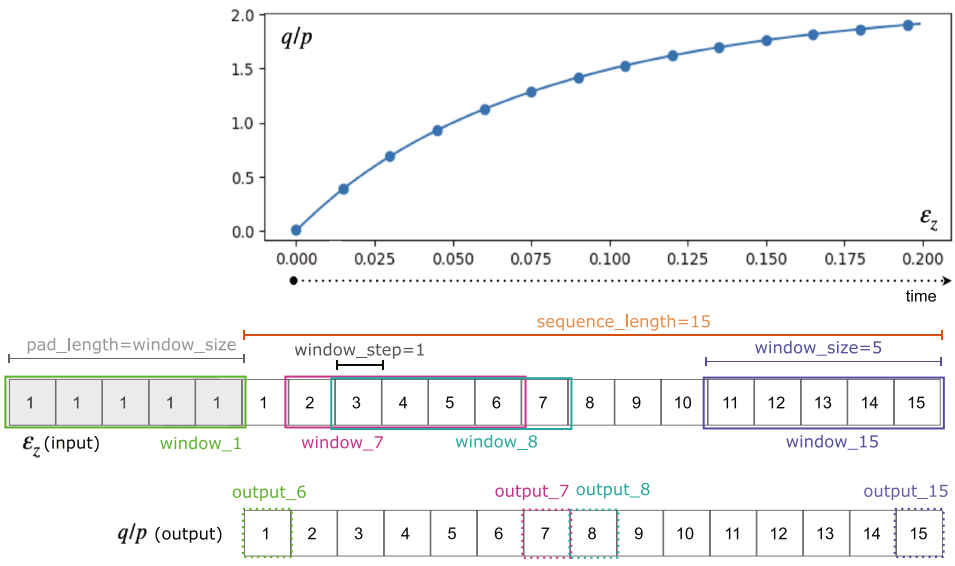


Fig. 15 LSTM-DEM workflow to predict stress–strain response

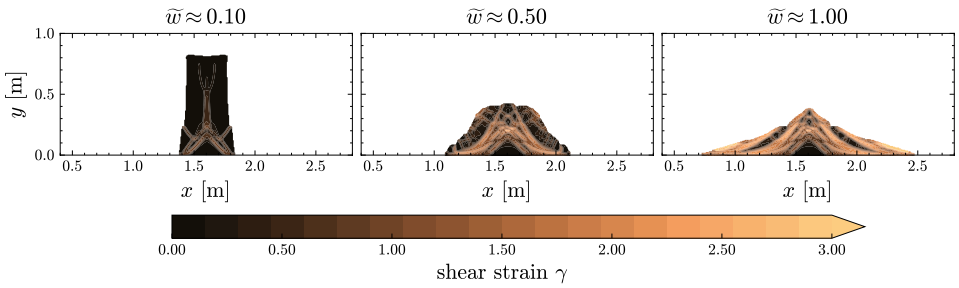


Fig. 16 Illustration of a typical MPM simulation of granular column collapse using the $\mu(I)$ rheology model [221]

the hyperparameter space. These configurations ensure the model is well-tuned, enhancing its predictive accuracy and generalizability. The optimised model and training metrics are saved for future use, facilitating reproducibility and further analysis.

During **evaluation**, the trained LSTM model accepts new sets of contact parameters and predicts the resulting stress–strain response over time. The predictions are evaluated against additional simulations to assess the model's accuracy and generalisability, such as DEM simulations performed with more particles. When **deployed**, LSTM-based surrogate models can efficiently yield physics-informed predictions for GMs under quasi-static loading. One significant advantage of LSTM-based surrogates is their integration with their physics-based counterparts in an uncertainty quantification or optimisation framework because the ML-based surrogates are much cheaper to run, and the error estimates of these surrogates can be computed for the Bayesian or probabilistic variants of these ML models.

5.2 Modelling Dynamic, Fluid-like Granular Flows

5.2.1 Collapse of Dense Granular Columns

Fluid-like granular regimes occur when materials flow rapidly—often down slopes or when columns collapse. Chute flows, for example, arise naturally in landslides and industrial transport processes and can vary from dense "liquid-like" to dilute "gas-like" states [14, 217] (see Fig. 16). Key factors such as channel inclination, particle shape, surface friction, and ambient fluid effects (e.g., air or water) all influence flow dynamics.

Contrary to the chute flows, the granular column collapse problem involves the sudden collapse of a vertical column of granular material, like sand, onto a flat surface, transitioning from a static to a dynamic state, as shown in Fig. 16. The scientific community studies this configuration [77, 218–220] to understand how granular materials transition to a rapid deformation and then to a deposit. The collapse sequence reveals how the grains spread and settle, which is crucial for understanding the extension of natural hazards, like landslides, avalanches, and debris flows, and designing safe and efficient storage and transport systems.

Modelling the granular column collapse [222, 223] is challenging due to the complex and nonlinear behaviour of GMs and their interaction with the surrounding media. Accurate simulations must capture intricate particle-particle and particle-fluid interactions, leading to non-trivial energy dissipation. Additionally, the influence of the surface on which the material collapses adds further complexity.

5.2.2 GNS-Based Surrogate Modelling of Granular Column Collapses

This section presents a concrete example of the machine learning workflow using the Graph Neural Network-based Simulator (GNS) for predicting granular column collapse trajectories and their runout distances. This example demonstrates how the generic framework outlined in Sect. 4.8 can be applied to fluid-like regimes in geotechnical contexts, as shown in Fig. 17. Different from Sect. 5.1.2, the numerical method used to generate training data is the Material Point Method (MPM) [66], and a GNS surrogate is trained on MPM data [224] to learn and predict granular flow dynamics. In general, the GNS runout prediction is within 5% of the MPM runout estimate. See Table 3. in [225]. The energy evolution profiles are also closely matched; for instance, in short column collapses, the predicted energy components are nearly identical to MPM. For more challenging extrapolations, such as tall column collapses (e.g., aspect ratio of 2.0, when the model was predominantly trained on aspect ratio 1.0), while the overall energy trends are well-replicated, the final normalized cumulative dissipated energy predicted by GNS showed a maximum difference of approximately 8-9% compared to MPM results, with GNS tending to predict slightly higher dissipation in such cases. Interested readers are referred to the examples in the GNS repository [226] .

Data simulation, understanding and preparation: This phase utilises MPM simulations to generate training data and identify Quantities of Interest (QoI). The input parameters include granular column geometry (e.g. aspect ratios from 0.5 to 4.0) and material properties (e.g. friction angles from 15° to 45°). A parametric sweep across these

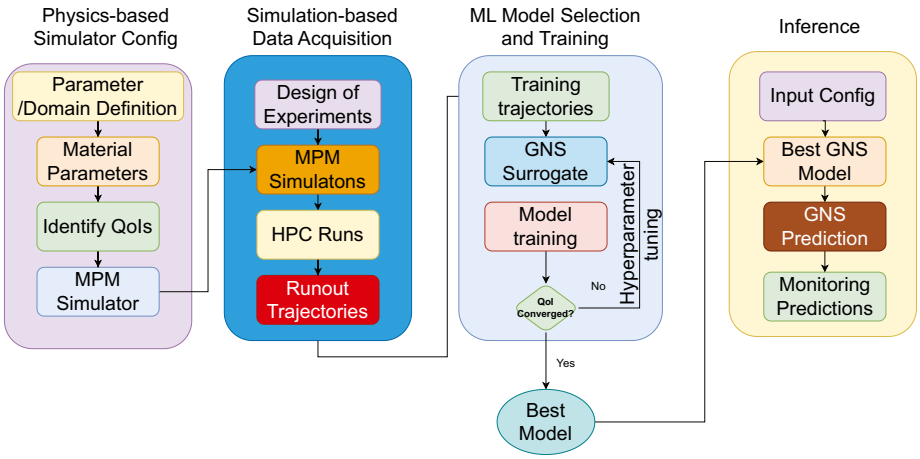


Fig. 17 GNS-MPM workflow to predict granular column collapse evolution

variables produces diverse granular flow trajectories. The primary Quantities of Interest (QoI) is the runout distance, with material point positions tracked over time. The simulations capture complex behaviours such as column collapse dynamics and barrier interactions. This phase establishes the ground truth data for training and validation, ensuring a comprehensive representation of granular flow physics across various configurations.

Modelling (GNS ML training): The GNS surrogate is trained on the MPM-generated trajectories. The GNS architecture incorporates strong inductive biases through its graph representation and message-passing operations. The model consists of an encoder, a processor with $M = 10$ message-passing steps, and a decoder. Domain knowledge is injected through an updater function that applies Euler integration for position updates, forcing the model to focus on learning unknown dynamics. Key hyperparameters include the connectivity radius R (e.g. 0.030 m for 2D, 0.025 m for 3D) and the number of message-passing steps. Training data is augmented with Gaussian noise to improve long-term prediction stability. The model is trained using an Adam optimiser with learning rate decay on a multi-GPU setup. Training progress is monitored using mean squared error loss on both training and validation sets, with the process continuing until convergence or a maximum of 5 million steps.

During **evaluation**, the trained GNS model, known for its adaptability, takes as input the initial geometry of the granular mass, material properties and boundary conditions. This input is encoded into a graph representation with N vertices representing material points. The GNS then performs $M = 10$ message-passing steps to predict particle dynamics. The updater function computes the next state based on these predictions. This process is repeated for K timesteps to simulate the entire granular flow evolution. Post-processing extracts relevant information such as runout distance, flow height and energy evolution. The GNS predictions are evaluated against MPM simulations or experimental data, assessing performance on various problem configurations, including different aspect ratios and upscaled domains. Metrics such as normalised runout distance and energy conservation are used to quantify the model's accuracy and generalizability. Once **deployed**, the GNS model's ability to generalise to unseen configurations and scales, leveraging the physics-informed architecture to make accurate predictions efficiently.

5.3 Open Infrastructure and Best Practices for Machine Learning in Granular Material Simulations

However, data requirements vary widely depending on the scientific challenges and applications presented in Sect. 5. To address this, regardless of the specific context, we also notice efforts towards establishing well-structured workflows, rigorous software engineering, and consistent data management principles that enhance the reproducibility and interoperability of ML approaches for modelling GMs. In particular, we observe approaches applying FAIR (Findable, Accessible, Interoperable, Reusable) data and software practices and leveraging existing frameworks like the Open Science Framework (OSF). These efforts are aimed at ensuring that ML tools remain transparent, reliable, and reusable, across diverse research groups and help organise and share data effectively. Hereby, hoping to establish a structured process for defining and transferring data from physics-based simulations to surrogate modelling and referencing benchmarks to ensure reproducibility and clarity. Below we briefly present these developments.

5.3.1 Data Management for Granular Material Simulations

Granular material simulations typically generate large data sets ranging from particle positions and velocities to boundary conditions and material parameters. Effectively handling

these datasets requires robust infrastructure (e.g., high-performance or parallel file systems), efficient and portable file formats (such as HDF5 or NetCDF), and comprehensive metadata schemas. Metadata should thoroughly record simulation parameters, temporal information, and the relationships between different data files or snapshots (App. A.1).

5.3.2 Best Software Practices

High-quality software is crucial for reproducible research in physics-informed data-driven modelling. By employing version control (e.g., Git), adopting a transparent release process, and publishing well-documented code in accessible repositories, developers enable the wider research community to verify results and build upon their prior work. Rigorous software testing, including unit and integration tests, helps prevent errors during collaborative development. Continuous integration that is deployable on many open platforms (e.g., GitHub and GitLab) can automate this testing. Lastly, it is essential to adhere to packaging conventions (e.g., PyPI for Python software) and consider code maintenance or deprecation strategies (App. A.2).

5.3.3 Data Preparation for Machine Learning Models

Raw simulation data often needs cleaning and normalisation to remove outliers or rescale variables before being fed into ML pipelines. Feature engineering transforms or extracts relevant attributes (e.g., packing density, velocity gradients) that help the ML model recognise underlying patterns. Appropriate data partitioning into training, validation, and test sets is crucial: it prevents overfitting, enhances model tuning reliability, and provides an unbiased performance estimate. Additionally, large and heterogeneous datasets may benefit from efficient data-loading and batching strategies—essential, for instance, when dealing with millions of particles in a single simulation. In some cases, data augmentation can increase the variety of training samples and improve model generalisation (App. A.3).

5.3.4 Training Machine Learning Models

Model training should be approached systematically to ensure clear documentation, reproducibility, and verifiable results. Key aspects include specifying the model architecture (e.g., the number and type of neural network layers) and clarifying input–output formats. Frequent monitoring of loss functions and metrics (e.g., mean squared error or accuracy) helps guide hyperparameter tuning, which can benefit from advanced methods like grid search, Bayesian optimisation, or other automated approaches. Techniques such as early stopping, cross-validation, and learning-rate scheduling bolster model reliability (App. A.4).

6 Conclusions and Outlook

The Lorentz Center workshop on "Machine Learning for Discrete Granular Media", brought together researchers from the granular materials and machine learning communities to discuss the state-of-the-art and challenges in an emerging research field that integrates machine learning and numerical simulations of granular materials. The workshop participants collectively created a knowledge base, which resulted in this position paper. First, the challenges in granular material simulations were reviewed, followed by a selective overview of ML approaches that proved useful in developing constitutive laws (theory), solving governing equations (numerical), and facilitating descriptive, predictive, and prescriptive digital twinning of granular systems. Synthesising on the current state-of-the-art and the discussions centered around the application, adaptation, and efficacy of ML in modelling granular materials, this section outlines future directions that touch on constitutive theory, neural operator learning, uncertainty quantification for granular materials, and the key issues on interpretability, scalability, and hybrid approaches within ML fields. The vision presented in this paper aims to serve research communities within granular mechanics and physics, as well as in various engineering applications.

6.1 Learning Material's Constitutive Laws

Materials' constitutive laws are largely based on classical theories of elasticity, plasticity and rheology, where physical constraints (e.g. thermodynamics) can be considered in handcrafting constitutive laws for granular materials. Advances in particle-scale methods now allow for the simulation of all the intricate features of granular material behaviour, including stress path dependency, fabric anisotropy (microstructure) and particle shapes, which can be expressed in stress and strain sequences. A major challenge here is to create ML-based constitutive laws based on particle-scale information that are more efficient and accurate (compared to experimental evidence) and provide insights into constitutive theories familiar to engineers. "Recurrent" black-box architectures (e.g., GRU, and LSTM) have shown their effectiveness in predicting stress and strain pairs, which are the macroscopic responses of granular materials. A future direction for learning constitutive models lies in the development of hybrid approaches, where ML models are guided by physical constraints and scarce experimental data, and complemented by abundant simulation data. For example, ML can assist in defining appropriate strain energy density functions or yield surfaces in stress space (augmented by other

variables), rather than relying on handcrafted models developed by domain experts. Symbolic regression can derive mathematical formulations directly from data, providing a middle ground between black-box ML models and fully physics-based approaches. These constitutive model surrogates should be general enough to cover both the solid-like (small strain, rate-independent) and fluid-like (large strain, rate-dependent) regimes of granular materials, incorporating physical principles (or biases) through regularisation. Lastly, it is essential to systematically verify ML-based constitutive surrogates by degenerating them into limiting cases, such as elastic solid and Newtonian fluid behaviour and benchmarking them against them.

6.2 Learning Field Evolutions of Underlying Governing Equations

Neural operators are capable of learning mappings between functional spaces. They are promising tools for bypassing conventional numerical methods to solve governing equations that exhibit strong nonlinearity and discontinuity, without the issues of numerical instabilities due to spatial or temporal discretisation. These methods allow generalisation across different problem geometries and material parameters without requiring handcrafted theories at a specific length or timescale. However, when applied to granular materials, neural operators face scalability issues due to the dynamic particle-scale interactions that exhibit macroscopic material behaviour transitioning between fluid-like and solid-like. Future directions should focus on refining encoder/decoder architectures to express spatial and temporal patterns in both the physical regime and on utilising transfer learning techniques to specific engineering tasks, e.g., geometries, materials, etc. Learning the multi-scale features of granular systems relies on latent state representations that can handle strong discontinuity and nonlinearity at the lower scales while capturing macroscopic fields of interest under various initial and boundary conditions. One example of circumventing the scalability issue is training on the underlying fields instead of fine-grained particle kinematics and interaction features in graph neural networks. However, the physically admissible microstructures, ready to be plugged again into numerical solvers like DEM, are difficult to reconstruct.

6.3 Learning Uncertainties from Limited Real-World Data

Many engineering problems in industry involve risk assessment and/or design optimisation, which integrate real-world uncertainty (although limited) into the physical processes represented by their predictive models. Given the stochastic nature of granular systems and the variability in material properties and conditions, uncertainty quantification is crucial in granular simulations and practical engineering applications. Data-driven surrogates, either based on classical machine learning or deep learning (see Sect. 4), are essential in constraining uncertainties with real-world measurement data or design objectives (e.g. through Bayesian updating and optimisation). Although deep learning models can learn to generalise across unseen parameter combinations, their probability landscape over the parameters is likely to be different from that of the numerical models, introducing additional uncertainties that are difficult to trace. A promising direction would be to add model uncertainties to deep learning models (e.g., Bayesian neural networks). The stochastic behaviour in real-world applications could be approached through ensembles: running multiple predictions with different parameters and averaging the results. Future work should also explore the trade-off between large and small neural networks for general or specific engineering tasks, allowing for more effective use of computational resources. Techniques such as Bayesian inference and Gaussian processes can be instrumental in understanding the robustness of ML-based constitutive models and identifying part of ML predictions that might no longer be reliable. By incorporating uncertainty quantification through the combined use of physics-based models and their ML surrogates into digital twin applications, researchers can enable more informed decision-making for real-world engineering processes.

6.4 Open Science Infrastructure to Facilitate "GranML"

The importance of data quality, sharing and reproducibility was discussed extensively in the workshop. For ML models to be broadly applicable and trusted by the community, they must be developed using standardised datasets and workflows that ensure reproducibility against well-known engineering benchmark problems. Open science practices can provide the infrastructure necessary for developing models and data resources collaboratively. To facilitate progress, future efforts should establish community-wide standards for data sharing and model benchmarking, such as the Open Network on DEM Simulations (ON-DEM), funded by the European Cooperation in Science and Technology, and DesignSafe, funded by the US National Science Foundation. An open-access platform akin to ImageNet could serve as a repository for training datasets, pre-trained models, and evaluation metrics. Such a platform would not only enhance reproducibility but also promote collaboration, allowing researchers from diverse disciplines to contribute to and benefit from shared data and methodologies.

In addition to the topics addressed during the workshop and in this position paper, several broader challenges emerged that are particularly relevant to machine learning

for granular materials. These include issues of scalability when dealing with large-scale systems involving millions of particles, the development of hybrid approaches that combine physics-based and data-driven models, and the need for improved interpretability to gain engineering insights. The potential applications of ML models for granular materials extend far beyond civil and mechanical engineering. More accurate, scalable, and interpretable ML-based approaches would also benefit fields such as pharmaceutical manufacturing, food processing, particle technology, and agriculture. Progress in the research area addressed in this position paper and beyond will enable the transition of ML from being a novel, exploratory tool to becoming an integral part of the assessment and design toolbox across multiple industries. Ultimately, the key to progress lies in fostering collaboration across domains–bringing together data scientists, engineers, and domain experts–to develop trustworthy, scalable models that can address the pressing challenges of granular material applications.

6.5 Cross-scale Generalisation and Foundation Models for Granular Materials

The machine learning community is increasingly focused on developing large-scale "foundation models" capable of generalising across diverse physical systems, domains, and boundary conditions [227]. Recent developments, such as NeuralDEM [71], show promise in learning underlying field equations directly from particle-scale data while capturing coupled multi-physics phenomena. For granular materials, these models could serve as unified surrogates across regimes and scales. However, their ability to capture transient behaviours—such as transitions between solid-like and fluid-like states in complex geometries—and to represent emerging features across varying length and time scales remains largely unexplored. Furthermore, the extent to which such models can be adapted to specific engineering tasks via transfer learning is still an open question. Given that training foundation models requires extensive and diverse datasets, it is essential to assess whether their benefits justify the development cost, particularly in engineering applications, where ML surrogates are often deployed in "edge cases" that are critical for DT tasks such as uncertainty quantification and optimisation. Addressing these questions calls for sustained cross-disciplinary collaboration at the intersection of ML and GM.

Appendix A Suggestions for Open Infrastructure and Best Practices for Material Learning in Granular Material Simulations

Best practices regarding FAIR data and software in this context include the creation of a standardised pre-processing pipeline to produce input for ML models, the establishment of a reproducible software environment that enables users to replicate workflows using identical computational setups, and the adoption of domain metadata standards covering vocabularies, formats, and data structures, as discussed in Sect. 5.3. Our practices demonstrated through the examples in Sect. 5 align with these principles by ensuring that ready-to-use input data for ML models is standardised, shared, and well-documented according to domain standards. The following appendices provide further details on these areas, offering more specific recommendations and strategies.

Data Management for Granular Material Simulations

- Describe the storage infrastructure used for handling large datasets, emphasising the use of high-performance systems such as parallel file systems.
- Specify the data formats employed (e.g. HDF5, NetCDF), justifying their selection based on factors like portability and efficiency.
- Detail the metadata schema, focusing on how it captures simulation parameters, particle-scale material properties, and spatio-temporal information in a structured manner.
- Outline the version control strategy for simulation codes and datasets, emphasising its role in ensuring reproducibility.
- If applicable, discuss any data compression techniques used, focusing on how they balance storage efficiency with data integrity.

Best Software Practices

It is important to recognise that different types of software vary in complexity. For instance, a DEM (Discrete Element Method) code contains multiple modules and greater complexity compared to a set of scripts designed for running simulations or machine learning model training. The following recommendations apply broadly to different types of software, though the extent and specifics may differ based on the software's nature and purpose [228].

- Use version control
  - Adopt a version control system (e.g. Git) to track code changes, ensure collaboration, and manage releases effectively.

  - Use tags to mark specific releases according to a clear versioning scheme (e.g. semantic versioning). This helps users understand compatibility between software versions and enhances reproducibility by enabling users to replicate results with a particular version.

- Repository publication
  - Host the software in a publicly accessible remote repository (e.g. GitHub, GitLab) to promote reusability, transparency and collaboration.
  - Include license (e.g. MIT, Apache2, GPL) to specify terms of use and reusability.
  - For scholarly recognition, publish your software in a service that will store snapshots of your software and assign a DOI (e.g Zenodo, Software Heritage, OSF - Open Science Framework, SourceForge) and ensure citation files (e.g. Citation File Format - CFF) are available so that your software can be correctly credited.
- Documentation
  - Provide user documentation that explains what the software does and how to use it, ideally in a Markdown format (e.g. README.md). This reduces barriers for new users, making the software more approachable.
  - Developer documentation (e.g. docstrings, inline comments, contributing guidelines) should explain how to extend or modify the software.
  - Deployment documentation should clearly outline system requirements, dependencies, and detailed installation instructions.
- Software testing
  - Implement a rigorous testing strategy, including unit tests, integration tests and regression tests, to ensure the software functions as intended in different environments.
  - Use code quality tools (e.g. linters) and coverage tools to assess how much of the code is being tested.
  - Set up CI tools (e.g. Github Actions, GitLab CI, Travis CI) to automatically run tests on each commit. This ensures that new changes do not introduce errors and maintain software quality over time.
- Modular code and software engineering standards:
  - Ensure code adheres to best practices in terms of modularity, readability and code reusability. Encourage collaboration by setting guidelines for contributing and reviewing code (e.g. pull requests, code reviews).
  - Follow recognised software engineering standards for packaging and distributing software using package managers like PyPI (Python), so that users can easily install the software.
- Maintenance and Sustainability:
  - Plan for the long-term maintenance of the software by ensuring there are sufficient resources and support structures. This could involve creating a developer community or ensuring future projects continue the software's maintenance.
  - Develop a retirement strategy if the software is no longer actively maintained, ensuring that users are aware of its status.
- Ensure that the software is secure against potential vulnerabilities such as dependency management, vendor lock-in and cross-platform compatibility.

Data Preparation for Machine Learning Models

- Describe the preprocessing pipeline for particle data, including cleaning, normalisation and augmentation methods.
- Explain the feature engineering process, detailing how relevant features are extracted and selected from raw particle data.
- Present the methodology for partitioning data into training, validation and test sets, justifying the chosen approach.
- Discuss efficient data loading and batching strategies, particularly for handling large-scale particle datasets during model training.
- If used, explain any data augmentation techniques applied to enhance model generalisation, providing the rationale for their selection.

Training Machine Learning Models

- Model description
  - Architecture: Document the model architecture (e.g. neural network layers, types of layers, recurrent/convolutional, etc.)
  - Inputs and outputs: Define the format, type and sizes of input data and the expected outputs.
  - Model size: Specify the number of parameters, memory footprint and compute requirements.
  - Regularisation techniques: Document the techniques used (e.g. L1/L2, dropout, batch normalisation) and their place in the pipeline.

- Loss functions and error metrics: Describe the loss and error metrics used and how are they monitored (i.e. per batch, epoch).
- Hyperparameter tuning
  - Document which techniques (e.g grid search, Bayesian optimisation, automated hyperparameter tuning tools) and in which order are they applied.
  - Report on which additional techniques such as: early stopping, cross-validation, learning rate schedules used and their parameters.
- Training–validation–test workflow: Document the training-validation-test workflow applied and the splitting of the data for each stage.

Acknowledgments The Lorentz Center workshop organisers, Hongyang Cheng, Marc Fransen, and Bojana Rosic would like to thank the Lorentz Center for providing the venue and facilitating the organisation. The main organiser, Hongyang Cheng, would like to acknowledge the financial support from the Netherlands eScience Center under grant number NLESC.OEC.2021.032. The materials used to create Figs. 2 and 3 are kindly provided by Saeed Mahdavy and Balázs Füvesi from the University of Twente.

Author Contributions All authors contributed equally to the first draft. D. Barreto, M.A. Cabrera, H. Cheng, D. Daniel, M. Fransen, B. Kieskamp, J. Nuttall, J. Ooi, S. Papanicolopulos, D. Schott and T. Weinhart wrote the first draft of Sec. 3. B. Alkin, J. Brandstetter, H. Cheng, A. Fürst, M. Guo, K. Kumar, T. Qu, T. Shuku, W. Sun, D. R. Tunuguntla, D.N. Wilke, and D. Ye wrote the first draft of Sec. 4. X. Fan, J. Morrissey, and L. Orozco wrote the first draft of Sec. 5. H. Cheng, D. R. Tunuguntla, and D.N. Wilke wrote the first draft of Sec. 1. H. Cheng wrote the first draft of Sec. 2 and Sec. 6. D. R. Tunuguntla led the revision of Sec. 1. M. Fransen led the revision of Sec. 3, A. Fürst and D. R. Tunuguntla led the revision of Sec. 4, H. Cheng, D. R. Tunuguntla and D.N. Wilke led the revision of Sec. 5. X. Fan and J. Morrissey wrote and reviewed Appendix. A. H. Cheng carried out the final edits of all the sections.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. O'Sullivan C (2011) Particulate discrete element modelling: a Geomechanics Perspective. Taylor & Francis, Milton Park, p 574
2. Montellà EP, Chauchat J, Bonamy C, Weij D, Keetels GH, Hsu TJ (2023) Numerical investigation of mode failures in submerged granular columns. Flow 3:28. https://doi.org/10.1017/flo.2023.23
3. Aguilar-López JP, Warmink JJ, Schielen RMJ, Hulscher SJMH (2018) Piping erosion safety assessment of flood defences founded over sewer pipes. Eur J Environ Civ Eng 22(6):707–735. https://doi.org/10.1080/19648189.2016.1217793
4. Parteli EJR, Pöschel T (2016) Particle-based simulation of powder application in additive manufacturing. Powder Technol 288:96–102. https://doi.org/10.1016/j.powtec.2015.10.035
5. Roy S, Weinhart T (2024) The role of granular matter in additive manufacturing. Granular Matter 26(4):102
6. Alvarez JE, Cheng H, Luding S, Weinhart T (2024) Densification of visco-elastic powders during free and pressure-assisted sintering. Int J Solids Struct 294:112786. https://doi.org/10.1016/j.ijsolstr.2024.112786
7. Horabik J, Molenda M (2016) Parameters and contact models for DEM simulations of agricultural granular materials: areview. Biosys Eng 147:206–225. https://doi.org/10.1016/j.biosystemseng.2016.02.017
8. Shmulevich I, Asaf Z, Rubinstein D (2007) Interaction between soil and a wide cutting blade using the discrete element method. Soil Tillage Res 97(1):37–50. https://doi.org/10.1016/j.still.2007.08.009
9. Schott D, Mohajeri J, Jovanova J, Lommen S, de Kluijver W (2021) Design framework for DEM-supported prototyping of grabs including full-scale validation. J Terrramech 96:29–43. https://doi.org/10.1016/j.jterra.2021.04.003
10. Fransen MP, Langelaar M, Schott D (2022) Including stochastics in metamodel-based DEM model calibration. Powder Technol 406:117400. https://doi.org/10.1016/j.powtec.2022.117400
11. Lommen S, Mohajeri M, Lodewijks G, Schott D (2019) DEM particle upscaling for large-scale bulk handling equipment and material interaction. Powder Technol 352:273–282. https://doi.org/10.1016/j.powtec.2019.04.034
12. Li C, Umbanhowar PB, Komsuoglu H, Koditschek DE, Goldman DI (2009) Sensitive dependence of the motion of a legged robot on granular media. Proc Natl Acad Sci U S A 106(9):3029–3034. https://doi.org/10.1073/pnas.0809095106. arXiv:0905.4950
13. Matl C, Narang Y, Bajcsy R, Ramos F, Fox D (2020) Inferring the material properties of granular media for robotic tasks. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 2770–2777. https://doi.org/10.1109/ICRA40945.2020.9197063
14. Forterre Y, Pouliquen O (2008) Flows of dense granular media. Ann Rev Fluid Mech 40(1):1–24
15. Li XS, Dafalias YF (2002) Constitutive modeling of inherently anisotropic sand behavior. J Geotech Geoenvironmental Eng 128(10):868–880. https://doi.org/10.1061/(ASCE)1090-0241(2002)128:10(868)
16. Gao Z, Zhao J, Li XS, Dafalias YF (2014) A critical state sand plasticity model accounting for fabric evolution. Int J Numer Anal Methods Geomech 38(4):370–390. https://doi.org/10.1002/nag.2211
17. Tunuguntla DR, Bokhove O, Thornton AR (2014) A mixture theory for size and density segregation in shallow granular free-surface flows. J Fluid Mech 749:99–112

18. Kamrin K, Hill KM, Goldman DI, Andrade JE (2024) Advances in modeling dense granular media. Ann Rev Fluid Mech 56(1):215–240
19. Cheng H, Orozco L, Lubbe R, Jansen A, Hartmann P, Thoeni K (2024) Grainlearning: a Bayesian uncertainty quantification toolbox for discrete and continuum numerical models of granular materials. J Open Source Softw 9(97):6338. https://doi.org/10.21105/joss.06338
20. Fransen MP (2024) Unravelling gravel: including stochastic behaviour of granular materials in design of bulk handling equipment. Dissertation (tu delft), Delft University of Technology. https://doi.org/10.4233/uuid:ad7999d2-5058-4a9d-8688-be6c91f41485
21. Vlahinić I, Kawamoto R, Andò E, Viggiani G, Andrade JE (2016) From computed tomography to mechanics of granular materials via level set bridge. Acta Geotech 12(1):1–11. https://doi.org/10.1007/s11440-016-0491-3
22. Cundall PA, Strack ODL (1979) A discrete numerical model for granular assemblies. Geotechnique 29:47–65
23. Kieckhefen P, Pietsch S, Dosta M, Heinrich S (2020) Possibilities and limits of computational fluid dynamics-discrete element method simulations in process engineering: a review of recent advancements and future trends. Annu Rev Chem Biomol Eng 11(2020):397–422. https://doi.org/10.1146/annurev-chembioeng-110519-075414
24. Khosravi A, Martinez A, Dejong JT (2020) Discrete element model (DEM) simulations of cone penetration test (CPT) measurements and soil classification. Can Geotech J 57(9):1369–1387. https://doi.org/10.1139/cgj-2019-0512
25. Schott DL, Mohajeri J (2023) 4. Multibody dynamics and discrete element method co-simulations for large-scale industrial equipment, pp. 107–143. John Wiley & Sons, Ltd. https://doi.org/10.1002/9783527835935.ch4
26. Fransen MP, Langelaar M, Schott DL (2023) Deterministic vs. robust design optimization using DEM-based metamodels. Powder Technol 425:118526. https://doi.org/10.1016/j.powtec.2023.118526
27. Abousleiman R, Walton G, Sinha S (2020) Understanding roof deformation mechanics and parametric sensitivities of coal mine entries using the discrete element method. Int J Min Sci Technol 30(1):123–129. https://doi.org/10.1016/j.ijmst.2019.12.006. (**Special issue on ground control in mining in 2019**)
28. Golshan S, Sotudeh-Gharebagh R, Zarghami R, Mostoufi N, Blais B, Kuipers JAM (2020) Review and implementation of CFD-DEM applied to chemical process systems. Chem Eng Sci 221:115646. https://doi.org/10.1016/j.ces.2020.115646
29. Shaheen MY, Thornton AR, Luding S, Weinhart T (2021) The influence of material and process parameters on powder spreading in additive manufacturing. Powder Technol 383:564–583
30. Vescovi D, Luding S (2016) Merging fluid and solid granular behavior. Soft Matter 12:8616–8628. https://doi.org/10.1039/C6SM01444E
31. Guo N, Zhao J (2014) A coupled FEM/DEM approach for hierarchical multiscale modelling of granular media. Int J Numer Meth Eng 99(11):789–818. https://doi.org/10.1002/nme.4702
32. Cheng H, Thornton AR, Luding S, Hazel AL, Weinhart T (2023) Concurrent multi-scale modeling of granular materials: Role of coarse-graining in FEM-DEM coupling. Comput Methods Appl Mech Eng 403:115651. https://doi.org/10.1016/j.cma.2022.115651
33. Wang M, Kumar K, Feng Y, Qu T, Wang M (2024) Machine learning aided modeling of granular materials: a review. Arch Comput Methods Eng 1–38
34. Bahmani B, Suh HS, Sun W (2024) Discovering interpretable elastoplasticity models via the neural polynomial method enabled symbolic regressions. Comput Methods Appl Mech Eng 422:116827. https://doi.org/10.1016/j.cma.2024.116827
35. Qu T, Zhao J, Guan S, Feng Y (2023) Data-driven multiscale modelling of granular materials via knowledge transfer and sharing. Int J Plast 171:103786
36. 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
37. Hartmann P, Cheng H, Thoeni K (2022) Performance study of iterative Bayesian filtering to develop an efficient calibration framework for DEM. Comput Geotech 141:104491. https://doi.org/10.1016/j.compgeo.2021.104491
38. Thuerey N, Holl P, Mueller M, Schnell P, Trost F, Um K (2021) Physics-based deep learning. arXiv preprint arXiv:2109.05237
39. Brunton SL, Kutz JN (2023) Machine learning for partial differential equations
40. Guo X, Li W, Iorio F (2016) Convolutional neural networks for steady flow approximation. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 481–490
41. Kochkov D, Smith JA, Alieva A, Wang Q, Brenner MP, Hoyer S (2021) Machine learning–accelerated computational fluid dynamics. Proc Natl Acad Sci 118(21):2101784118. https://doi.org/10.1073/pnas.2101784118
42. Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, Anandkumar A (2021) Fourier neural operator for parametric partial differential equations
43. Gupta JK, Brandstetter J (2023) Towards multi-spatiotemporal-scale generalized PDE modeling. Trans Mach Learn Res 2023
44. Alkin B, Fürst A, Schmid S, Gruber L, Holzleitner M, Brandstetter J (2024) Universal physics transformers: a framework for efficiently scaling neural operators. Adv Neural Inf Process Syst 37:25152–25194
45. Rasp S, Thuerey N (2021) Data-driven medium-range weather prediction with a resnet pretrained on climate simulations: a new model for weatherbench. J Adv Model Earth Syst 13(2):2020–002405
46. Weyn JA, Durran DR, Caruana R (2020) Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. J Adv Model Earth Syst 12(9):2020–002109
47. Sønderby CK, Espeholt L, Heek J, Dehghani M, Oliver A, Salimans T, Agrawal S, Hickey J, Kalchbrenner N (2020) Metnet: a neural weather model for precipitation forecasting. arXiv preprint arXiv:2003.12140
48. Pathak J, Subramanian S, Harrington P, Raja S, Chattopadhyay A, Mardani M, Kurth T, Hall D, Li Z, Azizzadenesheli K, Hassanzadeh P, Kashinath K, Anandkumar A (2022) FourCastNet: a global data-driven high-resolution weather model using adaptive fourier neural operators. arXiv preprint arXiv:2202.11214
49. Lam R, Sanchez-Gonzalez A, Willson M, Wirnsberger P, Fortunato M, Pritzel A, Ravuri S, Ewalds T, Alet F, Eaton-Rosen Z et al. (2022) GraphCast: learning skillful medium-range global weather forecasting. arXiv preprint arXiv:2212.12794
50. Nguyen T, Brandstetter J, Kapoor A, Gupta JK, Grover A (2023) Climax: a foundation model for weather and climate. In: Krause A, Brunskill E, Cho K, Engelhardt B, Sabato S, Scarlett J (eds.) International Conference on Machine Learning, ICML 2023. Proceedings of Machine Learning Research, vol. 202, pp. 25904–25938. PMLR
51. Bodnar C, Bruinsma WP, Lucic A, Stanley M, Brandstetter J, Garvan P, Riechert M, Weyn J, Dong H, Vaughan A et al (2024) Aurora: A foundation model of the atmosphere. arXiv preprint arXiv:2405.13063

52. Batzner S, Musaelian A, Sun L, Geiger M, Mailoa JP, Kornbluth M, Molinari N, Smidt TE, Kozinsky B (2022) E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. Nat Commun 13(1):2453
53. Batatia I, Kovacs DP, Simm G, Ortner C, Csányi G (2022) Mace: higher order equivariant message passing neural networks for fast and accurate force fields. Adv Neural Inf Process Syst 35:11423–11436
54. Zeni C, Pinsler R, Zügner D, Fowler A, Horton M, Fu X, Shysheya S, Crabbé J, Sun L, Smith J et al (2023) Mattergen: a generative model for inorganic materials design. arXiv preprint arXiv:2312.03687
55. Merchant A, Batzner S, Schoenholz SS, Aykol M, Cheon G, Cubuk ED (2023) Scaling deep learning for materials discovery. Nature 1–6
56. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A et al (2021) Highly accurate protein structure prediction with alphafold. Nature 596(7873):583–589
57. Brandstetter J, Worrall DE, Welling M (2022) Message passing neural PDE solvers. In: The tenth international conference on learning representations, ICLR
58. McCabe M, Blancard BR-S, Parker LH, Ohana R, Cranmer M, Bietti A, Eickenberg M, Golkar S, Krawezik G, Lanusse F et al (2023) Multiple physics pretraining for physical surrogate models. arXiv preprint arXiv:2310.02994
59. Herde M, Raonić B, Rohner T, Käppeli R, Molinaro R, Bézenac E, Mishra S (2024) Poseidon: Efficient foundation models for pdes. arXiv preprint arXiv:2405.19101
60. Quarteroni A, Valli A (2008) Numerical approximation of partial differential equations, vol 23. Springer, Berlin
61. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2008) The graph neural network model. IEEE Trans Neural Netw 20(1):61–80
62. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: International conference on learning representations
63. 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
64. Pfaff T, Fortunato M, Sanchez-Gonzalez A, Battaglia PW (2020) Learning mesh-based simulation with graph networks. arXiv preprint arXiv:2010.03409
65. Choi Y, Kumar K (2023) Graph Neural Network-based surrogate model for granular flows
66. 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
67. Choi Y, Kumar K (2024) Inverse analysis of granular flows using differentiable graph neural network simulator. arXiv preprint arXiv:2401.13695
68. Raonic B, Molinaro R, De Ryck T, Rohner T, Bartolucci F, Alaifari R, Mishra S, Bézenac E (2024) Convolutional neural operators for robust and accurate learning of pdes. Adv Neural Inf Process Syst 36:77187–77200
69. Cao S (2021) Choose a transformer: Fourier or galerkin. Adv Neural Inf Process Syst 34:24924–24940
70. Li Z, Meidani K, Farimani AB (2023) Transformer for partial differential equations' operator learning. Trans Mach Learn Res 2023
71. Alkin B, Kronlachner T, Papa S, Pirker S, Lichtenegger T, Brandstetter J (2024) NeuralDEM – real-time simulation of industrial particulate flows. arXiv preprint arXiv:2411.09678
72. IBM: What is a digital twin? https://www.ibm.com/think/topics/what-is-a-digital-twin. Accessed: 2025-06-22
74. Zhao J, Guo N (2013) Unique critical state characteristics in granular media considering fabric anisotropy. Géotechnique 8(8):695–704. https://doi.org/10.1680/geot.12.P.040
75. Cheng H, Yamamoto H, Guo N, Huang H (2017) A simple multiscale model for granular soils with geosynthetic inclusion. In: Li X, Feng Y, Mustoe G (eds.) Proceedings of the 7th international conference on discrete element methods, pp. 445–453. Springer, Singapore
76. Budhu M (2010) Soil mechanics and foundations. John Wiley and Sons, Hoboken
77. Staron L, Duchemin L, Lagrée P-Y (2023) Cohesive granular columns collapsing: numerics questioning failure, cohesion, and friction. J Rheol 67(5):1061–1072
78. Zhao T (2017) Coupled DEM-CFD analyses of landslide-induced debris flows, English edn. Springer, Singapore. Hardcover
79. Bishop CM (2007) Pattern recognition and machine learning (information science and statistics), 1st edn. Springer, Berlin
80. Haghighi EM, Na S (2022) A single Long Short-Term Memory network for enhancing the prediction of path-dependent plasticity with material heterogeneity and anisotropy. arXiv:2204.01466
81. Wang K, Sun W (2018) A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning. Comput Methods Appl Mech Eng 334:337–380
82. 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 Geotech 17(8):3463–3478
83. Wang K, Sun W (2019) Meta-modeling game for deriving theory-consistent, microstructure-based traction-separation laws via deep reinforcement learning. Comput Methods Appl Mech Eng 346:216–241
84. Qu T, Di S, Feng Y, Wang M, Zhao T (2021) Towards data-driven constitutive modelling for granular materials via micromechanics-informed deep learning. Int J Plast 144:103046
85. 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. Comput Geotech 152:105049
86. Ghane E, Fagerström M, Mirkhalaf M (2025) Multi-fidelity data fusion for inelastic woven composites: Combining recurrent neural networks with transfer learning. Compos Sci Technol 267:111163. https://doi.org/10.1016/j.compscitech.2025.111163
87. Ghaboussi J, Garrett J Jr, Wu X (1991) Knowledge-based modeling of material behavior with neural networks. J Eng Mech 117(1):132–153
88. Huang D, Fuhg JN, Weißenfels C, Wriggers P (2020) A machine learning based plasticity model using proper orthogonal decomposition. Comput Methods Appl Mech Eng 365:113008
89. 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. J Rock Mech Geotech Eng 16(6):2198–218
90. Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686–707. https://doi.org/10.1016/j.jcp.2018.10.045
91. Lefik M, Schrefler BA (2003) Artificial neural network as an incremental non-linear constitutive model for a finite element code. Comput Methods Appl Mech Eng 192(28–30):3265–3283
92. Vlassis NN, Zhao P, Ma R, Sewell T, Sun W (2022) Molecular dynamics inferred transfer learning models for finite-strain hyperelasticity of monoclinic crystals: Sobolev training and validations against physical constraints. Int J Numer Meth Eng 123(17):3922–3949

93. Heider Y, Wang K, Sun W (2020) So (3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials. Comput Methods Appl Mech Eng 363:112875
94. Cai C, Vlassis N, Magee L, Ma R, Xiong Z, Bahmani B, Wong T-F, Wang Y, Sun W (2023) Equivariant geometric learning for digital rock physics: estimating formation factor and effective permeability tensors from morse graph. Int J Multiscale Comput Eng 21(5)
95. Sun X, Bahmani B, Vlassis NN, Sun W, Xu Y (2022) Data-driven discovery of interpretable causal relations for deep learning material laws with uncertainty propagation. Granular Matter 24:1–32
96. Wang K, Sun W, Du Q (2021) A non-cooperative meta-modeling game for automated third-party calibrating, validating and falsifying constitutive laws with parallelized adversarial attacks. Comput Methods Appl Mech Eng 373:113514
97. Phan NN, Sun W, Clayton JD (2025) Hydra: Symbolic feature engineering of overparameterized eulerian hyperelasticity models for fast inference time. Comput Methods Appl Mech Eng 437:117792
98. Golan A, Harte J (2022) Information theory: a foundation for complexity science. Proc Natl Acad Sci 119(33):2119089119
99. He X, Chen J-S (2022) Thermodynamically consistent machine-learned internal state variable approach for data-driven modeling of path-dependent materials. Comput Methods Appl Mech Eng 402:115348
100. Wang K, Sun W, Du Q (2019) A cooperative game for automated learning of elasto-plasticity knowledge graphs and models with ai-guided experimentation. Comput Mech 64:467–499
101. Murdoch WJ, Singh C, Kumbier K, Abbasi-Asl R, Yu B (2019) Definitions, methods, and applications in interpretable machine learning. Proc Natl Acad Sci 116(44):22071–22080
102. Vlassis NN, Sun W (2021) Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening. Comput Methods Appl Mech Eng 377:113695
103. Vlassis NN, Ma R, Sun W (2020) Geometric deep learning for computational mechanics part I: anisotropic hyperelasticity. Comput Methods Appl Mech Eng 371:113299
104. Cranmer M, Sanchez Gonzalez A, Battaglia P, Xu R, Cranmer K, Spergel D, Ho S (2020) Discovering symbolic models from deep learning with inductive biases. Adv Neural Inf Process Syst 33:17429–17442
105. Villarreal R, Vlassis NN, Phan NN, Catanach TA, Jones RE, Trask NA, Kramer SL, Sun W (2023) Design of experiments for the calibration of history-dependent models via deep reinforcement learning and an enhanced kalman filter. Comput Mech 72(1):95–124
106. 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. Int J Plast 164:103576
107. Su M, Guo N, Yang Z (2023) A multifidelity neural network (mfnn) for constitutive modeling of complex soil behaviors. Int J Numer Anal Meth Geomech 47(18):3269–3289
108. Zhang P, Yin Z-Y, Sheil B (2024) Multifidelity constitutive modeling of stress-induced anisotropic behavior of clay. J Geotech Geoenviron Eng 150(3):04024003
109. Nguyen PC, Vlassis NN, Bahmani B, Sun W, Udaykumar H, Baek SS (2022) Synthesizing controlled microstructures of porous media using generative adversarial networks and reinforcement learning. Sci Rep 12(1):9034
110. Vlassis NN, Sun W, Alshibli KA, Regueiro RA (2024) Synthesizing realistic sand assemblies with denoising diffusion in latent space. Int J Numer Anal Meth Geomech 48(16):3933–3956
111. Karapiperis K, Stainier L, Ortiz M, Andrade JE (2021) Data-driven multiscale modeling in mechanics. J Mech Phys Solids 147:104239. https://doi.org/10.1016/j.jmps.2020.104239
112. Karapiperis K, Ortiz M, Andrade JE (2021) Data-driven nonlocal mechanics: discovering the internal length scales of materials. Comput Methods Appl Mech Eng 386:114039. https://doi.org/10.1016/j.cma.2021.114039
113. Kirchdoerfer T, Ortiz M (2017) Data driven computing with noisy material data sets. Comput Methods Appl Mech Eng 326:622–641. https://doi.org/10.1016/j.cma.2017.07.039
114. Bergmann D (2025) What is latent space? https://www.ibm.com/think/topics/latent-space. Accessed: 17 June 2025
115. LeCun Y, Cortes C, Burges CJC (1998) The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/. Accessed: 2025-06-17
116. Tong A, Huang J, Wolf G, Dijk D, Krishnaswamy S (2020) Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In: International conference on machine learning, pp. 9526–9536. PMLR
117. Yang Z, Li X, Brinson LC, Choudhary AN, Chen W, Agrawal A (2018) Microstructural materials design via deep adversarial learning methodology. J Mech Des 140(11):111416
118. Grover A, Dhar M, Ermon S (2018) Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In: Proceedings of the AAAI conference on artificial intelligence, vol. 32
119. Wilke DN, Heyns PS, Schmidt S (2022) The role of untangled latent spaces in unsupervised learning applied to condition-based maintenance. In: Hammami A, Heyns PS, Schmidt S, Chaari F, Abbes MS, Haddar M (eds) Modelling and simulation of complex systems for sustainable energy efficiency. Springer, Cham, pp 38–49
120. Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, Mohamed S, Lerchner A (2017) beta-vae: Learning basic visual concepts with a constrained variational framework. In: International Conference on Learning Representations
121. Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686–707
122. Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P (2017) Geometric deep learning: going beyond euclidean data. IEEE Signal Process Mag 34(4):18–42
123. Wilke DN, Cleary PW, Govender N (2021) From discrete element simulation data to process insights. EPJ Web Conf 249:15001. https://doi.org/10.1051/epjconf/202124915001
124. Balshaw R, Heyns PS, Wilke DN, Schmidt S (2023) Latent indicators for temporal-preserving latent variable models in vibration-based condition monitoring under non-stationary conditions. Mech Syst Signal Process 199:110446. https://doi.org/10.1016/j.ymssp.2023.110446
125. Zhang X, Wang L, Helwig J, Luo Y, Fu C, Xie Y, Liu M, Lin Y, Xu Z, Yan K, Adams K, Weiler M, Li X, Fu T, Wang Y, Yu H, Xie Y, Fu X, Strasser A, Xu S, Liu Y, Du Y, Saxton A, Ling H, Lawrence H, Stärk H, Gui S, Edwards C, Gao N, Ladera A, Wu T, Hofgard EF, Tehrani AM, Wang R, Daigavane A, Bohde M, Kurtin J, Huang Q, Phung T, Xu M, Joshi CK, Mathis SV, Azizzadenesheli K, Fang A, Aspuru-Guzik A, Bekkers E, Bronstein M, Zitnik M, Anandkumar A, Ermon S, Liò P, Yu R, Günnemann S, Leskovec J, Ji H, Sun J, Barzilay R, Jaakkola T, Coley CW, Qian X, Qian X, Smidt, T, Ji S (2024) Artificial intelligence for science in quantum, atomistic, and continuum systems. arXiv:2307.08423
126. Jiang Y, Byrne E, Glassey J, Chen X (2024) Integrating graph neural network-based surrogate modeling with inverse design for granular flows. Indus Eng Chem Res 63(20):9225–9235

127. Bronstein MM, Bruna J, Cohen T, Veličković P (2021) Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. arXiv preprint arXiv:2104.13478
128. Satorras VG, Hoogeboom E, Welling M (2021) E (n) equivariant graph neural networks. In: International conference on machine learning, pp. 9323–9332. PMLR
129. Brandstetter J, Hesselink R, Pol E, Bekkers EJ, Welling M (2022) Geometric and physical quantities improve E(3) equivariant message passing. In: The tenth international conference on learning representations, ICLR
130. Sharma V, Fink O (2025) Dynami-CAL GraphNet: A physics-informed graph neural network conserving linear and angular momentum for dynamical systems. arXiv:2501.07373
131. Musaelian A, Batzner S, Johansson A, Sun L, Owen CJ, Kornbluth M, Kozinsky B (2023) Learning local equivariant representations for large-scale atomistic dynamics. Nat Commun 14(1):579
132. Bonnet F, Mazari J, Cinnella P, Gallinari P (2022) Airfrans: high fidelity computational fluid dynamics dataset for approximating reynolds-averaged navier-stokes solutions. Adv Neural Inf Process Syst 35:23463–23478
133. Kozinsky B, Musaelian A, Johansson A, Batzner S (2023) Scaling the leading accuracy of deep equivariant models to biomolecular simulations of realistic size. In: Proceedings of the International conference for high performance computing, networking, storage and analysis, pp. 1–12
134. Qi CR, Yi L, Su H, Guibas LJ (2017) Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems 30
135. Fortunato M, Pfaff T, Wirnsberger P, Pritzel A, Battaglia P (2022) Multiscale meshgraphnets. arXiv preprint arXiv:2210.00612
136. Lino M, Fotiadis S, Bharath AA, Cantwell CD (2022) Multi-scale rotation-equivariant graph neural networks for unsteady eulerian fluid dynamics. Phys Fluids 34(8):087110
137. Haeri A, Holz D, Skonieczny K (2024) Subspace graph networks for real-time granular flow simulation with applications to machine-terrain interactions. Eng Appl Artif Intell 135:108765
138. Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE (2019) Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. Nat Machine Intell 3:218–229
139. Li Z-Y, Kovachki NB, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart AM, Anandkumar A (2020) Neural operator: Graph kernel network for partial differential equations. ArXiv arXiv:abs/2003.03485
140. Kovachki N, Li Z, Liu B, Azizzadenesheli K, Bhattacharya K, Stuart A, Anandkumar A (2023) Neural operator: learning maps between function spaces with applications to pdes. J Mach Learn Res 24(89):1–97
141. Seidman J, Kissas G, Perdikaris P, Pappas GJ (2022) Nomad: nonlinear manifold decoders for operator learning. Adv Neural Inf Process Syst 35:5601–5613
142. Li Z, Kovachki NB, Choy C, Li B, Kossaifi J, Otta SP, Nabian MA, Stadler M, Hundt C, Azizzadenesheli K, Anandkumar A (2023) Geometry-informed neural operator for large-scale 3D PDEs
143. Hao Z, Wang Z, Su H, Ying C, Dong Y, Liu S, Cheng Z, Song J, Zhu J (2023) Gnot: A general neural operator transformer for operator learning. In: International conference on machine learning, pp. 12556–12569. PMLR
144. Wang T, Wang C (2024) Latent neural operator for solving forward and inverse PDE problems. arXiv:2406.03923
145. Wang S, Seidman JH, Sankaran S, Wang H, Pappas GJ, Perdikaris, P (2024) Cvit: Continuous vision transformer for operator learning arXiv:2405.13998 [cs.LG]
146. Hesthaven JS, Rozza G, Stamm B (2016) Certified reduced basis methods for parametrized partial differential equations, vol 590. Springer, Berlin
147. Quarteroni A, Manzoni A, Negri F (2015) Reduced basis methods for partial differential equations: an introduction. Springer, Berlin
148. Benner P, Gugercin S, Willcox K (2015) A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev 57(4):483–531
149. Lee K, Carlberg KT (2020) Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. J Comput Phys 404:108973
150. Hesthaven JS, Ubbiali S (2018) Non-intrusive reduced order modeling of nonlinear problems using neural networks. J Comput Phys 363:55–78
151. Guo T, Rokoš O, Veroy K (2021) Learning constitutive models from microstructural simulations via a non-intrusive reduced basis method. Comput Methods Appl Mech Eng 384:113924
152. Kabalan A, Casenave F, Bordeu F, Ehrlacher V, Ern A (2025) Elasticity-based morphing technique and application to reduced-order modeling. Appl Math Model 141:115929. https://doi.org/10.1016/j.apm.2025.115929
153. Ye D, Krzhizhanovskaya V, Hoekstra AG (2024) Data-driven reduced-order modelling for blood flow simulations with geometry-informed snapshots. J Comput Phys 497:112639. https://doi.org/10.1016/j.jcp.2023.112639
154. Matray V, Amlani F, Feyel F, Néron D (2024) A hybrid numerical methodology coupling reduced order modeling and graph neural networks for non-parametric geometries: applications to structural dynamics problems. Comput Methods Appl Mech Eng 430:117243. https://doi.org/10.1016/j.cma.2024.117243
155. Champion K, Lusch B, Kutz JN, Brunton SL (2019) Data-driven discovery of coordinates and governing equations. Proc Natl Acad Sci 116(45):22445–22451
156. Yu J, Yan C, Guo M (2019) Non-intrusive reduced-order modeling for fluid problems: a brief review. Proc Instit Mech Eng Part G J Aerospace Eng 233(16):5896–5912
157. Murphy KP (2012) Machine learning: a probabilistic perspective. MIT press, Cambridge
158. Berkooz G, Holmes P, Lumley JL (1993) The proper orthogonal decomposition in the analysis of turbulent flows. Ann Rev Fluid Mech 25:539–575
159. Liang YC, Lee HP, Lim SP, Lin WZ, Lee KH, Wu CG (2002) Proper orthogonal decomposition and its applications–Part I: Theory. J Sound Vib 252(3):527–544
160. Fresca S, Manzoni A (2022) POD-DL-ROM: enhancing deep learning-based reduced order models for nonlinear parametrized pdes by proper orthogonal decomposition. Comput Methods Appl Mech Eng 388:114181
161. Botteghi N, Guo M, Brune C (2022) Deep kernel learning of dynamical models from high-dimensional noisy data. Sci Rep 12:21530
162. Chen W, Wang Q, Hesthaven JS, Zhang C (2021) Physics-informed machine learning for reduced-order modeling of nonlinear problems. J Comput Phys 446:110666
163. Swischuk R, Mainini L, Peherstorfer B, Willcox K (2019) Projection-based model reduction: formulations for physics-based machine learning. Comput Fluids 179:704–717
164. Xiao D, Fang F, Buchan AG, Pain CC, Navon IM, Muggeridge A (2015) Non-intrusive reduced order modelling of the navier-stokes equations. Comput Methods Appl Mech Eng 293:522–541

165. Guo M, Hesthaven JS (2018) Reduced order modeling for nonlinear structural analysis using gaussian process regression. Comput Methods Appl Mech Eng 341:807–826
166. Guo M, Hesthaven JS (2019) Data-driven reduced order modeling for time-dependent problems. Comput Methods Appl Mech Eng 345:75–99
167. Kast M, Guo M, Hesthaven JS (2020) A non-intrusive multifidelity method for the reduced order modeling of nonlinear problems. Comput Methods Appl Mech Eng 364:112947
168. Cicci L, Fresca S, Guo M, Manzoni A, Zunino P (2023) Uncertainty quantification for nonlinear solid mechanics using reduced order models with gaussian process regression. Comput Math Appl 149:1–23
169. Ghattas O, Willcox K (2021) Learning physics-based models from data: perspectives from inverse problems and model reduction. Acta Numer 30:445–554
170. Peherstorfer B, Willcox K (2016) Data-driven operator inference for nonintrusive projection-based model reduction. Comput Methods Appl Mech Eng 306:196–215
171. Qian E, Kramer B, Peherstorfer B, Willcox K (2020) Lift & learn: physics-informed machine learning for large-scale nonlinear dynamical systems. Phys D 406:132401
172. Guo M, McQuarrie SA, Willcox KE (2022) Bayesian operator inference for data-driven reduced-order modeling. Comput Methods Appl Mech Eng 402:115336
173. Brunton SL, Proctor JL, Kutz JN (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc Natl Acad Sci 113(15):3932–3937
174. Schaeffer H (2017) Learning partial differential equations via data discovery and sparse optimization. Proc R Soc A Math Phys Eng Sci 473(2197):20160446
175. Chen RTQ, Rubanova Y, Bettencourt J, Duvenaud DK (2018) Neural ordinary differential equations. Advances in neural information processing systems 31
176. Zhuang Q, Lorenzi JM, Bungartz H-J, Hartmann D (2021) Model order reduction based on Runge-Kutta neural networks. Data-Centric Eng 2:e13
177. Conti P, Guo M, Manzoni A, Frangi A, Brunton SL, Nathan Kutz J (2024) Multi-fidelity reduced-order surrogate modelling. Proc R Soc A 480(2283):20230655
178. Botteghi N, Motta P, Manzoni A, Zunino P, Guo M (2024) Recurrent deep kernel learning of dynamical systems. arXiv preprint arXiv:2405.19785
179. Särkkä S (2013) Bayesian Filtering and Smoothing. Institute of mathematical statistics textbooks, vol. 3. Cambridge University Press, Cambridge, UK. https://doi.org/10.1017/CBO9781139344203
180. Cheng H, Shuku T, Thoeni K, Tempone P, Luding S, Magnanimo V (2019) An iterative Bayesian filtering framework for fast and automated calibration of DEM models. Comput Methods Appl Mech Eng 350:268–294. https://doi.org/10.1016/j.cma.2019.01.027
181. Andrieu C, Doucet A, Holenstein R (2010) Particle markov chain monte carlo methods. J R Stat Soc Ser B (Stat Methodol) 72(3):269–342. https://doi.org/10.1111/j.1467-9868.2009.00736.x
182. Giles MB (2015) Multilevel monte carlo methods. Acta Numer 24:259–328. https://doi.org/10.1017/S096249291500001X
183. Rasmussen CE, Williams CK (2006) Gaussian processes for machine learning, vol 2. MIT press, Cambridge
184. Fuhg JN, Marino M, Bouklas N (2022) Local approximate gaussian process regression for data-driven constitutive models: development and comparison with neural networks. Comput Methods Appl Mech Eng 388:114217
185. Guan S, Qu T, Feng Y, Ma G, Zhou W (2023) A machine learning-based multi-scale computational framework for granular materials. Acta Geotech 18(4):1699–1720
186. Raissi M, Perdikaris P, Karniadakis GE (2017) Machine learning of linear differential equations using gaussian processes. J Comput Phys 348:683–693
187. Pförtner M, Steinwart I, Hennig P, Wenger J (2022) Physics-informed gaussian process regression generalizes linear pde solvers. arXiv preprint arXiv:2212.12474
188. Chen Y, Hosseini B, Owhadi H, Stuart AM (2021) Solving and learning nonlinear pdes with gaussian processes. J Comput Phys 447:110668
189. Ye D, Guo M (2024) Gaussian process learning of nonlinear dynamics. Communications in nonlinear science and numerical simulation, 108184
190. Xu D, Shen Y (2022) An improved machine learning approach for predicting granular flows. Chem Eng J 450:138036
191. Cai C, Wang D, Wang Y (2021) Graph coarsening with neural networks, 1–26 arXiv:2102.01350
192. Husic BE, Charron NE, Lemm D, Wang J, Pérez A, Majewski M, Krämer A, Chen Y, Olsson S, De Fabritiis G, Noé F, Clementi C (2020) Coarse graining molecular dynamics with graph neural networks. J Chem Phys 153(19):241722. https://doi.org/10.1063/5.0026133. arXiv:2007.11412
193. Guan Q, Yang Z, Guo N, Chen L (2024) Deep learning-accelerated multiscale approach for granular material modeling. Int J Numer Anal Meth Geomech 48(5):1372–1389. https://doi.org/10.1002/nag.3688
194. Maia MA, Rocha IBCM, Kerfriden P, van der Meer FP (2023) Physically recurrent neural networks for path-dependent heterogeneous materials: embedding constitutive models in a data-driven surrogate. Comput Methods Appl Mech Eng 407:115934. https://doi.org/10.1016/j.cma.2023.115934
195. Storm J, Rocha IBCM, van der Meer FP (2024) A microstructure-based graph neural network for accelerating multiscale simulations. Comput Methods Appl Mech Eng 427:117001. https://doi.org/10.1016/j.cma.2024.117001
196. Chaouch S, Yvonnet J (2024) Unsupervised machine learning classification for accelerating fe2 multiscale fracture simulations. Comput Methods Appl Mech Eng 432:117278. https://doi.org/10.1016/j.cma.2024.117278
197. Linden JH, Narsilio GA, Tordesillas A (2016) Machine learning framework for analysis of transport through complex networks in porous, granular media: a focus on permeability. Phys Rev E 94:022904. https://doi.org/10.1103/PhysRevE.94.022904
198. Xie T, Grossman JC (2018) Hierarchical visualization of materials space with graph convolutional neural networks. J Chem Phys 149(17):174111. https://doi.org/10.1063/1.5047803
199. Tordesillas A, Zhou S, Bailey J, Bondell H (2022) A representation learning framework for detection and characterization of dead versus strain localization zones from pre- to post-failure. Granular Matter 24(3):75. https://doi.org/10.1007/s10035-022-01233-7
200. Kirchdoerfer T, Ortiz M (2016) Data-driven computational mechanics. Comput Methods Appl Mech Eng 304:81–101. https://doi.org/10.1016/j.cma.2016.02.001
201. Vlassis NN, Sun W (2021) Component-based machine learning paradigm for discovering rate-dependent and pressure-sensitive level-set plasticity models. J Appl Mech 89(2):021003. https://doi.org/10.1115/1.4052684
202. Li Z, Kovachki NB, Choy C, Li B, Kossaifi J, Otta SP, Nabian MA, Stadler M, Hundt C, Azizzadenesheli K et al (2023) Geometry-informed neural operator for large-scale 3d pdes. arXiv preprint arXiv:2309.00583
203. Xiao D, Yang P, Fang F, Xiang J, Pain CC, Navon IM (2016) Non-intrusive reduced order modelling of fluid–structure

interactions. Comput Methods Appl Mech Eng 303:35–54. https://doi.org/10.1016/j.cma.2015.12.029
204. Han R, Wang Y, Qian W, Wang W, Zhang M, Chen G (2022) Deep neural network based reduced-order model for fluid–structure interaction system. Phys Fluids 34(7):073610. https://doi.org/10.1063/5.0096432
205. Pirker S, Atzori M, Heinrich S, Lichtenegger T (2025) Particle size segregation in bi and penta-disperse gas–solid fluidized beds: Cfd-dem and recurrence cfd simulations. Chem Eng Sci 309:121469. https://doi.org/10.1016/j.ces.2025.121469
206. Leibfried F, Dutordoir V, John S, Durrande N (2022) A tutorial on sparse gaussian processes and variational inference. arXiv:2012.13962
207. Martínez-Plumed F, Contreras-Ochando L, Ferri C, Hernández-Orallo J, Kull M, Lachiche N, Ramírez-Quintana MJ, Flach P (2019) Crisp-dm twenty years later: from data mining processes to data science trajectories. IEEE Trans Knowl Data Eng 33(8):3048–3061
208. Chapman P, Clinton J, Kerber R, Khabaza T, Reinartz T, Shearer C, Wirth R et al (2000) Crisp-dm 1.0: Step-by-step data mining guide. SPSS inc 9(13):1–73
209. Feng YT (2023) Thirty years of developments in contact modelling of non-spherical particles in DEM: a selective review. Acta Mech Sin Xuebao 39(1):722343. https://doi.org/10.1007/s10409-022-22343-x
210. Weinhart T, Orefice L, Post M, van Schrojenstein Lantman MPMP, Denissen IFCIFC, Tunuguntla DRDR, Tsang JMF, Cheng H, Shaheen MYMY, Shi H, Rapino P, Grannonio E, Losacco N, Barbosa J, Jing LL, Alvarez Naranjo JEJE, Roy S, Otter WKWK, Thornton ARAR (2020) Fast, flexible particle simulations—an introduction to MercuryDPM. Comput Phys Commun 249:107129
211. Duriez J, Galusinski C (2021) A level set-discrete element method in YADE for numerical, micro-scale, geomechanics with refined grain shapes. Comput Geosci 157(September):104936. https://doi.org/10.1016/j.cageo.2021.104936
212. Zhao S, Zhao J (2021) SudoDEM: unleashing the predictive power of the discrete element method on simulation for non-spherical granular particles. Comput Phys Commun 259:107670. https://doi.org/10.1016/j.cpc.2020.107670
213. Wang S, Ji S (2022) A unified level set method for simulating mixed granular flows involving multiple non-spherical DEM models in complex structures. Comput Methods Appl Mech Eng 393:114802. https://doi.org/10.1016/j.cma.2022.114802
214. Wang K, Sun W (2018) A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning. Comput Methods Appl Mech Eng 334:337–380. https://doi.org/10.1016/j.cma.2018.01.036
215. Cheng H (2021) DEM simulation data containing micro- and macro-scale quantities of granular materials under triaxial compression. Dataset. Version 1. Available at https://doi.org/10.4121/16632559.v1. 4TU.ResearchData
216. Lubbe R, Xu W-J, Zhou Q, Cheng H (2022) Bayesian calibration of GPU–based DEM meso-mechanics part ii: Calibration of the granular meso-structure. Powder Technol 407:117666. https://doi.org/10.1016/j.powtec.2022.117666
217. fr/MIDI/, G.M.: On dense granular flows. The Eur Phys J E 14, 341–365 (2004)
218. Cabrera M, Estrada N (2019) Granular column collapse: analysis of grain-size effects. Phys Rev E 99(1):012905
219. Sarlin W, Morize C, Sauret A, Gondret P (2021) Collapse dynamics of dry granular columns: from free-fall to quasistatic flow. Phys Rev E 104(6):064904
220. Polanía O, Estrada N, Azéma E, Renouf M, Cabrera M (2024) Polydispersity effect on dry and immersed granular collapses: an experimental study. J Fluid Mech 983:40
221. Lubbe R, Cheng H, Gupta P, Luding S, Magnanimo V (2024) A comparison of regime-specific continuum models for granular flows. In: Proceedings of the 11th international conference on conveying and handling of particulate solids (CHoPS 2024), Edinburgh, UK
222. Webb W, Turnbull B, Johnson C (2024) Continuum modelling of a just-saturated inertial column collapse: capturing fluid-particle interaction. Granular Matter 26(1):21
223. Gans A, Abramian A, Lagrée P-Y, Gong M, Sauret A, Pouliquen O, Nicolas M (2023) Collapse of a cohesive granular column. J Fluid Mech 959:41
224. Vantassel J, Kumar K (2022) Graph network simulator datasets. Designsafe-CI. https://doi.org/10.17603/DS2-0PHB-DG64. https://www.designsafe-ci.org/data/browser/public/designsafe.storage.published/PRJ-3702
225. Choi Y, Kumar K (2024) Graph neural network-based surrogate model for granular flows. Comput Geotech 166:106015
226. Vantassel J, Kumar K (2022) Graph network simulator datasets. v1. DesignSafe-CI 8(88):5025. https://doi.org/10.21105/joss.05025
227. Choi Y, Cheung SW, Kim Y, Tsai P-H, Diaz AN, Zanardi I, Chung SW, Copeland DM, Kendrick C, Anderson W, Iliescu T, Heinkenschloss M (2025) Defining foundation models for computational science: a call for clarity and rigor. arXiv:2505.22904
228. Martinez-Ortiz C, Martinez Lavanchy P, Sesink L, Olivier BG, Meakin J, Jong M, Cruz M (2022) Practical guide to software management plans. Zenodo. https://doi.org/10.5281/zenodo.7248877

Authors and Affiliations

Marc Fransen[1] · Andreas Fürst[2] · Deepak Tunuguntla[3] · Daniel N. Wilke[4,15] · Benedikt Alkin[2,16] · Daniel Barreto[5] · Johannes Brandstetter[2,16] · Miguel Angel Cabrera[6] · Xinyan Fan[6] · Mengwu Guo[7] · Bram Kieskamp[8] · Krishna Kumar[9] · John Morrissey[10] · Jonathan Nuttall[1] · Jin Ooi[10] · Luisa Orozco[11] · Stefanos-Aldo Papanicolopulos[10] · Tongming Qu[12] · Dingena Schott[6] · Takayuki Shuku[13] · WaiChing Sun[14] · Thomas Weinhart[8] · Dongwei Ye[8] · Hongyang Cheng[8]

✉ Hongyang Cheng
h.cheng@utwente.nl

[1] Deltares, Delft, The Netherlands

[2] Johannes Kepler University Linz, Linz, Austria

[3] Saxion University of Applied Sciences, Enschede, The Netherlands

[4] University of Pretoria, Pretoria, South Africa

[5] Edinburgh Napier University, Edinburgh, UK

[6] Delft University of Technology, Delft, The Netherlands

[7] Lund University, Lund, Sweden

[8] University of Twente, Enschede, The Netherlands

[9] University of Texas at Austin, Austin, USA

[10] School of Engineering, The University of Edinburgh, Edinburgh, UK

[11] Netherlands eScience Center, Amsterdam, The Netherlands

[12] Hong Kong University of Science and Technology, Kowloon, Hong Kong

[13] Tokyo City University, Tokyo, Japan

[14] Columbia University, New York, USA

[15] University of the Witwatersrand, Johannesburg, South Africa

[16] Emmi AI GmbH, Linz, Austria