Integrating Graph Neural Network-based Surrogate Modeling with Inverse Design for Granular Flows

Yu Jiang[a,*], Edmond Byrne[a], Jarka Glassey[a,b], Xizhong Chen[c,*]

[a]Department of Process and Chemical Engineering, School of Engineering and Architecture, University College Cork, Cork, T12 K8AF, Ireland

[b]Merz Court, School of Engineering, Newcastle University, Newcastle upon Tyne, NE1 7RU, United Kingdom

[c] Department of Chemical Engineering, School of Chemistry and Chemical Engineering, Shanghai Jiao Tong University, Shanghai 200240, PR China

[*]Corresponding author: 121106593@umail.ucc.ie, chenxizh@sjtu.edu.cn

Abstract

Granular flows are central to a wide range of natural phenomena and industrial processes such as landslides, industrial mixing, and material handling, and present intricate particle dynamics challenges. This study introduces a novel approach utilizing a Graph Neural Network-based Simulator (GNS) integrated with inverse design for optimizing Discrete Element Method (DEM) parameters in granular flow simulations. The GNS model, trained on datasets generated from high-fidelity DEM simulations, exhibits enhanced predictive accuracy and generalization capabilities across various materials and granular collapse scenarios. Methodologically, the study contrasts the GNS approach with conventional Design of Experiment (DoE) methods, highlighting its enhanced computational efficiency and dynamic optimization capacity for complex parameter interactions in granular flows. The results demonstrate the GNS method superiority over DoE in terms of computational speed and handling intricate parameter relationships. This work offers an advancement in computational techniques for granular flow studies, showing the potential of using differential simulations for realistic problems.

Keywords: Graph Neural Networks, Discrete Element Method, Granular Collapse, Inverse Design, Optimization Efficiency, Surrogate Model

1 Introduction

Understanding and accurately modeling granular material flow is a cornerstone challenge, with profound implications across diverse sectors such as pharmaceuticals, food science, and geotechnical engineering [1–3]. The complexity arises from the unique behavior of granular materials, which exhibit properties of both solids and fluids under different conditions. The dual nature poses significant challenges in predicting flow patterns, especially in processes like mixing, segregation, and transport in industrial settings. A fundamental aspect in these studies is the investigation of granular column collapse, a model experiment that encapsulates the critical characteristics of granular flow, including the angle of repose – a key property defining the stability and movement of granular materials [4]. The insights gained from such studies are instrumental for the early warning and risk mitigation of natural disasters like landslides and avalanches, as well as for the optimization of industrial processes involving granular materials [5,6]. By bridging the gap between

theoretical models and real-world applications, studies in this domain are essential for developing more accurate and reliable predictive tools for granular flow dynamics.

In the quest to model these complex flows, various simulation methods like the Material Point Method (MPM), Particle Finite Element Method, Smoothed Particle Hydrodynamics (SPH), and Discrete Element Method (DEM) have offered valuable insights [7–11]. Particularly, DEM has achieved good accuracy for modeling granular flows at a microscopic level if it is properly calibrated, which depends on the accurate adjustment of model parameters. Traditionally, this parameter calibration has been handled by Design of Experiment (DoE), a structured approach for systematically exploring the effects of various parameter settings [12–16]. Despite its effectiveness in parameter calibration, the approach of DoE does not fully encapsulate the intricate, nonlinear interactions characteristic of granular materials as it typically focuses on overarching simulation outcomes.

The recent application of machine learning in engineering science, motivated by the need for process modeling, prediction, design, optimization, and control, has led to significant advancement in this field [17–19]. In the evolving landscape of process engineering, concepts like digital twins and surrogate model have become pivotal, offering dynamic virtual replicas of physical systems. These models, widely applied across various sectors, enable quick response and prediction, enhancing efficiency in pharmaceutical manufacturing and energy management [20–23]. This study's exploration of surrogate models in granular flow modeling aligns with this trend, demonstrating the potential of digital models in addressing complex engineering challenges.

Building upon this advancements, the recent developments in graph neural network based simulators (GNS) [24], present a novel alternative to the calibration of model parameters. GNS has been used to build learned simulators, or surrogate models to simulate fluid dynamics, manipulate granular materials, and speed up the simulation of granular flows with a high degree of accuracy [25,26]. When integrated with gradient-based optimization, it has been successfully applied to inverse design tasks, achieving favorable results compared to specialized optimizer [27–29]. Until now, this approach has not been used for studying realistic processes with experimental validation. This research therefore seeks to explore this integration further, employing GNS to optimize DEM parameter configurations for an experiment-validated granular collapse process, with comparison to the conventional optimization based on DoE.

In this paper, a GNS-based surrogate model trained on a dataset derived from high-fidelity DEM simulations in two-dimensional (2D) space is employed to effectively capture and analyze granular material behaviors. By synergizing this model with an optimizer based on L-BFGS-B [30], it facilitates an iterative refinement of DEM parameters, aiming to more closely emulate experimental results [31]. This method stands in contrast to the purely DoE-based approach, in terms of computational efficiency, capability in capturing granular collapse behaviors of the current studied cases, and predictive optimization. The results show this approach is capable of iteratively enhancing the DEM simulations with better accuracy within less offline optimization time compared to DoE-based approach.

The paper is structured to first elaborate on the methodologies of GNS and DEM, highlighting their integration through inverse design. It is followed by a comparative analysis with the DoE-based approach and then concluded by a discussion on the broader impacts, potential applications, and future research directions in the field of optimization of granular dynamics and particulate processes.

2 Methodology

2.1 DEM method

In general, existing approaches to modeling granular materials can be classified into two categories: the continuum approach at a macroscopic level, and the discrete approach at a microscopic level [3]. For discrete approaches, the motion of individual particles is simulated. A principal type of discrete approach is called Discrete Element Method (DEM), which describes every particle in a system using Newton's equations of motion [32]. DEM-based simulation has been recognized as an effective method to study the fundamentals of granular materials and assess the performance of industrial processes [33–35].

In the context of this research, the governing equations for translational and rotational motions of particle *i* are as follows.

$$\frac{d\boldsymbol{x}_i}{dt} = \boldsymbol{v}_i \tag{1}$$

$$m_i \frac{d\boldsymbol{v}_i}{dt} = m_i \boldsymbol{g} + \boldsymbol{F}_{c,i} \tag{2}$$

$$I_i \frac{d\boldsymbol{\omega}_i}{dt} = \boldsymbol{T}_i \tag{3}$$

where $\boldsymbol{x}_i, m_i, I_i, \boldsymbol{v}_i, \boldsymbol{\omega}_i$ are, respectively, the position, mass, moment of inertia, translational and rotational velocities of the i^{th} particle. The forces involved are: gravitational force, $m_i \boldsymbol{g}$, and the net contact force with surrounding particles, $\boldsymbol{F}_{c,i}$. T_i is the sum of all torques acting on the i^{th} particle.

In DEM simulation of particulate flows, the contact model plays an important role to ensure the accuracy of calculation [36]. In the context of this research, linear spring-dashpot-friction method is used as the contact model in DEM simulations [37]. A constant normal spring stiffness of 1000 N/m is used for both particle-particle and particle-wall interactions. The ratio of tangential to normal spring stiffness is maintained at 2/7 for all interactions. The damping coefficient used follows a tangential to normal ratio of 0.5 for both types of interactions, ensuring consistency across all simulation cases. In this study, a fixed packing fraction value of approximately 0.58 for the initial quartz sand column and 0.5 for the copper sand column is used, as provided by the experimental work [31]. Friction and restitution coefficients are two important parameters of this linear spring-dashpot method.

The restitution coefficient, e, is a measure of the elasticity of a collision. It determines how much energy is conserved during the impact between particles or between a particle and a wall, thereby influencing the post-collision velocity in the normal direction. The friction coefficient, μ, determines the resistance to relative tangential (lateral) motion at the contact point between particles or between a particle and a boundary. The friction coefficient affects the rotational behavior and the post-collision trajectory of particles. Higher friction leads to more resistance to tangential motion, affecting how particles slide past each other.

In this research, both coefficients, e and μ, were varied to generate different cases used as ground-truth trajectories for training the GNS model.

2.2 Graph Neural Network Simulations

A network graph, which can be represented as $G = (V, E)$, consists of nodes and edges. Graphs have been used to represent particle-based systems. As granular collapse involves many particles, the system can be represented by a network graph. The nodes $v_i \in V$ represent the particles in the granular collapse, and the edges $e_{i,j} \in E$ connecting a pair of nodes v_i, v_j signify the interactions between the particles.

The Graph Network-based Simulator (GNS) comprises two main parts, a parameterized function approximator d_θ, including *Encoder, Processor* and *Decoder*, followed by an update function [24]. Fig. 1 illustrates the process of simulating granular collapse data using d_θ and the update function at a high level. In GNS, the physical state of the granular particles at time t_k is denoted as X^{t_k}. This state is then converted to a graph $G_{in} = (V, E)$ using a nearest neighbor algorithm, which serves as the input for the *Encoder*. The *Encoder* generates an output as a latent graph G_0, which is subsequently input into the *Processor*. Here, it undergoes M steps of message-passing graph-net (GN) layers, in each step the node and edge embeddings are updated. The updates changes G_l to G_{l+1}, where l increases from *0* to *M-1*. Eventually the *Processor* outputs G_M, a graph with the same graph structure as G_0 but with different node and edge embeddings. Finally, the *Decoder* processes G_M and outputs $G_{out} = (V', E')$, containing dynamic information for all particles (nodes), such as the acceleration of the particles denoted as Y^{t_k}. An update function subsequently computes the physical state of the particles for the next time step $X^{t_{k+1}}$.

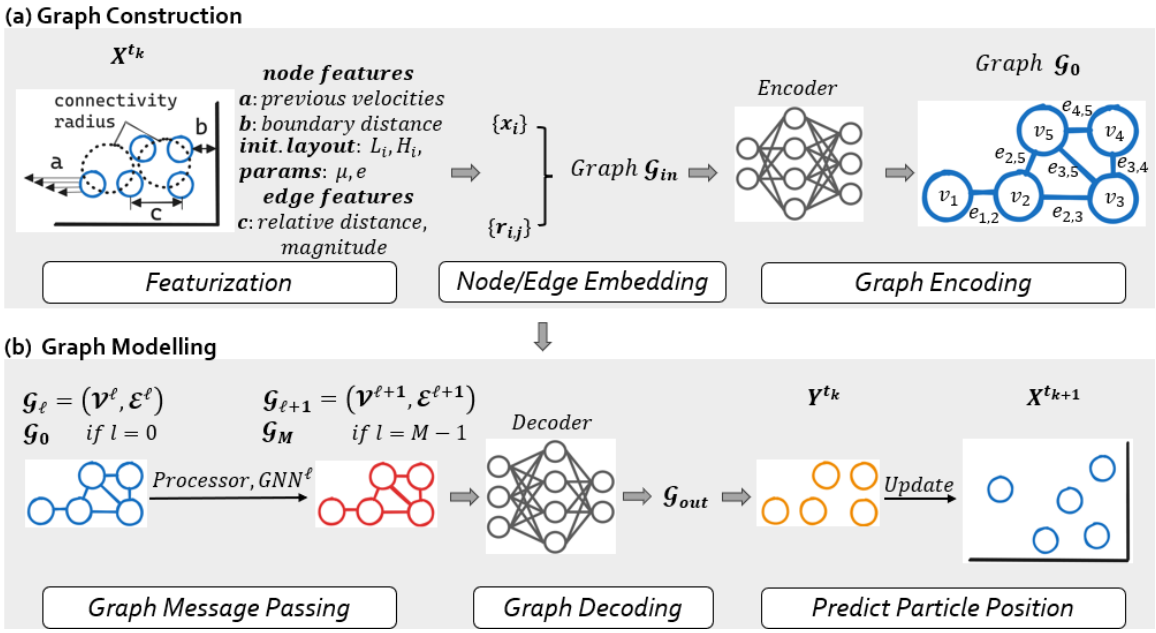


Figure 1: A GNS schematic (adapted from [24]) (a) Graph construction procedure (b) Graph modelling to predict particle position for next time step

To be more specific, the input state vector to the GNS model is shown in the top right part of Fig. 1a. The state of a particle i at time step t_k is a vector $x_i^{t_k}$, including position, velocity context, distance to the boundary, particle type of particle i, which can be concluded in eq (4).

$$x_i^{t_k} = \left[p_i^{t_k}, \dot{p}_i^{t_{k-C+1}}, \dots, \dot{p}_i^{t_k}, b_i^{t_k}, f_i, params, initial\ layout\right] \tag{4}$$

$p_i^{t_k}$ stands for the position information of particle i. The velocity context $\dot{p}_i^{t_{k-C+1}}, \dots, \dot{p}_i^{t_k}$ includes the current and previous velocities for C timesteps in total. $b_i^{t_k}$ represents the boundary information, which has four components for the 2D domain in this research, including the distance of particle i to the upper, bottom, left and right boundary. f_i is the embedding of the particle type, *params* stands for the corresponding parameter values for DEM model such as friction and restitution coefficients, *initial layout* represents the length L_i and height H_i of the initial column of particles.

An edge is built if two particles i and j are within a distance smaller than the *connectivity radius*. The edge features $r_{i,j}$ includes information about the interaction, including relative distance between these two particles and its magnitude, as shown in eq (5).

$$r_{i,j}^{t_k} = \left[(p_i^{t_k} - p_j^{t_k}), \left\|p_i^{t_k} - p_j^{t_k}\right\|\right] \tag{5}$$

Both $x_i^{t_k}$ and $r_{i,j}^{t_k}$ are input into two different two-layered 128-dimensional multi-layer perceptron (MLP) respectively, named as ε^v and ε^e, both constitutes the *Encoder*. The output of these two MLPs are two different latent vector of size 128, $v_i^{t_0}$ and $e_{i,j}^{t_0}$. Gathering all nodes and edges together leads to a latent graph $G_0 = (V_0, E_0)$.

The *Processor* includes M steps, with each step viewed as a GN block including two different two-layered 128-dimensional MLPs as well, each with 128-dimensional input and output layers. Specifically, the edge update MLP receives an input with dimensions Ne×384, and the node update MLP receives an input with dimensions Nv×256, where Ne and Nv denote the number of edges and nodes respectively. Each GN block takes in a latent graph and outputs the same structured latent graph with updated nodes embeddings V and edges embeddings E, conducting the message passing on these latent graphs to compute the particle interactions. After M steps of message passing, the input graph G_0 evolves to G_M, with enriched representations of particle dynamics.

The *Decoder* uses another MLP to decode the nodes embeddings V_M output by *Processor* to particle-wise accelerations $\ddot{p}_i^{t_k} \in Y^{t_k}$. These accelerations are then used to calculate future velocities and positions for each particle using Euler integration, as in eq (6) and (7).

$$\dot{p}_i^{t_{k+1}} = \dot{p}_i^{t_k} + \ddot{p}_i^{t_k}\Delta t \tag{6}$$

$$p_i^{t_{k+1}} = p_i^{t_k} + \dot{p}_i^{t_{k+1}}\Delta t \tag{7}$$

Based on the new position and velocity of the particles, $x_i^{t_k} \in X^{t_k}$ is updated to $x_i^{t_{k+1}} \in X^{t_{k+1}}$. The updated physical state $X^{t_{k+1}}$ is then used to predict the position and velocity for the next timestep. The entire simulation (Fig. 1b) involves running GNS surrogate model through K timesteps predicting from the initial state X^{t_0} to X^{t_K} $(X^{t_0}, X^{t_1}, \dots, X^{t_K})$, which is called the "rollout".

2.3 Inverse Design

Consider the optimization process depicted in Fig. 2. The goal is to find a pair of friction coefficient μ and restitution coefficient e, named as design parameters, that leads to best match between DEM simulation and experimental results in terms of runout distance of granular collapse. Runout distance L^* is a time dependent variable, defined as the dimensionless column length based on the length of the initial granular column $L^* = (L - L_i)/L_i$, where L is the time-dependent sand column length during the collapse. To evaluate the rollout, a loss function was used, which compared the runout distance L^* from both rollout and experiment using mean square error (MSE) across different time steps represented as t/t^* where $t^* = \sqrt{H_i/g}$ represents the free-fall time scale of the initial column. As depicted in Fig. 2, the optimization process starts from the black dot with loss value 0.66, where μ=0.55 and e=0.73, through several steps of optimization the loss reaches the optimized value 0.10 along the direction of those blue arrows, where μ=0.39 and e=0.54.

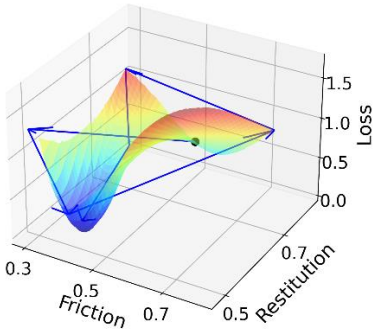


Figure 2: Sample optimization. Each step of optimization involves rolling out the simulation and then adjusting the designed DEM coefficients accordingly using an optimizer.

The pipeline of inverse design is depicted in Fig. 3: (1) embed design parameters to the initial input X^{t_0}, (2) rollout from the initial input using trained GNS surrogate model, which consists of $X^{t_0}, \widehat{X^{t_1}}, \ldots, \widehat{X^{t_K}}$, represented as $\hat{y}$, (3) evaluate the rollout by comparing predicted rollout $\hat{y}$ with experimental results as y, (4) adjust the design parameters. By automatically adjusting the friction coefficient and restitution coefficient, the DEM simulations were refined to $\hat{y}^*$, which matches best with the physical behavior of granular materials during collapse [31].

The design parameters were embedded as $params$ into the input of the GNS model, as shown in eq (4). To be more specific, it was embedded into the node embeddings V of the input graph $G_0 = (V, E)$, as shown in Fig. 1.

To adjust the design parameters, there are various optimization techniques available. In this research, the 'L-BFGS-B' approach implemented by Python library *scipy* was adopted, which is suitable for optimizing parameters that have physical bounds that they must stay within [38]. The optimization workflow is as follows:

1.Initialization: The process begins with initial design parameters, from which the mean squared error (MSE) between predicted and target runouts is calculated. These initial values serve as the starting point for optimization.

2.Gradient Approximation: By analyzing how slight modifications to parameters impact the MSE based on Eq. (8), the algorithm approximates the gradient of the MSE at a point (μ, e), which is the direction in which the parameters should be adjusted to minimize the MSE.

$$\nabla MSE(\mu, e) \approx \left(\frac{MSE(\mu+\epsilon, e) - MSE(\mu, e)}{\epsilon}, \frac{MSE(\mu, e+\epsilon) - MSE(\mu, e)}{\epsilon}\right) \quad (8)$$

where ϵ is a small number set to a default value typically around $1e^{-8}$, ensuring that the gradient approximation is both practical and reasonably accurate for optimization purposes.

3.Parameter Update: Utilizing its approximation of the gradient, the L-BFGS-B algorithm updates the design parameters towards values anticipated to reduce the MSE.

4.Iterative Refinement: Through iterative adjustments and evaluations, the algorithm refines its parameter estimates to minimize the objective function. This process involves a sophisticated mechanism that uses the algorithm's internal memory of past iterations to guide parameter updates efficiently.

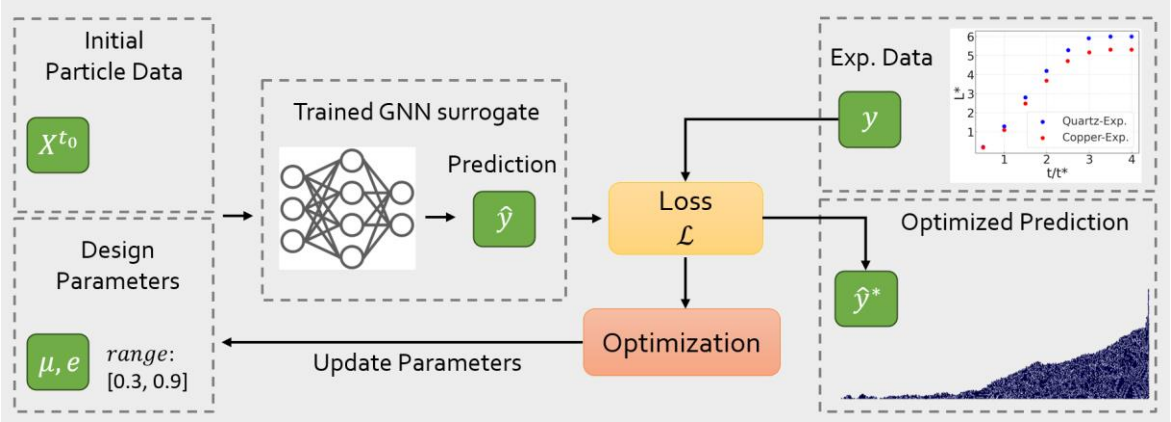


Figure 3: Schematic of the inverse design procedure for DEM parameters

2.4 Design of Experiment (DoE) for DEM Parameter Optimization

The Design of Experiment (DoE) method is a structured approach employed for optimizing parameters in the DEM simulations. DoE systematically explores and analyzes the effects of various parameter settings to optimize DEM models for more accurate simulations.

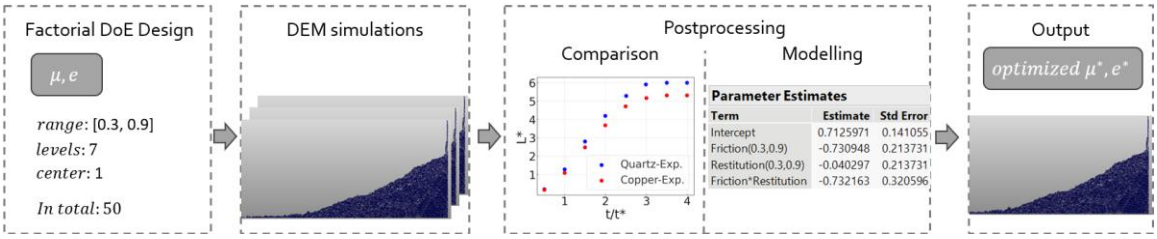


Figure 4: Schematic of DoE methodology for optimizing parameters in DEM simulations

The pipeline of DoE methodology is depicted in Fig. 4. Initially, key parameters influencing DEM simulation outcomes were identified, such as friction coefficient μ and restitution coefficient e. The range and levels for each parameter were established based on preliminary analysis. In this research, both parameters have 7 levels of values within the range of [0.3, 0.9], which means the values of μ and e are selected from {0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} for each design point in DoE where

"center" point is 0.6. A factorial DoE design was then selected to cover a broad spectrum of parameter combinations, ensuring a comprehensive exploration of the parameter space.

Once the DoE plan was set, a series of DEM simulations were run for each combination of parameters specified by the plan. This initial batch of simulations generated a dataset that reflects how different parameter settings affect simulation outcomes.

The results from these simulations were subjected to statistical analysis, typically using methods like Analysis of Variance. This analysis aimed to identify significant parameters and their interactions, informing the optimization process. As an output, DoE found the optimal set of parameters μ^*, e^* that best represent the real-world behavior of granular materials in DEM simulations.

In this study, the DoE-based method for DEM parameter optimization was utilized and compared with the GNS-based inverse design approach. The comparative analysis focused on efficiency, accuracy, and capability in handling complex simulation scenarios.

3 Experimental Setup

3.1 Sample Preparation

The target process stems from recent research on a granular collapse [31]. The same particle characteristics were used in this research, as listed in Table 1. Granular collapse was simulated by a DEM using the open-source MFiX software (version 23.2) on a desktop environment [39]. As with the literature, simulations were carried out in a computational domain of two-dimensional rectangular form of dimensions 500 x 350 mm.

Table 1. Properties of the materials simulated by DEM

Material	Size (μm)	Density (kg m^{-3})	Solid volume fraction
Copper sand	325	8960	50%
Quartz sand	425	2650	58%

In order to generate abundant data for various scenarios of granular collapse processes, a number of DEM simulation cases were conducted. These are summarized in Table 2.

For the GNS validation task, two datasets were generated corresponding to the two materials. For the quartz sand, 500 cases were simulated, each with different aspect ratios a (that is, the ratio between the initial height, H_i and length, L_i of the column) varying from 0.5 to 6, different friction coefficients, μ ranging from 0.3 to 0.9, and different initial column length, L_i. For the copper sand, 100 cases were simulated, each with same aspect ratio a = 5, but with different μ and L_i, as used for the quartz sand dataset.

For the inverse design task, a smaller dataset including 50 cases was generated for the quartz sand, with fixed initial length L_i and aspect ratio a, and varying friction coefficient μ and restitution coefficient e.

Table 2. DEM datasets configurations

Dataset	No. of training cases	Aspect ratio a	Initial length L_i	Friction μ	Restitution e
Quartz sand friction	500	0.5, 1, 2, 3, 4, 5, 6	[0.01, 0.02]	[0.3, 0.9]	Fixed, 0.65
Copper sand friction	100	Fixed, 5	[0.01, 0.02]	[0.3, 0.9]	Fixed, 0.65
Quartz sand multiple	50	Fixed, 6	Fixed, 0.02	[0.3, 0.9]	[0.3, 0.9]
Copper sand multiple	50	Fixed, 5	[0.01, 0.02]	[0.3, 0.9]	[0.3, 0.9]

3.2 Dataset Division

In this study, the datasets for both quartz sand and copper sand simulations were divided into training, validation, and testing subsets following an 8-1-1 split ratio. This distribution ensured comprehensive model training while retaining a substantial proportion of data for validation and unbiased testing.

Training Set: The training set, comprising 8 out of 10 of the datasets, was utilized to train the GNS model. This set included a wide range of scenarios covering different aspect ratios, friction coefficients and/or restitution coefficients, and initial column length.

Validation Set: Comprising 1 out of 10 of the datasets, the validation set was used to fine-tune the model parameters and prevent overfitting. This subset is crucial for adjusting the model during the training phase.

Testing Set: The remaining 1 out of 10 of the data, forming the testing set, was reserved for evaluating the model performance. This set is crucial for assessing the generalization capabilities of the model on unseen data.

The division strategy was chosen to ensure a balanced representation of different granular collapse scenarios across all subsets, facilitating robust model training and evaluation.

3.3 Model Configuration

The GNS model configuration involves setting up the values of its key parameters. Key aspects of the model configuration include:

Hyperparameters: Key hyperparameters such as the number of GN layers (M in Fig. 1), the *connectivity radius*, and the dimensionality of the hidden layers are optimized during the training process. An ablation study was conducted for *Quartz sand multiple* dataset in Table 2, of which the result is presented in Fig. 5. It shows that *connectivity radius* has a significant effect on the rollout MSE, in this research it was set at 0.004. The *message passing steps* and *input sequence length* were set at 5 and 6 respectively in the current research. It is worth noting that the first two hyperparameters were not selected based on the standard of MSE, but rather a balanced option between MSE and computation resources, as increased connectivity radius and message passing steps leads to larger graph that requires more computation resources to train the GNS model. Other hyperparameters are summarized in Table 3.

Training Procedure: The training for GNS models from each dataset involved at least 5 epochs, to ensure the model learns as much as from the training dataset. Regular validation checks were

performed every 500 training steps to monitor the model performance and that the best evaluated model was always stored throughout the training.

Repetitive Training: To avoid the effects of the network parameter initialization on the performance of the trained GNS models, the selected GNS models were trained twice independently for each combination of hyperparameters.

Software and Hardware Specifications: The GNS model was implemented using a deep learning framework PyTorch and trained on a Nvidia RTX 3060 GPU.

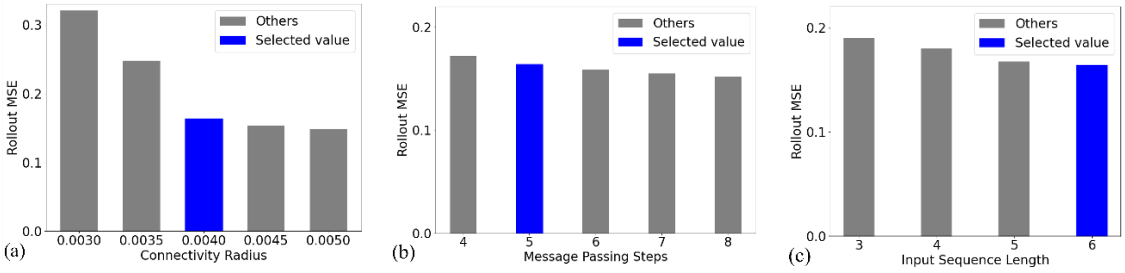


Figure 5. Effect of different ablations on the rollout error, a. Connectivity Radius; b. Message Passing steps; c. Input Sequence Length

Table 3. Hyperparameters of the GNS model

Module	Layers	Node MLP	Edge MLP
Encoder	Input	19	3
	Hidden	128, 2 layers	128, 2 layers
	Output	128	128
Processor	Input	256	384
(5 GN blocks)	Hidden	128, 2 layers	128, 2 layers
	Output	128	128
Decoder	Input	128	/
	Hidden	128, 2 layers	/
	Output	2	/

3.4 Evaluation Metrics

The evaluation of the GNS model performance is based on several metrics that collectively assess its accuracy and efficiency:

Mean Squared Error (MSE): MSE was used to quantify the difference between the predicted and actual runout distance of the granular collapse. It is used by the original GNS work, offering a quantitative measure of the model accuracy for the rollout process [24]. The use of MSE is also justified as it offers a clear and quantifiable way to assess the model precision in replicating the specific dynamics of granular collapse, which is central to the study's objective. It was also used in other granular system prediction models [40].

Computational Efficiency: The efficiency of the model in terms of simulation time and resource utilization was also evaluated. This metric is crucial for comparing the GNS model with traditional DEM simulations and other methods like DoE.

The chosen metrics provide a comprehensive evaluation of the model, ensuring its effectiveness and applicability in simulating granular material behaviors.

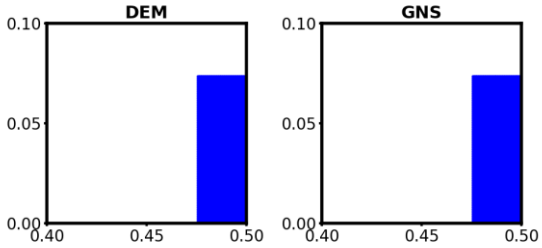
4. Results and Discussions

4.1. GNS Predictions of Granular Collapse

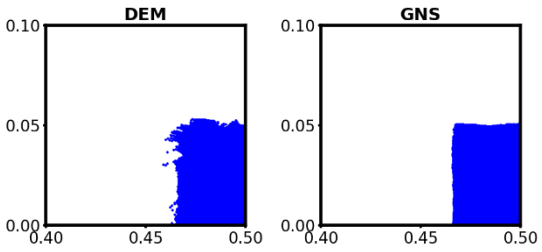
In this study, the predictive accuracy and efficiency of the GNS model was assessed for granular column collapse. The GNS model was validated against DEM simulations, considering various materials and analyzing the temporal evolutions of runout. This comparison highlights the model capacity for generalization and underscores its potential in the inverse design for optimizing DEM parameters.

4.1.1 Quartz Sand

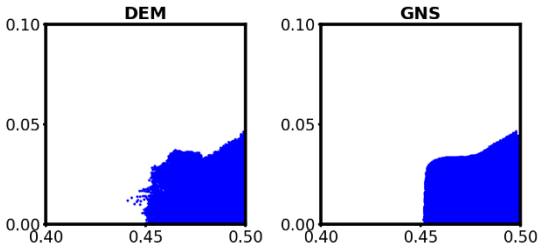
For quartz sand collapse, the time coordinate was normalized by the free-fall time scale $t^* = \sqrt{H_i/g}$. The free-fall time scale t^* serves as a critical normalization factor, providing a standardized reference for comparing the collapse dynamics across different scenarios. In Fig. 6, the collapse process is depicted through both DEM simulations and GNS predictions for a column with a height-length ratio a = 3.0. The progression of normalized time, as shown in (Fig. 6(a)-(e)), reveals a good visual and qualitative agreement between DEM and GNS results, validating the GNS model precision in simulating the granular collapse process. The collapse stops at t/t* = 4.



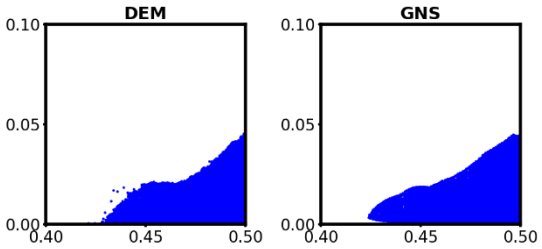
(a) t/t* = 0



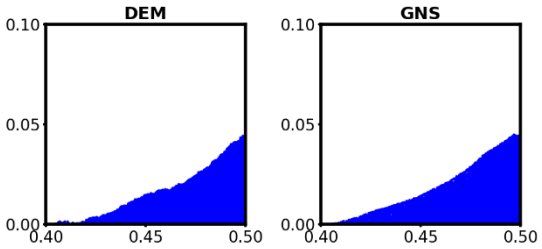
(b) t/t* = 1.0



(c) t/t* = 1.5



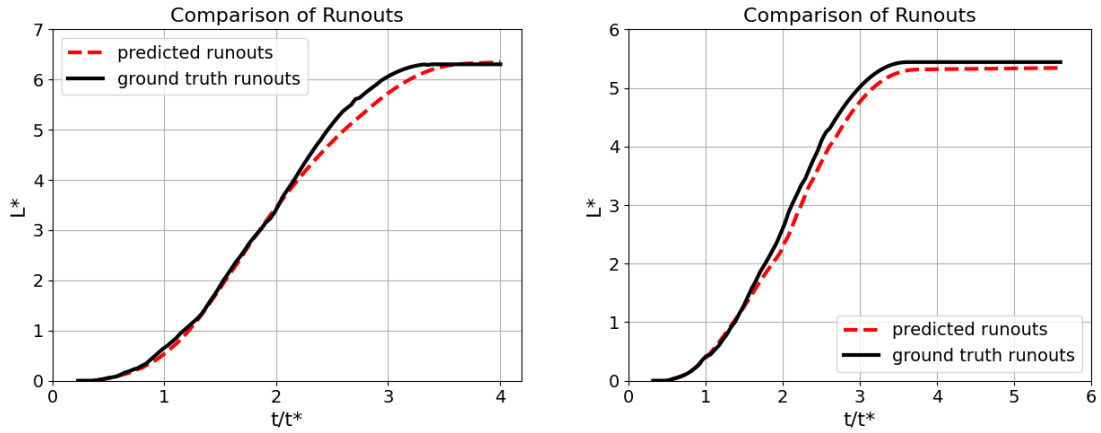
(d) t/t* = 2



(e) t/t* = 4

Figure 6: Evolution of flow with normalized time for GNS and DEM for the quartz sand, with initial aspect ratio = 3.

A deeper quantitative analysis in Fig. 7a underscores the model's effectiveness of the trained model in capturing the kinetic profile of the collapse. At each time step, the average runout distance of 8 leading particles is used to compute the dimensionless column length L^*. The temporal analysis, validated by a mean square error (MSE) of 0.172, indicates that the collapse process can be segmented into three stages: the initial fast-moving free-fall phase ($t/t^* \leqslant 1.0$), a rapid spreading phase ($1.0 < t/t^* \leqslant 2.5$), and the final stopping phase ($2.5 < t/t^* \leqslant 4.0$). In general, all the three phases were well captured while the GNS slightly underestimated the runouts compared with DEM.

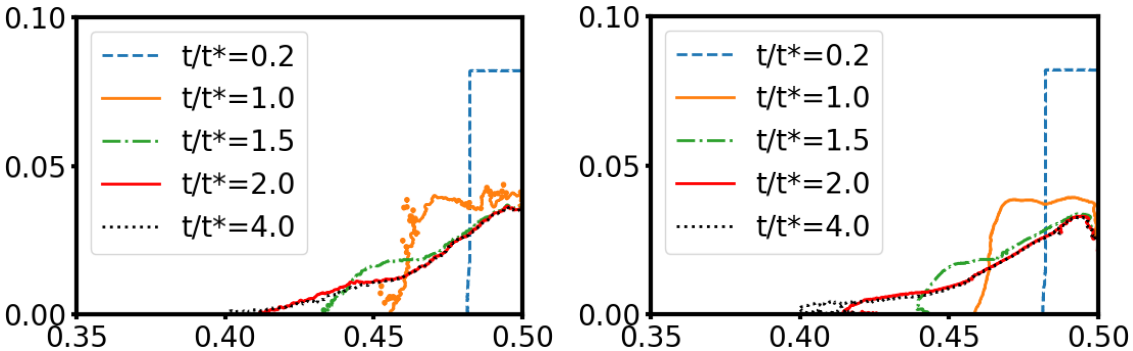


(a) Quartz sand (b) Copper sand

Figure 7: A comparison between the DEM and GNS results of the evolution of the column runout, $L^* = (L - L_i)/L_i$, where L is the time-dependent sand column length during the collapse. (a). Quartz sand; (b). Copper sand.

4.1.2 Copper sand

The adaptability of the GNS model is further exemplified in its predictions for copper sand collapses. Given the inherent differences in particle size and density compared to quartz sand, the copper sand collapse exhibits a different runout evolution. As an example, here the GNS prediction for a tall column with a = 5.0 is presented in Fig. 8. The model captures not only the overall trend of the runout but also the intricate details of the collapse at various stages.



(a) DEM rollout simulation (b) GNS rollout prediction

Figure 8: Evolution of flow with normalized time for GNS and DEM for the copper sand, with initial aspect ratio = 5, different color represents t/t* ranging from 0.2, 1.0, 1.5, 2.0, 4.0.

In Fig. 7b, the quantitative narrative of the runout evolution is presented. Similar to the quartz sand, the copper sand collapse follows the three-phase pattern. The GNS model results align well with the DEM results for copper sand, with an MSE of 0.167 between the two curves, confirming the high fidelity of the model in predicting granular collapse dynamics.

4.2 Inverse Design of Parameters of the DEM model

The efficacy of the GNS model, when combined with an optimizer, offers an excellent surrogate model for carrying out quick optimization and design tasks in realistic problems. As a first

demonstration, this study investigated the calibration of DEM parameter designs. All the results in this section were validated by the actual runout distances measured from experiment [31].

4.2.1 GNS-based Optimization for Friction and Restitution Coefficients

The integration of the learned simulator with a gradient-based optimizer has proven to be a powerful tool in design optimization [27]. Fig. 9 visualizes this iterative optimization process for quartz sand collapse with a = 6.0, which begins with preset initial values for the friction and restitution coefficients. The grey vertical dashed line in Fig. 9 marks the final runout $L_f^* = 0.154$ from experiment, positioning this line at a value of 0.346. The blue particles in Fig. 9 represent the final profile when t/t*=3.5. In the inverse design process, the predicted runout profile is compared to the target runout profile at 7 timesteps, where t/t* varies from [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5], to derive the MSE corresponding to the values for μ, e for that optimization step.

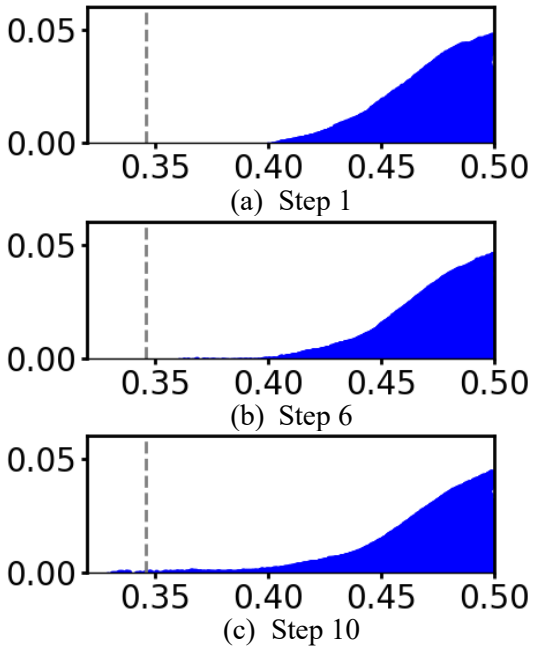
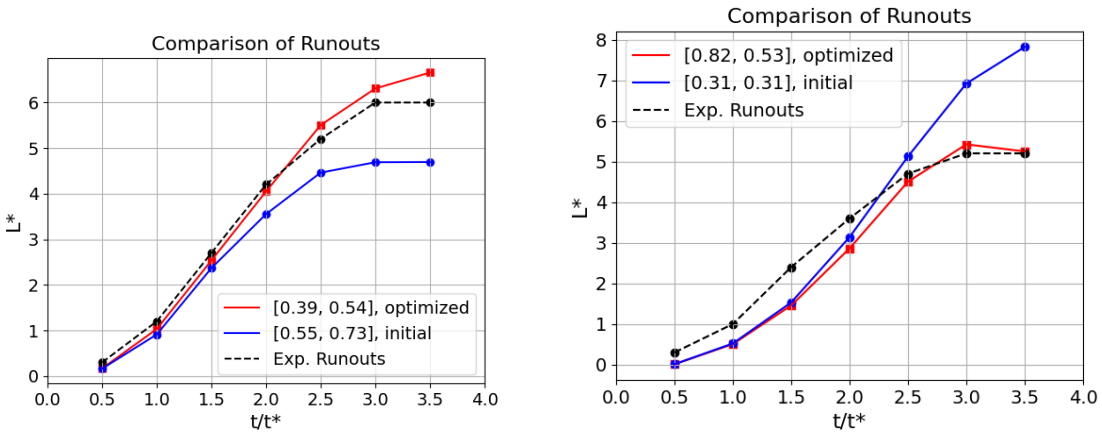


Figure 9: Evolution of friction and restitution coefficient found by GNS inverse design during optimization for quartz sand collapse task, (a) initial value, μ = 0.55, e =0.73 (b) values after 5 steps, μ = 0.45, e =0.54, (c) final (step 10) optimized value, μ = 0.39, e =0.54

Through successive iterations, the GNS model converged to an optimal coefficient pair of [0.39, 0.54]. With the final dimensionless runout distance L^*_f in Fig. 9c reaches 6.45, closely matching the experimental final runout distance of 6.0, the approach demonstrates its capability in parameter optimization. This close agreement is also visualized in Fig. 10a, where the GNS-optimized runout curve is compared with experimental observations. This optimization process notably improved the model accuracy, reducing the MSE from 0.66 to 0.10 based on experimental data. This substantial reduction illustrates the effectiveness of the GNS model in enhancing multi-parameter prediction precision. It is worth noting that the true values for μ and e are not available from the referenced paper [31], which motivates the current research to focus on comparison of the runout distance evolution.



(a) Quartz sand (b) Copper sand

Figure 10: A comparison of the GNS results of the evolution of the runout before and after optimization derived by GNS-based optimization with experimental data (Exp. Runouts). (a) Quartz sand. (b) Copper sand.

Additionally, the optimization of both DEM parameters for copper sand granular collapse was conducted. As displayed in Fig. 10b, the final optimized runout is approximately 5, smaller than that observed in the quartz sand granular collapse. The optimized coefficients lead to larger improvement compared to the initial values, with the MSE of GNS model predictions reduced from 1.61 to 0.26 based on experimental data.

4.2.2 Comparison with Design of Experiment Method

This section compares the inverse design speed using the learned simulator combined with an optimizer to another design method, Design of Experiment (DoE).

Besides the GNS-based inverse design, a full factorial DoE approach was applied to optimize the two DEM parameters, each tested at 7 levels ranging from 0.3 to 0.9. With an additional center testing point, a total of 50 cases were designed and run in DEM simulations. Table 4 summarizes the effect of both coefficients and their interaction on the accuracy of simulated runout distance evolution, where Log-Worth represents the negative logarithm (base 10) of the P-Value for easy interpretation. Main effects were observed for both factors, with friction exhibiting a higher statistical significance (P-Value = 0.00132) than restitution (P-Value = 0.85). An R^2 value of 0.27 implies that a large proportion of the variability in the response variable is not explained by the model, which means that the model is not capturing much of the complexity inherent in the data. The prediction expression is expressed in Eq. (9), from which the optimized parameter pair derived is [0.85, 0.64].

Table 4. Effect summary of the DEM parameters and their interaction on the MSE of runout distance

Source	Log-Worth	P-Value
Friction(0.3, 0.9)	2.879	0.00132
Friction*Restitution	1.568	0.02705
Restitution (0.3, 0.9)	0.070	0.85128

$$MSE = 0.713 - 0.731 \cdot \left(\frac{\mu - 0.6}{0.3}\right) - 0.04\left(\frac{e - 0.6}{0.3}\right) - 0.732 \cdot \left(\frac{\mu - 0.6}{0.3}\right) \cdot \left(\frac{e - 0.6}{0.3}\right) \quad (9)$$

In Fig. 11, the evolution of runout distance determined by the DoE optimization is compared with the experimental results. In the early stages of granular collapse, the DEM simulations using the DoE-optimized parameters capture the process as effectively as those using the GNS-optimized parameters. This observation is supported by the similarity between the first two data points in Figs.11 and 10a. However, in the final stages, there is a less accurate match between the experimental data and the DoE-optimized DEM simulation results, with an MSE of 0.17. The MSE is higher than that observed in the GNS outcomes depicted in Fig. 10a. This is due to the inaccuracy of the model used to capture the relationship between MSE and μ, e, which is a quadratic polynomial function as described in Eq. (9).

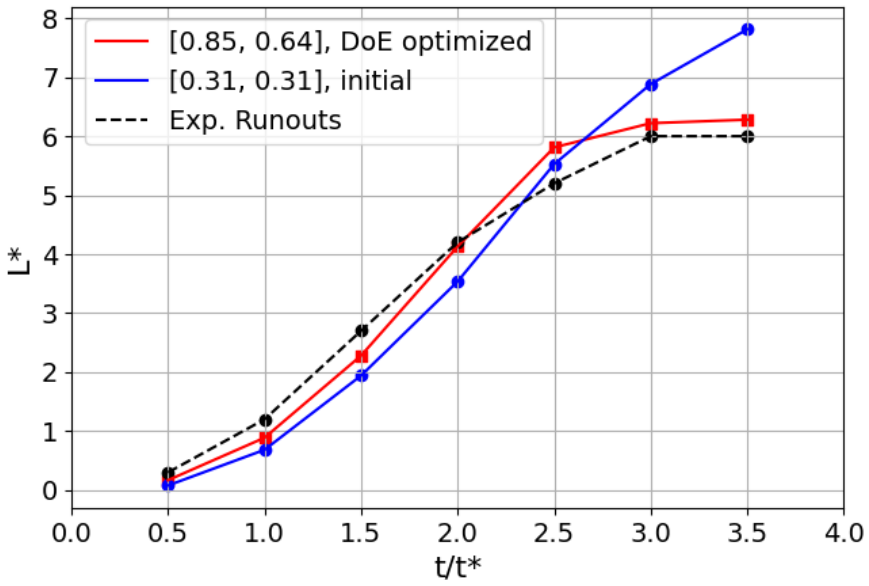


Figure 11: A comparison of evolution of the Quartz sand runout before and after optimization derived by DoE-based optimization with experimental data (Exp. Runouts)

Table 5 shows the quantitative comparison of the time cost in both design methods. The GNS method divides the inverse design process into two stages: offline (model training) and online (parameter inverse design). Once the GNS method is adequately trained, its online phase demonstrates superior efficiency over the DoE method, despite requiring the same number of simulation cases and additional training time initially, as illustrated in Table 5.

Table 5. Comparison of time and resource cost for both design approaches

Method(stage)	Cases	Resources	Task	Time
GNS (offline)	50	i5-12500K RTX 3060	dataset generation,	~8 hours

GNS (online)	/	RTX 3060	GNS model training optimization	~2 hours ~15 minutes
DoE	50	i5-12500K	dataset generation, optimization	~ 8 hours

The highly efficient online stage of GNS approach enables rapid inverse design, meeting the urgent demands of various industrial applications. Its efficacy in computational resource and time management underscores its significant potential for practical, real-world simulations and optimization in granular material studies.

5 Conclusion

This study has successfully demonstrated the efficacy of a Graph Neural Network-based Simulator (GNS) in predicting granular collapse dynamics and its integration with inverse design methods for optimizing Discrete Element Method (DEM) parameters. Our findings reveal that the GNS model exhibits high predictive accuracy for different materials and collapse scenarios. The model's ability to efficiently converge to optimal parameter values, as evidenced in both quartz sand and copper sand simulations, showcases its potential in refining DEM simulations for granular flows. Notably, the GNS approach, compared to traditional Design of Experiment (DoE) methods, offers superior computational efficiency and a more dynamic, continuous parameter optimization process. Additionally, this trained model could be used as a basis of further training once more high-fidelity DEM simulation data are available, thus improving the generalizability and predictability of the model.

While the study presents promising results, it is not without limitations. One significant challenge lies in the dependency on extensive training data for the GNS model in the offline stage of surrogate model development, which necessitates a considerable amount of high-fidelity DEM simulations. As with other applications of machine learning, the performance of the trained model is contingent on the quality of training data.

Looking ahead, the potential applications of GNS in industrial and research settings are extensive, especially in fields requiring rapid and accurate simulations of granular material behavior. Future work could focus on enhancing the GNS model training efficiency, perhaps by exploring more advanced machine learning techniques or hybrid models to handle time sequence information of the trajectories more efficiently [41–43]. Further research could also delve into expanding the model applicability to three-dimensional simulations and more complex material interactions that come along with polydisperse particle systems, enhance the model by incorporating a broader range of material properties like particle packing fraction, non-spherical particle shapes and complex domain boundaries [44].

Ultimately, the continued development of GNS models holds the promise of revolutionizing the way we simulate and understand granular flows, with wide-ranging implications for both industrial processes and natural phenomenon prediction. On top of that, the potential application of GNS-based inverse design is also extensive, including the hopper shape design [45], the design of pneumatic conveying process [33], particle drying process optimization [34], etc.

Nomenclature

Symbol	Explanation
$\boldsymbol{g}$	gravitational acceleration
$\boldsymbol{x}_i$	position of the i^{th} particle
m_i	mass of the i^{th} particle
I_i	moment of inertia of the i^{th} particle
$\boldsymbol{v}_i$	translational velocity of the i^{th} particle
$\boldsymbol{\omega}_i$	rotational velocity of the i^{th} particle
$\boldsymbol{F}_{c,i}$	net contact force with surrounding particles
T_i	sum of all torques acting on the i^{th} particle
e	restitution coefficient
μ	friction coefficient
G	graph representation of the particles
V	all the nodes/vertices of the graph G
E	all the edges of the graph G
v_i	node embedding for i^{th} node
$e_{i,j}$	edge embedding for the edge connecting a pair of nodes v_i, v_j
d_θ	parameterized function approximator
t_k	k^{th} time step of a sequence of data, k from 0 to K
X^{t_k}	the physical states of the granular particles at time t_k
Y^{t_k}	dynamics information of the particles at time t_k
$x_i^{t_k}$	the state of particle i at time step t_k
$r_{i,j}^{t_k}$	relative information of particles i, j at time step t_k
$p_i^{t_k}$	position information of particle i at time step t_k
$\dot{p}_i^{t_k}$	velocity of particle i at time step t_k
$\ddot{p}_i^{t_k}$	acceleration of particle i at time step t_k
$b_i^{t_k}$	the boundary information of particle i at time step t_k
f_i	particle type of particle i
L_i	length of the initial column of particles
H_i	height of the initial column of particles
ε^v	node embedding multilayer perceptron
ε^e	edge embedding multilayer perceptron
Nv	total number of nodes
Ne	total number of edges
L	time-dependent sand column length
L^*	time-dependent dimensionless sand column length
t^*	the free-fall time scale of the initial column
y	experimental results of runout distance
$\hat{y}$	rollout from the initial input using trained GNS surrogate model, which consists of $X^{t_0}, \widehat{X^{t_1}}, \ldots, \widehat{X^{t_K}}$
$\hat{y}^*$	refined rollout prediction by the trained GNS surrogate model
μ^*	optimal friction coefficient derived from inverse design process
e^*	optimal restitution coefficient derived from inverse design process
$\underline{a}$	the ratio between H_i and L_i of the initial column

CRediT authorship contribution statement

Yu Jiang: Conceptualization, Data curation, Methodology, Formal analysis, Visualization, Writing – original draft. **Edmond Byrne:** Supervision, Writing – review and editing. **Jarka Glassey:** Supervision, Writing – review and editing. **Xizhong Chen:** Supervision, Conceptualization, Writing – review and editing.

Acknowledgements

The financial supports from University College Cork Eli Lilly Research Scholarships are acknowledged gratefully. X. Chen gratefully acknowledge the financial support from National Natural Science Foundation of China Excellent Young Scientists Fund Program (Overseas), National Natural Science Foundation of China (grant no. 22308212).

Data Availability

The code that supports the findings of this study are openly available in Github at https://github.com/uccproc/gns4demdesign

References

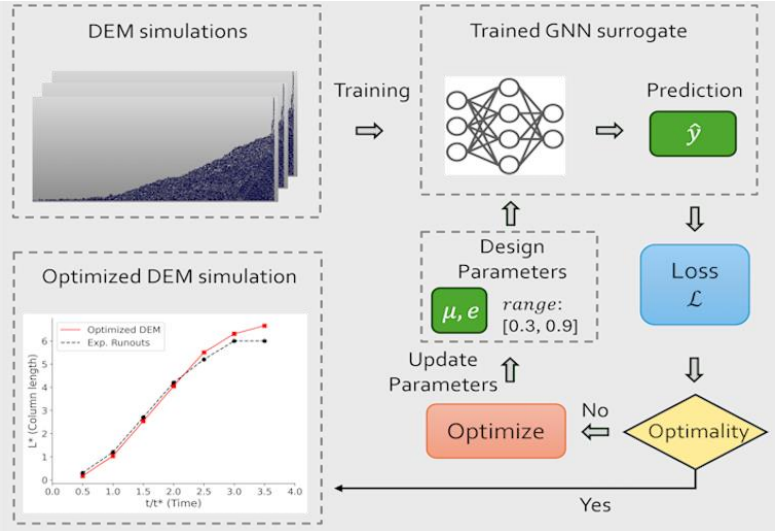
(1) Andreotti, B.; Forterre, Y.; Pouliquen, O. *Granular Media: Between Fluid and Solid*; Cambridge University Press, 2013.
(2) Duran, J. Granular Media in a State of Flow. In *Sands, Powders, and Grains: An Introduction to the Physics of Granular Materials*; Duran, J., Ed.; Partially Ordered Systems; Springer: New York, NY, 2000; pp 119–153. https://doi.org/10.1007/978-1-4612-0499-2_4.
(3) Yu, A. B. Powder Processing—Models and Simulations. In *Encyclopedia of Condensed Matter Physics (Second Edition)*; Chakraborty, T., Ed.; Academic Press: Oxford, 2024; pp 679–693. https://doi.org/10.1016/B978-0-323-90800-9.00118-9.
(4) Jing, L.; Yang, G. C.; Kwok, C. Y.; Sobral, Y. D. Flow Regimes and Dynamic Similarity of Immersed Granular Collapse: A CFD-DEM Investigation. *Powder Technol.* **2019**, *345*, 532–543. https://doi.org/10.1016/j.powtec.2019.01.029.
(5) Hoang, U. T.; Nguyen, N. H. T. Particle Shape Effects on Granular Column Collapse Using Superquadric DEM. *Powder Technol.* **2023**, *424*, 118559. https://doi.org/10.1016/j.powtec.2023.118559.
(6) Løvholt, F.; Bondevik, S.; Laberg, J. S.; Kim, J.; Boylan, N. Some Giant Submarine Landslides Do Not Produce Large Tsunamis. *Geophys. Res. Lett.* **2017**, *44* (16), 8463–8472. https://doi.org/10.1002/2017GL074062.
(7) Larsson, S.; Rodríguez Prieto, J. M.; Gustafsson, G.; Häggblad, H.-Å.; Jonsén, P. The Particle Finite Element Method for Transient Granular Material Flow: Modelling and Validation. *Comput. Part. Mech.* **2021**, *8*, 135–155. https://doi.org/10.1007/s40571-020-00317-6.
(8) Lube, G.; Huppert, H. E.; Sparks, R. S. J.; Freundt, A. Collapses of Two-Dimensional Granular Columns. *Phys. Rev. E* **2005**, *72* (4), 041301. https://doi.org/10.1103/PhysRevE.72.041301.

(9) Mast, C. M.; Arduino, P.; Mackenzie-Helnwein, P.; Miller, G. R. Simulating Granular Column Collapse Using the Material Point Method. *Acta Geotech.* **2015**, *10* (1), 101–116. https://doi.org/10.1007/s11440-014-0309-0.
(10) Seelen, L. J. H.; Padding, J. T.; Kuipers, J. A. M. A Granular Discrete Element Method for Arbitrary Convex Particle Shapes: Method and Packing Generation. *Chem. Eng. Sci.* **2018**, *189*, 84–101. https://doi.org/10.1016/j.ces.2018.05.034.
(11) Wang, G.; Riaz, A.; Balachandran, B. Smooth Particle Hydrodynamics Studies of Wet Granular Column Collapses. *Acta Geotech.* **2020**, *15* (5), 1205–1217. https://doi.org/10.1007/s11440-019-00828-4.
(12) Ben Turkia, S.; Wilke, D. N.; Pizette, P.; Govender, N.; Abriak, N.-E. Benefits of Virtual Calibration for Discrete Element Parameter Estimation from Bulk Experiments. *Granul. Matter* **2019**, *21* (4), 110. https://doi.org/10.1007/s10035-019-0962-y.
(13) Boikov, A. V.; Savelev, R. V.; Payor, V. A. DEM Calibration Approach: Design of Experiment. *J. Phys. Conf. Ser.* **2018**, *1015*, 032017. https://doi.org/10.1088/1742-6596/1015/3/032017.
(14) Chen, X.; Pei, C.; Elliott, J. A. A "Sequential Design of Simulations" Approach for Exploiting and Calibrating Discrete Element Simulations of Cohesive Powders. *Front. Chem. Sci. Eng.* **2022**, *16* (6), 874–885. https://doi.org/10.1007/s11705-021-2131-1.
(15) Forgber, T.; Khinast, J. G.; Fink, E. A Hybrid Workflow for Investigating Wide DEM Parameter Spaces. *Powder Technol.* **2022**, *404*, 117440. https://doi.org/10.1016/j.powtec.2022.117440.
(16) Zhang, S.; Tekeste, M. Z.; Li, Y.; Gaul, A.; Zhu, D.; Liao, J. Scaled-up Rice Grain Modelling for DEM Calibration and the Validation of Hopper Flow. *Biosyst. Eng.* **2020**, *194*, 196–212. https://doi.org/10.1016/j.biosystemseng.2020.03.018.
(17) Hwang, S.; Pan, J.; Sunny, A. A.; Fan, L.-S. A Machine Learning-Based Particle-Particle Collision Model for Non-Spherical Particles with Arbitrary Shape. *Chem. Eng. Sci.* **2022**, *251*, 117439. https://doi.org/10.1016/j.ces.2022.117439.
(18) Ren, Y. M.; Alhajeri, M. S.; Luo, J.; Chen, S.; Abdullah, F.; Wu, Z.; Christofides, P. D. A Tutorial Review of Neural Network Modeling Approaches for Model Predictive Control. *Comput. Chem. Eng.* **2022**, *165*, 107956. https://doi.org/10.1016/j.compchemeng.2022.107956.
(19) van de Berg, D.; Savage, T.; Petsagkourakis, P.; Zhang, D.; Shah, N.; del Rio-Chanona, E. A. Data-Driven Optimization for Process Systems Engineering Applications. *Chem. Eng. Sci.* **2022**, *248*. https://doi.org/10.1016/j.ces.2021.117135.
(20) Ghosh, D.; Datta, A. Deep Learning Enabled Surrogate Model of Complex Food Processes for Rapid Prediction. *Chem. Eng. Sci.* **2023**, *270*, 118515. https://doi.org/10.1016/j.ces.2023.118515.
(21) Mora-Mariano, D.; Flores-Tlacuahuac, A. Bayesian LSTM Framework for the Surrogate Modeling of Process Engineering Systems. *Comput. Chem. Eng.* **2024**, *181*, 108553. https://doi.org/10.1016/j.compchemeng.2023.108553.
(22) Wu, Z.; Wang, H.; He, C.; Zhang, B.; Xu, T.; Chen, Q. The Application of Physics-Informed Machine Learning in Multiphysics Modeling in Chemical Engineering. *Ind. Eng. Chem. Res.* **2023**, *62* (44), 18178–18204. https://doi.org/10.1021/acs.iecr.3c02383.
(23) Zhu, L.-T.; Chen, X.-Z.; Bo, O.; Yan, W.-C.; Lei, H.; Chen, Z.; Luo, Z.-H. Review of Machine Learning for Hydrodynamics, Transport, and Reactions in Multiphase Flows and Reactors. *Ind. Eng. Chem. Res.* **2022**, *61* (28), 9901–9949. https://doi.org/10.1021/acs.iecr.2c01036.
(24) Sanchez-Gonzalez, A.; Godwin, J.; Pfaff, T.; Ying, R.; Leskovec, J.; Battaglia, P. Learning to Simulate Complex Physics with Graph Networks. In *Proceedings of the 37th International Conference on Machine Learning*; PMLR, 2020; pp 8459–8468.

(25) Tuomainen, N.; Blanco-Mulero, D.; Kyrki, V. Manipulation of Granular Materials by Learning Particle Interactions. *IEEE Robot. Autom. Lett.* **2022**, *7* (2), 5663–5670. https://doi.org/10.1109/LRA.2022.3158382.
(26) Choi, Y.; Kumar, K. Graph Neural Network-Based Surrogate Model for Granular Flows. *Comput. Geotech.* **2024**, *166*, 106015. https://doi.org/10.1016/j.compgeo.2023.106015.
(27) Allen, K. R.; Lopez-Guevara, T.; Stachenfeld, K.; Sanchez-Gonzalez, A.; Battaglia, P.; Hamrick, J.; Pfaff, T. Physical Design Using Differentiable Learned Simulators. arXiv February 1, 2022. http://arxiv.org/abs/2202.00728 (accessed 2023-01-08).
(28) Choi, Y.; Kumar, K. Inverse Analysis of Granular Flows Using Differentiable Graph Neural Network Simulator. arXiv February 9, 2024. http://arxiv.org/abs/2401.13695 (accessed 2024-03-29).
(29) Kumar, K.; Choi, Y. 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*; ACM: Denver CO USA, 2023; pp 60–65. https://doi.org/10.1145/3624062.3626082.
(30) Zhu, C.; Byrd, R. H.; Lu, P.; Nocedal, J. Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization. *ACM Trans. Math. Softw.* **1997**, *23* (4), 550–560. https://doi.org/10.1145/279232.279236.
(31) Biroun, M. H.; Sorensen, E.; Hilden, J. L.; Mazzei, L. CFD Modelling of Powder Flow in a Continuous Horizontal Mixer. *Powder Technol.* **2023**, *428*, 118843. https://doi.org/10.1016/j.powtec.2023.118843.
(32) Herrmann, H. J.; Hovi, J.-P.; Luding, S. *Physics of Dry Granular Media*; Springer Science & Business Media, 2013.
(33) Kuang, S.; Zhou, M. CFD-DEM Modelling and Simulation of Pneumatic Conveying: A Review. *Powder Technol.* **2020**, *365*, 186–207. https://doi.org/10.1016/j.powtec.2019.02.011.
(34) Lan, B.; Zhao, P.; Xu, J.; Zhao, B.; Zhai, M.; Wang, J. CFD-DEM-IBM Simulation of Particle Drying Processes in Gas-Fluidized Beds. *Chem. Eng. Sci.* **2022**, *255*, 117653. https://doi.org/10.1016/j.ces.2022.117653.
(35) Zhou, Q.; Xu, W.-J.; Dong, X.-Y. SPH-DEM Coupling Method Based on GPU and Its Application to the Landslide Tsunami. Part I: Method and Validation. *Acta Geotech.* **2022**, *17* (6), 2101–2119. https://doi.org/10.1007/s11440-021-01388-2.
(36) Jordam Caserta, A.; Navarro, H. A.; Cabezas-Gómez, L. Damping Coefficient and Contact Duration Relations for Continuous Nonlinear Spring-Dashpot Contact Model in DEM. *Powder Technol.* **2016**, *302*, 462–479. https://doi.org/10.1016/j.powtec.2016.07.032.
(37) Cundall, P. A.; Strack, O. D. L. A Discrete Numerical Model for Granular Assemblies. *Géotechnique* **1979**, *29* (1), 47–65. https://doi.org/10.1680/geot.1979.29.1.47.
(38) Dalvand, Z.; Hajarian, M. Solving Generalized Inverse Eigenvalue Problems via L-BFGS-B Method. *Inverse Probl. Sci. Eng.* **2020**, *28* (12), 1719–1746. https://doi.org/10.1080/17415977.2020.1763982.
(39) NETL Multiphase Flow Science Team. *MFiX User Manual — MFiX 23.2 documentation*. https://mfix.netl.doe.gov/doc/mfix/23.2/html/index.html (accessed 2024-03-26).
(40) Zhang, H.; Li, X.; Li, Z.; Huang, D.; Zhang, L. Estimation of Particle Location in Granular Materials Based on Graph Neural Networks. *Micromachines* **2023**, *14* (4), 714. https://doi.org/10.3390/mi14040714.
(41) Han, X.; Gao, H.; Pfaff, T.; Wang, J.-X.; Liu, L.-P. Predicting Physics in Mesh-Reduced Space with Temporal Attention. arXiv May 26, 2022. http://arxiv.org/abs/2201.09113 (accessed 2022-08-04).
(42) Yu, Y.-Y.; Choi, J.; Cho, W.; Lee, K.; Kim, N.; Chang, K.; Woo, C.; Kim, I.; Lee, S.; Yang, J. Y.; Yoon, S.; Park, N. Learning Flexible Body Collision Dynamics with Hierarchical

Contact Mesh Transformer. arXiv December 19, 2023. http://arxiv.org/abs/2312.12467 (accessed 2024-01-21).
(43) Yun, S.; Jeong, M.; Kim, R.; Kang, J.; Kim, H. J. Graph Transformer Networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2019; Vol. 32.
(44) Mayr, A.; Lehner, S.; Mayrhofer, A.; Kloss, C.; Hochreiter, S.; Brandstetter, J. Boundary Graph Neural Networks for 3D Simulations. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37* (8), 9099–9107. https://doi.org/10.1609/aaai.v37i8.26092.
(45) Huang, X.; Zheng, Q.; Liu, D.; Yu, A.; Yan, W. A Design Method of Hopper Shape Optimization with Improved Mass Flow Pattern and Reduced Particle Segregation. *Chem. Eng. Sci.* **2022**, *253*, 117579. https://doi.org/10.1016/j.ces.2022.117579.

Abstract Graphics



For Table of Contents Only