

Granular Net: A Physics-Informed Neural Network for Continuum Modeling of Granular Segregation

Amir Nazemi
nazemi@umich.edu
University of Michigan
USA

Amirpasha Hedayat
ahedayat@umich.edu
University of Michigan
USA

ABSTRACT

Granular materials, ubiquitous in natural and industrial processes, exhibit complex segregation dynamics when flowing, leading to challenges in achieving uniform mixtures. Traditional continuum models capture segregation through advection–diffusion equations with closure laws for percolation and diffusion, but predictive accuracy is often limited by geometry- and material-specific uncertainties. This project explores Physics-Informed Neural Networks (PINNs) as a novel framework for modeling granular segregation at the continuum scale. By embedding the governing transport PDE into the training loss, we develop PINNs that both reproduce spatiotemporal concentration fields and identify data-driven segregation velocity closures, improving generalization across regimes and materials. Our goals are to (a) build a mesh-free forward solver that matches discrete element simulations and experiments, (b) discover physics-consistent models for segregation velocity from sparse experimental data, and (c) assess performance across key nondimensional parameters and geometries. The expected outcome is a physics-guided, data-efficient tool that enhances scientific understanding of segregation mechanisms while providing practical predictive capabilities for industrial bulk-solid handling systems.

KEYWORDS

Physics-Informed Neural Networks, AI for Science, Granular Materials, Segregation Modeling

1 INTRODUCTION

Granular materials are collections of macroscopic particles, like beads, grains, pills, pellets, or ore, found everywhere in nature and industry. When such mixtures flow, their behavior is neither that of simple solids nor Newtonian fluids. Energy is lost through friction and inelastic collisions, shear creates transient voids, and the flow strongly depends on the container and its geometry. One important consequence is segregation, the spontaneous demixing of species that differ in size, density, or shape under shear or vibration. In dense free-surface flows, “kinetic sieving” (small particles falling into shear-generated voids while large particles are forced upward) provides a dominant physical picture for size segregation and has been observed across chutes, tumblers, heaps, and hopper systems [25].

Segregation undermines product uniformity in sectors that handle bulk solids, such as polymers, pharmaceuticals, and agricultural products, where consistent composition is essential to downstream processing and final quality. In hoppers and heap flows used for storage and dosing, changing operating conditions or geometry can substantially alter discharge composition, making scale-up from lab to plant non-trivial. These sensitivities motivate predictive models

that capture the interplay of mean advection, percolation-driven segregation, and diffusive remixing [25].

Traditionally, the discrete element method (DEM), which is a type of particle simulation used in the field of granular mechanics, is used to numerically model such segregation phenomena. Also, a successful continuum framework has been developed that augments the advection–diffusion transport equation with a segregation flux in the direction normal to the free surface, producing a scalar transport PDE for each species’ concentration. This equation has achieved quantitative agreement with DEM and experiments in bounded heaps, rotating tumblers, inclined chutes, and quasi-2D hoppers. Its fidelity hinges on constitutive closures for (a) the segregation/percolation velocity, typically a function of local shear rate and composition; and (b) the diffusion coefficient, often shear-rate dependent [18, 25]. For polydisperse mixtures, the same transport structure extends by modeling segregation velocity as a superposition of pairwise interactions with other species, still driven by local shear. This generalization preserves accuracy across tridisperse/polydisperse heaps and chutes [4, 18].

Despite progress, accurate prediction across geometries and materials still faces two bottlenecks: (a) realistic kinematics are geometry and operating condition dependent; and (b) a universally reliable segregation law is elusive, especially beyond spherical, size-segregating systems to density contrast and shape effects (e.g., rods), where similar transport forms work but segregation velocity model must change [25, 28]. However, such models still benefit researchers in granular physics and continuum modeling to gain a systematic route to embed physics in learning-based solvers and to interrogate constitutive ambiguity. Practitioners in bulk-solids handling pharma, chemical, food, mining, etc., benefit from faster “what-if” prediction tools for hopper/heap operations and design rules that mitigate segregation at scale [18].

Physics-Informed Neural Networks (PINNs), introduced by Raissi, Perdikaris, and Karniadakis [16], are a recent class of machine learning models that incorporate governing PDEs directly into the training objective. By enforcing the residuals of the PDE along with boundary and initial conditions, PINNs can approximate solutions even with no labeled data. While PINNs have been used in fluid and solid mechanics, to date they have not been applied to continuum segregation models in granular flows.

Within this continuum picture, we target the forward prediction of spatiotemporal concentration fields in canonical free-surface flows in quasi-2D bounded heaps by firstly building a PINN that solves the advection–segregation–diffusion PDE using established segregation velocity forms and measured kinematics. Then, we utilize our learned PINN architecture to (a) discover material-related parameters in different empirical models developed for segregation

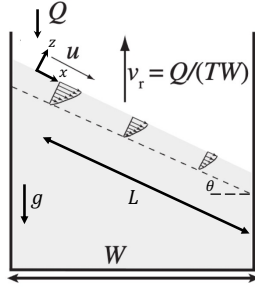


Figure 1: Schematic of a quasi-2D bounded heap of width W and thickness T with the inclination angle of θ . The effective length of the flowing layer from the right edge of the feed zone to the downstream bounding wall is L .

velocity, and (b) develop a differentiable segregation velocity surrogate model from limited experimental data while the PDE remains enforced, improving generalization across materials (size, density) and operating conditions.

We aim to (a) deliver a mesh-free, differentiable forward solver for segregation that matches DEM/experiment accuracy (forward modeling); (b) identify material-related parameters and data-driven surrogate models for segregation velocity that retain correct trends with shear and composition (inverse modeling); and (c) examine performance across regimes that will be identified by several key nondimensional parameters to clarify where PINNs are most effective and accurate in both forward and inverse modeling. Scientifically, this can test how much of segregation variability stems from closure uncertainty vs kinematic uncertainty; practically, it can shorten design cycles for hoppers/heaps and reduce off-spec product due to demixing. Societally, robust prediction and control of segregation improves quality and reduces waste in large-scale processing lines.

2 PROBLEM DEFINITION

We define our problem as a 50–50 mix of two particle sizes that is poured into a thin bin and allowed to flow down a sloped free surface (a *quasi-2D bounded heap*), as shown in Fig. 1. In this problem, the small particles tend to percolate downward while the large ones rise and are carried further downstream. The setting follows canonical bounded-heap studies validated against DEM and experiments [4, 18, 25, 28]. As the flow is quasi-2D, the feed rate is represented by the two-dimensional parameter $q = Q/T$, where T denotes the gap thickness between the sidewalls and Q is the volumetric feed rate. The flowing layer has a thickness denoted by δ , which is assumed to be constant across the entire layer.

Let $c_i(x, z, t)$ denote the volume concentration of species $i \in \{s, \ell\}$ (small, large) within the *surface flowing layer* of thickness δ and length L in a 2D frame aligned with the free surface, with streamwise coordinate $x \in [0, L]$ and surface-normal coordinate $z \in [-\delta, 0]$. The mean velocity field is considered as $\mathbf{u}(x, z) = u(x, z) \hat{\mathbf{x}} + w(x, z) \hat{\mathbf{z}}$. For the quasi-2D bounded heap, it is assumed that there is no net movement of species in the spanwise (y) direction ($v_i = 0$), and that both species move with the same velocity as the mean flow in the streamwise direction ($u_i = u$). The normal velocity of species i is expressed as $w_i = w + w_{s,i}$, where $w_{s,i}$ denotes the normal component of the segregation velocity of species i relative to the mean normal flow. The streamwise component

of the segregation velocity is assumed to be negligible. Thus, the advection–segregation–diffusion transport equation applied to the flowing layer is

$$\frac{\partial c_i}{\partial t} + \underbrace{\nabla \cdot (\mathbf{u} c_i)}_{\text{advection}} + \underbrace{\frac{\partial}{\partial z} (w_{s,i} c_i)}_{\text{segregation}} = \underbrace{\nabla \cdot (D \nabla c_i)}_{\text{diffusion}}, \quad (1)$$

where $\frac{\partial}{\partial z} (w_{s,i} c_i)$ represents the transport of species i due to *segregation* and $D(x, z)$ is the (collisional) diffusion coefficient. In equation (1), the local volume concentration of each species is governed by advection arising from the mean flow, segregation caused by percolation, and diffusion resulting from random particle collisions, consistent with [8, 21, 22, 24]. Equation (1) solves advection in x , segregation in z , and diffusion in x, z , which captures bounded-heap segregation patterns and has shown quantitative agreement with DEM and experiments [4, 28]. However, to achieve this, the velocity profiles, the segregation velocity, and the diffusion coefficient are required.

The normalized streamwise velocity profile at different streamwise locations collapses onto a single curve, as shown in Fig. 2 (a) [5]. This scaling is valid for different feed rates and particle size distributions, showing a universal functional form for the velocity field in bounded heap flow. Thus, the streamwise velocity profile in the flowing layer is in the form of

$$u(x, z) = U \left(1 - \frac{x}{L}\right) f(z) \quad (2)$$

where $f(z)$ characterizes the depth dependence with $f(0) = 1$, and $U = u(0, 0)$ is the velocity at the origin. Furthermore, Fan et al. showed that an exponential form of $f(z) = e^{kz/\delta}$ for the depth dependence term reasonably approximate the streamwise velocity profile observed in the experiment and DEM simulations [5, 11, 12]. Substituting Equation (2) into the mass conservation equation, and further mathematical manipulation related to normal velocity in the flowing layer, an analytical expression for the mean velocity field can be written as

$$\begin{aligned} u &= \frac{kq}{\delta(1 - e^{-k})} \left(1 - \frac{x}{L}\right) e^{kz/\delta}, \\ w &= \frac{q}{L(1 - e^{-k})} \left(e^{kz/\delta} - 1\right), \end{aligned} \quad (3)$$

Equation (3) inherently satisfies the boundary condition $w = -q/L = -v_r \cos \alpha$ at the base of the flowing layer ($z = -\delta$), where α is the dynamic angle of repose. In this study, the bottom of the flowing layer is defined as the depth where the streamwise velocity reaches 10% of the surface velocity, corresponding to $k = 2.3$. Equation (3) for the normal velocity also agrees well with the DEM simulation results in [4], as illustrated in Fig. 2 (a,b).

The segregation velocity represents the relative motion between species in the segregation direction. It depends on the particle size ratio, strain rate, and normal stress [1, 7, 9]. However, in heap flow, the flowing layer is only a few particle diameters thick (typically less than $10d_i$) [5], so the influence of normal stress on the segregation velocity can be neglected. Several models for the segregation velocity have been proposed [13, 14, 17, 19], yet none of them account for all these parameters or have been validated for multiple flow geometries. When the granular mixture consists of species with comparable volume fractions, the segregation velocity also

depends on the local concentration of each species, since the void size distribution is linked to local packing. For the first phase of our study, by leveraging data generated in [4], the local segregation velocity $w_{s,i}$ and the volume concentration c_i of each species are obtained from DEM simulations to analyze the dependence of segregation velocity on particle size ratio and shear rate in bounded heap flow. Fig. 2 (c) shows the segregation velocity (plotted negative for small particles) as a function of the local concentration of the other species. To collapse the data in Fig. 2 (c), the local segregation velocity is normalized by the local shear rate, $\dot{\gamma} = \partial u / \partial z$, as shown in Fig. 2 (d), since segregation occurs only when the material dilates under flow. The collapsed data can be described by

$$\begin{aligned} w_{s,l} &= S\dot{\gamma}(1 - c_l), \\ w_{s,s} &= -S\dot{\gamma}(1 - c_s), \end{aligned} \quad (4)$$

where S (with units of length) is the slope of the linear fit in Fig. 2 (d), representing the *segregation length scale*. S depends on both the particle size ratio and absolute size. Data with $1 - c_i > 0.9$ were excluded from the fit because the concentration of one species dominates, and the flux $c_i w_{s,i}$ becomes negligible. Equation (4) satisfies mass conservation since the total flux $c_s w_{s,s} + c_l w_{s,l} = 0$. As summarized in [4], the segregation length scale S is slightly smaller than the smallest particle diameter and remains nearly constant across different feed rates for the same mixture. For mixtures with identical size ratios but different absolute particle sizes, S increases with particle size. However, comparing S across size ratios is challenging since both relative and absolute sizes influence segregation behavior.

In granular mixtures, the collisional diffusion coefficient, D , contributes to the re-mixing of segregating particles. In the present study, the local diffusion coefficient of the mixture in the segregation (z -) direction is determined by calculating the mean squared displacement. For simplicity in modeling, a constant diffusion coefficient D is used, corresponding to the mean value measured over the entire flowing layer from DEM simulations in [4].

Combining Equations (1), (3), and (4) yields the transport equation for species i that explicitly accounts for the dependence of the segregation velocity on spatially varying shear rate and incorporates the full effects of flow kinematics on advection, which can be written as

$$\begin{aligned} \frac{\partial c_i}{\partial t} + u \frac{\partial c_i}{\partial x} + w \frac{\partial c_i}{\partial z} + S \frac{\partial}{\partial z} [\dot{\gamma} c_i (1 - c_i)] \\ = \left[\frac{\partial}{\partial x} \left(D \frac{\partial c_i}{\partial x} \right) + \frac{\partial}{\partial z} \left(D \frac{\partial c_i}{\partial z} \right) \right]. \end{aligned} \quad (5)$$

Equation (5) can be non-dimensionalized using

$$\tilde{x} = \frac{x}{L}, \quad \tilde{z} = \frac{z}{\delta}, \quad \tilde{t} = \frac{t}{\delta L / 2q}, \quad \tilde{u} = \frac{u}{2q/L}, \quad \tilde{w} = \frac{w}{2q/L}. \quad (6)$$

This transformation maps the flowing layer to a square domain $0 \leq \tilde{x} \leq 1$ and $-1 \leq \tilde{z} \leq 0$. The nondimensional governing equation becomes

$$\begin{aligned} \frac{\partial c_i}{\partial \tilde{t}} + \tilde{u} \frac{\partial c_i}{\partial \tilde{x}} + \tilde{w} \frac{\partial c_i}{\partial \tilde{z}} + \Lambda(1 - \tilde{x}) \frac{\partial}{\partial \tilde{z}} [g(\tilde{z}) c_i (1 - c_i)] \\ = \left(\frac{\delta}{L} \right)^2 \frac{\partial}{\partial \tilde{x}} \left(\frac{1}{Pe} \frac{\partial c_i}{\partial \tilde{x}} \right) + \frac{\partial}{\partial \tilde{z}} \left(\frac{1}{Pe} \frac{\partial c_i}{\partial \tilde{z}} \right), \end{aligned} \quad (7)$$

where $\Lambda = SL/\delta^2$, $Pe = 2q\delta/DL$, and

$$g(\tilde{z}) = \frac{1}{2} \frac{\delta f'(\delta \tilde{z})}{\int_{-1}^0 f(\delta \tilde{\xi}) d\tilde{\xi}}. \quad (8)$$

The functions $\tilde{u}$, $\tilde{w}$, and $g(\tilde{z})$ are determined by Equations (3), (6), and (8). For $\delta/L \ll 1$ (approximately $1/50$ in [4]), diffusion in the $\tilde{x}$ -direction is negligible, and the governing equation reduces to

$$\frac{\partial c_i}{\partial \tilde{t}} + \tilde{u} \frac{\partial c_i}{\partial \tilde{x}} + \tilde{w} \frac{\partial c_i}{\partial \tilde{z}} + \Lambda(1 - \tilde{x}) \frac{\partial}{\partial \tilde{z}} [g(\tilde{z}) c_i (1 - c_i)] = \frac{\partial}{\partial \tilde{z}} \left(\frac{1}{Pe} \frac{\partial c_i}{\partial \tilde{z}} \right). \quad (9)$$

The dimensionless parameters Λ and Pe have clear physical meanings; Λ represents the ratio of an advection timescale, $L/(2q/\delta)$, to a segregation timescale, $\delta/(2Sq/\delta^2)$, while Pe (the Péclet number) represents the ratio of a diffusion timescale, δ^2/D , to the advection timescale, $L/(2q/\delta)$. Both parameters depend solely on particle and flow properties (L , q , δ , S , and D).

In terms of boundary conditions, we focus on the *steady filling stage*, which occurs when the heap extends to the downstream endwall and rises at a uniform velocity. At the inlet boundary ($\tilde{x} = 0$), the particles are initially well mixed, so

$$c_s(0, \tilde{z}) = c_l(0, \tilde{z}) = 0.5. \quad (10)$$

At the top and bottom boundaries of the flowing layer ($\tilde{z} = 0$ and $\tilde{z} = -1$), the segregation flux balances the diffusive flux, following Gray and Chugunov [8],

$$\Lambda(1 - \tilde{x}) [g(-1) c_i (1 - c_i)] = \frac{1}{Pe} \frac{\partial c_i}{\partial \tilde{z}}. \quad (11)$$

This equation can be also written in a dimensional form as

$$(D \nabla c_i - w_{s,i} c_i \hat{z}) \cdot \hat{n} = 0. \quad (12)$$

This condition indicates that particles exit the heap through the bottom of the flowing layer ($w = -v_r \cos \alpha$) solely via advection due to the mean flow in the moving reference frame. At the downstream boundary ($\tilde{x} = 1$), the flow is parallel to the wall, so $\tilde{u}(1, \tilde{z}) = 0$. Because both diffusion and segregation act only in the $\tilde{z}$ -direction, no boundary condition is required there.

The flux boundary condition at $z = -\delta$ is essential since the streamwise velocity and shear rate are small but nonzero below $z = -\delta$. For $z < -\delta$, the streamwise velocity, segregation velocity, and diffusion are assumed negligible. Therefore, the concentrations directly below $z = -\delta$ are taken equal to those at $z = -\delta$. To ensure mass conservation in the heap, the mean concentration of material leaving the flowing layer must match that of the feed, meaning

$$\frac{\int_0^L w(x, -\delta) c_i(x, -\delta) dx}{\int_0^L w(x, -\delta) dx} = 0.5, \quad (13)$$

for equal concentrations of small and large particles entering the flowing layer from the feed zone. This requirement implies that diffusion and segregation fluxes must balance at $z = -\delta$, ensuring that particles exit the heap only through advection. At last, values of all parameters used in this study can be found in Table 1.

3 RELATED WORK

The framework of Physics-Informed Neural Networks (PINNs) was introduced by Raissi, Perdikaris, and Karniadakis in 2019 [16]. The central idea is to train a neural network to approximate the solution of a partial differential equation (PDE) by embedding the governing

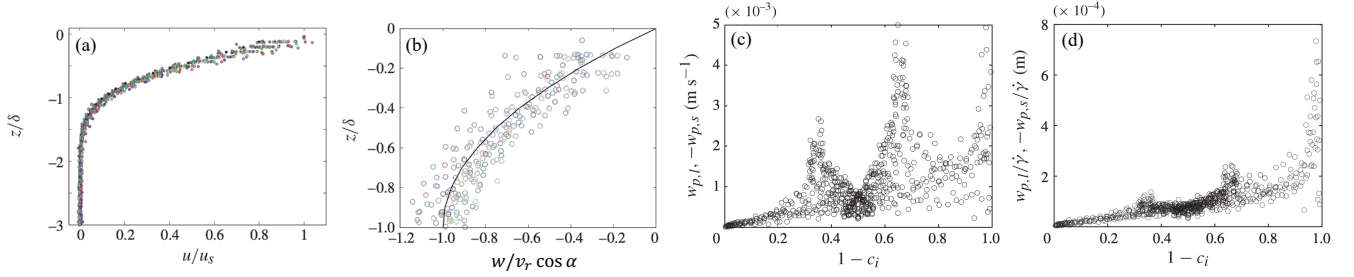


Figure 2: (a) Scaled streamwise velocity profiles u/u_{surface} in the depth direction z/δ collapse onto a single curve at different streamwise locations [4]; (b) Scaled normal velocity in the depth direction at different streamwise locations (different colors). The fitted curve represents the analytical form in Equation 4 [4]; (c) and (d) Dependence of segregation velocity ($w_{p,i}$ here) and its normalized form by shear rate on particle concentration [4].

Table 1: Variables and parameters used in the model, with units and values.

Symbol	Units	Value / Definition
c_i	–	$0 \leq c_i \leq 1$
Q	$\text{mm}^3 \text{s}^{-1}$	1.52×10^4
T	mm	12.7
W	mm	457.0
α	rad	0.38
δ	mm	14
L	mm	430.0
k	–	2.3
S	mm	0.18
D	$\text{mm}^2 \text{s}^{-1}$	2.83
R	–	$R = d_l/d_s = 2.0$
d_s	mm	1.0
d_l	mm	2.0

equations directly into the loss function. Given a network $f_\theta(x, t)$ that predicts a field variable (for example, concentration or velocity), automatic differentiation can be used to evaluate the PDE residual $\mathcal{R}(f_\theta)$. Training then involves minimizing both the residual at interior collocation points and the mismatch with boundary or initial data. In this way, PINNs can make accurate predictions even with no labeled data, provided that the PDE model is known. Since their introduction, PINNs have been extended and applied to a wide range of problems in fluid mechanics, solid mechanics, and beyond.

In mechanics and materials science, several studies have adopted PINNs to capture constitutive responses and multiscale behavior. For example, Su et al. [20] proposed a thermodynamics-informed neural network to model elastoplasticity in granular soils, embedding physical constraints such as energy consistency directly into the training. Zhang et al. [27] developed a physics-informed, multi-fidelity residual network to address consolidation and coupled hydromechanical problems in geotechnics. These studies demonstrate that neural networks guided by physics can generalize better and remain more robust than purely data-driven models. However, the focus in these works is typically on constitutive laws or relatively simple one-dimensional PDEs, rather than the transport equations that govern segregation in granular flows.

Most recent machine learning studies in the granular field have targeted particle-scale simulations. Graph neural networks (GNNs)

have been trained to emulate discrete element method (DEM) dynamics, allowing rollouts of thousands of particles with orders-of-magnitude speedups compared to classical solvers. For instance, Choi and Kumar [2] used GNNs to reproduce granular column collapse, while Mayr et al. [15] proposed boundary graph networks that generalize to arbitrary domain shapes such as hoppers and rotating drums. Other studies have combined surrogate models with optimization, such as Jiang et al. [10], who integrated GNN surrogates into inverse design workflows for granular processes. Several recent reviews have summarized these developments [6, 23], highlighting both the successes and the challenges in scaling ML methods across the particle-to-continuum divide.

The continuum description of segregation in heaps and hoppers is well established. Models based on an advection–diffusion–segregation transport equation for species concentration $c(x, z, t)$, coupled with closures for segregation velocity and diffusion, have been validated against DEM and experiments [4, 26]. Yet, despite the growing interest in machine learning for granular flows, we could not identify any peer-reviewed work that applies PINNs to these continuum segregation equations. Existing ML work focuses either on accelerating DEM or on data-driven constitutive laws. Our project addresses this gap, and investigates whether PINNs can solve the segregation PDEs directly, and identify the material-related parameters in different forms of segregation velocity field.

4 METHODOLOGY

This section details the implementation of physics-informed neural networks (PINNs) for solving the granular segregation equation. We proceed in two phases: (1) a forward PINN that solves the PDE with known parameters, and (2) inverse PINNs that learn unknown constitutive parameters or functional forms from sparse experimental data.

4.1 Forward PINN

In the forward phase, we construct a standard PINN to approximate the concentration field $c_\theta(\tilde{x}, \tilde{z}, \tilde{t})$, where θ denotes the neural network parameters. The network architecture consists of a fully connected multilayer perceptron (MLP) with 4 hidden layers of 64 neurons each, taking $(\tilde{x}, \tilde{z}, \tilde{t})$ as inputs and outputting the concentration c . The output layer uses a sigmoid activation to ensure $c \in [0, 1]$, while hidden layers employ tanh activations. Weights are initialized using Xavier uniform initialization.

The network is trained by minimizing a composite loss function that enforces the PDE residual, boundary conditions, and initial conditions:

$$\begin{aligned} \mathcal{L}(\theta) = & \lambda_{\text{pde}} \frac{1}{|S|} \sum_{s \in S} |R[c_\theta](s)|^2 + \lambda_{\text{bc,inlet}} \frac{1}{|B_{\text{inlet}}|} \sum_{b \in B_{\text{inlet}}} |c_\theta(b) - 0.5|^2 \\ & + \lambda_{\text{bc,top}} \frac{1}{|B_{\text{top}}|} \sum_{b \in B_{\text{top}}} \left| \frac{1}{Pe} \frac{\partial c_\theta}{\partial \tilde{z}}(b) - \Lambda(1 - \tilde{x})g(\tilde{z})c_\theta(1 - c_\theta)(b) \right|^2 \\ & + \lambda_{\text{bc,bottom}} \frac{1}{|B_{\text{bottom}}|} \sum_{b \in B_{\text{bottom}}} \left| \frac{1}{Pe} \frac{\partial c_\theta}{\partial \tilde{z}}(b) - \Lambda(1 - \tilde{x})g(\tilde{z})c_\theta(1 - c_\theta)(b) \right|^2 \\ & + \lambda_{\text{ic}} \frac{1}{|T_0|} \sum_{q \in T_0} |c_\theta(q) - 0.5|^2, \end{aligned} \quad (14)$$

where $R[c_\theta]$ is the PDE residual of Eq. (9), S denotes interior collocation points, B_{inlet} , B_{top} , and B_{bottom} denote boundary point sets, and T_0 denotes initial condition points. The loss weights are set to $\lambda_{\text{pde}} = 1.0$, $\lambda_{\text{bc,inlet}} = \lambda_{\text{bc,top}} = \lambda_{\text{bc,bottom}} = 10.0$, and $\lambda_{\text{ic}} = 10.0$ to balance the different constraint terms.

The PDE residual is computed using automatic differentiation (PyTorch's autograd) to evaluate the required partial derivatives $\partial c / \partial \tilde{t}$, $\partial c / \partial \tilde{x}$, $\partial c / \partial \tilde{z}$, and $\partial^2 c / \partial \tilde{z}^2$. The segregation velocity is fixed to the known analytical form $w_s = \Lambda(1 - \tilde{x})g(\tilde{z})(1 - c)$ with $\Lambda = 0.3949$ and $Pe = 27.5387$ from the literature [4].

Training uses the Adam optimizer with an initial learning rate of 10^{-3} and a cosine annealing schedule over 20,000 epochs. We employ gradient clipping with a maximum norm of 1.0 to stabilize training. Collocation points are sampled uniformly: 10,000 interior points, 1,000 points each for inlet, top, and bottom boundaries, and 2,000 initial condition points.

4.2 Inverse PINN: Learning Lambda

In the first inverse problem, we treat the segregation parameter Λ as a learnable parameter while maintaining the known functional form of the segregation velocity. The network architecture remains the same as the forward PINN, but now includes Λ as a trainable parameter. To ensure positivity, we parameterize Λ in log-space, $\Lambda = \exp(\log \Lambda)$, where $\log \Lambda$ is initialized to $\log(0.3949)$.

The loss function extends Eq. (14) to include a data-fitting term coming from experimental measurements:

$$\mathcal{L}(\theta, \Lambda) = \mathcal{L}_{\text{physics}}(\theta, \Lambda) + \lambda_{\text{data}} \frac{1}{|D|} \sum_{r \in D} |c_\theta(r) - c_{\text{exp}}(r)|^2, \quad (15)$$

where $\mathcal{L}_{\text{physics}}$ includes all terms from Eq. (14) (with Λ now being learnable), D denotes experimental data points, and $\lambda_{\text{data}} = 50.0$ to emphasize data fidelity. Both the network parameters θ and Λ are optimized jointly through backpropagation.

This approach allows the model to calibrate the segregation parameter directly from sparse spatiotemporal concentration measurements while maintaining physical consistency through the PDE residual. The learned Λ value can be compared against literature values to assess the model's ability to recover physical parameters.

4.3 Inverse PINN: Learning A and B

Recently, it has been observed that segregation flux has an underlying asymmetry that depends on local particle concentration [3]. Moreover, although it was shown that the segregation velocity depends on both size and density ratios simultaneously, Eq. (4) does not consider these dependencies. Thus, we consider a segregation velocity that includes additional concentration-dependent terms to account for the above-mentioned behaviors of segregation velocity:

$$w_s = \Gamma(1 - \tilde{x})g(\tilde{z})(1 - c) [A + B(1 - c)], \quad (16)$$

where Γ is a fixed non-dimensional length scale ($\Gamma = Ld_s/\delta^2$), and A and B are learnable parameters. This form generalizes the standard model by allowing nonlinear concentration dependence beyond the $(1 - c)$ factor.

Similar to the Λ learning case, we parameterize A and B in log-space to ensure positivity: $A = \exp(\log A)$ and $B = \exp(\log B)$, initialized to $\log(1.5)$ each. The loss function is identical to Eq. (15), but now both A and B are optimized along with the network parameters θ . The boundary conditions are updated to reflect the new segregation velocity form:

$$\frac{1}{Pe} \frac{\partial c}{\partial \tilde{z}} = \Gamma(1 - \tilde{x})g(\tilde{z})c(1 - c) [A + B(1 - c)] \quad \text{at } \tilde{z} = 0, -1. \quad (17)$$

This formulation enables the discovery of more complex segregation behaviors that may not be captured by the simple linear form, while still maintaining interpretability through the parametric structure.

4.4 Inverse PINN: Learning Neural Network Closure

In the most general case, we replace the entire segregation velocity functional form with a neural network $v_{\text{seg},\phi}(\tilde{x}, \tilde{z}, \tilde{t}, \dot{\gamma}, c)$, where ϕ denotes the closure network parameters. This network takes as inputs the spatial coordinates $(\tilde{x}, \tilde{z})$, time $\tilde{t}$, shear rate $\dot{\gamma}$ (computed from $g(\tilde{z})$), and concentration c , and outputs the segregation velocity directly.

The closure network architecture consists of 3 hidden layers with 32 neurons each, using tanh activations and no output activation (allowing both positive and negative velocities). The main concentration network remains as in the forward PINN. The two networks are trained jointly, with the total loss:

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{\text{physics}}(\theta, \phi) + \lambda_{\text{data}} \frac{1}{|D|} \sum_{r \in D} |c_\theta(r) - c_{\text{exp}}(r)|^2, \quad (18)$$

where the PDE residual now uses the neural network closure:

$$\frac{\partial c}{\partial \tilde{t}} + \tilde{u} \frac{\partial c}{\partial \tilde{x}} + \tilde{w} \frac{\partial c}{\partial \tilde{z}} + \frac{\partial}{\partial \tilde{z}} [v_{\text{seg},\phi}(\tilde{x}, \tilde{z}, \tilde{t}, \dot{\gamma}, c) \cdot c] = \frac{1}{Pe} \frac{\partial^2 c}{\partial \tilde{z}^2}. \quad (19)$$

The boundary conditions are similarly updated to use the learned closure:

$$\frac{1}{Pe} \frac{\partial c}{\partial \tilde{z}} = v_{\text{seg},\phi}(\tilde{x}, \tilde{z}, \tilde{t}, \dot{\gamma}, c) \cdot c \quad \text{at } \tilde{z} = 0, -1. \quad (20)$$

This approach provides maximum flexibility in discovering segregation laws that may not conform to standard analytical forms, while still enforcing the governing transport equation structure. The learned closure can be visualized and analyzed to understand the functional dependencies of segregation velocity on the input variables.

4.5 Training Details

All models are implemented in PyTorch and trained on one NVIDIA Tesla V100 GPU. The training procedure follows these steps:

- (1) Generate collocation points uniformly across the domain and boundaries.
- (2) Initialize network weights and learnable parameters.
- (3) For each epoch:
 - Compute PDE residual using automatic differentiation.
 - Evaluate boundary and initial condition losses.
 - (For inverse problems) Compute data-fitting loss using experimental data.
 - Backpropagate gradients and update parameters using Adam optimizer.
 - Apply gradient clipping and update learning rate via cosine annealing.
- (4) Monitor loss components and parameter evolution during training.

The same training infrastructure supports all phases, with the main difference being which parameters are trainable and the form of the segregation velocity used in the PDE and boundary conditions residual computation.

5 RESULTS AND DISCUSSION

We present comprehensive results from all four experimental phases: forward PINN, and three inverse PINN variants (learning Λ indicated by PINN_Lambda, learning A and B indicated by PINN_AB, and learning a neural network closure indicated by PINN_NN). The results demonstrate the effectiveness of physics-informed neural networks for both forward solution and inverse parameter identification in granular segregation modeling.

5.1 Training Convergence

Figure 3 shows the loss evolution during training for all four models over 20,000 epochs. All models demonstrate convergence with consistent decreases in both total loss and individual loss components, although they exhibit occasional fluctuations in the loss values. These fluctuations are likely due to the stochastic nature of the training process, and can be mitigated by using smaller learning rates. Nevertheless, the overall trend shows an acceptable convergence to small loss values.

The forward PINN (Figure 3a) achieves the lowest final total loss of 1.49×10^{-4} , as expected since it has no data-fitting constraints. The PDE residual loss decreases from 3.56×10^{-2} to 6.04×10^{-5} , while boundary condition losses converge to values below 1.0×10^{-5} , confirming accurate enforcement of physical constraints.

The inverse models show similar convergence patterns but with slightly higher final losses due to the additional data-fitting term. PINN_Lambda (Figure 3b) reaches a final total loss of 5.54×10^{-4} , with the data loss decreasing from 6.92×10^{-2} to 2×10^{-6} , indicating excellent agreement with experimental measurements. PINN_AB (Figure 3c) achieves a final total loss of 5.22×10^{-4} , while PINN_NN (Figure 3d) converges to 8.01×10^{-4} . The slightly higher loss for PINN_NN reflects the increased model complexity and the challenge of simultaneously learning both the closure law and the concentration field.

All models show low PDE residual losses at the end of training ($\sim 10^{-4}$), confirming that the learned solutions satisfy the governing transport equation. The boundary condition losses converge to similar levels across all models, demonstrating consistent enforcement of physical constraints regardless of whether parameters are fixed or learned.

5.2 Concentration Field Evolution

Figure 4 presents a comprehensive comparison of concentration fields predicted by all models at six time points ($\tilde{t} = 0, 0.5, 1.5, 3, 5, 20$) alongside ground truth data. The forward PINN plots, Figure 4 (a,b), use parameters $\Lambda = 0.3949$ and $Pe = 27.5387$, while the inverse models, Figure 4 (c-f), use $Pe = 4.0$ and learn $\Lambda = 1.2$ from experimental data.

The forward PINN successfully captures the expected segregation pattern: small particles (higher c_s) accumulate near the bottom of the flowing layer ($\tilde{z} \approx -1$), while large particles dominate near the surface ($\tilde{z} \approx 0$). The segregation becomes more pronounced downstream ($\tilde{x} \rightarrow 1$) and evolves over time from the initial well-mixed state ($c = 0.5$ everywhere at $\tilde{t} = 0$) to the final segregated profile at $\tilde{t} = 20$. The forward PINN predictions closely match the ground truth, validating the physics-informed approach for solving the continuum segregation equation.

The inverse models (PINN_Lambda, PINN_AB, and PINN_NN) all produce concentration fields that are visually indistinguishable from the ground truth. This demonstrates that the learned parameters and closure laws successfully capture the segregation dynamics. The concentration fields show the same qualitative features as the forward PINN: gradual development of segregation over time, with stronger segregation near the surface and downstream regions.

Notably, all three inverse approaches achieve similar accuracy despite their different levels of model complexity. PINN_Lambda learns a single scalar parameter, PINN_AB learns two parameters in an extended functional form, and PINN_NN learns a fully flexible neural network closure. The fact that all produce comparable results is due to the fact that all three models use the same experimental data for training (based on the same segregation velocity defined in Equation 4). Nevertheless, the neural network closure provides additional flexibility that may be valuable for extrapolation or more complex scenarios, for instance, when data is provided from a real-world experiment.

5.3 Prediction Accuracy

Figure 5 shows scatter plots comparing predicted versus actual concentration values for all four models. The forward PINN (Figure 5a) achieves excellent accuracy with $RMSE = 0.0030$ on 10,000 test points at $\tilde{t} = 20.0$ (steady-state). The points cluster tightly around the $y = x$ line, confirming that the PINN accurately reproduces the reference solution.

The inverse models achieve even higher accuracy on the experimental data. Both PINN_Lambda (Figure 5b) and PINN_AB (Figure 5c) achieve $RMSE = 0.0015$ on 1,000 experimental data points (sampled over the entire spatial and temporal domains). PINN_NN (Figure 5d) achieves $RMSE = 0.0028$, slightly lower than the parametric models but still demonstrating excellent agreement.

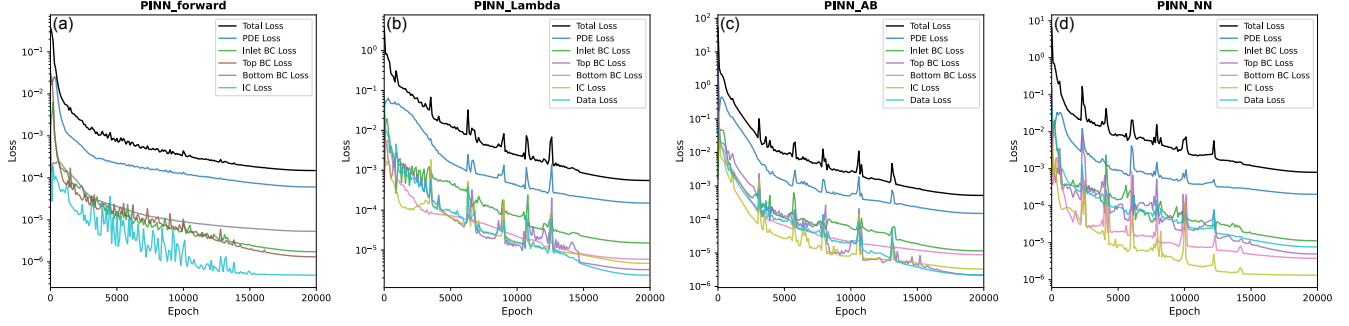


Figure 3: Training loss history for all four PINN models: (a) Forward PINN, (b) PINN with learnable Λ , (c) PINN with learnable A and B , and (d) PINN with neural network closure. Each subplot shows the total loss (black) and individual loss components (PDE, boundary conditions, initial condition, and data fitting where applicable) versus training epochs.

The superior accuracy of the inverse models on experimental data compared to the forward PINN is because the inverse models use sparse experimental data for training (supervised), while the forward PINN only satisfies the governing equation, without any knowledge of the experimental data (unsupervised). If we incorporate experimental data loss into the forward PINN, it will likely achieve accuracy similar to that of the inverse models.

5.4 Parameter and Closure Discovery

Figure 6 presents the evolution of learned parameters and a comparison of the neural network closure with the functional form. Panel (a) shows the evolution of Λ during training for PINN_Lambda. Starting from an initial value of $\Lambda = 0.3949$ (similar to the value used in the forward PINN), the parameter converges to $\Lambda = 1.2007$, matching the ground truth value of 1.2 with an error of only 0.0007 (0.06%). The convergence is smooth and stable, demonstrating that the physics-informed framework successfully identifies the segregation parameter from sparse experimental data.

Panel (b) shows the evolution of A and B parameters for PINN_AB. The parameter A converges from an initial value of 1.5 to 0.5254, close to the ground truth value of 0.5479 with an error of 0.0225 (4.1%). The parameter B decreases from 1.5 to 0.0536, while the ground truth is $B = 0$. Note that here we target the same linear segregation velocity (hence $B = 0$) function; the goal here is to show that the PINN framework is able to learn multiple parameters in a functional form. Therefore, if one uses a more complicated segregation velocity function with more parameters involved, the inverse setting is still expected to work well.

Panels (c)–(e) compare the neural network closure learned by PINN_NN with the functional form $\Lambda(1-\tilde{x})g(\tilde{z})(1-c)$ using $\Lambda = 1.2$. The neural network segregation velocity (panel c) shows a complex spatial pattern that depends on both position and local concentration. The functional form (panel d) exhibits similar overall structure, indicating that the neural network is able to learn the general trends of the segregation velocity function. The residual (panel e) reveals the differences, where the mean residual is -0.0052 with a standard deviation of 0.0318. In this setting, the target function is a linear function of the concentration. If real-world data is used, the

Table 2: Performance comparison of all PINN models. Forward PINN uses fixed parameters ($\Lambda = 0.3949$, $Pe = 27.5387$), while inverse models learn parameters from experimental data ($Pe = 4.0$).

Model	Final Loss	RMSE
Forward PINN	1.49×10^{-4}	0.0030
PINN_Lambda	5.54×10^{-4}	0.0015
PINN_AB	5.22×10^{-4}	0.0015
PINN_NN	8.01×10^{-4}	0.0028

target function is likely to be more complex, and the neural network is expected to learn the general trends of the segregation velocity function better, surpassing the empirical functional forms suggested by the literature. That said, it is worth mentioning that the neural network lacks expressivity of the functional forms, and therefore, would not be a suitable surrogate to gain insights into the segregation mechanism.

5.5 Comparative Analysis

Table 2 summarizes the key performance metrics across all four approaches. The forward PINN achieves the lowest total final loss, as expected since it has no data-fitting constraints. All approaches achieve similar RMSE ($\sim 1.5 - 3 \times 10^{-3}$), suggesting that the experimental data can be well-explained by any of these models within the measurement uncertainty.

The choice between parametric and neural network closures depends on the application requirements. Parametric models (learning Λ or A and B) provide interpretability and can be easily incorporated into existing continuum models. The learned parameter values can be compared against literature values or used for material characterization. The neural network closure offers maximum flexibility and may discover novel segregation mechanisms, but at the cost of reduced interpretability.

All approaches successfully demonstrate that physics-informed neural networks can solve inverse problems in granular segregation,

learning both the solution field and unknown constitutive parameters or laws from sparse experimental data while maintaining physical consistency through the governing PDE.

6 CONCLUSION

This work demonstrates the successful application of physics-informed neural networks (PINNs) to continuum modeling of granular segregation in bounded heap flows. We have shown that PINNs can serve as both forward solvers and inverse problem frameworks for discovering constitutive relationships from sparse experimental data.

6.1 Key Contributions

Our primary contributions are:

Forward Solution Capability: We established that PINNs can accurately solve the advection-segregation-diffusion equation for granular segregation without requiring labeled training data. The forward PINN achieved small PDE residuals while correctly capturing the expected segregation patterns: small particles accumulating near the bottom of the flowing layer and large particles dominating near the surface. The mesh-free, continuous solution representation provides advantages over traditional numerical methods, particularly for optimization and sensitivity analysis applications.

Parameter Identification: We demonstrated that PINNs can simultaneously learn both the concentration field and unknown constitutive parameters. The inverse PINN for learning Λ successfully recovered parameter values that explain experimental observations while maintaining physical consistency through the PDE constraints. Similarly, the approach for learning A and B parameters showed that more complex parametric forms can be identified from data, providing insights into the functional dependence of segregation velocity on concentration.

Closure Discovery: Most significantly, we showed that neural network closures can be learned directly from data while enforcing the governing transport equation. The learned closure automatically captures the complex dependencies of segregation velocity on spatial coordinates, shear rate, and concentration without assuming a specific analytical form. This capability opens new possibilities for discovering constitutive laws in granular flows and other complex systems where traditional models may be inadequate.

6.2 Limitations and Future Directions

Several limitations and opportunities for future work are worth noting:

Generalization: While the models show good agreement with experimental data, their ability to generalize to different geometries, particle size ratios, or operating conditions remains to be thoroughly tested. Future work should include cross-validation across different experimental configurations.

Uncertainty Quantification: The current framework provides point estimates but does not quantify uncertainty in the learned parameters or closure laws. Incorporating Bayesian approaches or ensemble methods could provide uncertainty estimates that are crucial for model validation and decision-making.

Interpretability: While the neural network closure provides flexibility, it sacrifices the interpretability of parametric models.

Future work could explore symbolic regression (e.g., SINDy) to discover interpretable functional forms.

6.3 Final Remarks

The combination of physics constraints and data-driven learning represents a promising direction for the continuum modeling of granular materials, where constitutive laws are often uncertain. As experimental techniques improve and computational resources become more accessible, we anticipate that physics-informed machine learning will play an increasingly important role in advancing our understanding and predictive capabilities for granular flows.

6.3.1 Extension to Real-World Applications. To deploy PINNs for granular segregation in real applications, they must be validated against established non-ML methods, similar to the current study that was done in the lab scale. This requires comparing accuracy, computational cost, and robustness with traditional numerical solvers and analytical solutions where available. Performance should be assessed on realistic geometries and parameter ranges. Advantages (e.g., parameter discovery from sparse data, mesh-free evaluation) and limitations (e.g., training time, generalization, uncertainty quantification) should be taken into account to evaluate the performance of the PINN in industrial applications. Scalability to 3D and multi-phase systems must be evaluated, and reliability under noisy or limited experimental data demonstrated. Benchmarks on industrial cases (e.g., hopper flows, chute flows) are needed to show where PINNs outperform or complement traditional methods, and to identify gaps (e.g., real-time prediction, handling of complex rheology) that require further development.

6.3.2 PINNs vs. Differentiable Solvers. Differentiable solvers offer an alternative to PINNs by making traditional numerical methods differentiable end-to-end. Rather than using a neural network to approximate the solution, a differentiable discretization solves the PDE directly, enabling gradient-based optimization for parameter estimation and inverse problems. This approach preserves the structure and interpretability of classical numerical methods while allowing automatic differentiation through the solver. Benefits include stronger physics constraints, improved convergence properties, and reduced training data requirements. However, challenges remain in implementing differentiable boundary conditions, handling complex geometries, and managing computational overhead, as it requires discretization, similar to numerical techniques, while PINNs are mesh-free. Moreover, when it comes to coupling the advection-segregation-diffusion governing equation with conservation of momentum and mass, differentiable solvers and PINNs offer different trade-offs. Differentiable solvers enforce coupling at the discrete level, enabling exact gradients through the coupled system and leveraging established stable schemes. However, they require domain discretization, and memory can grow with coupling iterations and time steps, which can limit scalability for strongly coupled, transient problems. PINNs are mesh-free and can represent multiple physics in a unified network, but coupling is enforced through loss terms rather than exact discrete coupling, which can lead to competing objectives and weaker enforcement. Overall, for well-understood, moderately complex coupled physics, differentiable solvers often provide more reliable gradients and better

interpretability. For complex geometries, moving boundaries, or partially unknown physics, PINNs may be more flexible.

We believe that the foundation established in this work provides a solid basis for these developments and demonstrates the potential of physics-informed machine learning for scientific discovery in granular materials.

DATA AVAILABILITY

Readers interested in reproducing the results or extending this work are encouraged to visit the following:

GitHub Repository: The source code, training scripts, and experimental data for this project are available at: <https://github.com/amirnz/GranularNet>.

Project Website: A website showcasing the project, including visualizations, results, and documentation, is available at: https://aphedayat.github.io/blog/2025/Granular_Net/.

REFERENCES

- [1] J Bridgwater, COOKE MH, and SCOTT AM. 1978. INTER-PARTICLE PERCOLATION: EQUIPMENT DEVELOPMENT AND MEAN PERCOLATION VELOCITIES. (1978).
- [2] Yongjin Choi and Krishna Kumar. 2024. Graph Neural Network-Based Surrogate Model for Granular Flows. *Computers and Geotechnics* 166 (2024), 106015. <https://doi.org/10.1016/j.compgeo.2023.106015>
- [3] Yifei Duan, Paul B Umbanhowar, Julio M Ottino, and Richard M Lueptow. 2021. Modelling segregation of bidisperse granular mixtures varying simultaneously in size and density for free surface flows. *Journal of Fluid Mechanics* 918 (2021), A20.
- [4] Yi Fan, Conor P Schlick, Paul B Umbanhowar, Julio M Ottino, and Richard M Lueptow. 2014. Modelling size segregation of granular materials: the roles of segregation, advection and diffusion. *Journal of fluid mechanics* 741 (2014), 252–279.
- [5] Yi Fan, Paul B Umbanhowar, Julio M Ottino, and Richard M Lueptow. 2013. Kinematics of monodisperse and bidisperse granular flows in quasi-two-dimensional bounded heaps. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 469, 2157 (2013), 20130235.
- [6] Marc Fransen, Andreas Fürst, Deepak Tunuguntla, Daniel N. Wilke, Benedikt Alkin, Daniel Barreto, Johannes Brandstetter, Miguel Angel Cabrera, Xinyan Fan, Mengwu Guo, Bram Kieskamp, Krishna Kumar, John Morrissey, Jonathan Nuttall, Jin Ooi, Luisa Orozco, Stefanos-Aldo Papanicolopoulos, Tongming Qu, Dingena Schott, Takayuki Shuku, WaiChing Sun, Thomas Weinhardt, Dongwei Ye, and Hongyang Cheng. 2025. Towards Scientific Machine Learning for Granular Material Simulations: Challenges and Opportunities. *Archives of Computational Methods in Engineering* (2025). <https://doi.org/10.1007/s11831-025-10322-8>
- [7] Laura A Golick and Karen E Daniels. 2009. Mixing and segregation rates in sheared granular materials. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 80, 4 (2009), 042301.
- [8] JMNT Gray and VA Chugunov. 2006. Particle-size segregation and diffusive remixing in shallow granular avalanches. *Journal of Fluid Mechanics* 569 (2006), 365–398.
- [9] KM Hill and Yi Fan. 2008. Isolating segregation mechanisms in a split-bottom cell. *Physical review letters* 101, 8 (2008), 088001.
- [10] Yu Jiang, Edmond Byrne, Jarka Glassey, and Xizhong Chen. 2024. Integrating Graph Neural Network-Based Surrogate Modeling with Inverse Design for Granular Flows. *Industrial & Engineering Chemistry Research* 63, 20 (2024), 9225–9235. <https://doi.org/10.1021/acs.iecr.4c00692>
- [11] H Katsuragi, AR Abate, and DJ Durian. 2010. Jamming and growth of dynamical heterogeneities versus depth for granular heap flow. *Soft Matter* 6, 13 (2010), 3023–3029.
- [12] Teruhisa S Komatsu, Shio Inagaki, Naoko Nakagawa, and Satoru Nasuno. 2001. Creep motion in a granular pile exhibiting steady surface flow. *Physical review letters* 86, 9 (2001), 1757.
- [13] Benjy Marks, Pierre Rognon, and Itai Einav. 2012. Grainsize dynamics of polydisperse granular segregation down inclined planes. *Journal of Fluid Mechanics* 690 (2012), 499–511.
- [14] Lindsay BH May, Laura A Golick, Katherine C Phillips, Michael Shearer, and Karen E Daniels. 2010. Shear-driven size segregation of granular materials: Modeling and experiment. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 81, 5 (2010), 051301.
- [15] Andreas Mayr, Johannes Brandstetter, Sepp Hochreiter, Thomas Pock, and Günter Klambauer. 2023. Boundary Graph Neural Networks for 3D Simulations. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 37, 9594–9602. <https://doi.org/10.1609/aaai.v37i8.26092>
- [16] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. 2019. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* 378 (2019), 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [17] SB Savage and CKK Lun. 1988. Particle size segregation in inclined chute flow of dry cohesionless granular solids. *Journal of fluid mechanics* 189 (1988), 311–335.
- [18] Conor P Schlick, Austin B Isner, Ben J Freireich, Yi Fan, Paul B Umbanhowar, Julio M Ottino, and Richard M Lueptow. 2016. A continuum approach for predicting segregation in flowing polydisperse granular materials. *Journal of Fluid Mechanics* 797 (2016), 95–109.
- [19] Kunio Shinohara, Kazunori Shoji, and T Tanaka. 1972. Mechanism of size segregation of particles in filling a hopper. *Industrial & Engineering Chemistry Process Design and Development* 11, 3 (1972), 369–376.
- [20] Mingming Su, Yang Yu, Tianhao Chen, Ning Guo, and Zixin Yang. 2024. A Thermodynamics-Informed Neural Network for Elastoplastic Constitutive Modeling of Granular Materials. *Computer Methods in Applied Mechanics and Engineering* 430 (2024), 117246. <https://doi.org/10.1016/j.cma.2024.117246>
- [21] Anthony Thornton, Thomas Weinhardt, Stefan Luding, and Onno Bokhove. 2012. Modeling of particle size segregation: calibration using the discrete particle method. *International journal of modern physics C* 23, 08 (2012), 1240014.
- [22] Anthony R Thornton, JMNT Gray, and AJ Hogg. 2006. A three-phase mixture theory for particle size segregation in shallow granular free-surface flows. *Journal of Fluid Mechanics* 550 (2006), 1–25.
- [23] M. Wang et al. 2024. Machine Learning Aided Modeling of Granular Materials: A Review. *Archives of Computational Methods in Engineering* (2024). <https://doi.org/10.1007/s11831-024-10199-z>
- [24] Sébastien Wiederseiner, Nicolas Andreini, Gaël Épely-Chauvin, Gaudenz Moser, Mathieu Monnerneau, JMNT Gray, and Christophe Ancey. 2011. Experimental investigation into segregating granular flows down chutes. *Physics of Fluids* 23, 1 (2011).
- [25] Hongyi Xiao, Yi Fan, Karl V Jacob, Paul B Umbanhowar, Madhusudhan Kodam, James F Koch, and Richard M Lueptow. 2019. Continuum modeling of granular segregation during hopper discharge. *Chemical Engineering Science* 193 (2019), 188–204.
- [26] Hongyi Xiao, Paul B. Umbanhowar, Julio M. Ottino, and Richard M. Lueptow. 2016. Modelling Density Segregation in Flowing Bidisperse Granular Materials. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 472, 2191 (2016), 20150856. <https://doi.org/10.1098/rspa.2015.0856>
- [27] Pin Zhang, Zhen Yu Yin, Yin Fu Jin, Jie Yang, and Brian Sheil. 2022. Physics-Informed Multifidelity Residual Neural Networks for Hydromechanical Modeling of Granular Soils and Foundation Considering Internal Erosion. *Journal of Engineering Mechanics* 148, 4 (2022), 04022015. [https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0002094](https://doi.org/10.1061/(ASCE)EM.1943-7889.0002094)
- [28] Yongzhi Zhao, Hongyi Xiao, Paul B Umbanhowar, and Richard M Lueptow. 2018. Simulation and modeling of segregating rods in quasi-2D bounded heap flow. *AIChE Journal* 64, 5 (2018), 1550–1563.

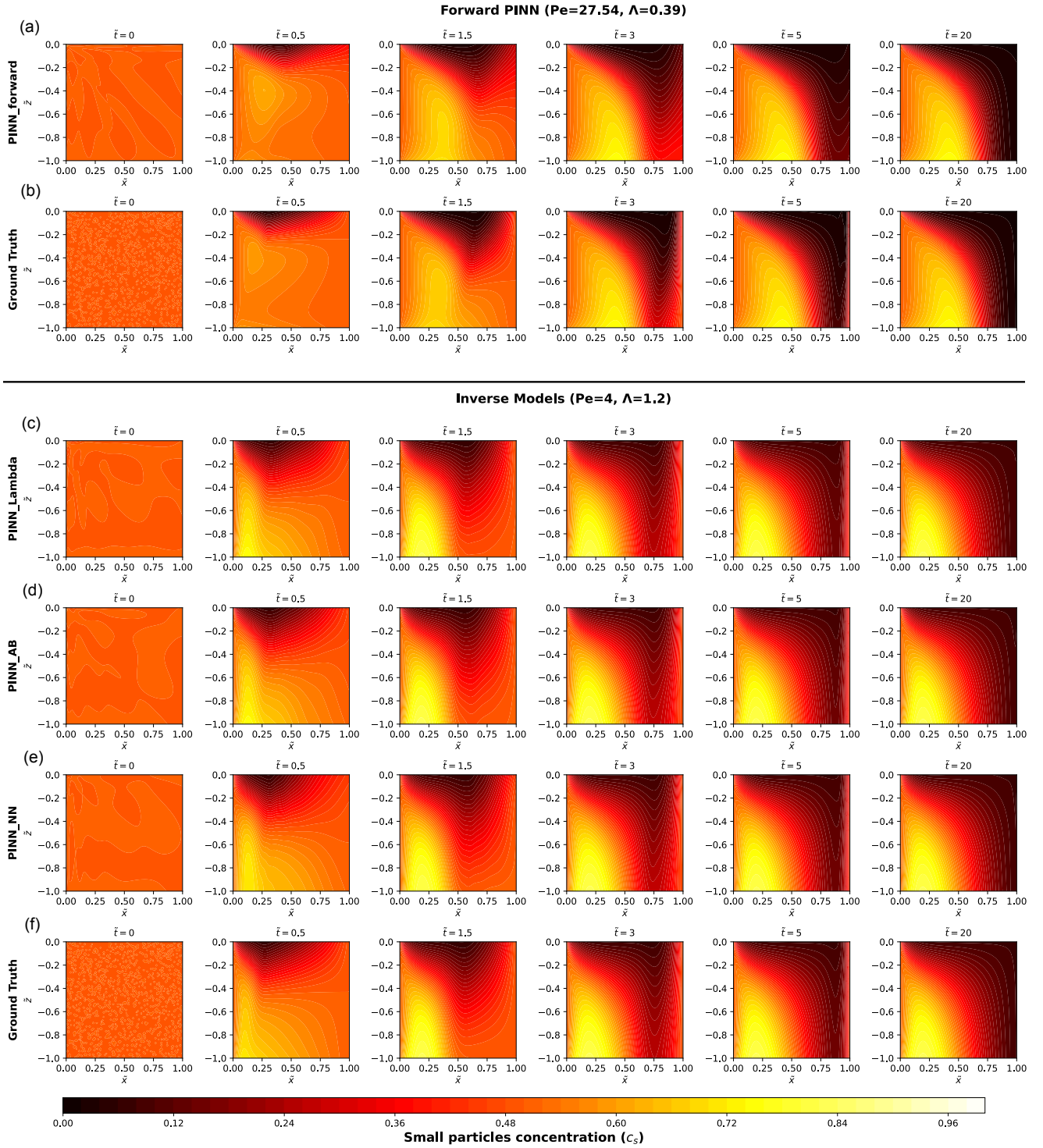


Figure 4: Concentration field evolution for all PINN models at six time points ($\bar{t} = 0, 0.5, 1.5, 3, 5, 20$); (a–b) Forward PINN predictions and ground truth (Pe=27.54, $\Lambda=0.39$); (c–f) Inverse model predictions (PINN_Lambda, PINN_AB, PINN_NN) and ground truth (Pe=4, $\Lambda=1.2$). All plots show small particle concentration $c_s = 1 - c$.

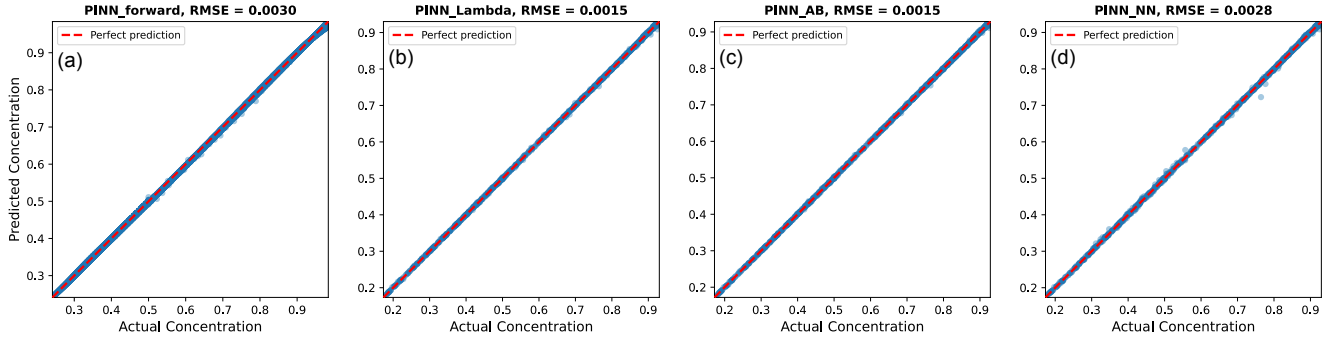


Figure 5: Predicted versus actual concentration scatter plots for all four PINN models: (a) Forward PINN, (b) PINN with learnable Λ , (c) PINN with learnable A and B , and (d) PINN with neural network closure. The red dashed line indicates perfect prediction ($y = x$). RMSE is shown in each subplot title.

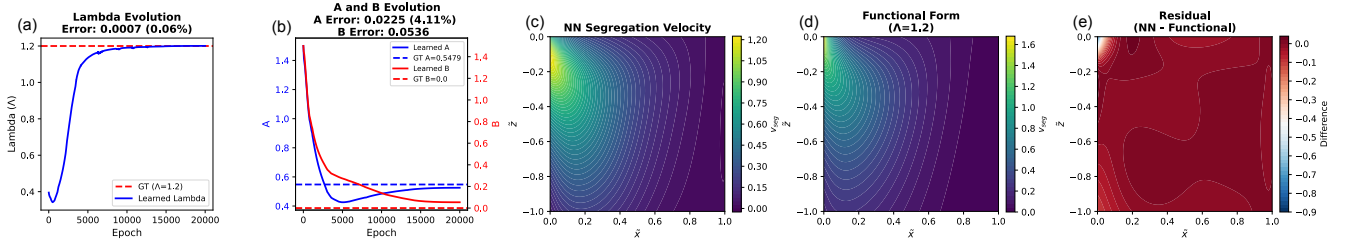


Figure 6: Parameter evolution and segregation velocity comparison: (a) Evolution of learned Λ parameter (ground truth $\Lambda = 1.2$ shown as red dashed line), (b) Evolution of learned A and B parameters (ground truth $A = 0.5479$, $B = 0$ shown as dashed lines), (c) Neural network segregation velocity field, (d) Functional form segregation velocity ($\Lambda(1 - \tilde{x})g(\tilde{z})(1 - c)$ with $\Lambda = 1.2$), and (e) Residual (difference between neural network and functional form).