

Supplementary Information for:

Data-driven discovery of governing equation for sheared granular materials in steady and transient states

Xu Han¹, Lu Jing², Chung-Yee Kwok^{1*}, Gengchao Yang³,
Yuri Dumaresq Sobral⁴

^{1*}Department of Civil Engineering, The University of Hong Kong, Hong Kong, China.

²Institute for Ocean Engineering, Shenzhen International Graduate School, Tsinghua University, Shenzhen, 518055, China.

³School of Aeronautics and Astronautics, Sun Yat-sen University, Guangzhou, 510275, China.

⁴Departamento de Matemática, Universidade de Brasília, Campus Universitário Darcy Ribeiro, Brasília DF, 70910-900, Brazil.

*Corresponding author(s). E-mail(s): fkwok8@hku.hk;
co-
ing authors: hanxu8@connect.hku.hk; lujing@sz.tsinghua.edu.cn;
yanggch8@mail.sysu.edu.cn; yurisobral@gmail.com;

Contents

18	1 Algorithm	2
19	1.1 PINNSR algorithms	2
20	1.1.1 Alternating Direction Optimization (ADO)	2
21	1.1.2 Residual-based adaptive sampling (RAS)	2
22	1.1.3 STRidge: Sequential Threshold Ridge regression	4
23	1.1.4 Hyperparameters of PINNSR	4
24	1.2 Machine learning-based dimensionless analysis (DA) algorithm	6
25	1.2.1 Two-steps optimization of dimensionless learning	6
26	1.2.2 Hyperparameters of DA	7
27	2 Results	9
28	2.1 Case-specific governing equation	9
29	2.1.1 Convergence history of PINNSR	10
30	2.1.2 Comparison between PINNSR and traditional SINDy	11
31	2.2 General governing equation	12
32	2.2.1 Validation under different setups	12
33	2.2.2 Steady state analysis	13
34	2.2.3 Transient state analysis	14
35	2.3 Granular relaxation characteristics implicit in the governing equation	15
36	2.3.1 Analytical Solution for Heaviside Loading Cases	16
37	2.3.2 Characteristic relaxation time	17
38	2.4 Micro-interpretations in terms of fabric tensor	18
39	2.4.1 Evolution of the force chain	18

40	2.4.2 Evolution of the fabric tensor	19
41	2.4.3 Existing evolution law of fabric tensor	20
42	3 Discussion	21
43	3.1 Comparison between PINNSR-DA and other methods	21
44	3.2 Influences of the number of loading sets on equation discovery	22
45	3.3 Influences of various parameters on the results	23
46	3.3.1 DEM simulation results	23
47	3.3.2 Equation coefficient results	24
48	3.4 Relationships between equation coefficients and relaxation time	25
49	3.5 Interpretations from critical state soil mechanics	26
50	4 Reference	26

51 1 Algorithm

52 1.1 PINNSR algorithms

53 PINNSR, which integrates physics-informed neural networks (PINNs) with sparse regression (SR), emerges as a quintessential approach to discover ordinary (or partial) differential
 54 equations even when data is limited and noisy¹. In this study, we utilize PINNSR as a basic
 55 methodology to discover case-specific governing equations. The total loss function of PINN-SR
 56 can be designed as follows:

$$57 \quad \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_m, \mathcal{D}_c) = \mathcal{L}_D(\boldsymbol{\theta}; \mathcal{D}_m) + \lambda \cdot \mathcal{L}_P(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_c) + \xi \cdot \|\boldsymbol{\Lambda}\|_0, \quad (1)$$

58 where \mathcal{D}_m and \mathcal{D}_c respectively denote the measurement data and data at collocation points;
 59 $\boldsymbol{\theta}$ and $\boldsymbol{\Lambda}$ stand for trainable parameters of the deep neural network (DNN) and the sparse
 60 identification of nonlinear dynamical systems (SINDy), respectively; Meanwhile, λ and ξ are
 61 hyperparameters that govern the weight of the residual physics loss and the penalty for
 62 equation complexity; $\|\cdot\|_0$ denotes the l_0 norm. Our optimization objective is to minimize
 63 this total loss, ultimately yielding a DNN model capable of mapping inputs to outputs and
 64 a SINDy-identified governing equation. The specific algorithms employed are elaborated in
 65 detail in Sections 1.1.1~1.1.3.

66 1.1.1 Alternating Direction Optimization (ADO)

67 Inspired by the seminal work¹, we employ alternating direction optimization (ADO) to
 68 train the PINNSR framework, thereby minimizing the total loss in Eq. (1). This algorithm
 69 decomposes the global optimization task into a series of handleable subproblems, as specified
 70 in Eqs (2) and (3), which in turn supports the methodical learning of the trainable parameters.

$$\boldsymbol{\Lambda}_{i+1}^* := \arg \min_{\boldsymbol{\Lambda}} \left[\|\dot{\boldsymbol{\tau}}(\boldsymbol{\theta}_i^*) - \boldsymbol{\Phi}(\boldsymbol{\theta}_i^*) \boldsymbol{\Lambda}\|_2^2 + \xi \cdot \|\boldsymbol{\Lambda}\|_0 \right], \quad (2)$$

$$71 \quad \boldsymbol{\theta}_{i+1}^* := \arg \min_{\boldsymbol{\theta}} \left[\mathcal{L}_d(\boldsymbol{\theta}; \mathcal{D}_m) + \lambda \cdot \mathcal{L}_p(\boldsymbol{\theta}, \boldsymbol{\Lambda}_{i+1}^*; \mathcal{D}_c) \right]. \quad (3)$$

72 Concurrently, we introduce an adaptive sampling strategy for collocation points, which
 73 optimizes the utilization of physical information, particularly in capturing localized sharp
 74 regions induced by shear reversal (see Algorithm 2 in Section 1.1.2). Overall structure of ADO
 75 is given in Algorithm 1.

76 1.1.2 Residual-based adaptive sampling (RAS)

77 We implement a residual-based adaptive sampling strategy to dynamically refine the spatial
 78 distribution of the collocation points in the training process. The core mechanism involves peri-
 79 odically monitoring the magnitude of the physics-informed residual (the discrepancy between
 80 the neural network's predictions and the governing PDEs). New collocation points are sub-
 81 sequently allocated within these high-error regions, while potentially coarsening sampling in

Algorithm 1 ADO training: $[\boldsymbol{\theta}, \boldsymbol{\Lambda}]_{best} = \text{ADO}(\mathcal{D}_m, \mathcal{D}_c, \lambda, \xi, \Delta\delta, N_{\text{ADO}}, N_{\text{STR},\vartheta})$

Input:

- \mathcal{D}_m : Measurement data
 - \mathcal{D}_c : Initial data at collocation points
 - λ : Weight of residual physics loss
 - ξ : Weight of penalty for equation complexity
 - $\Delta\delta$: Threshold tolerance increment for STRidge
 - N_{ADO} : Maximum number of ADO iterations
 - N_{STR} : Maximum number of STRidge cycles
 - ϑ : Coefficient for controlling the number of collocation points
- ▷ Input $\leftarrow \mathbf{t}, \dot{\gamma}_0$; Output $\leftarrow \boldsymbol{\tau}, \boldsymbol{\gamma}$
 $\mathcal{D}_c \leftarrow [\mathbf{t}_c, \dot{\gamma}_{0c}, \boldsymbol{\tau}_c, \boldsymbol{\gamma}_c] \in \mathbb{R}^{\vartheta \cdot \text{len}(\mathcal{D}_m) \times 4}$
 $\xi \leftarrow 1 \times 10^{-7}$ in pretraining

Phase 1: Data processing and pretraining

1. Split \mathcal{D}_m and \mathcal{D}_c into training and validation sets in a ratio of 0.8:0.2.
2. Initialize parameters for the neural network. ▷ Golorot Initialization
3. Perform global pre-training on the network via the L-BFGS optimizer, leveraging the training data $\{\mathcal{D}_m^{\text{tr}}, \mathcal{D}_c^{\text{tr}}\}$, and validate against the validation data $\{\mathcal{D}_m^{\text{val}}, \mathcal{D}_c^{\text{val}}\}$:
$$\{\hat{\boldsymbol{\theta}}_0, \hat{\boldsymbol{\Lambda}}_0\} \leftarrow \arg \min_{\{\boldsymbol{\theta}, \boldsymbol{\Lambda}\}} \{\mathcal{L}_D(\boldsymbol{\theta}; \mathcal{D}_m) + \lambda \cdot \mathcal{L}_P(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_c) + \xi \cdot \|\boldsymbol{\Lambda}\|_1\}.$$
4. Calculate $\dot{\boldsymbol{\tau}}_0(\hat{\boldsymbol{\theta}}_0; \mathbf{t}_c)$ by automatic differentiation and $\dot{\boldsymbol{\tau}}_0(\hat{\boldsymbol{\Lambda}}_0; \mathbf{t}_c)$ by equation results.

Phase 2: ADO cycles

- ▷ Alternating between STRidge and Adam
5. **while** $k < N_{\text{ADO}}$ **do**
 6. Update data at collocation points via adaptive sampling strategy in Algorithm 2:
 $\tilde{\mathcal{D}}_c = \text{RAS}(\dot{\boldsymbol{\tau}}_0, \hat{\boldsymbol{\tau}}_0, \vartheta).$
 7. Calculate first-order and higher-order temporal derivatives of output $\boldsymbol{\tau}, \boldsymbol{\gamma}$ at $\tilde{\mathbf{t}}_c$:
 $\dot{\mathbf{U}}_\tau^{\text{tr}} = \boldsymbol{\tau}(\hat{\boldsymbol{\theta}}_{k-1}; \tilde{\mathbf{t}}_c^{\text{tr}}), \dot{\mathbf{U}}_\tau^{\text{val}} = \boldsymbol{\tau}(\hat{\boldsymbol{\theta}}_{k-1}; \tilde{\mathbf{t}}_c^{\text{val}}), \dot{\mathbf{U}}_\gamma^{\text{tr}} = \boldsymbol{\gamma}(\hat{\boldsymbol{\theta}}_{k-1}; \tilde{\mathbf{t}}_c^{\text{tr}}), \dot{\mathbf{U}}_\gamma^{\text{val}} = \boldsymbol{\gamma}(\hat{\boldsymbol{\theta}}_{k-1}; \tilde{\mathbf{t}}_c^{\text{val}}).$
 8. Construct the candidate library matrices at collocation points \mathbf{t}_c :
 $\tilde{\boldsymbol{\Phi}} = \phi(\hat{\boldsymbol{\theta}}_{k-1}; \tilde{\mathbf{t}}_c), \tilde{\boldsymbol{\Phi}}^{\text{tr}} = \phi(\hat{\boldsymbol{\theta}}_{k-1}; \tilde{\mathbf{t}}_c^{\text{tr}}) \text{ and } \tilde{\boldsymbol{\Phi}}^{\text{val}} = \phi(\hat{\boldsymbol{\theta}}_{k-1}; \tilde{\mathbf{t}}_c^{\text{val}}).$
 9. Perform column-wise normalization on the candidate library $\tilde{\boldsymbol{\Phi}}, \tilde{\boldsymbol{\Phi}}^{\text{tr}}$ and $\tilde{\boldsymbol{\Phi}}^{\text{val}}$.
 10. Determine l_0 regularization parameters: $\xi \leftarrow \alpha \mathcal{L}_P(\hat{\boldsymbol{\theta}}_0, \hat{\boldsymbol{\Lambda}}_0; \tilde{\mathcal{D}}_c^{\text{tr}})$. ▷ $\alpha = 0.01$
 11. Initialize the threshold tolerance: $\delta_1 \leftarrow \Delta\delta$.
 12. Initialize the error indices:
 $\hat{\epsilon} \leftarrow \mathcal{L}_p(\hat{\boldsymbol{\theta}}_{k-1}, \hat{\boldsymbol{\Lambda}}_{k-1}; \tilde{\mathcal{D}}_c^{\text{val}}) + \xi \cdot \|\hat{\boldsymbol{\Lambda}}_{k-1}\|_0.$
 13. **for** $i = 1$ **to** N_{STR} **do**
 14. Execute STRidge as stated in Algorithm 3 to determine:
 $\hat{\boldsymbol{\Lambda}} = \text{STRidge}(\tilde{\boldsymbol{\Phi}}^{\text{tr}}, \tilde{\boldsymbol{\tau}}^{\text{tr}}, \xi, \delta, N_{\text{STR}}).$
 15. Evaluate the error metric:
 $\epsilon = \mathcal{L}_p(\hat{\boldsymbol{\theta}}_{k-1}, \hat{\boldsymbol{\Lambda}}; \tilde{\mathcal{D}}_c^{\text{val}}) + \xi \cdot \|\hat{\boldsymbol{\Lambda}}\|_0.$
 16. **if** $\epsilon \leq \hat{\epsilon}$ **then**
 17. Increase threshold tolerance with increment: $\delta_{i+1} = \delta_i + \Delta\delta$ ▷ $\delta_1 = \Delta\delta$
 18. **else**
 19. Reduce threshold tolerance increment: $\Delta\delta = \Delta\delta/\eta$. ▷ $\eta = 1.618$
 20. Adjust threshold tolerance according to the new increment:
 $\delta_{i+1} = \Delta\delta + \max\{0, \delta_i - 2\Delta\delta\}.$
 21. **end if**
 22. **end for**
 23. Return the current optimal solution from STRidge cycles: $\hat{\boldsymbol{\Lambda}}_k = \hat{\boldsymbol{\Lambda}}$.
 24. Train the network using Adam on $\{\mathcal{D}_m^{\text{tr}}, \mathcal{D}_c^{\text{tr}}\}$, and validate it on $\{\mathcal{D}_m^{\text{val}}, \mathcal{D}_c^{\text{val}}\}$:

$$\hat{\boldsymbol{\theta}}_k = \arg \min_{\boldsymbol{\theta}} \{\mathcal{L}_D(\boldsymbol{\theta}; \mathcal{D}_m) + \lambda \cdot \mathcal{L}_P(\boldsymbol{\theta}, \hat{\boldsymbol{\Lambda}}_k; \mathcal{D}_c)\}.$$
 25. **end while**

Output:

- $[\boldsymbol{\theta}, \boldsymbol{\Lambda}]_{\text{best}}$: Optimal parameters for the neural network and sparse regression
-

81 areas where the solution is well-resolved (i.e., low residual). This strategy significantly improving
 82 the fidelity of the PINN approximation, particularly for problems involving sharp gradients,
 83 discontinuities, or localized features.

84 Specifically, the collocation set \mathcal{D}_c is adjusted dynamically at each training step. Gaussian
 85 kernel density estimation (KDE) is deployed on the temporal domain, with weights being
 86 proportional to the residual physical losses at \mathcal{D}_m , so as to generate updated collocation
 87 points. Such adaptive sampling methods give priority to high-loss regions, boosting sampling
 88 density in areas where physical constraints are not well satisfied. The total number of samples
 89 is specified as $N_c = \vartheta N_m$, where N_m is the total count of measured data points and ϑ is a
 90 hyperparameter that governs collocation density. In this study, ϑ is set to 50, and more details
 91 can be found in Algorithm 2. The detailed procedure is provided in Algorithm 3.

Algorithm 2 Residual-based Adaptive Sampling (RAS): $\widetilde{\mathcal{D}}_c = \text{RAS}(\dot{\mathbf{U}}, \widehat{\mathbf{U}}, \vartheta)$

Input:

- $\dot{\mathbf{U}}$: The derivative vector as the true value calculated by automatic differentiation
- $\widehat{\mathbf{U}}$: The derivative vector as the predicted value calculated by identified equation
- ϑ : Coefficient for controlling the number of collocation points

1. Calculate the physical residual loss:

$$\mathbf{L} = (\dot{\mathbf{U}} - \widehat{\mathbf{U}})^2.$$

2. Determine the total number of collocation points:

$$N_{cp} = \vartheta \cdot (N_m), \quad N_m = \text{len}(\dot{\mathbf{U}}).$$

3. Generate new collocation points via Gaussian kernel density estimation (KDE) in the temporal domain (weighted by \mathbf{L}) with N_{cp} as total number.
4. Assemble new collocation set t_c as input and predict output via current neural network.

Output: $\widetilde{\mathcal{D}}_c$: Data at collocation points after residual-based adaptive sampling

92 **1.1.3 STRidge: Sequential Threshold Ridge regression**

93 In this study, we use the Sequential Threshold Ridge regression (STRidge) approach to
 94 tackle sparse regression by iteratively integrating ridge regression (Tikhonov regularization)
 95 with hard thresholding. Specifically, ridge regression is utilized to mitigate multicollinearity,
 96 thereby enhancing model stability when dealing with correlated features or noise. Concurrently,
 97 STRidge promotes sparsity in the coefficient matrix (Λ) by sequentially eliminating
 98 coefficients whose magnitudes fall below a dynamically adjusted threshold. This iterative
 99 thresholding process refines the model's active candidate terms, ultimately yielding a sparse
 100 and interpretable solution containing only the most significant contributing terms.

101 **1.1.4 Hyperparameters of PINNSR**

102 Regarding the PINNSR algorithm, there are primarily six hyperparameters. Among these,
 103 the selection of most parameters follows principles similar to those adopted in the seminal
 104 work by Chen et al.¹. Detailed specifications are provided as follows:

- 105 • **Weight of residual physics loss (λ):** This hyperparameter is employed to balance the
 106 contributions of data loss and physical constraints during the network training process.
 107 Its value is typically determined based on the scale ratio between the measured response
 108 (τ) and its temporal derivative ($\dot{\tau}$). Guided by this principle, we performed a global
 109 estimate in 625 computational cases with varying system variables. By setting $\lambda = 0.01$,
 110 favorable convergence performance was achieved under nearly all operating conditions.
- 111 • **Weight of penalty for equation complexity (ξ):** ξ denotes the coefficient of l_0
 112 regularizer for Λ . To determine the value that optimally balances physics loss and equation

Algorithm 3 STRidge: $\hat{\Lambda} = \text{STRidge}(\Phi, \dot{U}, \xi, \delta, N_{\text{STR}})$

Input:

- Φ : Candidate function library matrix
- \dot{U} : Derivative vector (left-hand side term)
- ξ : Ridge regression penalty weight
- δ : Sparsity threshold tolerance
- N_{STR} : Maximum number of iteration cycles

Initialize: $\hat{\Lambda}$ (inherited from pre-training or previous update)

1. **for** $n = 1$ to N_{STR} **do**
2. Identify indices of coefficients based on threshold δ :

$$\mathcal{S} = \left\{ s \in \mathcal{S} : |\hat{\Lambda}_s| < \delta \right\} \quad \text{and} \quad \mathcal{L} = \left\{ l \in \mathcal{L} : |\hat{\Lambda}_l| \geq \delta \right\}.$$

3. Apply sparsity by setting small coefficients to zero:

$$\hat{\Lambda}_{\mathcal{S}} \leftarrow 0.$$

4. Update non-zero coefficients via ridge regression:

$$\hat{\Lambda}_{\mathcal{L}} \leftarrow \arg \min_{\theta} \left\{ \left\| \Phi_{\mathcal{L}} \theta - \dot{U} \right\|_2^2 + \xi \cdot \|\theta\|_2^2 \right\}.$$

5. **end for**

Output: $\hat{\Lambda}$: Updated sparse coefficient vector ($\hat{\Lambda} = \hat{\Lambda}_{\mathcal{L}} \cup \hat{\Lambda}_{\mathcal{S}}$)

113 sparsity, a Pareto front analysis was conducted on a representative case. This analysis
114 identifies the range of 1 to 10 as optimal. For the vast majority of cases, ξ is set to 1,
115 which enables the identification of three terms that form the final equation. However,
116 in a small number of exceptional cases, either 4 or 2 terms may be identified. In such
117 instances, a re-conducted Pareto analysis is required, with ξ adjusted accordingly (i.e.,
118 decreased or increased) to align with the observed term count.

- 119 **• Threshold tolerance increment for STRidge ($\Delta\delta$):** The $\Delta\delta$ is to perform adaptive
120 calibration on δ , thereby introducing appropriate sparsity into the identified equations.
121 We adopt the same strategy as in reference¹ to automatically determine the optimal
122 threshold in STRidge. In most of our test scenarios, a small positive value ($\delta = 1$) is
123 sufficient when the number of iterations for STRidge is large.
- 124 **• Maximum number of ADO iterations (N_{ADO}):** This hyperparameter denotes the
125 total number of alternating cycles between Adam (employed to train neural networks) and
126 STRidge (utilized to train sparse regression model). In practice, its value is determined
127 dynamically: the training process is terminated as soon as the total loss, physical loss,
128 and penalty regulation loss all achieve convergence. We set $N_{\text{ADO}} = 10$, which proves
129 sufficient across nearly all numerical cases. Notably, the vast majority of cases reach
130 convergence within 3 cycles. Further details are available in the [Main Text](#) (refer to the
131 section titled *Learning case-specific governing equation*).
- 132 **• Maximum number of STRidge iterations (N_{STR}):** A sufficiently large count of
133 STRidge cycles (e.g., 100 iterations) is configured to guarantee that the algorithm can
134 heuristically identify the optimal threshold.
- 135 **• Coefficient for controlling the number of collocation points (ϑ):** The hyperpara-
136 meter ϑ regulates the number of collocation points, which are critical sampling nodes
137 where ODE (or PDE) residuals are evaluated to enforce physical constraints. As a scal-
138 ing factor, ϑ modulates collocation point density, balancing computational efficiency
139 (fewer points) and residual evaluation granularity (more points). Sensitivity analysis over

140 $\vartheta \in [10, 100]$ reveals a small impact on overall model performance on our problems, indicating
 141 a low sensitivity to this parameter within the tested range. Consequently, $\vartheta = 50$
 142 is adopted as the default setting for most experiments, achieving a practical trade-off
 143 between constraint enforcement reliability and computational cost.

144 1.2 Machine learning-based dimensionless analysis (DA) algorithm

145 1.2.1 Two-steps optimization of dimensionless learning

146 This dimensionless learning method is inspired by the reference². In this section, we will
 147 introduce the problem statement and two-step optimization algorithm in detail.

148 Within a system comprising n independent variables and a single dependent variable, the
 149 causal relationship can be expressed as:

$$p = f(p_1, p_2, \dots, p_n). \quad (4)$$

150 The n independent variables can be characterized by r fundamental physical dimensions,
 151 such as length, mass, time, and temperature. From this, a dimension matrix $\mathbf{D} \in \mathbb{R}^{r \times n}$ for
 152 the n independent variables can be constructed:

$$\mathbf{D} = [\mathbf{v}(p_1), \mathbf{v}(p_2), \dots, \mathbf{v}(p_n)]. \quad (5)$$

153 Based on classical dimensional analysis, Eq. (1) can be reformulated in a dimensionless
 154 expression:

$$\Pi = f(\Pi_1, \Pi_2, \dots, \Pi_k). \quad (6)$$

155 Buckingham's Pi theorem³ asserts that the quantity of dimensionless parameters k equals
 156 the count of variables n subtracted by the count of dimensions r , assuming the dimension
 157 matrix \mathbf{D} is of full rank. While the theorem provides a sufficient number of dimensionless
 158 parameters to characterize a system, it does not specify the necessary quantity. Consequently,
 159 the proposed dimensionless learning seeks to identify m ($m \leq k$) predominant dimensionless
 160 parameters from data. In Eq. (6), $f(\cdot)$ is the best representation (i.e., scaling law) of the
 161 outputs and inputs, and $\{\Pi_1, \Pi_2, \dots, \Pi_m\}$ are the input dimensionless numbers. Specifically,
 162 to find the i -th dimensionless number Π_i , we need to find a vector \mathbf{w}_i :

$$\Pi_i = p_1^{w_1^i} p_2^{w_2^i} \dots p_n^{w_n^i}. \quad (7)$$

163 We have the dimensional invariance for \mathbf{w}_i as the constraint:

$$\mathbf{D}\mathbf{w}_i = 0 \quad (8)$$

164 To solve Eq. (8), we can first calculate the K basis vectors of the solution w :

$$\mathbf{D}\mathbf{w}_{bk} = 0, k \in [1, K]. \quad (9)$$

165 Here, $\{\mathbf{w}_{b1}, \mathbf{w}_{b2}, \dots, \mathbf{w}_{bK}\}$ are linearly independent, and $K=n-\text{rank}(\mathbf{D})$. We can then
 166 express \mathbf{w}_i within a linear combination of basis vectors $\{\mathbf{w}_{bk}\}_{k=1}^K$. Then, Eq. (8) can be
 167 reformulated as:

$$\mathbf{D}\mathbf{w}_i = \mathbf{D} \sum_k^K \gamma_i^k \mathbf{w}_{bk} = 0, \quad (10)$$

168 where γ_i^k is the coefficient of the k th basis vector.

169 In our problem, we conduct an in-depth examination of system parameters including nor-
 170 mal stress (P), particle diameter (d_p), particle density (ρ), Young's modulus of particles
 171 (E_p), sinusoidal shear loading period (T), and height of setup (H). Specifically, the left-to-
 172 right entries in Eq. (11) correspond to the physical dimensions of these parameters in the
 173 aforementioned order. Thus, we can construct the dimension matrix as follows:

$$\mathbf{D} = \begin{bmatrix} -1 & 1 & -3 & -1 & 0 & 1 \\ -2 & 0 & 0 & -2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} [L] \\ [T] \\ [M] \end{bmatrix}. \quad (11)$$

¹⁷⁴ Using Eq. (11), the basis vectors $\{\mathbf{w}_{bk}\}_{k=1}^K$ can be derived. In the present case, three basis
¹⁷⁵ vectors are explicitly given in Eqs. (12)-(14):

$$w_{b1} = [-1, 0, 0, 1, 0, 0]^T, \quad (12)$$

$$w_{b2} = [0.5, -1, -0.5, 0, 1, 0]^T, \quad (13)$$

$$w_{b3} = [0, -1, 0, 0, 0, 1]^T. \quad (14)$$

¹⁷⁸ Our next objective is to determine the values of γ_i^k in the Eq. (10), thereby establishing the
¹⁷⁹ specific form of the dimensionless numbers. To derive the basis coefficients from the datasets,
¹⁸⁰ a representation learning model is needed to approximate $f_i(\cdot)$, where we assume that our
¹⁸¹ objective dimensionless number can be expressed as $C_i = f_i(\Pi_i)$. A variety of machine learn-
¹⁸² ing models, such as neural networks, support vector machines (SVM), decision trees, and
¹⁸³ XGBoost, are all viable options as representation models. In this work, to strike a balance
¹⁸⁴ in computational efficiency, we opt for twelfth-order polynomial functions. To determine γ_i
¹⁸⁵ and β_i , we employ an iterative machine learning-based optimization approach as detailed in
¹⁸⁶ Algorithm 4 (modified from the reference²).

$$C_i = \beta_{i0} + \beta_{i1}\Pi_i + \beta_{i2}\Pi_i^2 + \beta_{i3}\Pi_i^3 + \cdots + \beta_{i12}\Pi_i^{12}. \quad (15)$$

¹⁸⁷ where $\beta_i = [\beta_{i0}, \beta_{i1}, \beta_{i2}, \dots, \beta_{i12}]^T$ captures the scaling relationship between C_i and Π_i .

Algorithm 4 Grid search-based two-level optimization: $\tilde{\Pi} = G(\mathbf{p}, \mathbf{D}, N_{\text{in}}, Q, \delta)$

Input:

- \mathbf{p} : Input variables in the system
- \mathbf{D} : Dimension matrix of the input variables ▷ as shown in Eq. (9)
- N_{in} : Number of input dimensionless number ▷ 1 is enough, otherwise should >1
- Q : Degree of polynomial function ▷ $f(\cdot)$ represented by polynomial function
- δ : Grid interval ▷ default δ is 0.05

1. Calculate the basis vectors of the solution \mathbf{w} , according to the equation $\mathbf{D}\mathbf{w}_{bk} = 0$.
2. **for** $i \leftarrow 1$ to N_{in} **do**
3. set $\gamma_i^1 = 1$.
4. Uniformly discretize the interval $[-6, 6]$ with a grid interval δ for $\gamma_i^2, \gamma_i^3, \dots, \gamma_i^K$.
5. **end for**
6. Iterate over all basis vector combinations for the N_{in} dimensionless numbers.
7. Fit $\tilde{\Pi} = f(\Pi_1, \Pi_2, \dots, \Pi_{N_{\text{in}}})$ using a polynomial with degree Q .
8. Evaluate the fitting performance with R^2 score.

Output: $\tilde{\Pi}$: Output dominant dimensionless number

¹⁸⁸ **1.2.2 Hyperparameters of DA**

¹⁸⁹ In the machine learning-based dimensionless analysis (DA) algorithm, there are three
¹⁹⁰ hyperparameters:

- ¹⁹¹ • **Number of input dimensionless number (N_{in})**: The DA algorithm is designed to
¹⁹² identify the dominant dimensionless numbers that govern the coefficients of the identified
¹⁹³ equation (C_1, C_2, C_3). In this study, the dominant dimensionless forms to be excavated
¹⁹⁴ are treated as inputs to the representation model (herein a polynomial function) for
¹⁹⁵ predicting the equation coefficients as outputs. Initially, we may assume $N_{\text{in}} = 1$ to verify
¹⁹⁶ whether the predicted performance is good enough. If the R^2 score is less than 0.80, we

shall then consider scenarios where $N_{in} > 1$. In cases where the polynomial function is used as a representation model, the following forms need to be further considered:

$$C = \beta_0 + \sum_{i=1}^{N_{in}} \beta_i \Pi_i + \sum_{i=1}^{N_{in}} \sum_{j=1}^{N_{in}} \beta_{ij} \Pi_i \Pi_j + \sum_{i=1}^{N_{in}} \sum_{j=1}^{N_{in}} \sum_{k=1}^{N_{in}} \beta_{ijk} \Pi_i \Pi_j \Pi_k \dots, \quad (16)$$

where C denotes the output and β represents polynomial coefficients. It should be noted that Eq. 16 only explicitly presents polynomial terms up to the 3rd order, and additional higher order terms should be incorporated sequentially based on the complexity of the specific problem. In this study, setting $N_{in} = 1$ suffices, given that relatively high R^2 scores for C_1, C_2, C_3 (all larger than 0.97) can be achieved under this assumption, as elaborated in [Main Text](#) (see the section of *Learning equation coefficients*).

- **Degree of polynomial function (Q):** For polynomial functions, Q governs the complexity of the model. In addition to polynomials, neural networks, support vector machines (SVM), random forests, and other machine learning models can also be employed as representation models. Here, Q is analogous to the hyperparameter that controls the number of parameters in these machine learning models (e.g., the number of layers and neurons in a neural network). A smaller Q implies a weaker capacity to characterize the complex nonlinear relationships between inputs and outputs; however, a larger Q is not inherently superior, as it may make the model susceptible to overfitting. To investigate the impact of Q , the dataset is partitioned into training and testing subsets with an 8:2 ratio. Subsequent parameter analysis is then conducted to identify the optimal hyperparameter ranges for the prediction tasks of C_1, C_2 and C_3 . The results are illustrated in Figs. 1~3. In the present study, Q is set to 12.

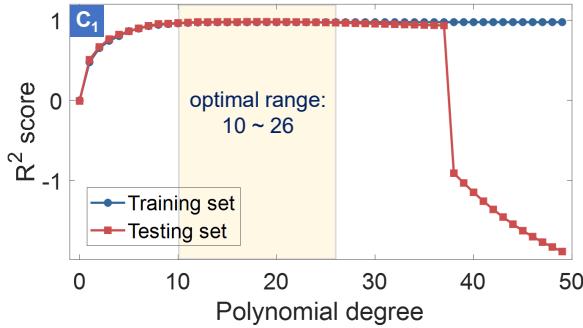


Fig. 1 Optimal polynomial degree range for C_1 .

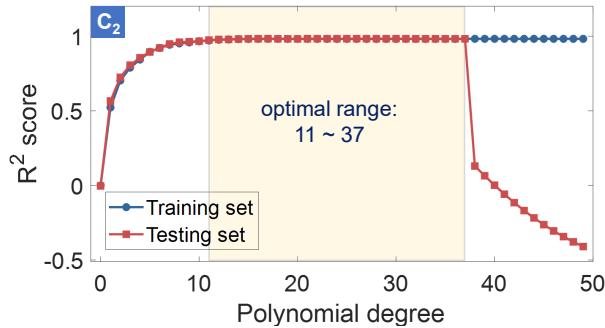


Fig. 2 Optimal polynomial degree range for C_2 .

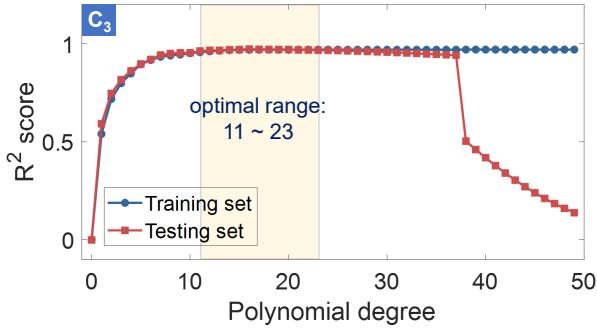


Fig. 3 Optimal polynomial degree range for C_3 .

- **Grid interval (δ):** We set $\delta = 0.05$ in our study, a value that ensures sufficient search precision in the dimensionless space while preventing blind search for excessively small variations in physical parameters. Practically, physical parameters characterized by fractional exponents often result in dimensionless numbers that lack physical meaning in most cases. This configuration effectively enhances the efficiency of the algorithm.

2 Results

2.1 Case-specific governing equation

This section outlines the results obtained via PINN-SR for the case-specific governing equation. We focus on a representative system defined by parameters $P = 10 \text{ kPa}$, $H = 15 d_p$, $T = 10 \text{ s}$, alongside particle properties $\rho = 2500 \text{ kg/m}^3$, $d_p = 0.005 \text{ m}$, and $E_p = 5 \times 10^7 \text{ Pa}$ (see Fig. 4). Under these DEM configurations, the amplitude of the shear rate $\dot{\gamma}_0 = v_0/H$ is varied within the range [0.1, 0.4] at intervals of 0.02, generating 16 cases of sinusoidal oscillatory shear with distinct loading amplitudes ($\dot{\gamma}_0 = [0.10, 0.12, \dots, 0.38, 0.40]$). These 16 datasets constitute the training data for the data-driven framework. Corresponding testing datasets are generated with $\dot{\gamma}_0 = [0.11, 0.13, \dots, 0.37, 0.39]$ at 0.02 increments. Notably, while the testing sets are merely interpolations of the training sets, our objective diverges from traditional machine learning, which prioritizes prediction. Instead, our focus is to extract an explicit governing equation. The genuine predictive capacity of this equation manifests in subsequent variations of load types: we derive the equation from sinusoidal cases and validate it against Heaviside cases, as well as across diverse simulation configurations.

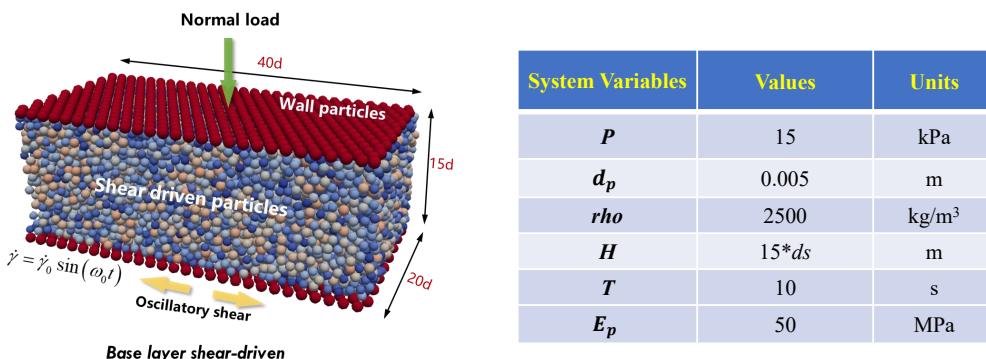


Fig. 4 Wall-confined oscillatory shear DEM simulation and parameter settings of the representative setup.

²³⁷ **2.1.1 Convergence history of PINNSR**

²³⁸ Training results are presented using data generated from this representative setup. The
²³⁹ ADO algorithm is employed to train the total loss function of PINNSR (Eq. (1)). Conver-
²⁴⁰ gence trajectories were documented from three perspectives: total loss, PINN loss, and sparse
²⁴¹ regression loss. Total loss has been reported in the [Main Text](#) (refer to the section titled *Learn-*
²⁴² *ing case-specific governing equation*), while the latter two are illustrated in Figs. 5~6. These
²⁴³ figures demonstrate a consistent reduction in total error until it reaches a stable state. Dur-
²⁴⁴ ing the PINNs training process, both DNN loss and physical loss exhibit a gradual decreasing
²⁴⁵ trend. Similarly, within the SINDy framework, both loss components stabilize—this observa-
²⁴⁶ tion indicates the successful identification of a unique equation structure. Ultimately, a highly
²⁴⁷ concise equation was identified, as shown in Eq. 17:

$$\frac{d\mu}{dt} = C_1 \cdot \frac{d\gamma}{dt} + C_2 \cdot \mu \left| \frac{d\gamma}{dt} \right| + C_3 \cdot \mu^2 \frac{d\gamma}{dt}. \quad (17)$$

²⁴⁸ with coefficients $C_1 = -5.98$, $C_2 = -28.05$, and $C_3 = -32.41$.

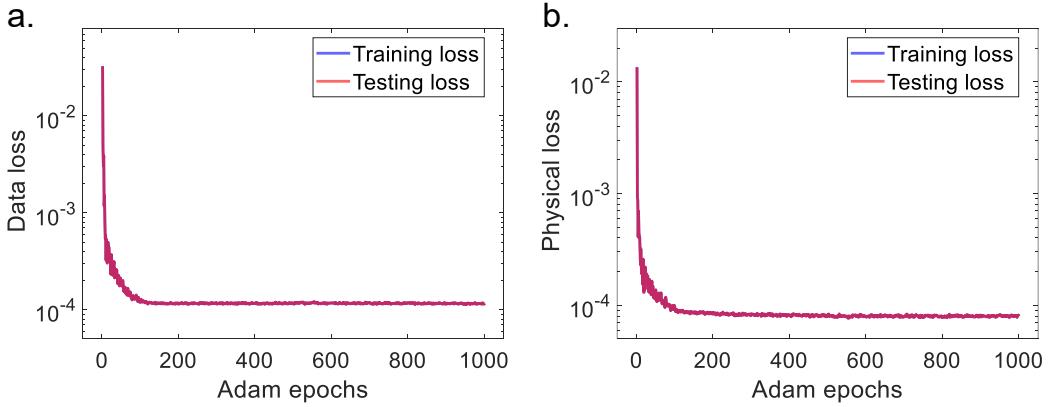


Fig. 5 The process of the PINN loss decreasing during training. **a** Data loss of Adam; **b** Physical loss of Adam.

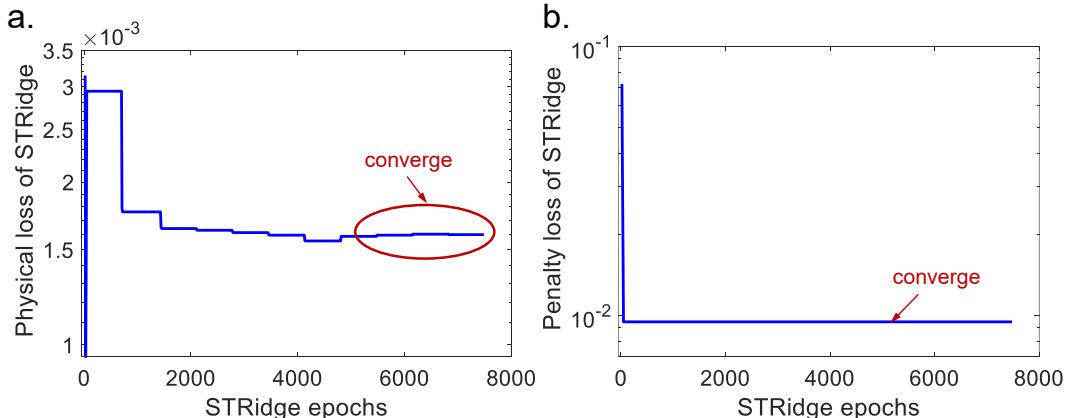


Fig. 6 The process of the sparse regression loss decreasing during training. **a** Physical loss of STRidge; **b** Penalty loss of STRidge.

2.1.2 Comparison between PINNSR and traditional SINDy

In this section, we compare outcomes of PINNSR and SINDy on the representative setup. Using the identical DEM configuration, traditional SINDy produces unstable equation results when applied to datasets that cover the same data range but vary in size. Specifically, adjusting the number of training cases (e.g., from 16 sinusoidal oscillatory shear cases to 9 or 21, with uniform distribution between the maximum and minimum $\dot{\gamma}_0$ amplitudes) leads to the identification of entirely distinct equation terms. In contrast, PINNSR produces consistent results across variations in training data size, reliably discovering a parsimonious three-term equation. This observation indicates that traditional SINDy is prone to overfitting, whereas PINNSR exhibits strong robustness in the presence of noisy data. For comparing with PINNSR's result, we selected the optimal equation identified by traditional SINDy:

$$\frac{d\mu}{dt} = A_1 \cdot \frac{d\gamma}{dt} + A_2 \cdot \mu \left| \frac{d\gamma}{dt} \right| + A_3 \cdot \mu \frac{d\gamma}{dt} + A_4 \cdot \mu^2 \left| \frac{d\gamma}{dt} \right| + A_5 \cdot \mu^2 \frac{d\gamma}{dt}, \quad (18)$$

with coefficients $A_1 = -4.524$, $A_2 = -2.501$, $A_3 = -2.016$, $A_4 = -0.004$, and $A_5 = 21.360$. The predictive results of PINNSR and SINDy are illustrated in Fig. 7.

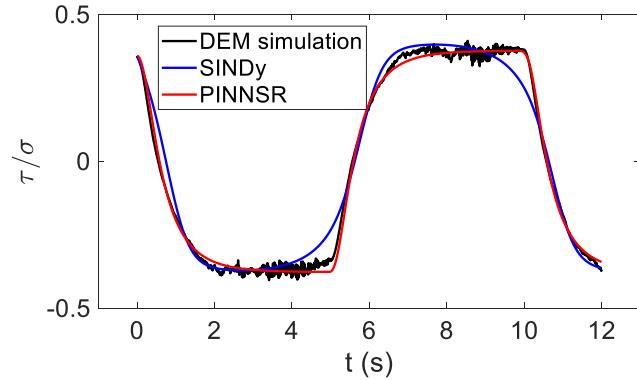


Fig. 7 Comparison between PINNSR and traditional SINDy. Red line represents the numerical solution results of PINNSR obtained via Runge-Kutta method; blue line indicates the numerical solution results of the traditional SINDy; and black line depicts the simulation results of DEM on the test set (regarded as the ground truth).

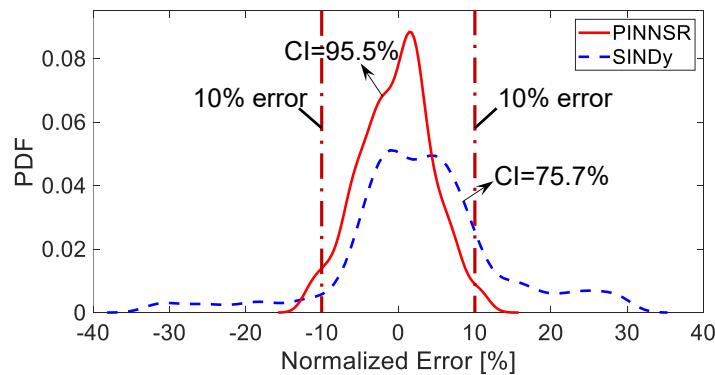


Fig. 8 Normalized error distribution of prediction using various methods. Red line represents PINNSR results; blue dashed line indicates traditional SINDy results.

262 To enhance clarity in illustrating the normalized prediction error for each method, we
 263 additionally employed a probability density function (PDF) from a probabilistic standpoint,
 264 defined as $\text{PDF} \left\{ \frac{y_i - y'_i}{\max|y_i|} \right\}$ (see Fig. 8). As shown in Fig. 8, the prediction errors of both
 265 PINNSR and traditional SINDy predominantly fall within the $\pm 10\%$ range, with corre-
 266 sponding 95.5% and 75.7% confidence intervals (CIs). These results clearly indicate that the
 267 predictive performance of our PINNSR model is substantially superior to that of SINDy.

268 2.2 General governing equation

269 Upon acquiring the coefficients through machine learning-based dimensional analysis, we
 270 define $\chi = \frac{H}{d_p} \frac{E_p}{P}$ as the dimensionless relaxation indicator (details can be found in the main
 271 text). Then the general governing equation considering system parameters can be formulated
 272 as follows:

$$\frac{d\mu}{dt} = f_1(\chi) \cdot \frac{d\gamma}{dt} + f_2(\chi) \cdot \mu \left| \frac{d\gamma}{dt} \right| + f_3(\chi) \cdot \mu^2 \frac{d\gamma}{dt}, \quad (19)$$

273 where $f_i(\chi) = a_i \log_{10}(\chi) + b_i$ for $i = 1, 2, 3$, with coefficients given by $a_1 = -1.98, b_1 = 3.87;$
 274 $a_2 = -15.01, b_2 = 45.65$; and $a_3 = -26.31, b_3 = 96.22$.

275 2.2.1 Validation under different setups

276 To investigate the applicability of Eq. (19) across varying configurations, particularly its
 277 extrapolative capability, we employ the setup with $d_p = 15$ mm for validation. It is notewor-
 278 thy that the DEM simulations utilized in deriving the general governing equation (Eq. (19))
 279 involved particle size variations within the range of $4 \sim 10$ mm. Furthermore, the values of
 280 E_p, P , and H/d_p lie outside the parameter space of the training dataset. The specific values
 281 of this parameter are detailed in Table 1. Within the ranges specified in Table 1, 67 sets of
 282 Heaviside cases are generated randomly to test the generalization capability of the general
 283 governing equation across both different configurations and varying loading conditions (as
 284 equation is discovered in the sinusoidal loading scenario).

Table 1 System variables of the oscillatory simple shear

System Variable	Symbol	Value	Unit
Pressure	P	12, 14, 16, 18, 150, 350, 450	kPa
Height	H	$(13, 17, 21) \cdot d_p$	mm
Period	T	15	s
Particle diameter	d_p	15	mm
Density	ρ	2500	kg m^{-3}
Particle modulus	E_p	60, 80, 400, 600, 800	MPa

285 Figure 9 presents the validation results for the general governing equation in these 67 test
 286 sets. Specifically, Fig. 9a shows that the correlation coefficient (r) between model predictions
 287 and DEM results exceeds 0.960 for all cases, indicating that the derived equation exhibits
 288 high predictive precision. Almost 50% of these cases yield $r > 0.995$, where the predictive
 289 performance is excellent; representative results in Case-1 (see Fig. 9b) demonstrate a near-
 290 complete overlap between DEM simulation curves and model predictions. Furthermore, Case-
 291 2 ($r = 0.960$) with the poorest performance is presented in Fig. 9c. Here, the model still
 292 captures the trend reasonably well, accurately predicting the plateau of μ (representing the
 293 static friction coefficient μ_s). Minor deviations in the curve during shear reversal in Case-2
 294 are attributed to the incomplete development of the plateau μ , which is different from the
 295 fully developed behavior observed in Case-1. Extending the DEM simulation period would
 296 significantly improve predictive performance.

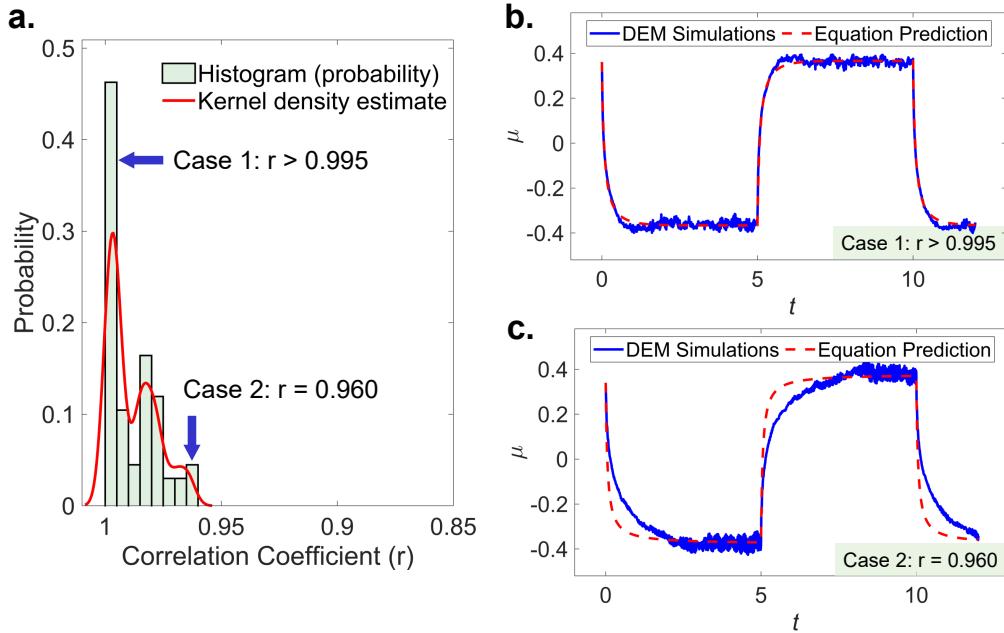


Fig. 9 Validation results for the general governing equation: **a** Regression analysis of all test cases, where the light green histogram height indicates the probability within each interval and the red line denotes the continuous kernel density estimate. **b** and **c** present two representative cases with $r > 0.995$ and $r = 0.960$, respectively. Solid blue lines represent DEM simulation data and the dashed red lines are the outcomes from the Runge-Kutta integration of the identified equation.

2.2.2 Steady state analysis

When our equation is subjected to the steady-state condition, defined by $\frac{d\mu}{dt} = 0$, it reduces to the form given in Eq. (20):

$$C_1 + C_2 \cdot \mu \cdot \text{sgn}\left(\frac{d\gamma}{dt}\right) + C_3 \cdot \mu^2 = 0. \quad (20)$$

In this state, stress becomes independent of the strain rate, depending solely on its direction. This finding aligns with established theoretical insights into transient flow within the quasi-static regime, where stress is known to be shear-rate-independent. Consequently, our model is fully consistent with current theoretical frameworks.

Moreover, solving the quadratic equation (Eq. (20)) yields the steady-state solution for Eq. (17). Within the quasi-static regime, this solution corresponds to μ_s in the well-known $\mu(I)$ rheology, which describes the static friction coefficient as the inertial number approaches zero ($I \rightarrow 0$). Analysis of the derived governing equation, incorporating coefficients $\{C_1, C_2, C_3\}$, reveals that μ_s values exhibit convergence across all scenarios, forming a roughly planar distribution in the three-dimensional (C_1, C_2, C_3) space. Fitting this planar distribution via the form of Eq. (20) yields a global μ_s value, denoted $\mu_s^{\text{global}} = 0.3664$. As a steady-state physical quantity derived from the governing equation, μ_s is formally the solution to a quadratic equation determined by the coefficients (C_1, C_2, C_3) —which, as noted earlier, are governed by the identified dimensionless number χ . However, our analysis indicates that μ_s shows minimal sensitivity to variations in χ , and instead consistently converges to a constant value.

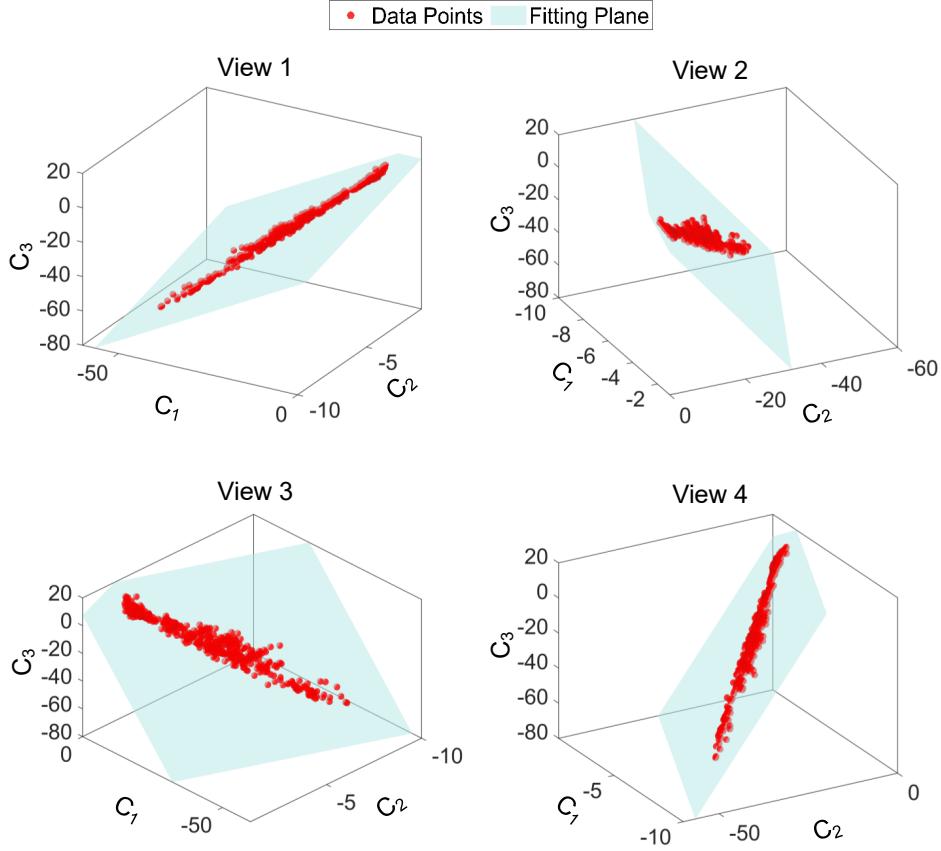


Fig. 10 Data points and fitted plane in (C_1, C_2, C_3) coordinates from different views. Each point represents a set of coefficients for case-specific governing equation; Almost all points lie on a single plane.

315 The equation of the plane in Fig. 10 is as follows.

$$C_1 - C_2 \cdot \mu_s + C_3 \cdot \mu_s^2 = 0, \quad (21)$$

316 where the fitting results of $\mu_s = 0.3664$.

317 2.2.3 Transient state analysis

318 We analyze the discovered governing equation (Eq. (17)) under transient conditions, aiming
 319 to elucidate the physical significance of each term. Figure 11 illustrates the contributions
 320 of individual terms in sinusoidal and Heaviside cases. We can observe that the rate of friction
 321 evolution ($\frac{d\mu}{dt}$) emerges as a balance of three distinct contributions, as observed therein.
 322 For simplicity, we elaborate on these terms using the Heaviside regime (Fig. 11a): (i) The linear
 323 response term ($C_1 \cdot \frac{d\gamma}{dt}$) dictates the immediate, proportionate adjustment of friction to
 324 the imposed rate $\dot{\gamma}$, exhibiting a profile congruent with that of $\dot{\gamma}$; (ii) The dissipative term
 325 ($(C_2 \cdot \mu \left| \frac{d\gamma}{dt} \right|)$) with ($C_2 < 0$) encapsulates energy dissipation, its contribution scaling monotonically
 326 with both the current friction magnitude μ and the absolute strength of the driving rate,
 327 manifesting as a progressively strengthening (or stabilizing) component that dampens excessive
 328 variations in μ over time; (iii) The nonlinear response regulatory term ($C_3 \cdot \mu^2 \frac{d\gamma}{dt}$) enables
 329 state-dependent modulation of friction evolution, with its contribution becoming significant
 330 only as μ accumulates—either amplifying or suppressing $\frac{d\mu}{dt}$ in a manner that regulates transitions
 331 between friction states, often driving the system toward (or away from) steady-state
 332 behavior depending on the sign of C_3 .

This further underscores the efficacy of data-driven approaches: the sinusoidal case depicted in Fig. 11b, despite its complexity and resistance to manual analysis, contains a wealth of inherent features that are easily discernible by the algorithm. The resulting machine-mined equation enables the analysis of simple human-interpretable scenarios to elicit the physical implications of each term. This process highlights the sophistication of our method, which holds promise as a new paradigm for future research in granular mechanics.

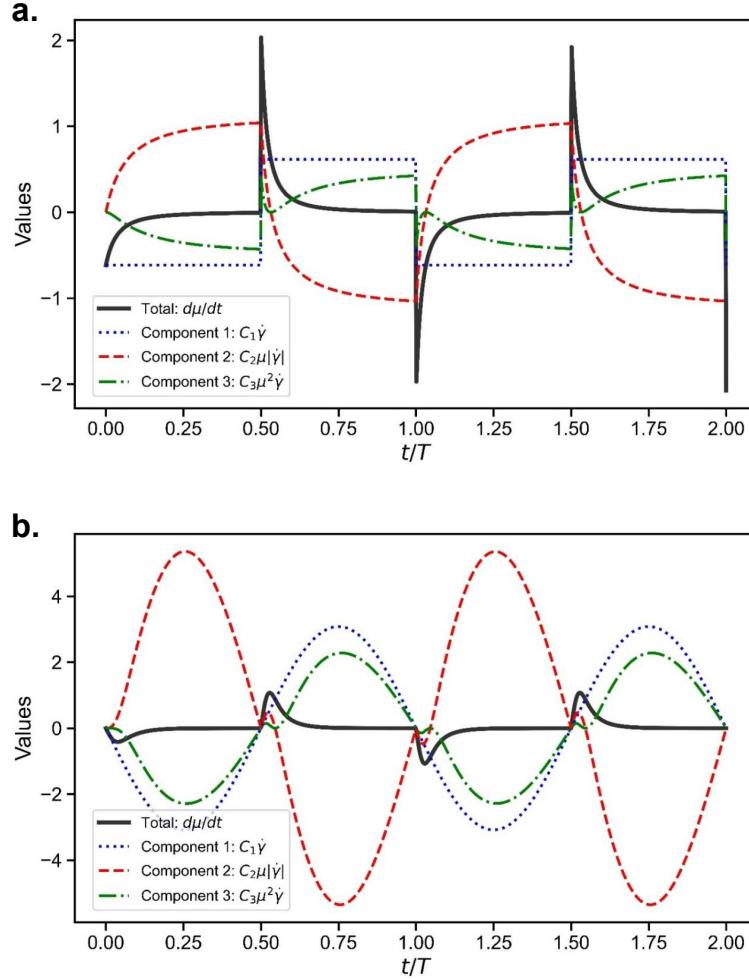


Fig. 11 Contributions of individual terms in the governing equation. **a** Heaviside case; **b** Sinusoidal case. The solid red line denotes the total derivative; the dashed blue line represents the first component ($C_1\dot{\gamma}$); the dashed red line corresponds to the second component ($C_2\mu|\dot{\gamma}|$); and the green dashed line indicates the third component ($C_3\mu^2\ddot{\gamma}$).

339 2.3 Granular relaxation characteristics implicit in the governing 340 equation

In this section, we investigate the granular relaxation characteristics inherent in the governing equation we have identified. For simplicity, we consider the Heaviside loading scenario: the relaxation process triggered by an instantaneous shear rate reversal, where the shear rate switches from $\dot{\gamma}_0$ to $-\dot{\gamma}_0$ (with $\dot{\gamma}_0 > 0$). As detailed in Section 2.3.1, we can derive an analytical solution to characterize this relaxation process. We also define a characteristic relaxation time on a dimensionless time axis ($\dot{\gamma}_0 t$), as described in Section 2.3.2.

2.3.1 Analytical Solution for Heaviside Loading Cases

For the Heaviside loading case, where the shear rate is constant ($\dot{\gamma}_0$ or $-\dot{\gamma}_0$), Eq. (17) mathematically takes the form of a Riccati equation. We can derive an analytical solution to this equation by applying the corresponding boundary conditions. The Riccati equation is solved via the standard particular solution method. Given a particular solution $y_1(x)$, the substitution $y(x) = y_1(x) + v(x)^{-1}$ reduces the equation to a first-order linear differential equation. Here, the particular solution is $y_1 = \mu_s$. Thus, we introduce the transformation:

$$\mu = \mu_s + \frac{1}{v}, \quad (22)$$

where $v = v(t)$ is an auxiliary function.

Step 1: Derivative calculation

Differentiate (22) with respect to time:

$$\frac{d\mu}{dt} = \frac{d}{dt} (\mu_s + v^{-1}) = -v^{-2} \frac{dv}{dt}. \quad (23)$$

Step 2: Substitution into governing equation

The original Riccati equation is:

$$\frac{d\mu}{dt} = \dot{\gamma}_0 (-C_1 + C_2\mu - C_3\mu^2). \quad (24)$$

Substitute (22) into the right-hand side of (24):

$$\text{RHS} = \dot{\gamma}_0 \left[-C_1 + C_2 (\mu_s + v^{-1}) - C_3 (\mu_s + v^{-1})^2 \right]. \quad (25)$$

Expanding the quadratic term:

$$(\mu_s + v^{-1})^2 = \mu_s^2 + 2\mu_s v^{-1} + v^{-2}, \quad (26)$$

and utilizing the steady-state condition, we simplify (25) to:

$$\text{RHS} = \dot{\gamma}_0 [(C_2 - 2C_3\mu_s)v^{-1} - C_3v^{-2}]. \quad (27)$$

Step 3: Equation for auxiliary function

Equating (23) and (27) yields the following:

$$-v^{-2} \frac{dv}{dt} = \dot{\gamma}_0 [(C_2 - 2C_3\mu_s)v^{-1} - C_3v^{-2}]. \quad (28)$$

Multiplying both sides by $-v^2$:

$$\frac{dv}{dt} = -\dot{\gamma}_0 [(C_2 - 2C_3\mu_s)v - C_3]. \quad (29)$$

This represents a first-order linear ODE of the form $dv/dt = av + b$, where:

$$a = -\dot{\gamma}_0(C_2 - 2C_3\mu_s), \quad b = \dot{\gamma}_0C_3. \quad (30)$$

Step 4: General solution

The solution to (29) is:

$$v(t) = e^{at} \left(\int be^{-at} dt + A \right), \quad (31)$$

with A denoting the integration constant. Solving the integral:

$$\int be^{-at} dt = -\frac{b}{a}e^{-at} \quad (a \neq 0), \quad (32)$$

³⁶⁹ we obtain:

$$v(t) = Ae^{at} - \frac{b}{a}. \quad (33)$$

³⁷⁰ Substituting coefficients from (30) and defining $D \equiv C_2 - 2C_3\mu_s$:

$$v(t) = Ae^{-\dot{\gamma}_0 D t} + \frac{C_3}{D}. \quad (34)$$

³⁷¹ Note: The case $D = 0$ requires separate treatment and is excluded here.

³⁷² Step 5: Back-substitution for stress

³⁷³ Reintroduce (34) into (22):

$$\mu = \mu_s + \frac{1}{v} = \mu_s + \frac{D}{C_3 + AD e^{-\dot{\gamma}_0 D t}}. \quad (35)$$

³⁷⁴ Defining $B \equiv AD$ simplifies (35) to:

$$\mu(t) = \mu_s + \frac{D}{C_3 + Be^{-\dot{\gamma}_0 D t}}. \quad (36)$$

³⁷⁵ Step 6: Boundary conditions

³⁷⁶ Condition 1: $\lim_{t \rightarrow \infty} \mu(t) = \mu_s$. For $\dot{\gamma}_0 > 0$, stability requires $D < 0$, ensuring the exponential term vanishes at infinity.

³⁷⁸ Condition 2: $\mu(0) = -\mu_s$. Applying to (36):

$$-\mu_s = \mu_s + \frac{D}{C_3 + B} \implies -2\mu_s = \frac{D}{C_3 + B}. \quad (37)$$

³⁷⁹ Solving for B :

$$B = -\frac{D}{2\mu_s} - C_3. \quad (38)$$

³⁸⁰ Step 7: Constant simplification

³⁸¹ Simplifying (38):

$$B = -\frac{C_2 - 2C_3\mu_s}{2\mu_s} - C_3 = -\frac{C_2}{2\mu_s}. \quad (39)$$

³⁸² Substituting (39) into (36) yields the analytical solution:

$$\mu(t) = \mu_s + \frac{C_2 - 2C_3\mu_s}{C_3 - \frac{C_2}{2\mu_s} \exp[-\dot{\gamma}_0(C_2 - 2C_3\mu_s)t]}. \quad (40)$$

³⁸³ 2.3.2 Characteristic relaxation time

³⁸⁴ For the Heaviside loading scenario, Fig. 12 presents discrete element method (DEM) simulation results for the typical configuration ($P = 10$ kPa, $H = 15$ d_p , $\rho = 2500$ kg/m³, $d_p = 0.005$ m, $E_p = 5 \times 10^7$ Pa) in varying amplitudes of $\dot{\gamma}_0$. These results demonstrate a ³⁸⁵ consistent relaxation behavior on dimensionless-time axis ($\dot{\gamma}_0 t$), as evidenced by all curves ³⁸⁶ collapsing into a single curve during relaxation following the sudden shear reversal at $t = 0$. ³⁸⁷

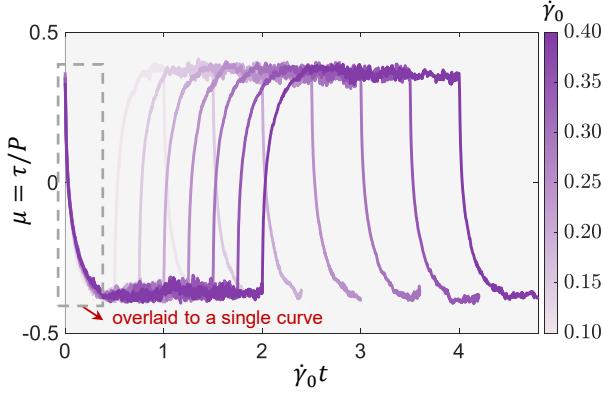


Fig. 12 Comparison of force chains at representative instants for $H/d_p = 10$ and 50 .

389 To quantify the relaxation process on this axis ($\dot{\gamma}_0 t$), we introduce a characteristic relax-
 390 ation time \hat{t} . Guided by the stretched exponential Kohlrausch-Williams-Watts law (Eq. (41)),
 391 with $t = t_0$, \hat{t} is defined as the dimensionless time at which $\mu = (2e^{-1} - 1)\mu_s$. This yields the
 392 analytical expression for \hat{t} in Eq. (42).

$$\mu(t) = \mu_2 - (\mu_2 - \mu_1) \cdot e^{-t_0/\hat{t}}. \quad (41)$$

$$\hat{t} = \frac{1 + \ln \left(C_2 - 2(1 - e^{-1}) \frac{C_3 \mu_s}{C_2} \right)}{|C_2 - 2C_3 \mu_s|}. \quad (42)$$

393 Given the governing equations of a granular system, the characteristic relaxation time can
 394 be derived via Eq. (42) using the equation coefficients (C_1 , C_2 and C_3) and its steady-state
 395 solution μ_s .

396 2.4 Micro-interpretations in terms of fabric tensor

397 2.4.1 Evolution of the force chain

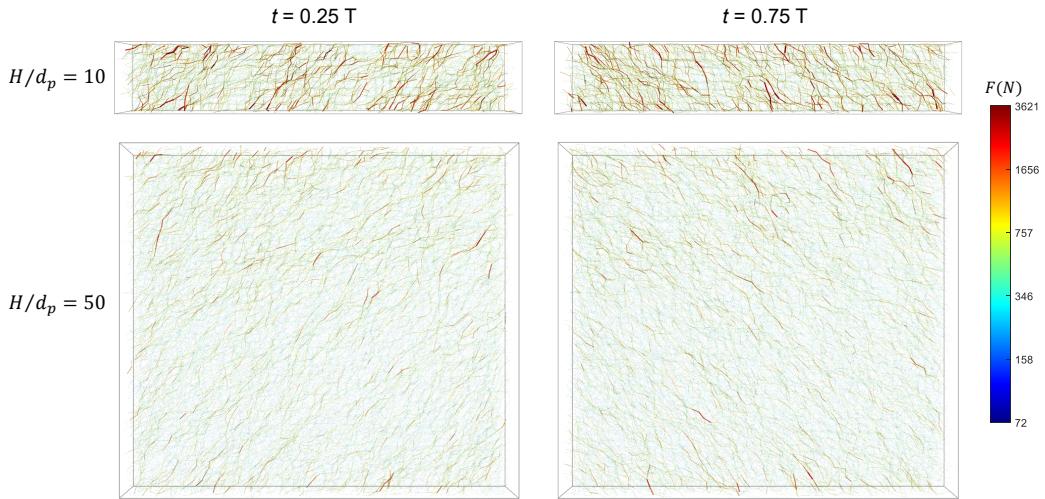


Fig. 13 Comparison of force chains at representative instants for $H/d_p = 10$ and 50 .

In this section, we primarily compare the effects of different H/d_p configurations on force chain structures. Figure 13 compares force chain distributions for $H/d_p = 10$ and $H/d_p = 50$ at two representative instants: $t = 0.25T$ and $t = 0.75T$.

Color encodes contact force magnitude, and all plots normalized to the same colorbar. From these, it is evident that for smaller H/d_p , during transient evolution, the force chain scale (L_{fc}) is comparable to H or even spans the entire system, resulting in greater strength (manifested as higher viscosity). Conversely, for larger H/d_p , L_{fc} is smaller than H ; interparticle force chains break and rearrange more easily, facilitating the system's rapid attainment of a new equilibrium. In summary, a reduced system size enhances the strength of the granular system and prolongs the relaxation times on the dimensionless time axis ($\dot{\gamma}_0$).

2.4.2 Evolution of the fabric tensor

The fabric tensor is a second-order tensor that quantitatively characterizes the anisotropy of particulate microstructures by encoding the statistical distribution of contact normals. Rose diagrams are polar plots that visually represent this directional information: the radial extent at any angle corresponds to the relative frequency of contact normals aligned with that direction. They serve as an intuitive geometric counterpart to the fabric tensor, translating the compact mathematical description of the tensor into a more accessible visualization of preferred orientations within the contact network.

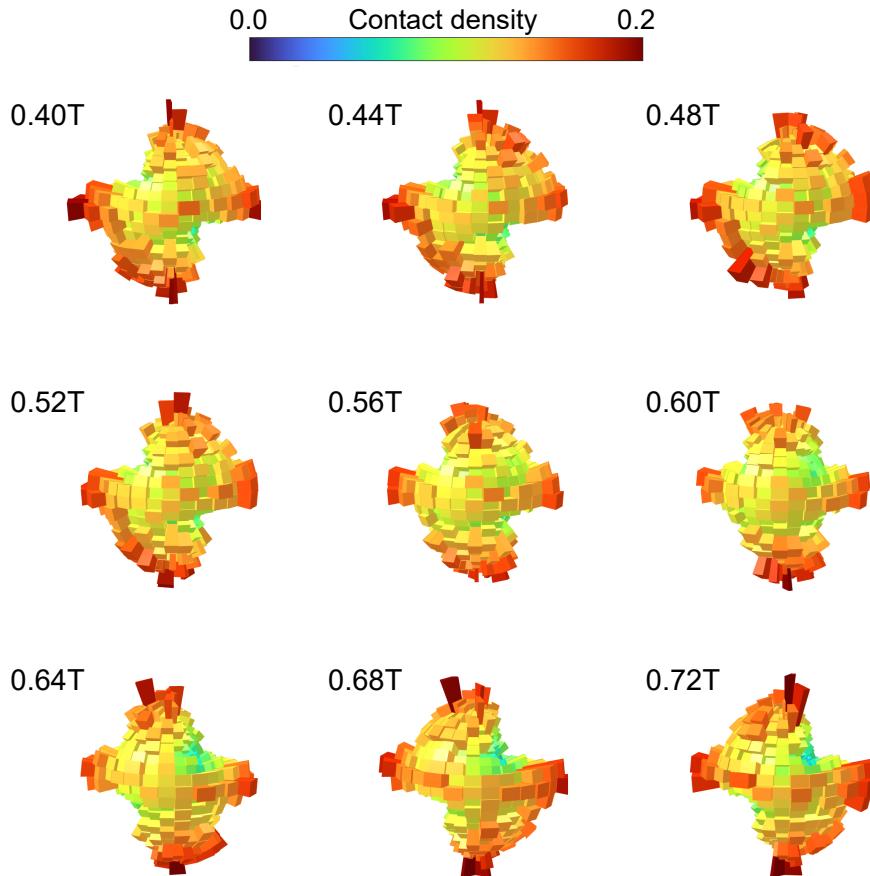


Fig. 14 Evolution of rose diagrams of contact normals for all particles at different time instants.

Figure 14 presents the temporal evolution of rose diagrams of contact normals for all particles in a representative sinusoidal case at various time instants. Shear loading is reversed at $t = 0.5T$. From $0.40\sim0.48T$, the rose diagrams exhibit a consistent orientation (northeast-southwest). The period $0.52\sim0.60T$ corresponds to the adjustment process of contact normals

throughout the system after shear reversal, gradually shifting from one predominant direction to another. Between $0.64T$ and $0.72T$, they stabilize in the southeast-northwest direction. The entire process exhibits a distinct evolutionary pattern, particularly the post-reversal adjustment. The evolutionary behavior of contact normals at the particle scale is of paramount importance, as it governs the macroscopic behaviors of the granular system.

2.4.3 Existing evolution law of fabric tensor

In this study, we introduce the fabric tensor \mathbf{A} as a traceless symmetric second-order tensor (see Eq. (43)) to quantify the anisotropy of microstructural features. The eigenvectors of \mathbf{A} correspond to the principal directions of the average orientations of the contacts, whereas their eigenvalues measure the degree of alignment of these contacts along these principal axes. Consequently, the level of anisotropy can be evaluated by the difference between the maximum and minimum eigenvalues.

$$\mathbf{A} = \frac{1}{N_C} \sum_{i=1}^{N_C} \left(\mathbf{n}^i \mathbf{n}^i - \frac{1}{3} \mathbf{I} \right). \quad (43)$$

Here, \mathbf{n}^i represents the i -th unit contact vector in a granular assembly containing N_C contacts, and \mathbf{I} denotes the identity tensor. In the reference⁴, a governing equation was proposed to describe the evolution of the fabric tensor, as given by Eq. (44).

$$\overset{\circ}{\dot{\mathbf{A}}} = c_1 \mathbf{S} + c_2 |\mathbf{D}| \mathbf{A} + c_3 (\mathbf{A} : \mathbf{S}) \mathbf{A}. \quad (44)$$

In this context, $\overset{\circ}{\dot{\mathbf{A}}}$ denotes the Jaumann derivative of \mathbf{A} , defined as $\overset{\circ}{\dot{\mathbf{A}}} = \dot{\mathbf{A}} + \mathbf{A} \cdot \mathbf{W} - \mathbf{W} \cdot \mathbf{A}$. Here, \mathbf{W} is the spin tensor, expressed as $\mathbf{W} = \frac{1}{2} (\nabla \mathbf{v} - (\nabla \mathbf{v})^T)$ with \mathbf{v} representing the velocity field, and $\dot{\mathbf{A}}$ signifies the material time derivative of \mathbf{A} . We define the strain-rate tensor and the deviatoric strain-rate tensor as $\mathbf{D} = \frac{1}{2} (\nabla \mathbf{v} + (\nabla \mathbf{v})^T)$ and $\mathbf{S} = \mathbf{D} - \frac{1}{3} \text{tr}(\mathbf{D}) \mathbf{I}$, respectively. The notation $|\cdot|$ indicates the tensor norm such that $|\mathbf{D}| = \sqrt{\mathbf{D} : \mathbf{D}}$. The parameters $c_1 \sim c_3$ are distinct from our $C_1 \sim C_3$ in Eq. (17). They are defined as material constants in the theoretical framework⁴.

Within our discrete element method (DEM) simulations, particulate velocities along the y - and z -axes manifest as negligible fluctuations centered at zero. This results in the approximations $\mathbf{D} \approx \mathbf{S} \approx \frac{1}{2} \dot{\gamma} (\hat{\mathbf{i}} \otimes \hat{\mathbf{j}} + \hat{\mathbf{j}} \otimes \hat{\mathbf{i}})$. Consequently, despite Eq. (44) is a tensor-based governing equation, its mathematical structure exhibits fundamental alignment with Eq. (17). Furthermore, both our DEM simulations and the data from the literature⁴ reveal that the values of A_{xx} and A_{zz} are negligible compared to A_{xz} , with the components A_{yy} , A_{xy} , and A_{yz} approach nullity.

We next seek to reduce the tensor equation to a scalar form, facilitating a direct comparison with our derived governing equation. Given that all y -axis related components of the fabric tensor are effectively zero, we simplify subsequent mathematical derivations by neglecting the y -dimension and working with a 2×2 tensor. The detailed derivation proceeds as follows.

Step 1 Express each tensor in Eq. (44) in its scalar form:

$$\mathbf{A} = \begin{bmatrix} A_{xx} & A_{xz} \\ A_{zx} & A_{zz} \end{bmatrix}, \quad (45)$$

$$\nabla \mathbf{v} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial z} \end{bmatrix} = \begin{bmatrix} 0 & \dot{\gamma} \\ 0 & 0 \end{bmatrix}, \quad (46)$$

$$\begin{aligned} \mathbf{D} &= \mathbf{S} = \frac{1}{2} (\nabla \mathbf{v} + (\nabla \mathbf{v})^T) \\ &= \begin{bmatrix} 0 & \frac{1}{2} \dot{\gamma} \\ \frac{1}{2} \dot{\gamma} & 0 \end{bmatrix}, \end{aligned} \quad (47)$$

$$|\mathbf{D}| = \sqrt{\mathbf{D} : \mathbf{D}} = \sqrt{D_{ij} D_{ij}} = \frac{\sqrt{2}}{2} |\dot{\gamma}|, \quad (48)$$

$$\overset{\circ}{\mathbf{A}} = \begin{bmatrix} \dot{A}_{xx} - \dot{\gamma} A_{xz} & \dot{A}_{xz} + \frac{1}{2} \dot{\gamma} (A_{xx} - A_{zz}) \\ \dot{A}_{zx} + \frac{1}{2} \dot{\gamma} (A_{xx} - A_{zz}) & \dot{A}_{zz} + \dot{\gamma} A_{xz} \end{bmatrix}. \quad (49)$$

Step 2 Substitute Eqs. (45)~(49) into Eq. (44):

The left-hand side is

$$\text{LHS} = \begin{bmatrix} \dot{A}_{xx} - \dot{\gamma} A_{xz} & \dot{A}_{xz} + \frac{1}{2} \dot{\gamma} (A_{xx} - A_{zz}) \\ \dot{A}_{zx} + \frac{1}{2} \dot{\gamma} (A_{xx} - A_{zz}) & \dot{A}_{zz} + \dot{\gamma} A_{xz} \end{bmatrix}, \quad (50)$$

and the right-hand side is

$$\text{RHS} = \begin{bmatrix} \left(\frac{\sqrt{2}}{2} c_2 |\dot{\gamma}| + c_3 \dot{\gamma} A_{xy}\right) A_{xx} & \left(\frac{\sqrt{2}}{2} c_2 |\dot{\gamma}| + c_3 \dot{\gamma} A_{xy}\right) A_{xy} + \frac{1}{2} c_1 \dot{\gamma} \\ \left(\frac{\sqrt{2}}{2} c_2 |\dot{\gamma}| + c_3 \dot{\gamma} A_{xy}\right) A_{yx} + \frac{1}{2} c_1 \dot{\gamma} & \left(\frac{\sqrt{2}}{2} c_2 |\dot{\gamma}| + c_s \dot{\gamma} A_{xy}\right) A_{yy} \end{bmatrix}. \quad (51)$$

From Eqs. (50) and (51), we ultimately obtain three scalar equations:

$$\frac{dA_{xx}}{dt} = A_{xz} \frac{d\gamma}{dt} + \frac{\sqrt{2}}{2} c_2 \left| \frac{d\gamma}{dt} \right| A_{xx} + c_3 \frac{d\gamma}{dt} A_{xz} A_{xx}, \quad (52)$$

$$\frac{dA_{zz}}{dt} = - \frac{d\gamma}{dt} A_{xz} + \frac{\sqrt{2}}{2} c_2 \left| \frac{d\gamma}{dt} \right| A_{zz} + c_3 \frac{d\gamma}{dt} A_{xz} A_{zz}, \quad (53)$$

$$\frac{dA_{xz}}{dt} = \frac{1}{2} (A_{zz} - A_{xx} + c_1) \frac{d\gamma}{dt} + \frac{\sqrt{2}}{2} c_2 \left| \frac{d\gamma}{dt} \right| A_{xz} + c_3 \frac{d\gamma}{dt} A_{xz}^2. \quad (54)$$

Given the small magnitudes of A_{xx} and A_{zz} under the shear-dominated conditions investigated herein, our focus centers on Eq. (54). Our analyses reveal that the influence of $A_{zz} - A_{xx}$ on the coefficient of $\frac{d\gamma}{dt}$ remains consistently below 1%, warranting its omission. Upon neglecting this term, Eq. (17) and Eq. (54) exhibit identical structural forms, with the former describing the evolution of μ and the latter indicating that of A_{xz} .

3 Discussion

3.1 Comparison between PINNSR-DA and other methods

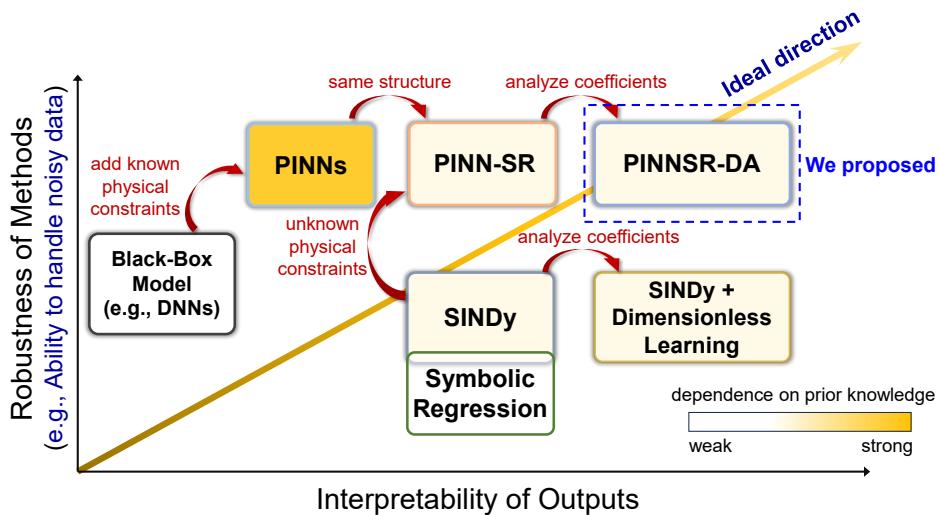


Fig. 15 Comparison between PINNSR-DA and other methods.

465 As shown in Fig. 15, a comparative analysis of various methods is conducted along two
 466 dimensions: interpretability, represented by the horizontal axis, and robustness, which denotes
 467 the capacity to accommodate noisy data and is represented by the vertical axis. Color coding
 468 is employed to delineate the extent of dependence on prior knowledge. Ideally, the objective is
 469 to pursue models characterized by high interpretability and robustness, coupled with minimal
 470 reliance on prior knowledge. Accordingly, the direction indicated by the yellow diagonal arrow
 471 represents the desired trajectory for model advancement. It is observed that black-box models,
 472 exemplified by deep neural networks, are situated in the relatively lower-left region. In con-
 473 trast, Physics-Informed Neural Networks (PINNs) can enhance interpretability and mitigate
 474 overfitting by integrating established physical constraints.

475 Methods such as SINDy and symbolic regression, because of their ability to derive explicit
 476 equations, demonstrate strong interpretability; however, both are highly susceptible to noise.
 477 Compared to symbolic regression, SINDy demands more prior knowledge, as it requires the
 478 pre-construction of a SINDy library. Nevertheless, this library can be easily constructed for
 479 the rheological governing equations. Notably, constructing such a library is straightforward
 480 for rheological governing equations, which typically comprises stress and shear rate deriva-
 481 tives up to the third order, as well as their polynomial combinations of degree three or lower.
 482 The PINNSR-DA framework proposed in this work integrates the advantages of PINNs and
 483 SINDy, thereby achieving a superior position in terms of the desired attributes. Further-
 484 more, by discovering the governing equations, we delve into the dimensionless expressions of
 485 the equation coefficients by machine learning-based dimensional analysis (DA), which fur-
 486 ther enhances interpretability. In summary, PINNSR-DA demonstrates significant advantages
 487 over existing methods and is expected to be applied in multiple domains to identify general
 488 governing equations.

489 3.2 Influences of the number of loading sets on equation discovery

490 This section presents the effects of the number of loading groups on equation discovery
 491 results. As observed in Fig. 16, the mean squared error (MSE) and mean absolute percentage
 492 error (MAPE) are relatively large when only two training groups are used. This arises because
 493 the structure of the equation is not correctly identified, with only two terms captured, as
 494 evidenced in Table 2. Here, the MSE is computed by comparing the Runge-Kutta numerical
 495 solutions of the mined equations with the discrete element method (DEM) data (Heaviside
 496 case). Incorporating more than 3 distinct loading amplitude datasets for equation discovery
 497 leads to diminishing returns in predictive accuracy, reflected by the plateau in MSE. Although
 498 3 groups may fail in certain cases, 4 groups consistently yield stable results.

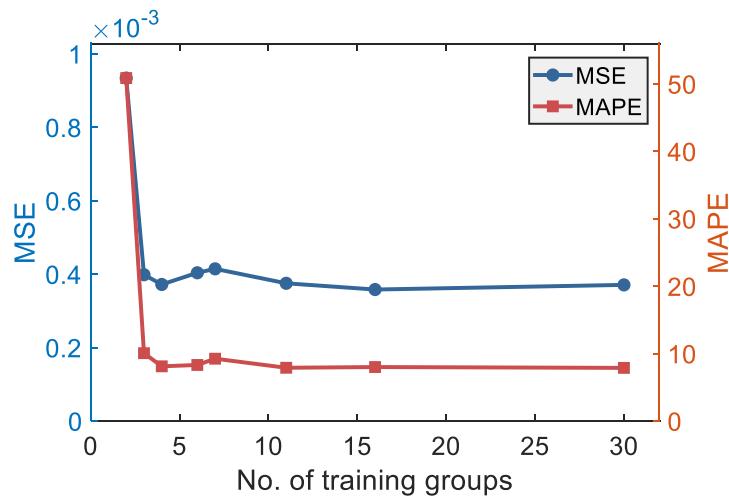


Fig. 16 Influence of the number of training groups on equation discovery results (evaluated by MSE and MAPE).

Table 2 Discovered equation by different sizes of data

NO. of Groups	MSE	RMSE	MAE	MAPE	Discovered Equation
2	0.00093	0.03056	0.02166	50.86761	$\sigma_t = -5.70\gamma_t - 15.91\sigma \gamma_t - 34.16\sigma^2\gamma_t$
3	0.00040	0.01996	0.01703	10.06807	$\sigma_t = -6.00\gamma_t - 28.58\sigma \gamma_t - 35.39\sigma^2\gamma_t$
4	0.00037	0.01930	0.01628	8.11644	$\sigma_t = -6.09\gamma_t - 29.25\sigma \gamma_t - 35.57\sigma^2\gamma_t$
6	0.00040	0.02010	0.01699	8.31772	$\sigma_t = -6.07\gamma_t - 29.27\sigma \gamma_t - 35.59\sigma^2\gamma_t$
7	0.00041	0.02036	0.01732	9.24736	$\sigma_t = -6.02\gamma_t - 29.04\sigma \gamma_t - 35.36\sigma^2\gamma_t$
11	0.00038	0.01937	0.01634	7.90100	$\sigma_t = -6.10\gamma_t - 29.35\sigma \gamma_t - 35.59\sigma^2\gamma_t$
16	0.00036	0.01893	0.01595	8.00868	$\sigma_t = -6.10\gamma_t - 29.25\sigma \gamma_t - 35.33\sigma^2\gamma_t$
30	0.00037	0.01926	0.01623	7.88448	$\sigma_t = -6.10\gamma_t - 29.37\sigma \gamma_t - 35.67\sigma^2\gamma_t$

Note: MSE (Mean Squared Error), RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error) are common metrics for evaluating regression model performance.

499 3.3 Influences of various parameters on the results

500 3.3.1 DEM simulation results

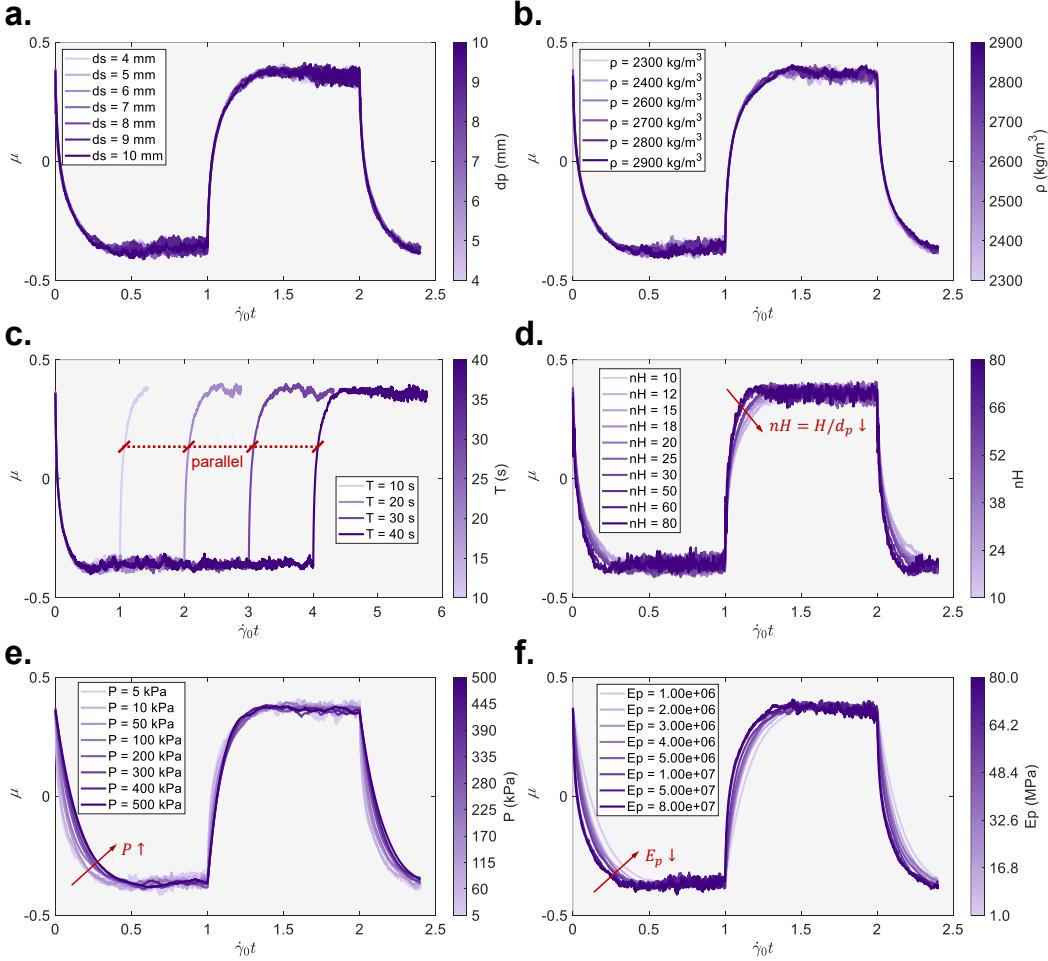


Fig. 17 Parameters with significant influence on DEM simulation results: a. Particle diameter (d_s), b. Particle density (d_s), c. Loading period (T), d. Number of flow layers (H/d_p), e. Normal stress (P), f. Young's modulus (E_p).

501 Herein, we illustrate the effects of varying parameters on the outcomes of discrete element
502 method (DEM) simulations, as presented in Fig. 17. It should be noted that all results depicted

in this figure correspond to the Heaviside loadings, chosen to afford a straightforward and intuitive demonstration of how each parameter influences the relaxation time. Our analysis commences with a representative case defined by the following parameters: $P=10$ kPa, $P = 10$ kPa, $d_p = 4$ mm, $\rho = 2500 \text{ kg} \cdot \text{m}^{-3}$, $H/d_p = 15$, $T = 10$ s, $E_p = 50$ MPa. For each subfigure, only the focal variable is adjusted while all other parameters remain unchanged. Furthermore, the results shown in Figs. 17a-e are normalized with respect to the $\dot{\gamma}_0 t$ axis. It can be observed that the diameter of particles (d_p), the density of particles (ρ), and the loading periods (T) exert virtually no influence on the relaxation processes after the instantaneous shear rate reversal revealed by DEM simulations. In contrast, P , E_p and H/d_p do affect these processes, manifested specifically as follows: Relaxation time decreases with smaller H/d_p , softer particles (lower E_p) and higher pressure P .

Furthermore, this study also investigates the influence of the inter-particle friction coefficient (μ_p) on the results. Specifically, the range of μ_p values examined in this work is 0.3 to 0.6, which is consistent with typical μ_p of most non-metallic mineral particles (e.g., quartz sand and limestone) in practical engineering scenarios. As illustrated in Fig. 18, the evolution of the effective friction coefficient exhibits insensitivity to μ_p . All results presented in this work are simulated using a fixed μ_p value of 0.5.

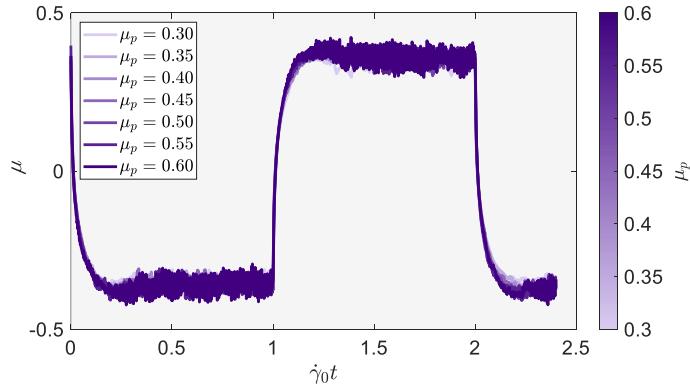


Fig. 18 DEM simulation results under different inter-particle friction coefficients (μ_p).

3.3.2 Equation coefficient results

In our study, the governing equations we derived have the same structure, but their coefficients vary depending on the system variables. This section provides a straightforward demonstration of how different system variables influence the coefficients of the equations. Consistent with prior findings, Fig. 19 indicates that (d_p), (ρ) and (T) have virtually no impact on the coefficients $\{C_1, C_2, C_3\}$, as their trends remain largely horizontal. Minor fluctuations in these coefficients emerge only when $T < 15$ s, stemming from insufficient development of μ within a cycle for such a small period value. This is manifested in two ways: the curves in Fig. 17 lack a plateau region, and the quadratic equation describing the return to steady state (see Eq. 21) has no steady-state solution, with its discriminant $\Delta = C_2^2 - 4C_1C_3 < 0$. For $T > 15$ s, increasing T no longer modifies the coefficient values, which means that the period does not affect the results. To avoid the confounding effect of small periods, in the main text section *Learning the equation coefficients*, these anomalous data (where μ is underdeveloped) have been excluded, with analysis confined to cases where μ is fully developed.

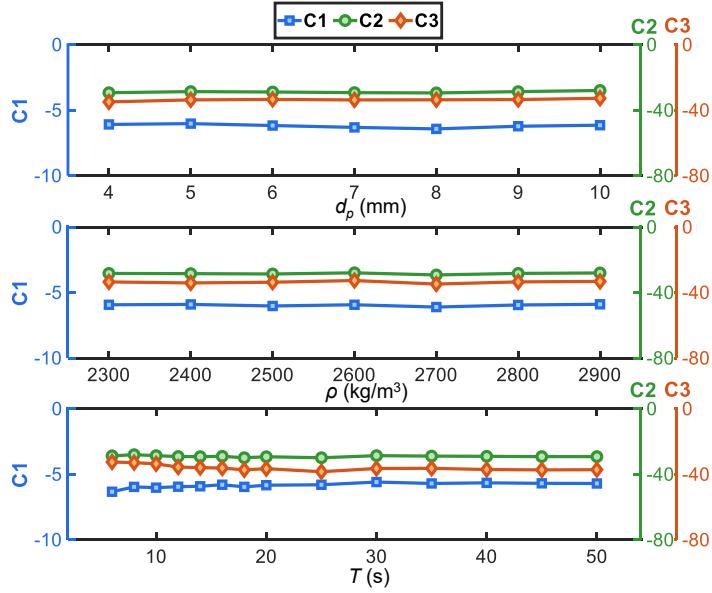


Fig. 19 Parameters with minor influence on equation coefficients: d_p , ρ , T .

Figure 20 illustrates the parameters that affect the coefficients: normal stress (P), Young's modulus of particles (E_p) and number of flow layers (H/d_p). It is observed that the variation trends of the three coefficients remain consistent across these influencing factors, a phenomenon arising from their non-independence, which will be elaborated in the subsequent Section 3.4. Specifically, the values of these three coefficients increase with rising P , while decreasing as E_p and H/d_p increase.

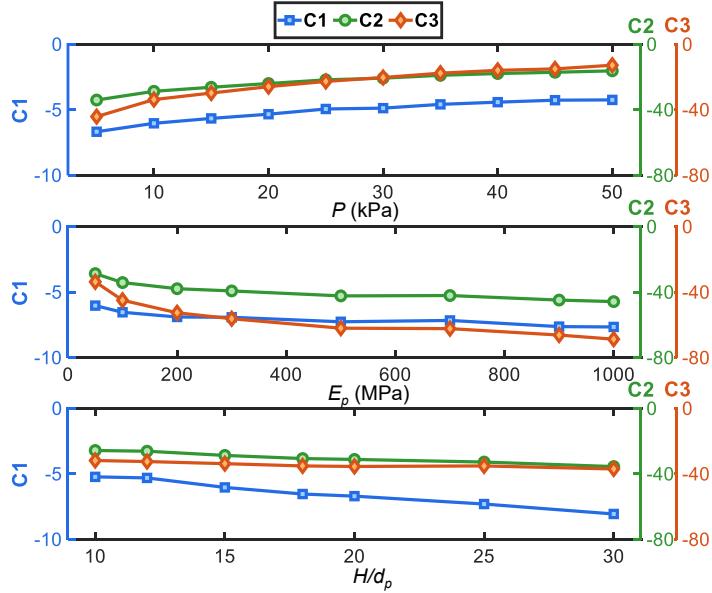


Fig. 20 Parameters with significant influence on equation coefficients: P , E_p , H/d_p .

3.4 Relationships between equation coefficients and relaxation time

As demonstrated in the main text, C_1 , C_2 , and C_3 consistently exhibit a high degree of pairwise correlation regardless of the values assigned to the system variables. In fact, the

543 magnitudes of these three coefficients are all related to the relaxation properties of the sys-
 544 tem. Specifically, each of C_1 , C_2 , and C_3 maintains a one-to-one correspondence with the
 545 characteristic relaxation time as we have defined it (see Fig. 21).

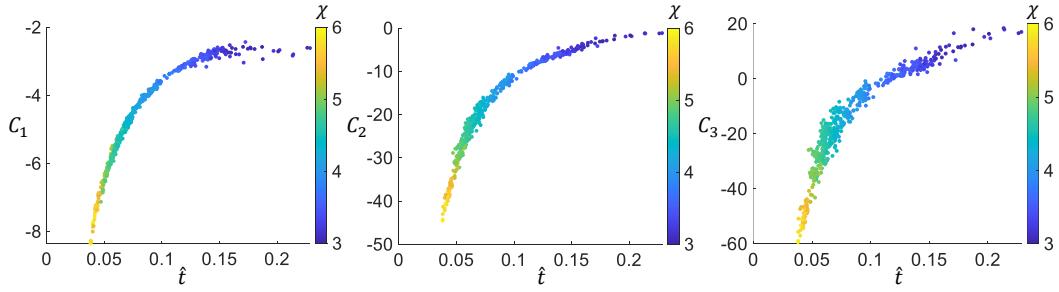


Fig. 21 The one-to-one correspondence between C_1 , C_2 , C_3 and the characteristic relaxation time.

546 3.5 Interpretations from critical state soil mechanics

547 In this section, the results are interpreted within the framework of critical state soil mechan-
 548 ics (CSM). Consistent with the scope of our quasi-static study, void ratio emerges as one of
 549 the core factors describing the state of soil particles and defining the critical state in CSM.

550 The relationships between the solid fraction (ϕ) and key system variables are illustrated in
 551 Fig. 22. As is evident from this figure, the solid fraction (ϕ) exhibits sensitivity to variations in
 552 E_p/P and H/d_p . This observation corroborates the validity of our findings, as the dependence
 553 of the void ratio (whose role is analogous to that of ϕ) on the constitutive models of granular
 554 systems under quasi-static conditions has been well established in the existing literature.

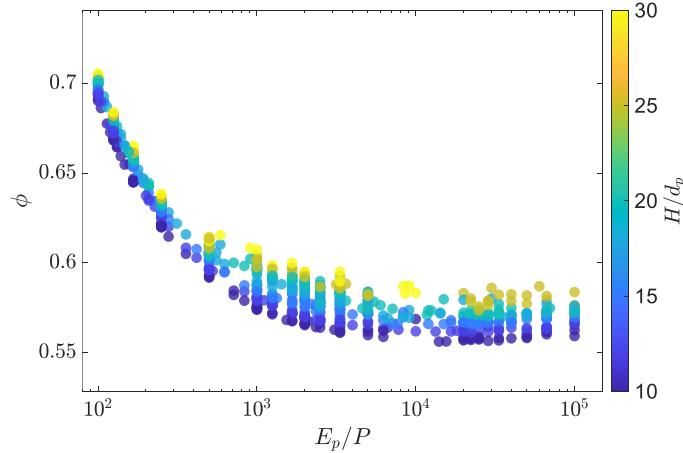


Fig. 22 Relationships between solid fraction (ϕ) and system variables: E_p/P and H/d_p .

555 4 Reference

556 References

- 557 [1] Chen, Z., Liu, Y. & Sun, H. Physics-informed learning of governing equations from scarce
 558 data. *Nat. Commun.* **12** (2021).
- 559 [2] Xie, X., Samaei, A., Guo, J., Liu, W. K. & Gan, Z. Data-driven discovery of dimensionless
 560 numbers and governing laws from scarce measurements. *Nat. Commun.* **13** (2022).

- ⁵⁶¹ [3] Buckingham, E. On physically similar systems; illustrations of the use of dimensional
⁵⁶² equations. *Phys. Rev.* **4**, 345–376 (1914).
- ⁵⁶³ [4] Sun, J. & Sundaresan, S. A constitutive model with microstructure evolution for flow of
⁵⁶⁴ rate-independent granular materials. *J. Fluid Mech.* **682**, 590–616 (2011).