UNSUPERVISED LEARNING BASED INTERACTION FORCE MODEL FOR NONSPHERICAL PARTICLES IN INCOMPRESSIBLE FLOWS

Report Type: Final Technical Report

Reporting Period: 7/1/2020 – 6/30/2023

Principal Investigator: Liang-Shih Fan

Email: **fan.1@osu.edu**

Phone: 614-688-3262

Report Issue Date: 9/30/2023

Award Number: DE-FE0031905

Submitting Organization: Department of Chemical and Biomolecular Engineering

The Ohio State University

151 West Woodruff Avenue

Columbus, OH 43210

Submitted by: Andrew Tong

Email: tong.48@osu.edu

Phone: 614-688-3262

DUNS Number: 832127323

Project Manager: **Dunst Richard**

Project Period: 7/1/2020 – 6/30/2023

Signature: _____ Date: 9/30/2023

Liang-Shih Fan

Table of Contents

Disclaimer:

Summary:

This project provides a neural network-based interaction force model for gas-solid flows from low to intermediate Reynolds numbers and concentration, which can be linked to MFiX-DEM. We have constructed a database of the interaction force between the irregular-shaped particles using a spherical harmonic method and the fluid phase based on the particle-resolved direct numerical simulation (PR-DNS) with immersed boundary-based gas kinetic scheme. Unsupervised learning method, i.e., variational auto-encoder (VAE) has been applied to extract the primitive shape factors determining the drag force, lifting forces, and torque. The interaction force model has been trained and validated with a simple but effective multi-layer feed-forward neural network: multi-layer perceptron (MLP), which will be concatenated after the encoder of the previously trained VAE for geometry feature extraction for single, irregular particles. We have trained transpose convolutional neural networks with the PR-DNS data to predict the velocity and pressure gradient of the single particle systems and utilized them to calculate drag force of multi-particle systems. This model can provide high computational efficiency because it does not require collecting multiparticle system data from PR-DNS.

Project Schedule

	Tasks	Start Data	End Data	Year1				Year2				Year3			
				Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
1	**Project Management and Planning**	8/1/2020	8/1/2023												
2	**Developing Code for Non-spherical Particles.**	8/1/2020	2/1/2021												
2.1	*Coupling the non-spherical particle-particle collision module.*	8/1/2020	11/1/2020												
2.2	*Efficiency improvement of the code.*	11/1/2020	2/1/2021												
	Milestone A. Code acceleration and verification for non-spherical particles completed		2/1/2021												
	Decision Point 1: Validation of highly efficient simulation for gas-particle systems with non-spherical particles		2/1/2021												
3	**Geometric Database for Non-spherical Particles.**	8/1/2020	8/1/2021												
3.1	*Collecting 3-D Shapes of Particles*	8/1/2020	2/1/2021												
3.2	*Generation of Diverse Non-spherical Shapes.*	2/1/2020	8/1/2021												
3.3	*Generation of Lagrangian Markers.*	2/1/2020	8/1/2021												
	Milestone B. Generative model for Non-spherical particle shape libraries completed		8/1/2021												
4	**Data collection by PR-DNS Method.**	2/1/2021	2/1/2022												
4.1	*Simulation in Low-Reynolds Numbers.*	2/1/2021	2/1/2022												
	Milestone C. Database for Gas-solid flows of non-spherical particles in low Reynolds numbers constructed		2/1/2022												
4.2	*Simulation in Moderate-Reynolds Numbers.*	2/1/2022	2/1/2023												
	MileStone E. Database for Gas-solid flows of non-spherical particles in moderate Reynolds numbers constructed		5/1/2023												
	Decision Point 3. Validation of MLP Architecture for interaction for interaction force model in moderate Reynolds number		5/1/2023												
5	**Extracting Geometrical Features.**	2/1/2022	8/1/2022												
5.1	*Construction of the VAE architecture.*	2/1/2022	5/1/2022												
5.2	*Training, validation and Optimization of the VAE model.*	5/1/2022	8/1/2022												
6	**Training of MLP-based Regressor and Final Reporting.**	2/1/2022	8/1/2023												
6.1	*Construction of the MLP architecture.*	2/1/2022	2/1/2023												
	Milestone D: Interaction force model for non-spherical particles in low Reynolds number completed		8/1/2022												
	Decision Point 2. Validation of MLP Architecture for interaction force model in low Reynolds number		8/1/2022												
6.2	*Training, Validation and Optimization of MLP.*	2/1/2023	8/1/2023												
6.3	*Linking of the model to MFiX and Final Reporting.*	5/1/2023	8/1/2023												
	Milestone F: Training and verification of interaction force model and Likning to MFiX		8/1/2023												

Task 2 Developing Code for Non-spherical Particles.

Subtask 2.1. Coupling the non-spherical particle-particle collision module.

The module to simulate the non-spherical particles is now integrated into the developed code. The collision module of non-spherical particles will be implemented in the further development of the project since we found this is not an essential part of the machine learning training process. Instead of simulating a fully fluidized bed with non-spherical particles to generate the required data for the drag and lifting force, we proposed to separate the model development into two independent steps. Firstly, the fluid flow prediction and interaction force prediction model for a single non-spherical particle in an infinite domain will be developed based on the un-supervised auto-encoder based machine learning method. Secondly, the force model which could consider the neighboring effect and Stokes number effect will be developed based on superposition, e.g., the linear Pair-wise interaction force model. Through such two progressive steps, the training process and data generation are greatly simplified, and we can be firstly focused on the force model for single non-spherical particles.

Subtask 2.2. Efficiency improvement of the code.

The original in-house IB-LBM code can only handle the 3-D structured grids. However, for a single-particle simulation, gradients of the flow field mainly concentrate around the near boundaries. Adaptive mesh refinement technique can greatly reduce the computational grid number and improve the efficiency of the simulation code. Thus, an adaptive mesh refinement (AMR) based immersed boundary (IB) framework was first developed. Considering the robustness, instead of the lattice-Boltzmann method, a finite volume method with gas-kinetic flux solver is coupled into the AMR framework. The efficiency of the developed method is evaluated through benchmark tests.

2.2.1 Theory of the Gas-kinetic scheme.

Boltzmann equation for the molecules is the starting point of the incompressible solver for our project. The Bhatnagar–Gross–Krook (BGK) collision model is the most widely used one described as follows:

$$\frac{\partial f}{\partial t} + \vec{\xi} \cdot \nabla f = -\frac{f-g}{\tau} \tag{1}$$

where f is the gas distribution function and g is the equilibrium state approached by f through particle collisions within a collision time scale τ. $\vec{\xi}$ is the particle velocity in the phase space. The mass and momentum conservation law could be obtained by multiplying Eq. (2-1) with $\phi = \{1, \xi_1, \xi_2, \xi_3\}^T$ on both sides and integrating over the velocity space, i.e.,

$$\int \left(\frac{\partial f}{\partial t}\right) \mathrm{d}\vec{\xi} + \nabla \cdot \int \left(\vec{\xi} f\right) \mathrm{d}\vec{\xi} = 0 \Longrightarrow \frac{\partial U}{\partial t} + \nabla \cdot \left(\vec{F}\right) = 0 \tag{2}$$

where $U = \{\rho, \rho u_1, \rho u_2, \rho u_3\}$, and $\vec{F}$ is the corresponding flux $\vec{F} = \int \vec{\xi} f \mathrm{d}\vec{\xi}$.

Different from the LBM method, in the gas-kinetic scheme, macroscopic variables *U* are stored and evolved along time. The flux of computational grids can be obtained by firstly reconstructing the distribution function through Champman-Enskog expansion near cell interfaces and then integrating over the velocity space.

2.2.2 Gas kinetic scheme based on continuous or discrete velocity space.

Instead of considering the Maxwellian distribution function for the gas phase, in this work, a simplified sphere function-based model which is continuous in velocity space and a D3Q15 lattice model which is discrete in velocity space are considered to construct the flux solver of the incompressible flow.

Sphere function-based gas kinetic scheme

Firstly, a simplified sphere function-based model proposed by Yang [Phys. Fluids 29, 083605 (2017)] is considered. The equilibrium distribution function can be expressed by

$$g = \begin{cases} \frac{\rho}{4\pi} & if \ \left|\vec{\xi} - \vec{u}\right| = c \\ 0 & elsewise \end{cases}, \tag{3}$$

and the integration over velocity space can be simplified by integration over the surface of the space, i.e.,

$$\int (\cdot) \mathrm{d}\vec{\xi} = \int_0^{2\pi} \int_0^{\pi} \sin(\psi) \, (\cdot) d\psi d\theta,$$

with $\{\xi_1, \xi_2, \xi_3\}$ redefined by

$$\{\xi_1, \xi_2, \xi_3\} = \{u_1 + c\sin\psi\cos\theta\,, u_2 + c\sin\psi\sin\theta\,, u_3 + c\cos\psi\}. \quad (4)$$

In addition, through Chapman-Enskog analysis, the relationship between the kinematic viscosity and collision time scale can be expressed as:

$$\tau = \frac{3\,\nu}{c^2}.$$

Given the distribution of macroscopic variables at t^n, the initial equilibrium gas distribution function can be expressed as:

$$g(\vec{\xi}) = \begin{cases} g^L & if\ \left|\vec{\xi} - \vec{u}\right| = c\ and\ x < 0 \\ g^R & if\ \left|\vec{\xi} - \vec{u}\right| = c\ and\ x > 0 \end{cases},$$

assuming that the cell interface is defined at x = 0.

At time $t^n + \delta t$, gas distribution function at cell interface can be expressed by

$$f\left(\vec{\xi}, x = 0, t^n + \delta t\right) = \left(g - \tau\left(\frac{\partial g}{\partial t} + \vec{\xi}\cdot\nabla g\right)\right)\Big|_{\left(\vec{\xi}, x=0, t^n+\delta t\right)},$$

which can be further approximated by:

$$f\left(\vec{\xi}, x = 0, t^n + \delta t\right) = g\left(\vec{\xi}, 0, t^n + \delta t\right) - \frac{\tau}{\delta t}\left(g\left(\vec{\xi}, -\vec{\xi}\delta t, t^n\right) - g\left(\vec{\xi}, 0, t^n + \delta t\right)\right). \quad (5)$$

Eq. (5) can be further simplified into Eq. (6) with the simplified sphere function-based distribution function, considering that the macroscopic velocities $u \ll c$ in incompressible regime,

$$\begin{aligned} f(\psi, \theta, x = 0, t^n + \delta t) &= g(\psi, \theta, 0, t^n + \delta t) \\ -\frac{\tau}{\delta t}&\left(g\left(\psi, \theta, -\vec{\xi}\delta t, t^n\right) - g(\psi, \theta, 0, t^n + \delta t)\right). \end{aligned} \quad (6)$$

Further, in Eq. (6), $g(\psi, \theta, 0, t^n + \delta t)$ can be obtained through the compatibility condition, i.e.,

$$\begin{aligned} \int g(\psi, \theta, 0, t^n + \delta t)\sin(\psi)\,d\psi d\theta & \\ = \int_{\vec{\xi}\cdot n>0} & g^L\left(\psi, \theta, -\vec{\xi}\delta t, t^n\right)\sin(\psi)\,d\psi d\theta \\ + \int_{\vec{\xi}\cdot n<0} & g^R\left(\psi, \theta, -\vec{\xi}\delta t, t^n\right)\sin(\psi)\,d\psi d\theta. \end{aligned}$$

Then the flux at cell interface $\vec{F} = \int \vec{\xi} f \mathrm{d}\vec{\xi} = \int_0^{2\pi} \int_0^{\pi} \sin(\psi) \left(\vec{\xi} f\right) d\psi d\theta$ can be obtained eventually. The detailed form of $\vec{F}$ can be found in the reference work by Yang [Phys. Fluids 29, 083605 (2017)] and the parameter $\delta t = 0.4 \frac{\Delta x}{\max(u_1, u_2, u_3) + c}$.

Discrete velocity-based gas-kinetic scheme.

The simplification from Eq. (5) to Eq. (6) means that, at low Mach number region, the following terms:

$$\{u_1^I + c \sin\psi \cos\theta, u_2^I + c \sin\psi \sin\theta, u_3^I + c \cos\psi\}$$

$$\{u_1^{II} + c \sin\psi \cos\theta, u_2^{II} + c \sin\psi \sin\theta, u_3^{II} + c \cos\psi\}$$

are treated as the same in the velocity space where the superscript I and II denotes macroscopic variables at different spatial locations even though the direction of them are slightly different. As will be shown later in the benchmark test, such simplification leads to a smaller stable CFL number which is only 0.5 for a single step scheme. As an alternative approach, the discrete velocity-based scheme, i.e., the lattice Boltzmann flux solver, avoids such an incompatibility and greatly improves the upper limit of the stable CFL number.

The velocity space is first discretized by the Taylor expansion near zero-velocity. In this project, the D3Q15 model is adopted. The discrete velocities can be expressed as:

$$\vec{e}_\alpha = \begin{cases} 0 & \alpha = 0 \\ (\pm 1,0,0)c, (0,\pm 1,0)c, (0,0,\pm 1)c, & \alpha = 1, \cdots 6, \\ (\pm 1, \pm 1, \pm 1)c & \alpha = 7,14 \end{cases} \tag{7}$$

and the equilibrium density distribution is

$$g_i = \rho \omega_i \left(1 + \frac{\vec{e}_i \vec{u}}{c_s^2} + \frac{1}{2}\left(\frac{\vec{e}_i \vec{u}}{c_s^2}\right)^2 - \frac{\vec{u} \cdot \vec{u}}{2c_s^2}\right), \tag{8}$$

where the weights are expressed as $\omega_0 = \frac{2}{9}, \omega_{1-6} = \frac{1}{9}, \omega_{7-14} = \frac{1}{72}$ and $c_s = \frac{c}{\sqrt{3}}$.

Then the integration over the velocity space can be replaced by the summation over discrete velocities, e.g.,

$$\int f \mathrm{d}\vec{\xi} \doteq \sum_{i=0}^{14} f_i \omega_i$$
$$\int f \vec{\xi} \mathrm{d}\vec{\xi} \doteq \sum_{i=0}^{14} f_i \vec{e}_i \quad .$$
$$\int f \vec{\xi} \vec{\xi} \mathrm{d}\vec{\xi} \doteq \sum_{i=0}^{14} f_i \vec{e}_i \vec{e}_i$$

The construction of the flux is the same as in Eq. (5), i.e.,

$$f_i(0, t^n + \delta t) = g_i(0, t^n + \delta t) - \frac{\tau}{\delta t}\big(g_i(-\vec{e}_i \delta t, t^n) - g(0, t^n + \delta t)\big). \tag{9}$$

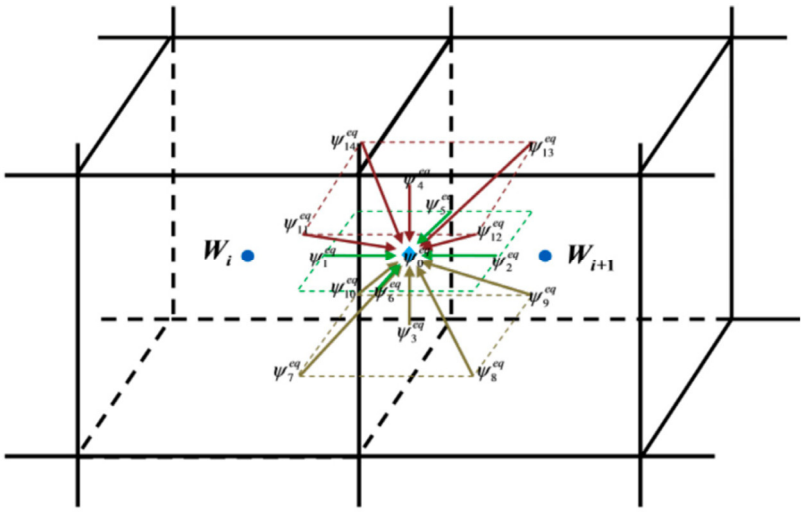


Figure 1. Illustration of the D3Q15 discrete velocity model.

At time t^n, the distribution of the macroscopic variables can be firstly obtained through reconstruction. $g_i(-\vec{e}_i \delta t, t^n)$ is calculated according to the direction of e_i, i.e.,

$$\hat{g}_\alpha(-\vec{e}_\alpha \delta t, t^n) = \begin{cases} g_\alpha^L(-\vec{e}_\alpha \delta t, t^n) & if\ e_{\alpha,1} > 0 \\ g_\alpha^R(-\vec{e}_\alpha \delta t, t^n), & if\ e_{\alpha,1} < 0 \\ \frac{1}{2}(g_\alpha^L + g_\alpha^R)(0, t^n) & if\ e_{\alpha,1} = 0 \end{cases}. \tag{10}$$

At time $t^n + \delta t$, the conserved variables at the cell interface are obtained through compatibility equations, i.e.,

$$U = \sum_{i=0}^{14} \hat{g}_i(-\vec{e}_\alpha \delta t, t^n)\phi\,, \tag{11}$$

with $U = \{\rho, \rho u_1, \rho u_2, \rho u_3\}$ and $\phi = \{1, \xi_1, \xi_2, \xi_3\}^T$.

Then $f_i(0, t^n + \delta t)$ can be obtained through Eq. (9), and the flux can be further derived. To have a second order of accuracy in a single time step, when then time is Δt, $\delta t = \frac{\Delta t}{2}$ in Eq. (9).

Benchmark tests of Taylor-Green vortex.

The efficiency and stability of the two schemes are studied and compared through a *Re* = 300 Taylor-Green vortex tests. The computational domain is a 3-D periodic domain with length *L*. The initial condition is:

$$u_1(\vec{x}, t = 0) = u_0 \cos\left(\frac{2\pi x}{L}\right) \sin\left(\frac{2\pi y}{L}\right) \sin\left(\frac{2\pi z}{L}\right),$$
$$u_2(\vec{x}, t = 0) = -u_0 \sin\left(\frac{2\pi x}{L}\right) \cos\left(\frac{2\pi y}{L}\right) \sin\left(\frac{2\pi z}{L}\right),$$
$$u_3(\vec{x}, t = 0) = 0.$$

Reynolds number is defined as

$$Re = \frac{u_0 L}{2\pi\nu}.$$

Defining the reference time as

$$\hat{t} = \frac{L}{2\pi u_0},$$

the evolution of the dimensionless kinetic energy $k^* = \frac{k}{k_0}$ along dimensionless time $t^* = \frac{t}{\hat{t}}$ is shown in Fig. 2. The maximum stable CFL number for both schemes and the computational time consumptions for a single step are all listed in Tab. 1 and 2. As shown, the dissipation of the continuous velocity-based flux is smaller than the discrete velocity-based scheme. However, for the discrete velocity-based gas-kinetic scheme, a stable single step scheme can be obtained even with CFL number to be 1. On the other hand, the continuous velocity-based gas-kinetic scheme,

i.e., the scheme based on the spherical distribution function, a stable single step scheme can be obtained only with a maximum CFL number to be 0.5. The computational consumptions in a single step for the continuous velocity-based flux solver is smaller than the discrete velocity-based solver because for the continuous velocity-based scheme, we do not need to calculate the discrete velocity distribution one by one. However, if a larger CFL number, i.e., CFL >= 0.73 is adopted, the efficiency of the discrete velocity-based scheme is higher than the continuous model. Thus, to improve the efficiency for our later simulation, the discrete velocity-based gas kinetic scheme is chosen in this project.

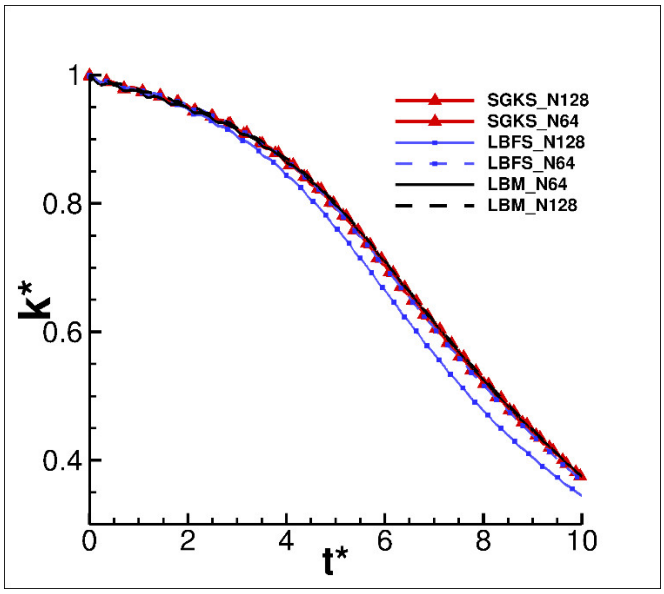


Figure 2. Comparison of the evolution curves between the spherical function-based gas kinetic scheme (SGKS) and the discrete velocity-based gas kinetic scheme (LBFS) under CFL = 0.5.

Table 1. Comparison of the simulation time [s] between different schemes for a single step.

	LBM	SGKS	LBFS	CLBFS
N=64	0.095	0.095	0.134	0.147
N=128	0.696	0.706	1.03	1.1
N=203	2.67	2.72	4.07	4.29

Table2. Maximum stable CFL number for different scheme.

	SGKS	LBFS	CLBFS
CFL_{max}	0.5	1.0	0.5

2.2.3 Implementation of Adaptive mesh refinement (AMR).

Several AMR techniques exist in the literature, e.g., the patch-based AMR technique, the cell-based AMR technique. in this project, the fine-grained cell-based AMR library, i.e., the p4est is used for the underline grid management. The p4est is based on octree data structure, where each node can be divided into eight children and the cells are encoded in an unstructured way. The AMR process of p4est can be separated into four steps. Firstly, a refinement indicator is calculated in each cell. Secondly, based on the indicator, the cells will be merged together if all children are tagged as to be coarsened. Thirdly, if a cell is tagged to be refined, it will be divided into eight children's cells and the physical variables will be mapped into these children cells. Fourthly, a balance procedure will be done to ensure the maximum 2:1 level ratio between neighboring cells.

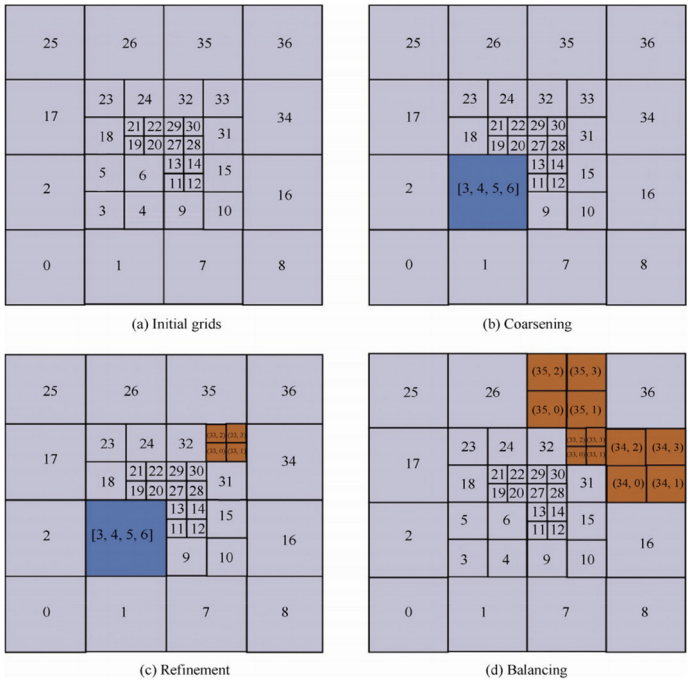


Figure 3. Illustration of the AMR process.

2.2.4 Reconstruction matrix free method coupling with the AMR technique.

When on the structured Cartesian grids, the gradient reconstruction is quite simple, like,

$$W_x = (W_{i+1} - W_{i-1})/(2\Delta x). \quad (12)$$

However, when coupling with adapted unstructured grids like in Fig. 3, special care should be taken for the reconstruction procedure. Different strategies exist for the reconstruction procedure. Take Cell 13 in Fig. 3(d) as an example.

In the first approach, virtual children's cells are generated in Cell 6 firstly. After that, the reconstruction in Cell 13 can be implemented like in the structured grids. However, this requires generation and management of virtual grids between different layers, which could be quite cumbersome when coding. In the second approach, instead of Eq. (12), a least squares problem is solved for each cell, i.e.,

$$\mathrm{a} = \mathrm{argmin}(Ma - b)^2,$$

where $a = \{W_x^i, W_y^i, W_z^i\}^T$ are gradients to be determined for Cell *i*; M is a matrix of dimension $n \times 3$ and $M_{j1} = \frac{1}{V_j}\int_{cv_j}(x - x_i)dV_j$, $M_{j1} = \frac{1}{V_j}\int_{cv_j}(y - y_i)dV_j$ and $M_{j1} = \frac{1}{V_j}\int_{cv_j}(z - z_i)dV_j$, $j = 1,2,\cdots n$ and n is the number of face neighboring cells for Cell *i*; b is a vector of dimension $n \times 1$ and $b_j = W_i - W_j$. However, for this approach, the least squares reconstruction matrix, and a list of reconstruction stencils should be managed, which could also be very memory consuming and cumbersome in coding especially when the mesh is dynamically refined and coarsened.

Thus, in this project, a novel finite volume method is proposed which can be applied in the cartesian based AMR framework while eliminating the management of virtual grids or least-squares reconstruction matrices. The method is based on the second order time stepping properties of the discrete velocity-based gas-kinetic scheme.

In addition to the cell averaged physical variables defined for each cell like in the traditional FV method, 6 point-values defined at the geometrical center of each face are introduced for each face.

Take the 2-D cases as an example in Fig. 4, 0 is the cell center; 1-4 are points where the point values are defined. Meanwhile, 1-4 are also the flux points for the second-order scheme.

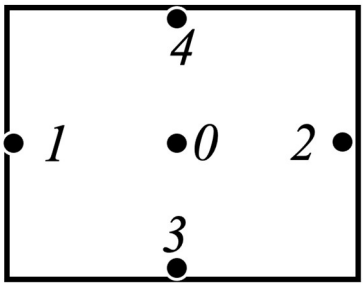


Figure 4. Illustration of the reconstruction matrix free FV method.

The updating steps are listed as:

1. Obtaining the cell averaged gradients using Gaussian theorem, i.e.,

$$\int W_x^c dV = \oint_{\partial V} W n_x dS \,, \int W_y^c dV = \oint_{\partial V} W n_y dS \,, \int W_z^c dV = \oint_{\partial V} W n_z dS.$$

2. For the flux calculation of the left edge in Fig. 4, the conserved variables defined at point 1 are directly used. The gradients are calculated by

$$\begin{aligned} W_x &= \frac{2(W_0 - W_1)}{\Delta x}, \\ W_y &= W_y^c, \\ W_z &= W_z^c. \end{aligned}$$

For the flux calculation of the bottom edge, the conserved variables defined at point 3 are directly used. The gradients are calculated by

$$\begin{aligned} W_x &= W_x^c, \\ W_y &= \frac{2(W_0 - W_3)}{\Delta y}, \\ W_z &= W_z^c. \end{aligned}$$

The same approach is applied for the right and upper edges.

3. Updating the cell averaged value by

$$W_0^{n+1} = W_0^n - \frac{\Delta t}{\Delta x}\left(\mathrm{F}_{\mathrm{i}+\frac{1}{2},j} - \mathrm{F}_{\mathrm{i}-\frac{1}{2},j} + \mathrm{F}_{\mathrm{i,j}+\frac{1}{2}} - \mathrm{F}_{\mathrm{i,j}-\frac{1}{2}}\right).$$

$$W_k^{n+1} = 2W_k^n - W_k^*, k = 1,2,3,4.$$

where W_k^* are the conserved values obtained with Eq. (11) by setting $\delta t = \frac{\Delta t}{2}$, which is second order of accuracy in time.

Such a scheme is stable under the maximum CFL number of 0.5, which corresponds to CFL = 1 considering the sub-cell resolution, i.e., the point values at each cell interface when combined with the discrete velocity-based gas-kinetic scheme.

The benchmark result of Taylor-Green vortex is shown in Fig. 5. and the computational time are also listed in Table. 1 in the last column. When compared with the traditional FV scheme, the proposed scheme is a little more dissipative with the same number of cell numbers while maintaining the same computational time for a single step.

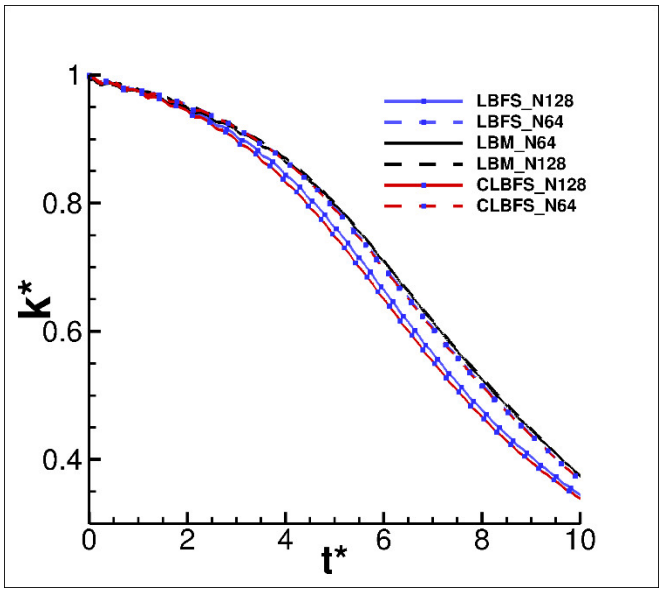


Figure 5. Comparison of the Taylor-Green Vortex result between the LBFS of traditional finite volume method and the proposed scheme, i.e., compact LBFS (CLBFS).

2.2.5 Coupling the immersed boundary method into the AMR framework.

The direct forcing based immersed boundary method is coupled with the proposed FV scheme and the AMR framework. To keep the simplicity of the scheme, a finest grid level is ensured around the Lagrangian marker.

In the proposed scheme, the inner degrees of freedom can be utilized in the immersed boundary method.

Firstly, for cells that will be used as the stencil of surface Lagrangian points, physical variables are firstly interpolated to eight children's cells, which is shown as the cells separated by dashed lines in Fig. 6. Then interpolate back and forth the physical variables from the children's cells to the Lagrangian points and correct the velocities in the children's cells. Eventually, interpolated back the cell averaged cells and face variables from the eight children's cells. The directly forcing scheme is the same as the work proposed by Uhlmann [J. Comput. Phys. 209, 448 (2005)] and will not be further reviewed in this report.

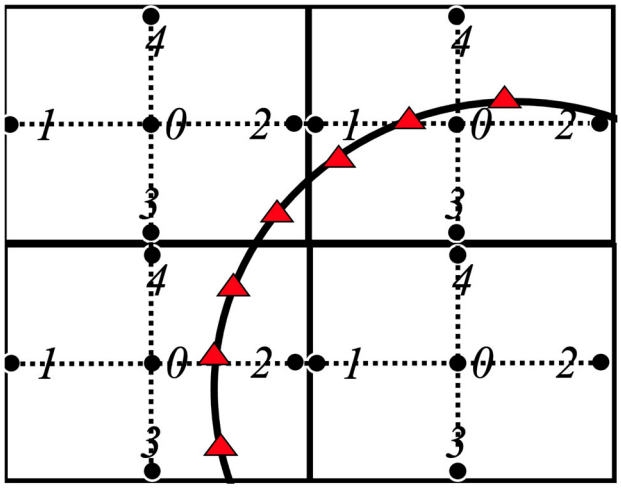


Figure 6. Illustration of the proposed immersed boundary method.

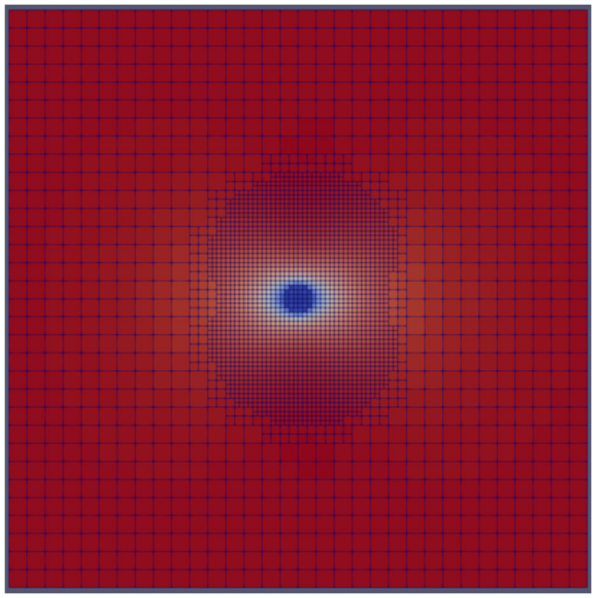


Figure 7. Illustration of the AMR result. Contours are the velocity magnitude.

Fig. 7 shows the result of flows around a spherical particle when the Reynolds number is 0.2. The developed AMR framework can capture the gradients of the flow around particles. The ratio between the cell number with and without AMR technique is approximately 1:22.4, which means that the computational efficiency is greatly improved through the AMR technique.

2.2.6 Efficiency Verification

Table 3 lists the computational consumptions for a single particle with computational domain being 20 diameters with the AMR technique based on the SGKS method. The non-spherical particle with the spherical descriptor, 0.5, is used. The *Re* is 10 and the CFL number is 0.2. The total time steps used here is 5,000. The required time steps should be large enough to achieve the steady state. The steps depend on the particle resolution and the Reynolds number.

Table 3. Computation time for the different particle resolutions.

Particle Resolutions (*D/dx*)	Computation time (min/CPU)
6	55.7
9	85.6
12	131.3

Task 3 Geometric Database for Non-spherical particles

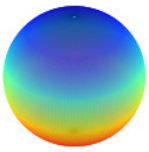
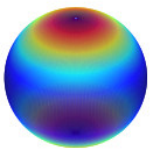
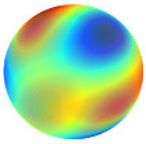
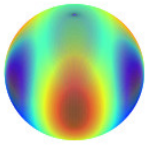
In this Task, collecting 3-D shapes of non-spherical particles was finished. The spherical harmonic method was used to represent the irregular surface shape of the particles.

Subtask 3.1 – Collecting 3-D Shapes of Particles

Non-spherical particles to be used in the neural network model and the numerical simulation are created using the spherical harmonic method. A variety of non-spherical particles can be generated by using the super-ellipsoid equation, but it is difficult to describe the non-smooth surface shape of particles. However, by using the spherical harmonic method, the shape, roundness, and compactness of the particles can be described in various ways with few parameters.

Spherical harmonic functions(Y) are composed of functions that are orthogonal to each other on a spherical surface, and the real values of some functions are shown in the table below. In the table below, the jet scale represents the real value of the Y. l and m denote the degree and order of the spherical harmonic function (Y). As the l value increases, a more localized surface shape is expressed. In this task, the maximum value (l_{max}) of l is set to 8.

Table 4. Real value distribution for some spherical harmonic functions.

l/*m*	0/0	1/0	2/0	3/1	3/3
Real Y values on the sphere surface					

As can be seen from the table above, each of the Y values has a distribution that is orthogonal to each other on a spherical surface, and various non-spherical surfaces can be described by adding them with appropriate weights. Here, we used three parameters, elongation index (EI), flatness index (FI) and spherical descriptor (d). The d value ranges from 0 to 1, and the closer it is to 1, the higher the degree is weighted, which results in a rougher surface particle. The basic coordinates to

be converted by the spherical harmonic function are obtained by dividing the faces based on an icosahedron. The more the faces are divided, the more points are obtained, and it will be checked in the future by simulation how many points are needed.

Even if the same d value is used, particles of the exact same shape cannot be obtained because random numbers are used when calculating the spherical harmonic coefficients. The volume of the particle, the center and area of each face are calculated using the "Trimesh" library. The center points are the Lagrangian points, and the face area will be considered for IBM force density calculation. In addition, voxel data is obtained, and it is used for getting the inertia and primary axes of the particle.

Subtask 3.2 – Generation of Diverse Non-spherical Shapes

In this Subtask, d values (0.0 ~ 0.5), *EI*s (0.5 ~ 1.0) and aspect ratios (1.0 ~2.0) of non-spherical particles were randomly chosen, and *FI*s were calculated from *EI*s and aspect ratios. In addition, the rotation matrix was used to orientate the particle and the Euler angles, α, β and γ, were randomly chosen. The orientation can be included in the latent space by using the rotation matrix. This strategy is beneficial in terms of reducing the number of DNS datasets because the rotational angles do not need to be included in the DNS sensitivity analysis.

If the particle rotation is included in the latent space, a new latent vector needs to be updated at every time step in CFD-DEM application. If the surface coordinates are used for the particles, it is required to convert them into 48^3 voxels. By testing through i7-7700 3.6 GHz CPU, it took about 10 to 30 seconds depending on the *EI* and *FI* indexes used. This can be heavy for simulation with many particles. Therefore, it would be better to use voxels instead of the surface coordinates for the particles in the DEM calculation.

The VAE model developed was used to get the latent vectors to represent the shapes of the particle. By using 1,200 datasets for the training and 400 validation datasets, the reconstruction error has been reduced to 0.9%.

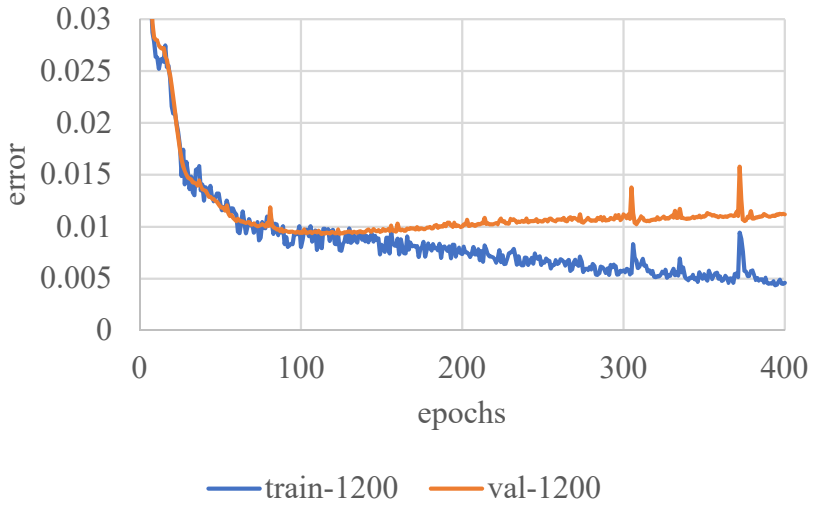


Figure 8. Reconstruction error for the decoded particles of train and validation data.

Fig. 9 shows the validation error according to the number of training datasets. The error converged when the number of datasets is in between 1,200 and 2,000.

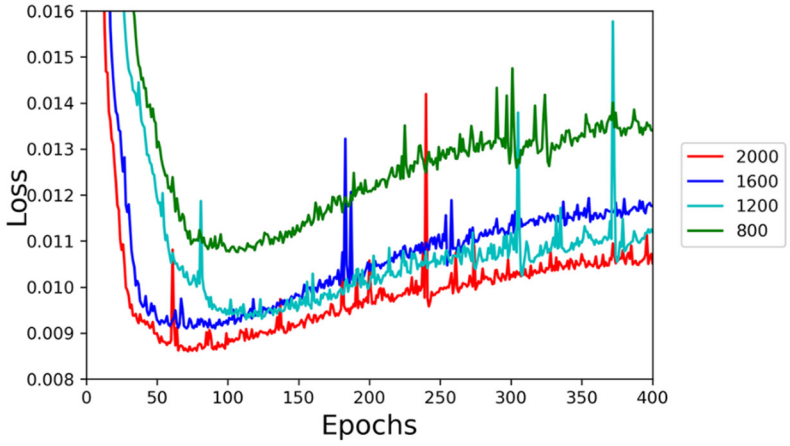


Figure 9. Validation error according to the number of datasets.

By using the trained VAE model from the 1,200 training datasets, the particles from the validation datasets can be reproduced as follows. The VAE model represents orientation, elongation, and flatness well. However, the roughness tended to be smoothened.

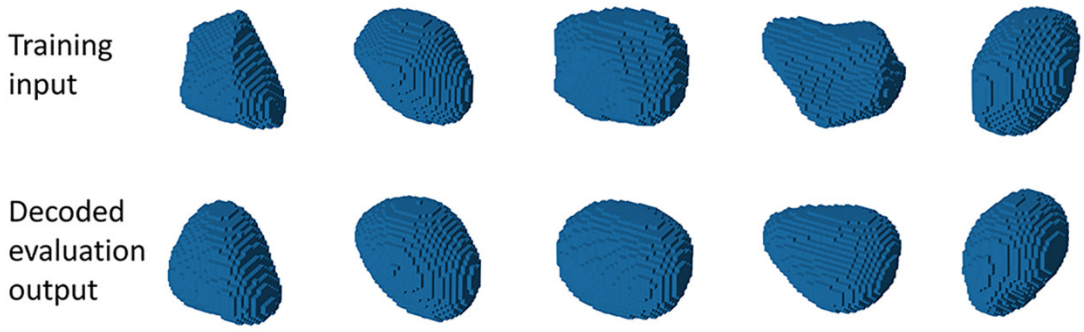


Figure 10. Input and output voxel data of validation datasets.

Subtask 3.3 – Generation of Lagrangian Markers

Lagrangian points with the tri-faces areas and volume were generated in the configuration file form for the DNS calculation. We collected the database for the 2,000 particles based on the method explained in the previous section. 1,200 particles were sufficient to train the VAE model, but we generated more particles to assign different Re numbers to the different shapes.

Task 4 Data collection by PR-DNS Method

Subtask 4.1 – Simulation in Low-Reynolds Numbers

The AMR method was used to reduce the computation cost with the IGKS method. The calculation speed is faster by using GPU, but it is more economical not to use GPU because of the much higher cost of the OSC GPU node. Furthermore, the LBFS was stable for Taylor-Green vortex experiment, but it turned out unstable for low Re flows. It will be studied further to be optimized in the future, but for the low Re flow, the IGKS was used because it is sufficiently fast to collect the. In the later

part of this Subtask, convergences according to the grid resolution were studied for a spherical particle.

4.1.1. AMR method and computation conditions

To use the AMR method, the following equations were used to quantify the degree of refinement.

$$A_{ij} = \frac{1}{2}\left(\frac{du_i}{dx_j} + \frac{du_j}{dx_i}\right), B_{ij} = \frac{1}{2}\left(\frac{du_i}{dx_j} - \frac{du_j}{dx_i}\right), \qquad \omega = \frac{\sum_i \sum_j {B_{ij}}^2}{\sum_i \sum_j {A_{ij}}^2 + \sum_i \sum_j {B_{ij}}^2}$$

ω value in the above equation decreases to zero as the flow field becomes uniform. The mesh will move to the higher level and be divided into 8 cells when the cell has higher ω than the threshold. We set the threshold as 0.04 and the highest and lowest levels are 4 and 0.

The length of the entire domain is 20D(diameter) and the far field boundary condition with the dimensionless velocity, 1, in x direction is used. The Re numbers ranged between 0.1 and 10 by changing viscosity. The CFL number and Mach number are 0.1 and 0.15, respectively. 1280 Lagrangian points are used with the corresponding face areas and the retraction lengths is 0.1665 to consider the length of the sub-cells.

The following modified convergence limit is used to determine the ending point of the calculation.

$$residual = \frac{\sum_{cells} \sum_{subcells} \sum_{xyz} (u_{n+1} - u_n)^2}{\sum_{cells} \sum_{subcells} \sum_{xyz} (u_{n+1})^2} \frac{\delta t_{ref}}{\delta t}$$

u is the macroscopic velocity and t in the equation is dimensionless time scaled by the reference length over the reference velocity. We decided to use t to consider the time span difference according to the Re numbers and the resolutions. The simulation was conducted without the AMR until the residual approached 2×10^{-6} and used AMR until to the residual, 1×10^{-6}.

4.1.2. Convergence Studies for spherical particles

Before collecting the flow field database, we needed to confirm the particle resolution. Therefore, drag coefficients for Re =1 and 10 were calculated according to the various grid resolutions (*D/dx* = 3 ~ 20). The drag coefficients (C_D) were obtained through the equation below. $F_{l,x}$ and ΔV_l indicate the force in x direction and volume of the Lagrangian point, respectively.

$$C_D = \frac{-\sum_l F_{l,x}\Delta V_l}{0.5\pi\rho v^2 A}$$

By applying the Richardson extrapolation method using the coefficients at *D/dx* = 6, 12 and 20, the following error plots were obtained. *D/dx* = 4 and 6 make the error below 2% for Re =1 and 10. Due to sub-cell points, low error can be obtained with relatively low grid resolutions. However, if a particle is not spherical, elongated and or flattened, higher resolution may be required.

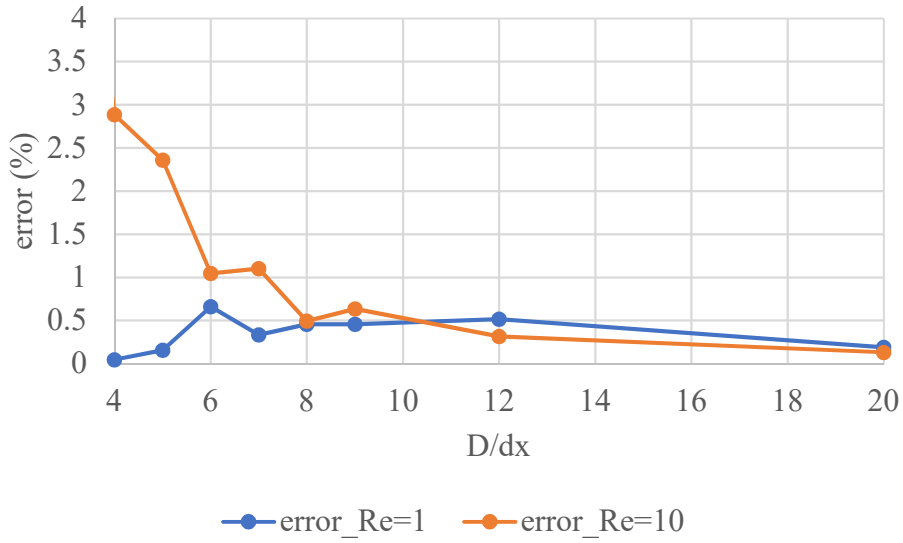


Figure 11. Errors of the drag force coefficients of the spherical particle according to the grid resolutions at Re = 1 and 10.

The drag force coefficients for the some Re numbers below 10 were obtained using the *D/dx* as 6 as shown below. The coefficients (CD_DNS) are close to the reference values (CD_ref) [Kaskas, Doctoral dissertation, 1970].

$$C_D = \frac{24}{Re} + \frac{4}{\sqrt{Re}} + 0.4, \qquad Re \leq 2{\sim}4 \times 10^5$$

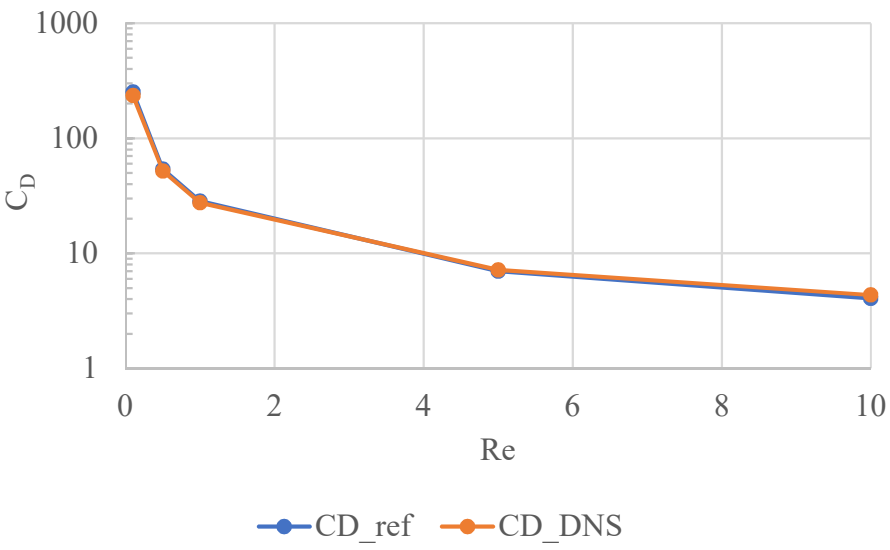


Figure 12. Drag coefficients for low Re numbers.

4.1.3. Convergence study of non-spherical particles.

For non-spherical particles, lifting force and torque should be calculated with the drag force. The coefficients of lifting force and torque were calculated by the equation below. The y and z components of the force coefficients vector ($\boldsymbol{C}_f$) are the lifting force coefficients, and the x component is the drag force coefficient. D_{eq} is the equivalent spherical diameter. For this convergence study, we used the absolute value of the force and torque coefficients vectors, but the components will be used as the target outputs for the MLP study in the future.

$$\boldsymbol{C}_f = \frac{-\sum_l \boldsymbol{F}_l \Delta V_l}{\frac{\pi}{2}\rho v^2 (\frac{D_{eq}}{2})^2}, \qquad \boldsymbol{C}_t = \frac{-\sum_l \boldsymbol{r} \times \boldsymbol{F}_l \Delta V_l}{\frac{\pi}{2}\rho v^2 (\frac{D_{eq}}{2})^3}$$

We have found the grid resolutions required for low Re region based on the drag force. However, when the particle is elongated or flattened, the required grid resolution can be increased. To figure out this, a randomly oriented particle having the aspect ratio 2 was generated and the convergence study was conducted considering drag force, lifting force and torque. The length of the entire domain is $20D_{eq}$ and the other conditions remained the same as the previous studies. Fig. 13 shows

the Lagrangian points of the particle and the velocity magnitude on the center x-z plan with *D/dx* = 12 at Re = 10.

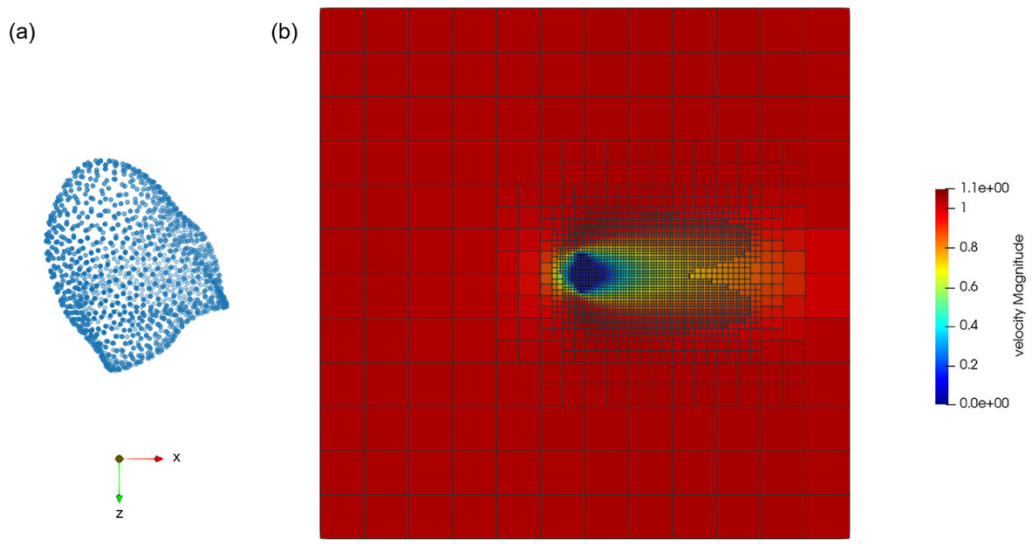


Figure 13. (a) Lagrangian points of the particle having d = 0.5 and AR = 2 and (b) the velocity magnitude of the flow at Re = 10 with D/dx = 12.

We applied the Richardson extrapolation method using the datasets at D/dt = 6, 12 and 20. D is the longest length of principal axis before the spherical harmonic transformation above order of 1. Fig. 14 indicates the errors of the force coefficients. Like the spherical particle, the errors of the force reduced exponentially according to the grid resolution. When D/dx = 6 was used, the error of forces was higher than the error of the spherical particle's case. For future data collection, the grid resolution will be multiplied by the aspect ratio so that the resolution can be based on the shortest dimension of the particle and the error can be below 3%.

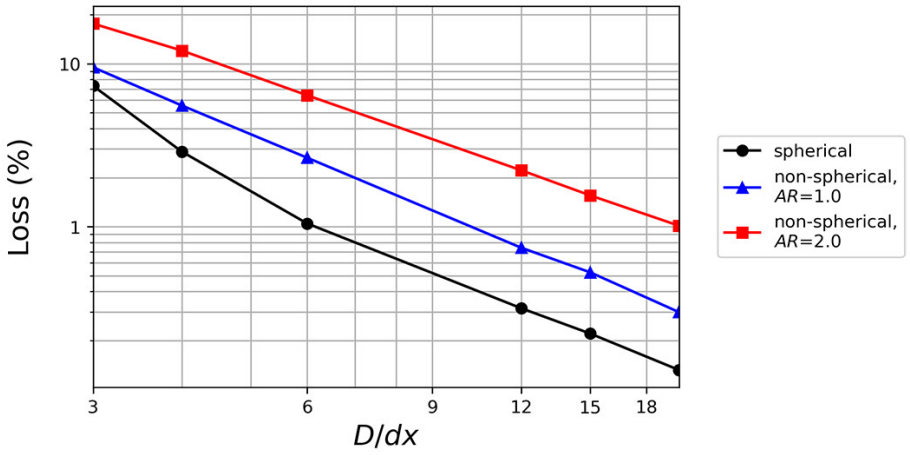


Figure 14. Errors of the force coefficients of the non-spherical particle having d = 0 and 0.5 and AR = 1 and 2 according to the grid resolutions at Re = 10.

4.1.4. Data collection from a low to moderate Re numbers

The grid resolution of the PR-DNS varies from 6 to 12 according to the aspect ratios of spherical harmonic particles. We collected 5,200 DNS datasets with the random *Re* number between 0.1 and 10. The shape and rotation angle of the particles are random and different from the particles used in VAE calculation. Therefore, when we develop the ANN model, the calculated latent vector will not be in the training datasets of VAE. The number of the datasets was decided based on the error of the ANN model which will be described at the end of the section.

Fig. 15 shows the drag force coefficients according to the *Re* numbers. The color bar indicates the *d* values of the particles. The overall trend (a) follows the typical drag force – *Re* number correlation. When we see the coefficients in detail (b), there is about 20% variation between data points which are caused by the particle shapes and orientation. The particles with higher *d* values tend to have higher drag force which makes sense because the rougher surface will make higher resistance against the flow.

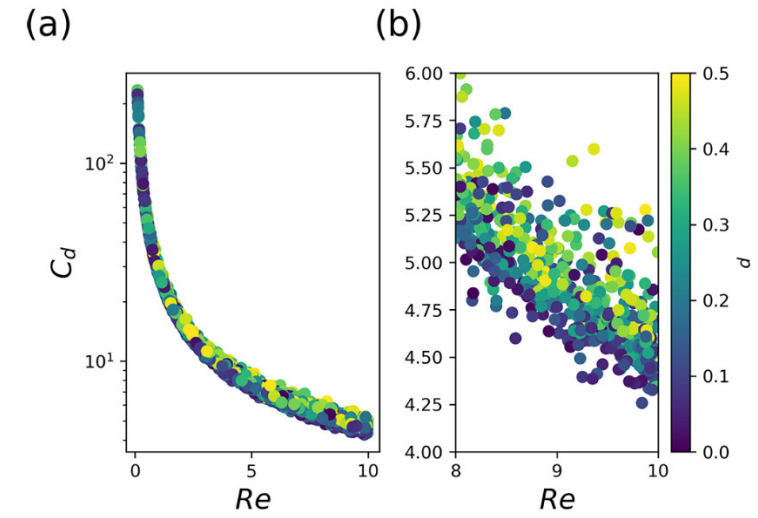


Figure 15. Drag force coefficients according to *Re* numbers.

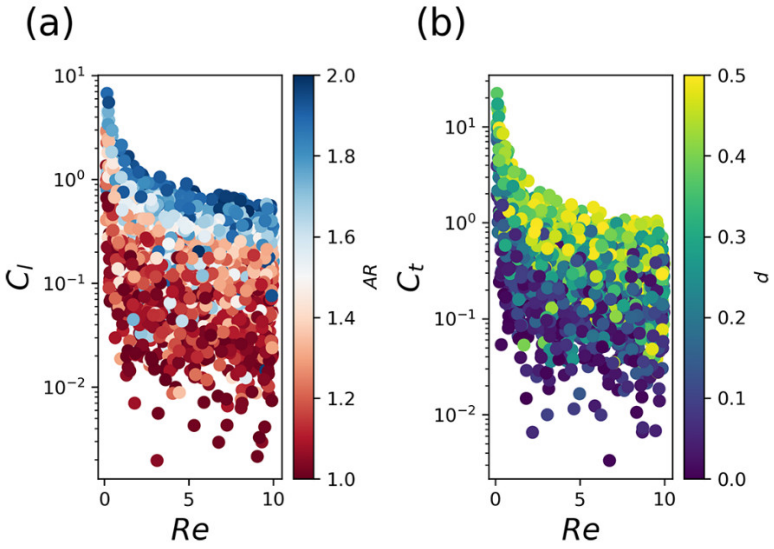


Figure 16. Lifting force and torque coefficients according to *Re* numbers.

Fig. 16 indicates the lifting force (C_l) and torque coefficients (C_t) from the PR-DNS calculation according to the *Re* numbers. The lifting force coefficient is the magnitude of the *y* and *z* components of force coefficient vector, and the torque coefficient is the magnitude of the torque coefficient vector. Unlike the drag force, lifting force coefficients are proportional to the aspect ratio (*AR*) and the variations are from the orientation. On the other hand, the torque coefficient is

affected by the *d* values, and this is because the magnitude of the drag force is larger than the lifting force and it affects to the outer product calculation for the torque coefficient.

4.1.5. The ANN (MLP) model development

As we explained earlier, the ANN calculation was conducted to correlate the particle and flow condition with the force and torque coefficient. By doing that, we found the number of datasets we need to develop the ANN model. The ANN model development was planned to be done in Task 6 but we conducted in advance to figure out the number of the datasets we needed. In Task 6, we will evaluate the ANN model with the high *Re* number datasets and develop a new model which can predict the interaction force for multiple particles system.

The ANN model includes two hidden layers with the dimensions of 32, 8. The input has the dimensions of 129 which contains the *Re* number and the latent vector of the non-spherical particles. The latent vector has the dimension of 128 and was calculated through the encoder of the trained VAE model. The hidden layers use the exponential linear units for the activation and the output layer uses the linear function for the regression. The output has the dimension of 6 including all direction components for the force and torque. We used the Adam optimizer and the mean square error (MSE) for the loss function.

We changed the number of the training datasets from 1,000 to 4,000 to check the loss according to the size of the datasets. 1,200 datasets were used for the validation and evaluation datasets. The validation datasets have been used to determine the parameter showing the minimum MSE. Note that loss of the lifting force and torque coefficient are the average values of the components. As the training proceeded, the validation losses were reduced, and overfitting was not observed. As we used the larger datasets, the minimum loss tended to be lower, and 4,000 datasets shows the converged minimum values. When the spherical particle is assumed, the lifting force and torque are zero if the fluid flows symmetrically. The dashed line in Fig. 17 indicates the MSE when we assume there is no lifting force and torque.

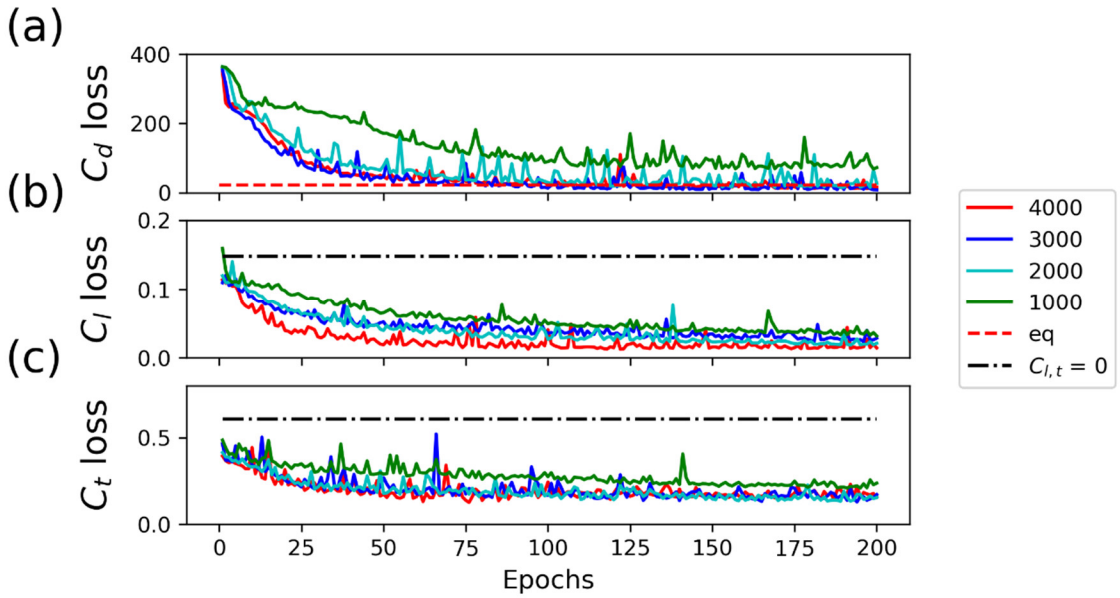


Figure 17. MSE of the validation datasets for the drag force, lifting force and torque coefficients according to the training steps and the number of the training datasets.

4.1.6. The ANN model evaluation

To evaluate the accuracy of the ANN model, we applied the following drag force model by G.H. Ganser [Powder technology, 77(2), 143-152 (1993)] for the non-spherical particle.

$$C_d = \frac{24}{ReK_1}\{1 + 0.1118(ReK_1K_2)^{0.6567}\} + \frac{0.4305K_2}{1+\frac{3305}{ReK_1K_2}}$$

$$K_1 = [\left(\frac{d_n}{3D_{eq}}\right) + 2/3\psi^{-0.5}]^{-1},\ K_2 = 10^{1.8148(-log\psi)^{0.5743}}$$

The above equation considers the sphericity (ψ) and the diameter of the projected area (d_n) of the non-spherical particle against the flow direction which means it considers the orientation of the particle. D_{eq} is the equivalent volume diameter of the non-spherical diameter. These parameters for the datasets we used for the ANN were calculated and the drag force coefficients for the validation datasets were obtained. As a result, the MSE was calculated as 21.8 which is higher than

the minimum value from the ANN, 12.7. We can see that Fig. 17 shows that accurate models can be obtained when the number of datasets is large enough.

Fig. 18 shows the parity plots for the drag force and *y*-direction lifting force and *z*-direction torque. The first plot (a) indicates the ANN model shows higher accuracy than the drag force model from the literature. Furthermore, lifting force and torque coefficients show low variation from the DNS calculation. Compared to the lifting force, torque coefficients indicate higher error. It seems because when we extract the geometrical features through the VAE model, the sharp shapes of the particles were smoothened.

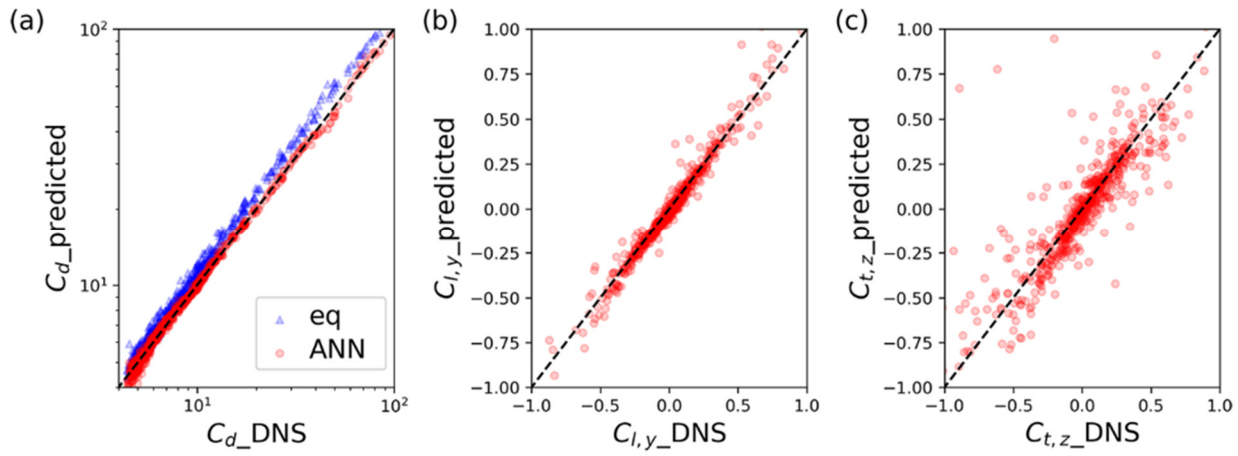


Figure 18. The parity plots for the drag force, y-lifting force and z-torque from the ANN and DNS.

To evaluate the accuracy of the lifting force and torque coefficients from the ANN, we utilized the following equations from another literature by M. Zastawny [International Journal of Multiphase Flow, 39, 227-239. (2012)]. In the equation, θ indicates the incidence angle which is the angle between the flow direction and the longest axis of a particle.

$$C_d = \frac{a_1}{Re^{a_2}} + \frac{a_3}{Re^{a_4}} + \left(\frac{a_5}{Re^{a_6}} + \frac{a_7}{Re^{a_8}} - \frac{a_1}{Re^{a_2}} - \frac{a_3}{Re^{a_4}}\right) sin(\theta)^{a_9}$$

$$C_l = \left(\frac{b_1}{Re^{b_2}} + \frac{b_3}{Re^{b_4}}\right) sin(\theta)^{b_5 + b_6 Re^{b_7}} cos(\theta)^{b_8 + b_9 Re^{b_{10}}}$$

$$C_t = \left(\frac{c_1}{Re^{c_2}} + \frac{c_3}{Re^{c_4}}\right) sin(\theta)^{c_5 + c_6 Re^{c_7}} cos(\theta)^{c_8 + c_9 Re^{c_{10}}}$$

In the literature, the authors fitted the coefficients with the results from DNS calculations. They used four different non-spherical particles. In this study, we selected one of them, the ellipsoidal particle having the *EI* of 0.8. For the evaluation, we collected DNS datasets for the non-spherical particles with the same *EI* value, not only a smooth particle but also rough particles with *d* values of 0.25 and 0.5. The *Re* number for the evaluation is 1. We conducted the calculation by rotating the particles in α direction and applied the ANN model to the rotated particles with the *Re* number of 1. Fig. 19 indicates the drag force and lifting force and torque coefficient prediction from the ANN, the PR-DNS, and the literature according to the incidence angle at $Re = 1$.

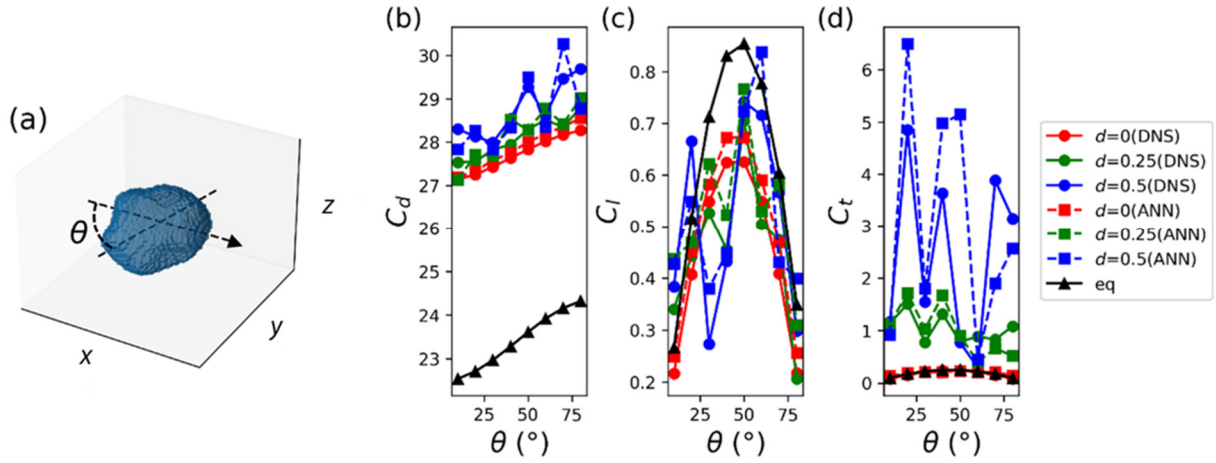


Figure 19. (a) The non-spherical particle with an incidence angle, (b) drag force, (c) lifting force, and (d) torque coefficients from the equation, the DNS and the ANN according to the incidence angle.

Fig. 19 demonstrates the ANN can predict the drag force coefficients for the particles much more accurately than the equations. The ANN model can predict the lifting force and torque including the surface shape effect which contributes to the deviations according to the incidence angles, even

though input data does not belong to the training and validation data. The main limitation of the equations is that new constants or equations are required for particles with higher d values since it is fitted for the ellipsoid particle with $d = 0$. On the other hand, this study has the advantage of making a more universal model through the neural network approach.

Subtask 4.2 – Simulation in Moderate-Reynolds Numbers

4.2.1. PR-DNS data collection

To identify grid resolutions, we perform PR-DNS and apply the Richardson extrapolation for the datasets at $D/dt = 6$, 12 and 20. We assume the error based on the magnitude of force at $D/dt = 30$. Fig. 20 indicates the errors of the force coefficients. Like the data for low-*Re* flows, the errors increase as the aspect ratio (*AR*) and roughness increase. On the other hand, the trend toward decreasing errors is not stable. For the ANN model development, the grid resolutions are chosen by multiplying the aspect ratio with 6 so that the grid resolution can be based on the shortest dimension of the particle and the error can be below 3%. The required grid resolution for PR-DNS does not increase compared to the low-*Re* conditions.

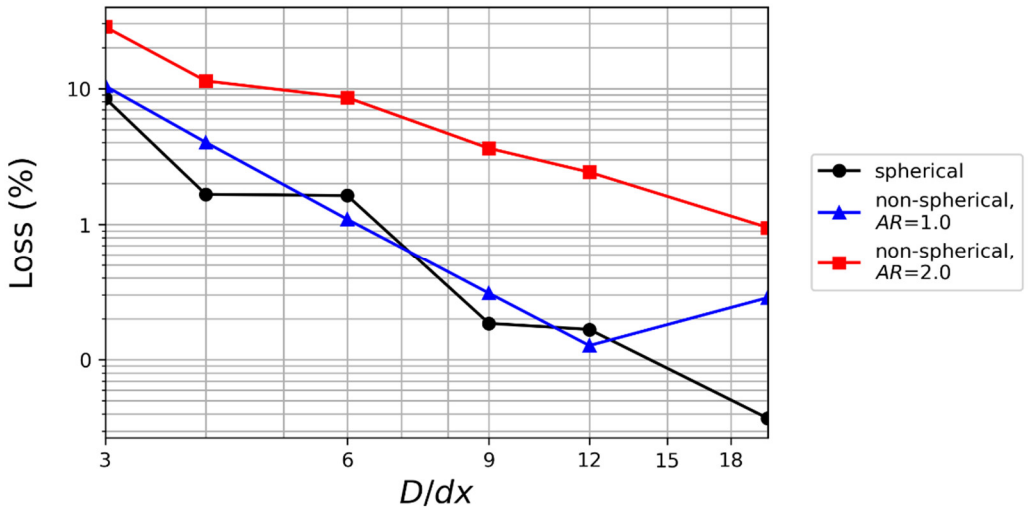


Figure 20. PR-DNS errors according to grid resolutions.

With the grid resolution conditions, we collect 5,200 DNS datasets with random *Re* numbers between 10 and 100. The random shape and rotation angle of the particles are chosen, and the particles for the VAE are excluded. Fig. 21 show the force coefficients from PR-DNS according to the *Re* numbers. Unlike the low-*Re* conditions, the force coefficients are less dependent on *Re* numbers, but are still affected by the orientations and shapes (spherical descriptor (*d*) and *AR*).

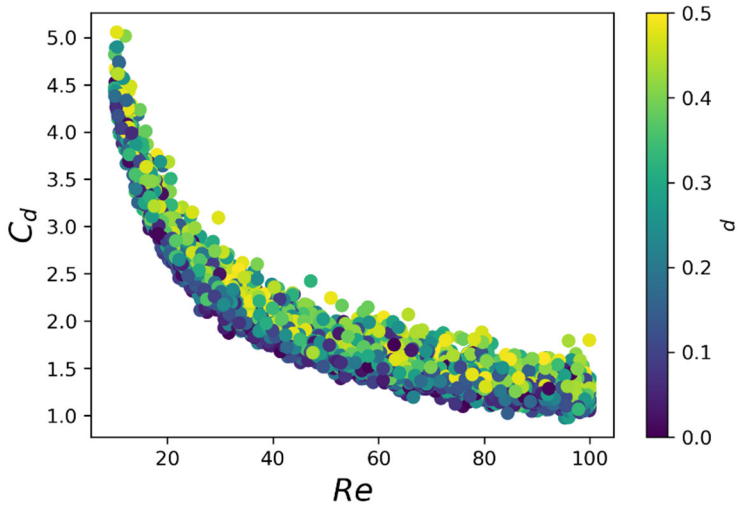


Figure 21. Drag force coefficients according to *Re* numbers.

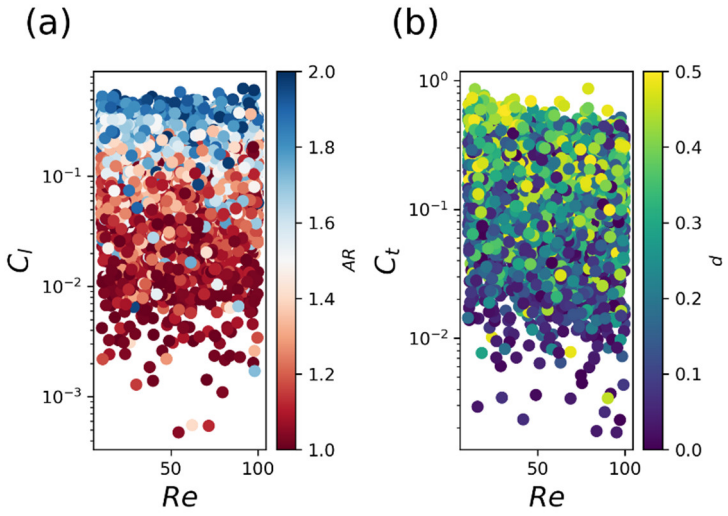


Figure 22. (a) Lifting force and (b) torque coefficient according to *Re* numbers.

4.2.2. ANN models for a single, non-spherical particle in moderate-Re flows

Using the PR-DNS datasets, the ANN model is developed. The same procedure and structure from the low-*Re* conditions are utilized to train the model. 4,000 datasets are used as training data, and 1,200 datasets are applied as validation and evaluation data. The number of epochs is decided by the minimum mean square error of the validation datasets. Fig. 23 shows the parity plots for the force coefficients, and they indicate high accuracy of prediction. Equation (1) indicates the drag force model by G.H. Ganser. The torque model showed relatively higher error for the model in low *Re* flows, but it shows more accurate result for the moderate *Re* region. The same result can be observed by comparing it to Equation (2) by M. Zastawny. When high *d* is applied, it shows force coefficients that deviate from the trend from Equation (2), but the ANN model can predict it. MSEs of drag force, lift force, and torque coefficients are 0.0014, 0.00085, and 0.00095, respectively. A MAPE of the prediction on C_ds is 1.1%, which is lower than the MAPE for the low *Re* regime (4.5%). The overall MAPE of the *Cd* prediction for low to moderate *Re* region is 2.8%.

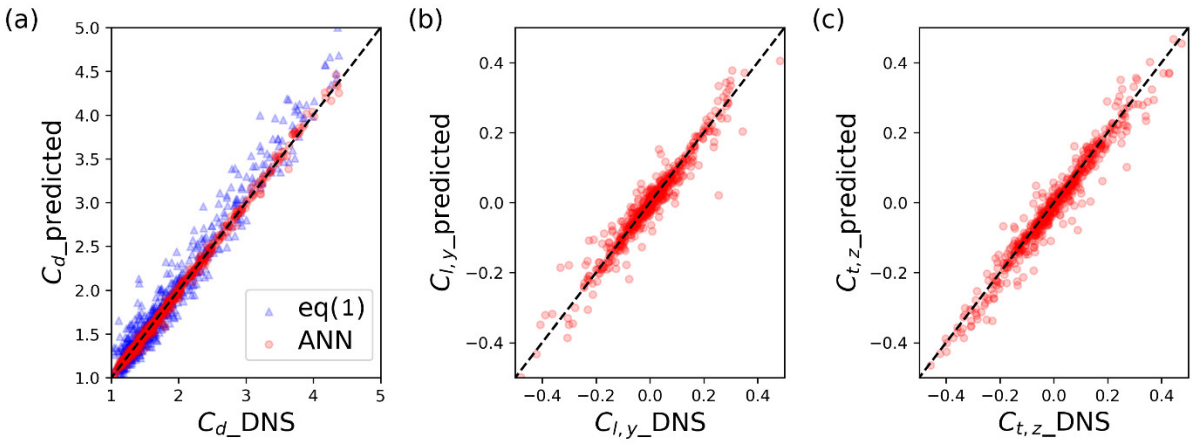


Figure 23. Parity plots for the force coefficients.

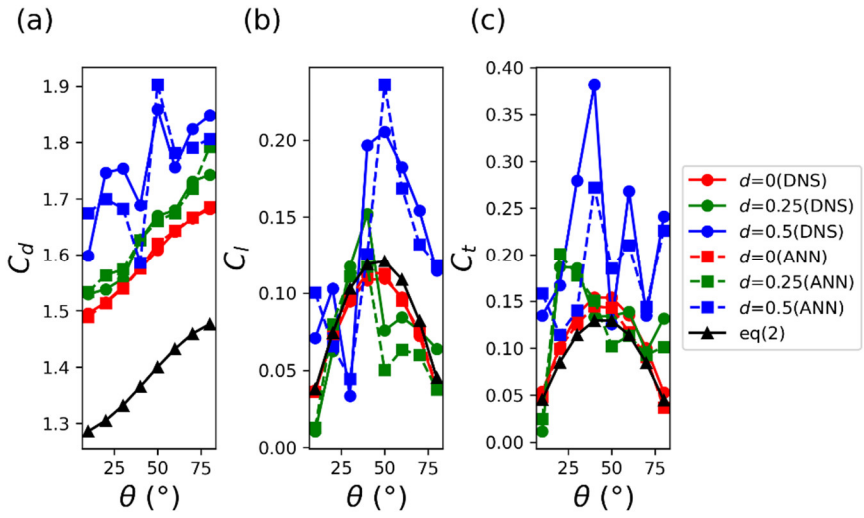


Figure 24. Force coefficients at Re = 50 according to incidence angles.

Task 6 Training of MLP-based Regressor and Final Reporting

Subtask 6.1 – Construction of the MLP architecture

6.1.1. PIEP method for multi-particles system

The PIEP model is a method which can consider the neighboring effect. This model evaluates undisturbed flows as a linear superposition induced by the neighboring particles. Here, the undisturbed flow is defined as the flow that would exist at the particle location in the absence of that particle with neighboring particles. Based on the PIEP model by G. Akiki [Journal of Computational Physics, 351, 329-357. (2017)], the drag force of the *i*th particle ($\tilde{F}_{drag,i}$) can be expressed as follows:

$$\tilde{F}_{drag,i} = \bar{F}_{drag}(Re_i, \varphi) + \{3\pi\mu d_i \sum_{\substack{j=1 \\ j\neq i}}^{N} \overline{u_{j\to i}}^{S} (1 + 0.15Re_i^{0.687})\}$$

In the above equation, $\bar{F}_{drag}(Re_i, \varphi)$ is the mean drag force of the *i*th particle with the Reynold number, Re_i and the volume fraction, φ. $\overline{u_{j\to i}}^{S}$ is the velocity perturbation due to *j*th neighbor, and

it is averaged over the surface of the *i*th particle. The equation of the drag force can be reformulated in terms of the drag force coefficient (C_D) as follows.

$$\tilde{C}_{D,i} = C_D(Re_i)\left\{f(\varphi, Re_i) + \frac{u_{x,mic}(\boldsymbol{r}_i)}{u_{mac}}\right\} + \frac{V_{p,i}[-\nabla_x p_{mic}(\boldsymbol{r}_i) + \nabla^2 u_{mic,x}(\boldsymbol{r}_i)/Re_i]}{\pi/8\rho {d_{eq,i}}^2 {u_{mac}}^2}$$

$$\approx C_D(Re_i)\left\{f(\varphi, Re_i) + \frac{\sum_{\substack{j=1 \\ j\neq i}}^{N} \overline{u_{x,J\to\iota}}^{S}(\boldsymbol{r}_i)}{u_{mac}}\right\}$$

$$- \frac{4d_{eq,i}\left[\sum_{\substack{j=1 \\ j\neq i}}^{N} \overline{\nabla_x p_{J\to\iota}}^{V}(\boldsymbol{r}_i) - \frac{1}{Re_i}\sum_{\substack{j=1 \\ j\neq i}}^{N} \overline{\nabla^2 u_{x,J\to\iota}}^{V}(\boldsymbol{r}_i)\right]}{3\rho {u_{mac}}^2}$$

In the above equation, we know the drag force coefficient for a single particle ($C_D(Re_i)$) from the previous studies, and assume that the mean drag force can be separated independently into $C_D(Re_i)$ and a function (*f*) in respect to φ and/or Re_i. In this study, the macroscopic velocity (u_{mac}) of the flow is dimensionless unity in *x*-direction, and the *x*-components of the velocity perturbation is only considered. The velocity perturbation will be calculated from superposition and an ANN model, which can predict a flow field of a single particle based on *Re* numbers and shapes (latent vectors). Because we can obtain the drag force coefficient of the *i*th particle in the poly-dispersed particle array from DNS, $f(\varphi, Re_i)$ can be calculated. The second term of the above equation indicates an undisturbed flow force, which requires superposed pressure fields. For the DNS simulation, mean *Re* numbers will be applied based on the mean diameter of the particles, and then, Re_i will be calculated with the corresponding diameter (d_i). The next section shows how to configure the ANN to predict the flow-fields from latent vectors and the *Re* numbers. Fig. 25 shows the scheme of this study with the ANN models and PIEP method.

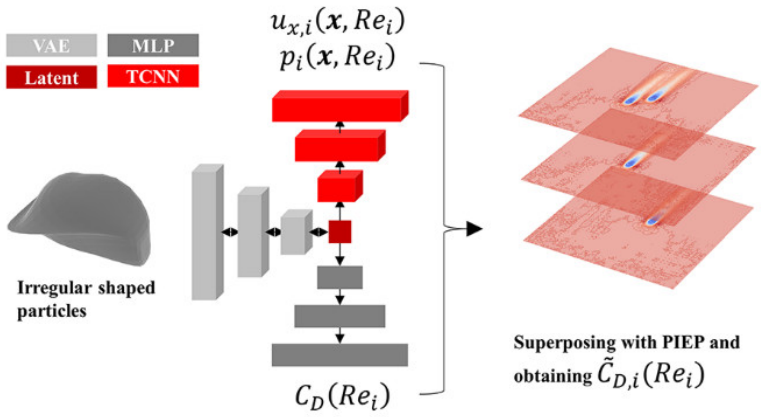


Figure 25. The scheme of PIEP method for the drag force with neighboring effects.

6.1.2. Development of a neural network model for flow-field prediction

To predict the flow-fields, we develop multi-layer neural network model using de-convolutional layers. The input of the model has 132 parameters including 128 parameters from the latent vector, a spherical descriptor (d), EI, FI and a Re number. The latent vector is obtained through the encoder of the VAE model. For the velocity fields, the ANN model includes five de-convolutional hidden layers with the number of channels of 256, 128, 64, 32, and 8, respectively. Each layer uses ELU for the activation function and the number of strides of 2. The input layer is fully connected and utilizes ELU to convert the input to the 3-D datasets. The output layer is also a fully connected layer but uses linear activation function because this is a regression model to predict the continuous data of the velocity. On the other hand, the pressure model uses the output dimension of 40^3, which is smaller than that of the velocity (160^3) because the pressure gradient is only effective near the particles. The training is conducted with the ADAM optimizer by using Python and Keras library, and the datasets from the previous studies (10,400 datasets, $0.1 < Re < 100$) are utilized. Note that the particles for the ANN training are independent of the particles for the training of the VAE. The output data are from DNS and are converted into the dimension of 160^3 or 40^3 based on the finest resolution of the adaptive refinement method. Because this dimension is too large to be loaded, data generator is utilized. The length of axis of the DNS data depends on the size of the particle. This is because the length was specified as 20 times the average diameter for the DNS. In this

study, to match the length of the training data equally, the median value of 160 is chosen. For data with a length of less than 160 on the data axis, edge data are calculated through extrapolation to match the dimension for the training, but they will not be used for PIEP because this study ignores the neighboring effect farther than 10 times the average diameter. With being shuffled, 8000, 1200, and 1200 datasets are used for training, validation, and evaluation, respectively. Mini-batch training is applied with the batch size of 5, and the optimum model is chosen based on the accuracy of the validation data in 10 epochs with an early stopping method. The VAE model to extract the latent vector is the same model with a model which was used for the prediction of $C_D(Re_i)$.

Fig. 26 shows the cross-sectional flow-fields ($z = z_{max}/2$) from the evaluation data and the predictions from the trained ANNs (TCNN 1 and TCNN 2). The ANN models show the mean absolute error of 0.00091 and 0.013 for the velocities in the x direction and pressure, respectively. The accuracy of the overall model needs to be confirmed after applying the superposition in terms of the drag force coefficient.

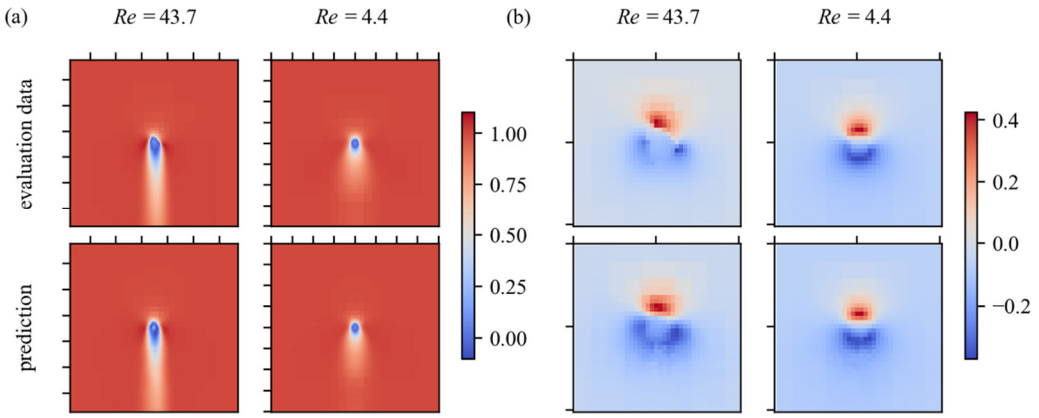


Figure 36. Evaluation and predicted flow fields from the DNS and the ANN models, respectively.

When the *Re* number is close to 100, the wake of the flow appears long to the boundary. This means that neighboring particles can affect the centered particle even from far distance, but in this study, the effect of particles farther than 10 times the average diameter is ignored.

We have tried to obtain the flow filed model using the decoded 3D voxel data from the VAE model. However, it has turned out that applying latent vector shows better accuracy. Another strategy we

have tried is applying VAE to the flow fields, but the latent vector to represent the fields only has shown about 95% of reconstruction error. Because the total error will be higher than the reconstruction error after correlating with the particle shapes, we choose the de-convolutional ANN to model the flow fields.

6.1.3. PIEP application for sparse multi-particle systems

Five irregular-shaped particles are randomly chosen from the evaluation data of the ANN. The PR-DNS, in this study, adjusts the viscosity of the flow (μ_f) with the density (ρ_f) and the macroscopic flow velocity (u_{mac}) of unities. Each particle has a different *Re* number, and the averaged *Re* numbers for the tests are 5 (2 particles) and 50 (3 particles) based on the following expressions. The individual *Re* number (Re_i) of each particle is applied to obtain the drag force coefficient and the flow field from the ANN models.

$$Re = \frac{\rho_f u_{mac} \langle d \rangle}{\mu_f}, \qquad Re_i = \frac{\rho_f u_{mac} d_i}{\mu_f}$$

Fig. 27 shows the cross sectional ($z = 0.5L$, L: length of domain) flow fields of 2 (Case 1, $L = 112$) and 3 particles (Case 2, $L = 144$) from the PR-DNS. Far-field boundary conditions are applied for the cases. For Case 1, the first (1) and the second particles (2) are located at ($0.5L$, $0.5L$, $0.5L$) and ($0.6L$, $0.5L$, $0.5L$), respectively. The three particles for Case 2 are located at ($0.5L$, $0.5L$, $0.5L$), ($0.6L$, $0.6L$, $0.5L$), and ($0.8L$, $0.55L$, $0.5L$), respectively.

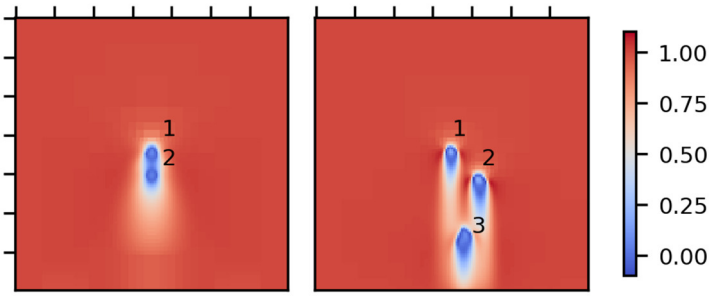


Figure 27. DNS results of Case 1 (left) and Case 2 (right). The color bar indicates flow velocity in *x*-direction.

To predict the drag force coefficients, the interaction force ANN model (MLP) is applied, and the flow field ANNs (TCNN 1, 2) are utilized to superpose the flow field in the flow direction (x). MLP and TCNN utilize the latent vector from the VAE, which was trained with independent particles sets.

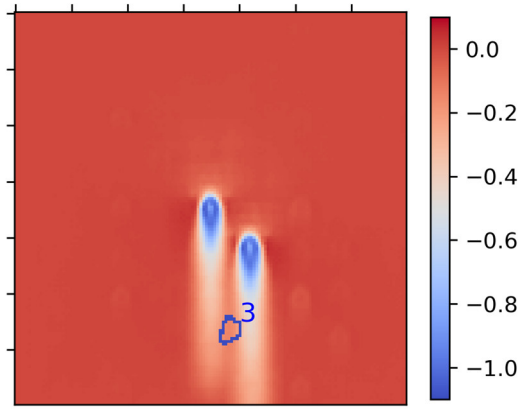


Figure 28. A superposed flow field (x-direction) of particle 1 and particle 2 of Case 3 using TCNN 1. Note that the color bar varies between 0.1 and -1.1.

Fig. 28 shows the superposed velocity field of particle 1 and particle 2 of Case 3 with TCNN. This perturbed velocity field is used to calculate the averaged velocity of the x-component on the surface of particle 3. Even though the resolution of the grids is not high enough to represent the detailed shape of particle 3, the resolution of the $48 \times 48 \times 48$ from the input data of the VAE is reflected to obtain the averaged velocity. Due to the sparse concentration of the system, the solid volume fraction can be ignored. Therefore, the following equation is applied to obtain the predicted force coefficients ($\tilde{C}_{D,i}$). We applied the same method for obtaining the superposed pressure field in the following equation.

$$\tilde{C}_{D,i} = C_D(Re_i)\left\{f(\varphi, Re_i) + \frac{u_{x,mic}(\boldsymbol{r}_i)}{u_{mac}}\right\} + \frac{V_{p,i}[-\nabla_x p_{mic}(\boldsymbol{r}_i) + \nabla^2 u_{mic,x}(\boldsymbol{r}_i)/Re_i]}{\pi/8\rho {d_{eq,i}}^2 {u_{mac}}^2} \approx$$

$$C_D(Re_i)\left\{f(\varphi, Re_i) + \frac{\sum_{\substack{j=1 \\ j\neq i}}^{N} \overline{u_{x,J\to\iota}}^{S}(\boldsymbol{r}_i)}{u_{mac}}\right\} - \frac{4d_{eq,i}\left[\sum_{\substack{j=1 \\ j\neq i}}^{N} \overline{\nabla_x p_{J\to\iota}}^{V}(\boldsymbol{r}_i) - \frac{1}{Re_i}\sum_{\substack{j=1 \\ j\neq i}}^{N} \overline{\nabla^2 u_{x,J\to\iota}}^{V}(\boldsymbol{r}_i)\right]}{3\rho {u_{mac}}^2}$$

The following table shows the drag force coefficients from the PR-DNS, and the results from the ANN models and the PIEP model. The DNS results for a single particle regards the individual *Re*

number, Re_i. The fourth row indicates the drag force coefficients just based on the MLP without the neighboring effect. The table shows percentage errors of cases relative to the multi-particle results from the DNS, and they are decreased after considering the neighboring effect. For the particles located at the front, it can be observed that the error is not large even just with ANN 1 because of the least influence from neighboring particles. Whereas particle 2 of Case 1 shows a higher error, the error of particle 2 of Case 2 is smaller due to the narrow wake. Instead, particle 3 of Case 2 indicates the highest error due to the neighboring effect. When the PIEP with TCNNs is applied, the overall error reduces. A MAPE of 5 cases is reduced to 21.5% to 1.7%, and the errors of particles under strong neighboring effect are significantly reduced. Due to the error of the MLP (2.8%), TCNNs (0.5%), and the superposition assumption, a certain amount of error is inevitable.

Table 5. The drag force coefficients and MAPEs of the sparse multi-particle systems, Case 1 and Case 2.

C_D / error	Case 1 (Re = 5.0)		Case 2 (Re = 50.0)		
	1	2	1	2	3
DNS (multi-particle)	6.66	4.48	1.73	1.68	1.40
DNS (single particle)	7.48	7.21	1.75	1.71	1.63
MLP	7.53/13.1%	7.37/64.5%	1.75/1.11%	1.71/2.3%	1.62/16.3%
MLP+ TCNNs/PIEP	6.64/0.2%	4.23/5.6%	1.73/0.1%	1.70/1.7%	1.38/1.0%

If the multi-particle system contains many particles, solid volume fraction needs to be considered. Several studies [Chemical Engineering Science, 192, 1157-1166. (2018)] have suggested the model to regard the effect but there is no reference for the irregular-shaped particles.

6.1.4. Solid volume fraction effect

There are several studies to include the solid volume fraction (φ) effect on the drag force, but the separate term is going to be applied to utilize the individual drag force model (MLP) in this study.

Hilton et al. [Chem. Eng. Sci., 65 (5) (2010)] showed the solid volume effect for particle assemblies of ellipsoids and cuboids as follows:

$$f(Re_i, \varphi) = (1-\varphi)^{-3.7+0.65exp[0.5(1.5-logRe_i)^2]} \quad (13)$$

Equation 13 is for an average drag force, but it will be investigated whether it can describe the individual particle of this study. Fig. 29 shows the solid volume fraction effect according to Re_i and φ. When φ is 0.5% with Re =50.0, the volume fraction factor is only about 1.4%. As the solid fraction increases, the effect of the fraction and the dependence of Re increases. We employ one hundred irregular particles with $\varphi = 0.005$ to ignore solid volume fraction effect and investigate neighboring particle effect.

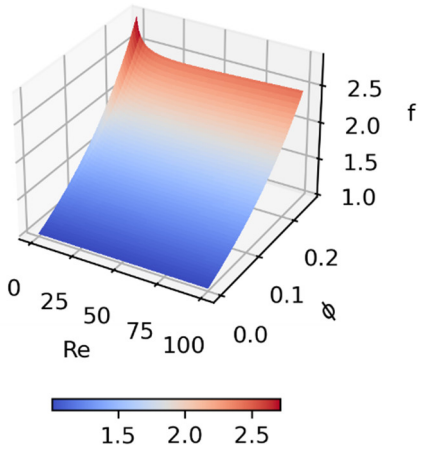


Figure 29. Volume fraction factors according to Re and φ.

6.1.5. Dilute multi-particle system

The PR-DNS is applied to collect the drag force coefficients of sparse systems with $\varphi = 0.005$. 100 irregular particles in the evaluation data of TCNNs are included. Particles have a length of 6 for the smallest axis, and as in training the ANN models, the AR is between 1 and 2. The particles are located randomly, and the particle voxel data are utilized to inhibit overlaps between particles. For the DNS, the AMR update is applied after the velocity residual approaches to 1.2×10^{-6}, and then the simulation stops when the residual is 1.0×10^{-6}. Periodic boundary conditions are applied

for all directions, and the driving force is applied based on Equation (14), where p_c is a parameter used to adjust the magnitude of the pressure gradient.

$$\nabla p_e = \boldsymbol{e}_x p_c \rho v^2 / \langle d \rangle^3 \tag{14}$$

For the systems with the initial *Re*s of 1, 5, 25 and 50, 0.05, 0.5, 5 and 15 are chosen for ∇p_e, respectively. The first row in Fig. 30 indicates the velocity field in *x*-direction with $z = 0.5z_{max}$ from the DNS. Converged *Re*s shows the average values of 0.5, 4.2, 24.0 and 49.8 of *Re*s. The second row of Fig. 30 shows the predicted, superposed flow fields from TCNN 1 using the PIEP method.

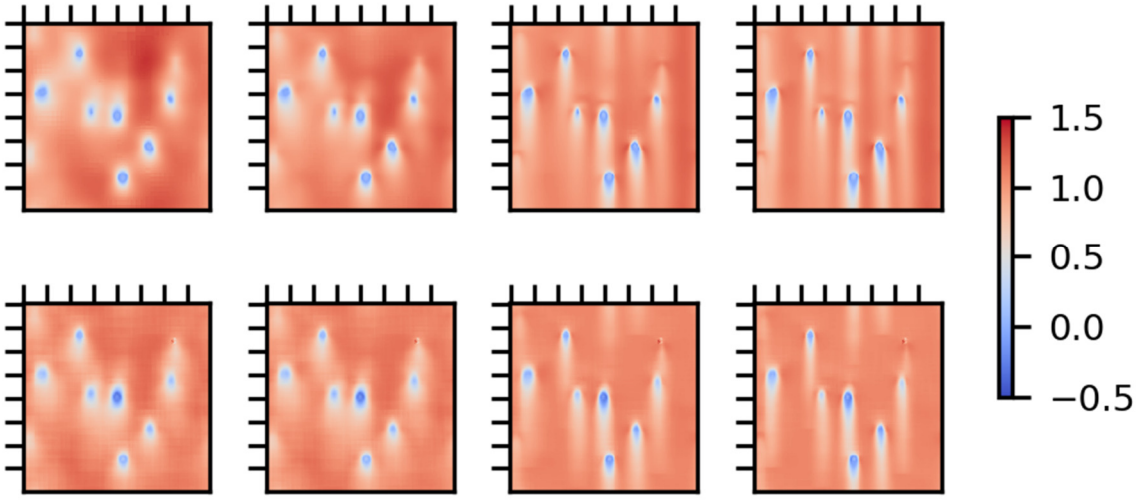


Figure 30. Velocity fields (*x*-direction) from DNS (top) and TCNN 1/PIEP (bottom) for 100 irregular shaped particles with *Re* = 0.5, 4.2, 24.0, and 49.8 (left to right).

For the 100 irregular-shaped particles, the MLP is applied to obtain C_D, and TCNNs and PIEP are utilized to regard the neighboring effect ($\widetilde{C_D}$). In this study, we first find the sum of the velocity fields for all particles, and then subtract only the velocity field of the corresponding particle. Because the predicted flow fields from ANN 2 have limited domain size, linear summation of the fields accumulates an error, which cannot be ignored when there are a lot of particles. Therefore, incidence velocity of the combined domain is utilized to compensate for the error as follows.

$$f_e = \langle \sum_{i=1}^{N} (u_{i,ANN2} - u_{mac}) \rangle_{x=0} \quad (15)$$

$$\sum_{j=1, j\neq i}^{N} u_{j\to i} = \sum_{j=1}^{N} (u_{j,ANN2} - u_{mac}) - (u_{i,ANN2} - u_{mac}) - f_e \quad (16)$$

In Fig. 31, the prediction with the MLP, TCNNs and PIEP shows higher accuracy than the model only with the MLP. The data are arranged in ascending order based on the true values, and when only the MLP is used, the plot shows a small tendency according to the index due to the absence of the neighbor effect, but in the case of PIEP, positive correlations can be observed. Some studies [Big Data, 8(5), 431-449 (2020)] used a moving average to smoothen the noise of the prediction according to the particle index, but this approach is impossible without the true values. Table 6 shows MAPEs and R^2 scores of the cases. It shows higher accuracy with TCNNs/PIEP than the prediction only with the MLP. Predictions for smaller *Re* show higher accuracy because the flow domain from TCNNs is large enough to include the flow pattern. However, for higher *Re*, the wake is larger than the domain, and it makes strong interaction over far distance. Because this study only considers the interaction within 10 d_{eq}, high *Re* shows relatively higher error. If solid volume fraction is higher, the effective distance can be smaller, but the solid volume fraction effect needs to be investigated according to *Re*, φ, *AR* and *d*.

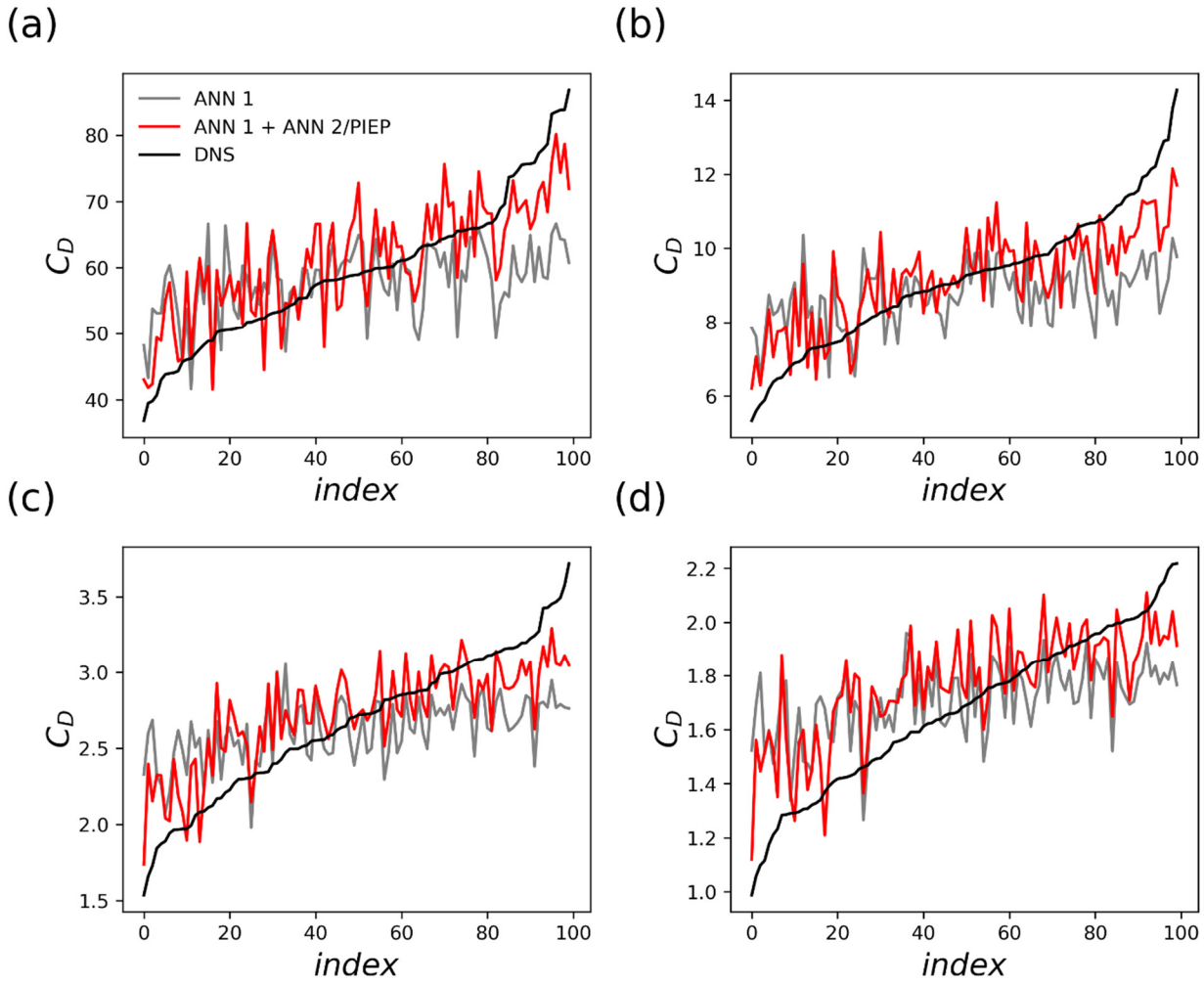


Figure 41. C_ds from DNS, MLP, and TCNNs/PIEP for 100 irregular shaped particles at Re = 0.5 (a), 4.2 (b), 24.0 (c), and 49.8 (d).

Table 6. R^2 score and MAPE with MLP, TCNNs/PIEP.

<*Re*>	Re_{min}	Re_{max}	R^2 (MLP/TCNNs)	MAPE (%, MLP/TCNNs)
0.51	0.42	0.68	0.19/0.62	12.8/9.8
4.2	3.5	5.7	0.16/0.65	14.5/9.8
24.0	20.0	32.3	0.26/0.62	13.4/9.5
49.8	40.8	66.9	0.24/0.48	13.4/11.1

Subtask 6.2 – Training, Validation, and Optimization of MLP

6.2.1. Solid volume effect formulation

Because the existing solid volume effect factors (f) are for spherical or smooth particles, we need to collect the data for high d particles and validate the factors. To do that, we have created four particles as follows. Please note that the spherical harmonic method provides different shapes with the same shape factors, so we create Particle 4 to compare with Particle 1.

Table 7. Shape factors of the particles for validating solid volume effect factor.

Particle	D	AR
1	0.5	2
2	0	1
3	0.3	2
4	0.5	2

To generate packed systems, we apply a simple Monte Carlo packing algorithm to most cases except the case with $\varphi = 0.2$. For the high concentration condition, The Monte Carlo-based algorithm shows extremely heavy computation time. Therefore, we utilize the advancing packing algorithm for the systems with $\varphi = 0.2$. To achieve even higher concentration, additional methods such as a minimum potential method can be applied, but we just applied the simple advancing method to obtain the concentration up to $\varphi = 0.215$.

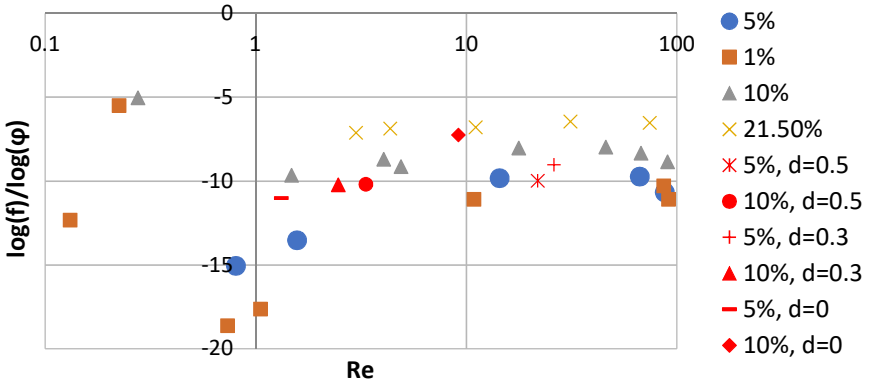


Figure 32. Solid fraction factors of homogeneous particle systems according to Re, d, and φ.

By using *Re*s between 0.1 and 100 and φ between 0.01 and 0.215 for the PR-DNS, we plot the results in Fig. 32 based on the equation in the literature [Powder Technology, 401, 117303 (2023)].

$$\overline{F_D} = F_D f(Re,\ \varphi,\ d)\ = F_D \varphi^{-3.7+0.65\exp(-0.5(1.5-\log(Re))^2)} \quad (17)$$

$$\frac{\log(f)}{\log(\varphi)} = (-3.7 + 0.65\exp(-0.5(1.5 - \log(Re))^2)) \quad (18)$$

The plots for Particle 1 in Fig. 32 (blue, orange, gray, and yellow markers) show solid fraction dependency as opposed to the tendency shown in Equation 2. Moreover, the factors vary much more than the range from Equation 18. This implies that we need to modify Equation 17 and 18. In addition, the trend below Re = 1 does not follow the relationship of Equation 18. We will narrow down the *Re* range between 1 and 100 because it seems we need to collect intensive data for the low *Re* regime. It has shown a much higher computational cost for the *Re* regime around 0.1. Lastly, it seems there is no big impact of roughness on the solid fraction effect. Particle 4 also does not show much difference to other particles including Particle 1 having the same *d* value. Note that we are applying the same orientation for all same particles to obtain the *f*. We have assumed that *f* depends on *Re*, φ, and averaged *d* of surrounding particles. If it turns out that *d* does not affect the *f*, we will not need to add any additional parameters to Equation 17 and 18, but just modify them based on the PR-DNS results.

We have shown that particle shape does not affect the solid fraction effect much. With the data of Particle 2, 3, and 4, we have formulated the solid fraction effect. We collect more data of Particle 1 using the PR-DNS, and Fig. 33. shows the result. Note that the red and black boarder indicate the particles other than Particle 1 and we have ignored the low *Re* data for simplicity. Particles with back border have *AR* of one instead of two. These data show smaller *f* factors and cause error on the following PIEP results. It will be discussed further in the next section. Note that the *Re* number is the particle *Re* number based on superficial velocity.

Based on the data collected, we assume the following formulation by referring to the literature [International journal of multiphase flow, 20(1), 153-159. (1993)].

$$x = (\ln(Re) - 3.5)^2 \quad (19)$$

$$\frac{\log(f)}{\log(1-\varphi)} = a(\varphi)x + b(\varphi) \quad (20)$$

$$a(\varphi) = 0.18\ln(\varphi) + 0.23 \quad (21)$$

$$b(\varphi) = 18.1\varphi - 10.4 \quad (22)$$

The regression using Equation (20) shows R2 scores between 0.44 and 0.99, and it shows 0.97 and 0.99 with Equation (21) and (22), respectively. Using the regression parameters, Equation (19-22) shows mean absolute percentage error of 5.3% and R2 score of 0.97.

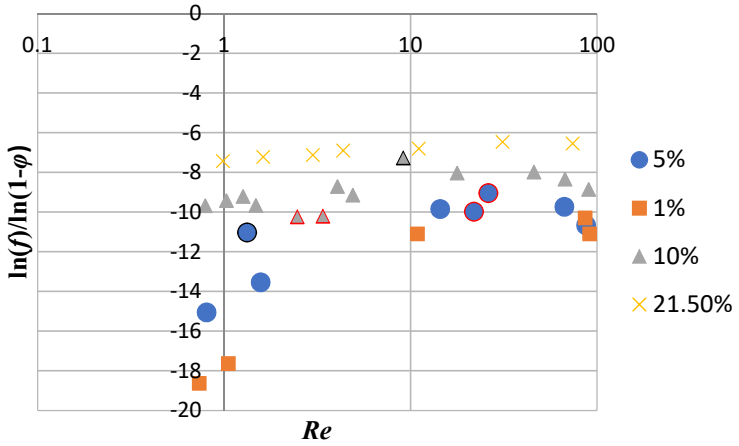


Figure 33. Solid fraction effect (homogeneous) according to concentration and *Re*.

6.2.2. High concentration systems

To test the solid fraction effect from the previous study, we generate three high concentration systems (Case 1: 6.1% and Case 2: 5.1%, Case 3: 13.2%) using random packing method. We assume we can apply the previous model for *f* factor can be applied to random, polydisperse systems because the particle shape does not affect *f* factor much. With *Re* numbers of 1.81, 2.25 and 49.1, respectively, we compare the predicted results from the PIEP-based method with the results from the PR-DNS. Case 2 uses monodisperse system using particles with $d = 0.49$ and $AR = 1.1$. Case 1 and 3 utilize polydisperse particles with $d = 0 – 0.5$ and $AR = 1$ - 2. Fig. 34. shows

the x-velocity fields from the PR-DNS and TCNN for Case 1, and Fig. 35. shows the ascending order data of the drag force coefficients of Case 1, 2 and 3. Note that for calculating the *f* factor, we have used the average *Re* instead of individual *Re*. The results show MAPEs of 12%, 7% and 24.9%, and R2 scores of 0.46, 0.7 and -0.56, respectively. Without the PIEP method, the prediction shows 4 – 9% lower accuracy. Case 3 shows low accuracy because the *f* factor prediction is not accurate, and when we apply a modified value from the PR-DNS, it shows much better prediction. It implies that *f* factor prediction is important, and we believe the reason of error is due to the high *AR* of the monodisperse particles using for the *f* factor regression. It makes aligned arrangement of the particles, and it shows higher *f* factors than those of random particles. As a result, we can see lower *f* factors for low *AR* particles in Fig. 33 because they do not make strong alignment. Therefore, the *f* factor model should be re-formulated with random oriented particles, and this will be provided in the next section.

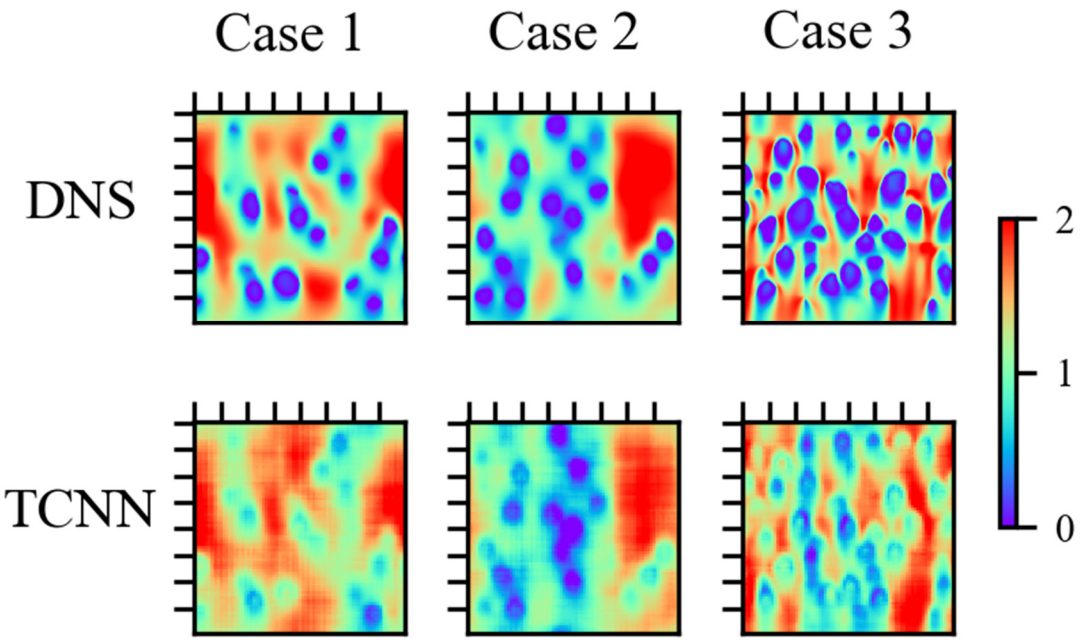


Figure 54. DNS and TCNN results for high concentration systems.

Table 8. Prediction results with MLP, TCNN and PIEP for high concentration systems.

	Case 1	Case 2	Case 3	Case 3-1
dispersity	poly-	mono-	poly-	poly-
conc. (%)	6.1	5.1	13.2	13.2
Re	1.82	2.25	49.1	49.1
f	2.02	1.84	2.98	2.46
MAPE (MLP+PIEP/MLP, %)	11.2/20.3	7.0/13.2	24.9/28.6	12.7/17.4
R2 (MLP+PIEP/MLP, %)	0.53/-0.46	0.70/-0.06	-0.56/-1.01	0.48/0.06

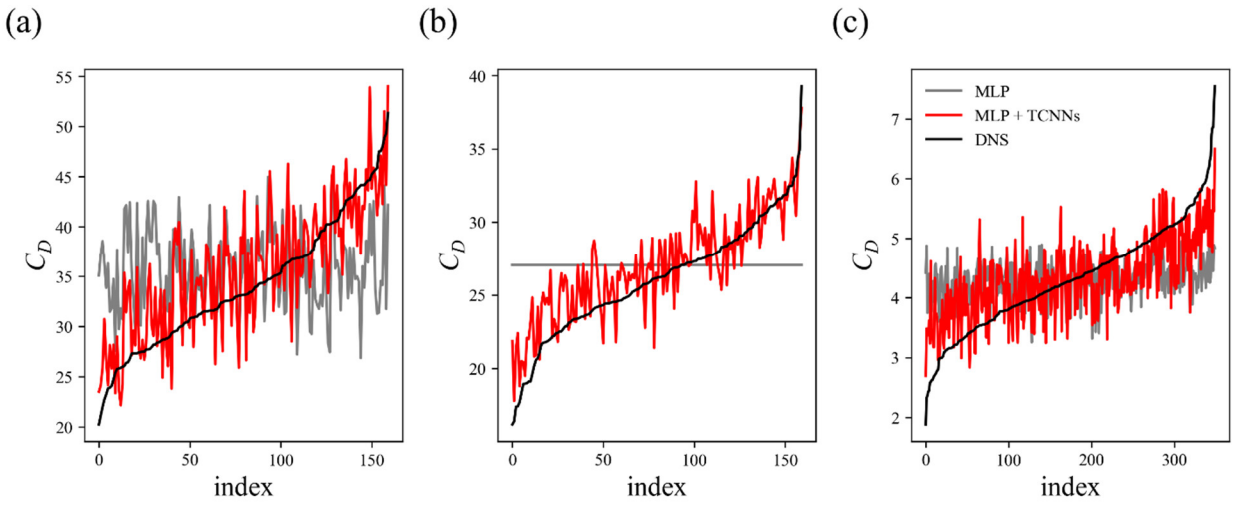


Figure 35. Prediction results using MLP or/and TCNN+PIEP method for (a) Case 1, (b) Case 2, and (c) Case 3-1. Note that (c) is using the modified *f* factor driven from the PR-DNS.

6.2.3. Solid volume effect for random particles and evaluation

To modify the solid volume effect factors with random particles (d = 0~0.5, AR = 1~2), we have performed the PR-DNS with the concentrations (φ) of 0.5%, 2.0%, 6.1%, 10.8%, 14.7%, and 20.0%. We have applied random packing algorithm for the systems except the system with 20.0%, which need to use the advanced packing method. Resolution has been adjusted to contain at least one hundred particles for all cases. The MLP is utilized to calculate the C_D for each particle for given *Re*s rather than applying the PR-DNS to all individual particles. It is because the MLP has shown good predictions. Once the PR-DNS is converged with a certain value of *Re*, then the MLP

is applied to obtain the average C_D taking the particle size into consideration, and the f can be calculated by dividing the C_D with the true average C_D from the PR-DNS. Table 9 shows the f factors, and the f exceeds 10% when the solid volume fraction is higher than 2%.

Table 9. Solid volume fraction effect factors according to Re and volume fraction.

Re	0.5%	Re	1.0%	Re	2.0%	Re	6.1%
0.96	1.102	1.286	1.077	0.9216	1.358	1.12	1.993
1.751	1.061	2.238	1.1	1.7313	1.266	8.62	1.878
5.718	1.092	15.28	1.042	11.4	1.237	22.13	1.621
10.164	1.101	38.93	1.006	44.832	1.146	54.74	1.568
46.71	1.026	59.1	1.011	57.774	1.14		
57.3	1.017						
Re	10.8%	Re	14.7%	Re	20.0%		
1.17	2.84	1.25	3.594	0.8218	4.457		
2.278	2.691	12.64	2.928	1.6156	3.933		
11.17	2.415	49.65	2.612	11.677	3.573		
15.48	2.262	69.84	2.637	29.895	3.218		
26.74	2.157			52.935	3.26		
43.47	2.125						
69.47	2.144						

Fig. 36 shows $\ln(f)/\ln(1-\varphi)$ according to Re, and it shows φ-dependency like homogeneous systems, but also shows linearity depending on log (Re) when φ is equal to or higher than 2%. It represents packing or alignment of particle array affects the interaction force, and to evaluate the PIEP method for random arrays, we need to re-formulate the solid volume effect factors.

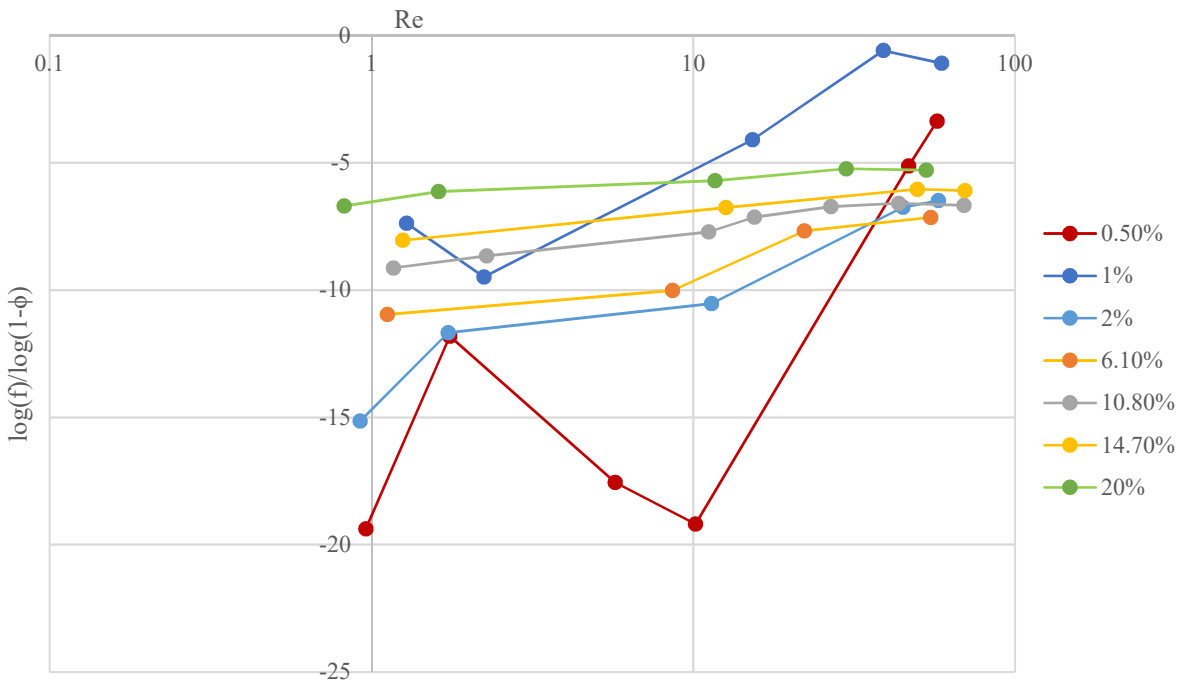


Figure 37. Solid fraction effect (random arrays) according to concentration and *Re*.

When φ is lower than 2%, the results are inconsistent with the tendency when the volume fraction is higher than 2%. Using the data between 2 and 20.0% and the same approach in the previous section, the *f* factor can be formulated as follows:

$$\frac{\log(f)}{\log(1-\varphi)} = a(\varphi)\ln(Re) + b(\varphi) \tag{23}$$

$$a(\varphi) = -0.67\ln(\varphi) - 0.79 \tag{24}$$

$$b(\varphi) = -\sqrt{-68.35\ln(\varphi) - 65.62} \tag{25}$$

The regression shows good fit, and the R2 scores for Eq. (23) are all above 0.89. Eq. (24, 25) show R2 scores above 0.99. Using Eq. (23-25), we apply the MLP, TCNN, and PIEP model to the cases in the previous section. Table 10 shows that Eq. (23-25) can provide better prediction than the results in the previous section for all cases without modifying the average drag force coefficient. If the solid volume fraction is lower than 2%, an error up to 10% should be tolerated, and to prevent this, it is necessary to develop another formulation for the *f* factors for low concentration conditions.

Table 10. Prediction results with MLP, TCNN and PIEP for high concentration systems with Eq. (23-25).

	Case 1	Case 2	Case 3
dispersity	poly-	mono-	poly-
conc. (%)	6.1	5.1	13.2
Re	1.82	2.25	49.1
f	1.89	1.77	2.53
MAPE (%)	10.3	6.1	12.2
R2 (%)	0.60	0.76	0.47

To sum up, the results show that the neural network-based model can predict high concentration, poly-dispersed systems with irregular particles. Besides providing a drag force model for irregular particles, this study has the great advantage of being efficient in terms of data collection. The current study has mainly required to collect single particle data to predict multi-particle system, and it makes this method significantly efficient compared to other machine learning studies for multi-particle systems, which require huge data collection [Big Data, 8(5), 431-449 (2020), Powder technology, 345, 379-389 (2019)]. The current study can be coupled with a collision model for CFD-DEM calculation, and our group published another machine learning-based collision model, which can be coupled for irregular particles [Chemical Engineering Science, 251, 117439 (2022)].

Outcomes

The current study provides a comprehensive drag force model for irregular-shaped particles in gas-solid flows. By applying the VAE, we have been able to describe the irregular particles from spherical harmonics with random numbers. Even though the VAE has the tendency to smoothen the particle shapes, the trained MLP has shown good predictions on interaction forces, especially on the drag force. Instead of setting orientation angles as input parameters for the MLP, we have included the angles in the latent vector of the VAE, so that we don't need to make additional combinations other than *Re* and shapes. The MLP has been evaluated with datasets, which have not been used for training in terms of incidence angle. It matches well with the PR-DNS results and allows us to predict the interaction forces of the irregular particles. Whereas existing models [Chemical Engineering Science, 192, 1157-1166 (2018)] are only able to predict the interaction forces of spherical or regular-shaped particles, the current model can cover a variety of particle

shapes. Furthermore, the existing models need additional data to obtain new parameters for the prediction, but the current approach can be applied to other data if we train the model with proper range of conditions. This single particle study was published in Powder Technology at the beginning of the second year of this project [Powder Technology, 392, 632-638 (2021)].

To expand the scope of this study to multi-particle systems, we have chosen the PIEP method to apply linear superposition. This approach has an advantage in terms of data collection. Because the superposition method does not require the data for multi-particles systems. Only small datasets have been collected to evaluate the PIEP model in this study, and we have used the same data from the previous, single particle study. To apply the PIEP method, velocity field in flow-direction and pressure field of single particle are required. Therefore, we have developed the TCNN model, which can predict the flow fields from the latent vector and *Re*. The TCNN model predicts the flow pattern well with low error and shows the asymmetric flows for oriented, irregular particles. TCNN has been utilized to calculate the undisturbed flow force, and low concentration systems up to 0.5% are calculated. Because the average drag force is affected by the concentration, we choose to study low concentration systems first to explore the eligibility of PIEP method. By assuming the solid fraction factor as unity, we have been able to predict the low concentration systems with mean absolute percentage error between 9 and 10% and R2 score between 0.56 and 0.62. Due to the nonlinear nature of the multi-particles flows and error from the machine learning-based models, the prediction is not perfect, but this method is much more efficient because the models can be trained with small number of datasets. This is because we put the PIEP, physics-based model at the end of the whole model structure, and it gives us much more flexibility in leveraging the neural network models. This study about low concentration systems was published in Chemical Engineering Science around the middle of the third year of this project [Chemical Engineering Science, 266, 118299 (2023)].

We have collected some datasets for higher concentration systems up to 21.5% and developed the model for the solid fraction effect. For homogeneous systems, the PR-DNS results have shown that the particle shapes do not affect much the solid fraction effect. However, it has shown that it is affected much by voidage, so we need to modify the existing model [International journal of multiphase flow, 20(1), 153-159. (1993), International Journal of Multiphase Flow, 39, 227-239 (2012)]. to satisfy the PR-DNS results. We assume the modified, regressed model can be applied

to polydisperse systems as well, and it shows moderate prediction on the systems with errors between 7 and 13% and R2 score between 0.48 and 0.7. Unlike the previous models, we only provide the model, which can predict the flows with *Re* numbers between 1 and 100 because the high concentration flows with *Re* below than 1 have shown complicated trends of the *f* factor, and computational cost is expensive for data collection. However, the above results show a high probability that a wider range of flows can be predicted with the PIEP model if there is an appropriate model for the solid fraction effect.

This study provides neural network-based, interaction force model including an unsupervised model for irregular-shaped particles in incompressible flows. From low to moderate *Re* numbers and solid volume fraction, we have been able to develop a cost-effective prediction model for drag force. This approach can be applied to a wider range of flow conditions and other force components including lift force and torque we have presented for single particle systems. This study is significant in that it increases the efficiency of data collection by utilizing machine learning-based models, which are considered a black box, to obtain intermediate physical values required for a physics-based model. For the future study, we can:

1. Expand the model by collecting more data of other *Re* number regime and lower *St* number,
2. Develop a TCNN model for *y*, *z*-direction velocity to model lifting force and torque, and
3. Resolve nonlinear behavior of PEIP model to improve the accuracy,

to broaden the scope of the current study. Because all the machine learning study has been conducted with TensorFlow package using Python, the model should be able to be incorporated in MFiX AI platform just with simple data transformation algorithm.