DISCLAIMER

Department of Energy/National Energy Technology Laboratory

DOE Grant Number: DE-FE0031904

Development and Evaluation of a General Drag Model for Gas-Solid Flows via Deep Learning

Principal Investigator
Dwayne McDaniel, Ph.D., P.E.
Associate Professor
(305) 348-6554

Submission Date: September 30, 2025

DUNS Number: 071298814

Recipient Organization:
Mechanical and Materials Engineering
Florida International University
10555 W. Flagler Street EC2100
Miami, FL 33174

Project Period: 08/1/2020 – 05/31/2025

Reporting Period End Date: 5/31/2025

Final Report

Signature of Submitting Official

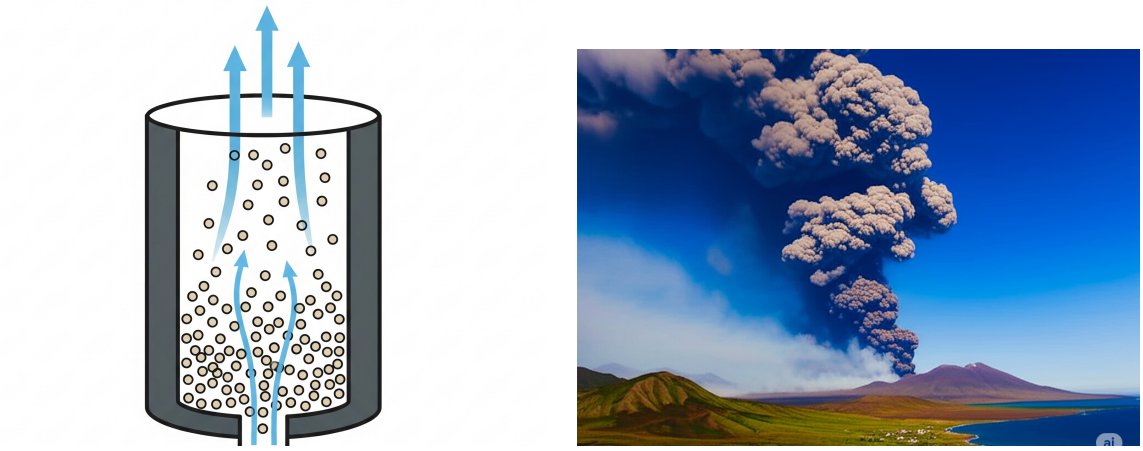
Abstract

This project presents the development and evaluation of a general drag model for gas–solid multiphase flows using deep learning techniques. A comprehensive database of more than 4,000 experimental and numerical data points for spherical and non-spherical particles was compiled, incorporating geometric features such as sphericity, aspect ratio, and orientation. Several predictive approaches—including traditional empirical correlations, machine learning, and deep neural networks—were benchmarked, with the proposed Drag Coefficient Correlation-aided Deep Neural Network (DCC-DNN) demonstrating superior accuracy. To account for particle–particle interactions, additional drag data were generated using CFD-based simulations of packed and fluidized beds, leading to the development of a retrained model capable of incorporating volume fraction effects. Integration of the trained model with the MFiX CFD solver was achieved using FTorch, enabling drag predictions during discrete element method (DEM) simulations. Validation against experimental data for single particles and fluidized beds confirmed the model's improved predictive ability, particularly for non-spherical geometries. While the model performed strongly under fluidized conditions, limitations remained in unfluidized regimes, suggesting a need for expanded datasets. Overall, this study demonstrates the feasibility of combining deep learning with physics-informed CFD to improve drag modeling for gas–solid flows, with promising implications for scaling multiphase simulations in industrial applications.

Contents

1 Introduction

Particles are encountered in a wide array of engineering and natural processes, ranging from industrial systems to environmental and geophysical phenomena. Among these, multiphase energy systems stand out due to their importance in energy conversion and environmental sustainability. For example, fluidized bed technologies—widely used in combustion and gasification—exploit the intense mixing and excellent heat transfer properties of particulate flows to convert solid fuels into syngas, thereby enabling cleaner energy production and reduction in carbon emissions. Beyond energy systems, particle-laden flows appear in



(a) Fluidized Bed (b) Volcanic Ash

Figure 1: Example of particle flow in the fluidized bed and volcanic debris transport.

applications such as sand ingestion in air-breathing jet engines, the ballistic transport of volcanic ash in the atmosphere, and sedimentation in rivers and ocean. Figure-1 showing two examples of the particle-fluid interactions in the industrial and natural process. These processes involve a large number of particles interactions which makes the associated flow problems highly complex and computationally intensive to model. To manage the computational burden, simplifications and modeling assumptions are often made, including the use of averaged continuum models and empirical correlations [12, 16]. At the core of accurately modeling these systems lies the need to predict the drag force acting on individual particles. This force plays a crucial role in determining particle dynamics such as settling velocity, residence time, and dispersion. The drag coefficient (C_D) is a non- dimensional number that relates the drag force on a particle to reference quantities like fluid density, particle velocity, and projected area. However, while considerable research has provided data and empirical models for C_D, its dependence on particle shape, flow regime, and fluid properties remains uncertain—especially for non-spherical particles at moderate to high Reynolds numbers. Understanding the drag force on non-spherical particles is inherently more complex than for spherical ones due to the geometric irregularities that influence the surrounding flow field. While the drag force on a smooth, isolated sphere in a uniform flow has been studied extensively and can be described with well-established correlations across various Reynolds number regimes, the situation is more nuanced for non-spherical particles. Their

asymmetric shapes lead to non-uniform pressure distributions and induce additional flow separation and wake effects, which, in turn, alter the magnitude and direction of the drag force. The drag coefficient (C_D) for non-spherical particles depends not only on the Reynolds number but also on shape descriptors such as aspect ratio, sphericity, and orientation relative to the flow. Sphericity (Φ), defined as the ratio of the surface area of a sphere (with equivalent volume) to the surface area of the particle, is commonly used to characterize the deviation from a spherical form. As Φ decreases (i.e., as the particle becomes less spherical), the drag generally increases due to enhanced flow separation and form drag. In addition to shape, particle orientation plays a pivotal role. Non-spherical particles, such as ellipsoids or cylinders, can experience significant variation in drag depending on their alignment with the flow. For example, a prolate spheroid aligned with the flow will experience less drag compared to the same particle oriented perpendicular to it. In turbulent or complex flows, particles continuously rotate, resulting in time-averaged drag that is orientation-dependent and difficult to capture with simplified models. Several empirical and semi-empirical correlations have been proposed to estimate the drag on non-spherical particles. These models often apply correction factors to standard drag correlations for spheres, incorporating particle shape parameters to better align with experimental observations. Notably, the empirical formulation by Haider and Levenspiel [18] introduces the concept of sphericity to account for deviations from spherical geometry. Similarly, Ganser [15] extended the Stokes drag model by incorporating shape effects through sphericity-based correction factors. Another key parameter in drag modeling for non-spherical particles is particle orientation, which can significantly influence drag, especially for particles with low sphericity. Hölzer and Sommerfeld [21] addressed this by introducing lengthwise and crosswise sphericity parameters that explicitly capture orientation effects. Their proposed correlation uses three sphericity values, offering improved accuracy over simpler models. However, the additional complexity in calculating these sphericities poses a challenge, particularly in simulations requiring real-time orientation tracking. Specifically, the lengthwise sphericity can be computationally demanding, as it involves integrating the projected sectional area along the particle's axis Despite these advances, modeling drag forces for non-spherical particles remains a challenge, particularly in multiparticle systems or flows with high turbulence intensity. The lack of generalizable, first-principles-based models compels researchers to rely on empirical data, which may not cover all particle shapes and flow regimes of practical interest. Consequently, continued efforts are directed toward improving drag predictions through high-fidelity simulations, machine learning techniques, and targeted experiments focusing on realistic particle geometries. We present a model capable of accurately predicting the drag coefficient (C_D) by leveraging multiple flow and particle features, while also maintaining robustness to previously unseen data. Neural networks have demonstrated strong potential in capturing the complex interplay between flow conditions and particle geometry, achieving high predictive accuracy for C_D values [6, 57]. Unlike prior studies that rely on fixed or limited datasets, this work introduces a more generalizable approach—termed the Drag Coefficient Correlation-aided Deep Neural Network (DCC-DNN)—which integrates an expanded set of geometric and flow-related features. Our key contributions are summarized as follows:

- We propose a novel DNN architecture that blends domain knowledge from empirical drag correlations with deep learning to construct a generalized drag prediction model.

- To the best of our knowledge, the dataset employed in this study is the most extensive to date for drag modeling, and includes a wider range of physically relevant features, as supported by prior literature.
- Through comprehensive ablation studies, we demonstrate that the combination of a large, diverse dataset, regularization techniques, and a meta-learning framework enables the proposed model to achieve high accuracy and generalize effectively to new, unseen cases.

2 Literature Survey

To date, there exists no general analytical solution for the drag coefficient of particles beyond Stokes' law, which is valid only for spherical particles in creeping flow conditions (very low Reynolds number, Re). At higher Re, and especially for non-spherical particles, drag coefficients must be determined experimentally. These experimental results have led to the development of shape-specific empirical correlations. Consequently, each particle shape typically has its own drag correlation, and a unified drag model that applies accurately across a wide range of shapes and flow regimes remains elusive.

In recent years, Machine Learning (ML) has emerged as a powerful alternative for problems where the relationships between inputs and outputs are complex or unknown. Bhattacharya et al. [3] showed ML models can effectively improves sediment transport model by incorporating measurement data. Yoon et al. [58] develops an Artificial Neural Network (ANN) model to predict time-dependent sediment suspension in the surf zone, using data from a large-scale laboratory experiment. Oehler et al. [32] compared Boosted Regression tree and ANN method to predict resuspension of sediment particles for nearshore study. Goldstein et al. [17] demonstrated ML model in predicting particle's settling velocity using the test data from the literature.

With the advancement of Deep Learning (DL), more sophisticated models have been developed to estimate drag-related parameters. He and Tafti [19] applied a single-hidden-layer Artificial Neural Network (ANN) to predict the drag coefficient of spherical particles using Reynolds number, volume fraction, and relative particle positions derived from numerical simulations. Their model, trained with the Levenberg–Marquardt algorithm, achieved a relative error below 10% for 52% of the predictions. Yan *et al.*[57] applied both Radial Basis Function (RBF) and Back Propagation Neural Networks (BPNN) to predict the drag coefficient of non-spherical particles using Reynolds number and sphericity as inputs. While the results were accurate for spherical particles, the prediction performance degraded significantly for particles with lower sphericity. In contrast, Chen and Li[6] employed a deeper neural network (three hidden layers) to predict particle drift velocity and drag force using volume fraction, slip velocity, and pressure gradient as inputs. Their model, trained using the ADAM optimizer, yielded satisfactory results across varying conditions. Balachandar [2] used a simple linear regression model to estimate the drag force in a fluidized bed, using Reynolds number and volume fraction. Although the approach was straightforward, it achieved only a moderate performance with an Re value of 0.64. Zhu [60] developed a three-layer DNN to predict the drag correction factor using features such as particle size,

pressure gradient, and volume fraction. Their model achieved a Mean Absolute Percentage Error (MAPE) of 9.85%.

More recent efforts have explored the use of advanced architectures and input representations. Hwang *et al.* [22] proposed a Convolutional Neural Network (CNN) framework to predict drag, lift, and torque coefficients of particles in low-Re regimes based on Particle-Resolved Direct Numerical Simulation (PR-DNS) data. To handle the complexity of particle shape and orientation while minimizing computational cost, a Variational Autoencoder (VAE) was employed to extract and compress geometric information for use in the force prediction model. Similarly, Luo *et al.* [26] developed a feedforward neural network trained on DNS data to predict drag forces using Reynolds number, volume fraction, and velocity/position fluctuations of particles. Their model was validated against empirical correlations and also yielded a simplified expression based on trained weights. Rushd *et al.* [39] performed a comparative analysis of multiple ML models for drag prediction and assessed the importance of different input features, highlighting the sensitivity of ML models to input selection and data range.

Despite these advancements, drag data for particles at finite volume fractions remains scarce, limiting most studies to single-particle drag predictions. Furthermore, the development of a general-purpose drag model that can adapt across a wide variety of particle shapes, orientations, and flow conditions—while maintaining robustness to previously unseen scenarios—has not yet been fully realized in the literature.

3 Data Collection

Machine learning model's efficacy depends on the database quality and size. The non-spherical particle's drag can depends on the various geometric parameter such as sphericity, aspect ratios, circularity, regular or irregular shapes etc. To understand this, it is important to gather data of various shapes from the litrature which can cover large number of shapes for a wide range of Reynold number. For this work single particle's drag data reported by the researchers using the experiments are considered. This experiments generally includes single non-spherical particles drop in a vertical cylinder filled with liquid and the drag forces were calculated using the particle's terminal velocity. Song et al. [46] used similar setup where a vertical glass chamber is filled with glycerine and particles with different shapes such as sphere, cube, and cylinder with various dimensions and densities are tested for the settling velocities. Additionally, the fluid properties such as density and viscosity are varied by mixing glycerine and water. Kale [24] conducted experiments on the coal particles which are of irregular shape. Their study uses wind tunnel testing where non-spherical particle is suspended magnetically that allow particle to freely rotate by the action of the aerodynamic moment. Aditionally, the report include key details such as particle volume and photograph of the settling orientations which can help in determining important geometric details. Baba and Komar [1] studied irregular grain particles by experimenting with larger glass particles of sphere and ellipsoid shape to mimic the grain partilces. The experiments reports settling behaviour of various ellipsoid. Chhabra et al. [7] studied the disk and plate-shaped particles with glycerine, silicon oil, carboxymethyl cellulose and Methocel HG90 to realize various kinematic conditions. The study reports highly non-spherical particles' drag

cofficients for the Reynolds number range 1 to 100. Pettyjohn and Christiansen [33] have summarized non-spherical particles drag data in a single chart. This paper also helped in referring to the original papers that have reported experimental data. Figure 2 shows the non-linear behaviour of the particles drag coefficient due to shape. Highly non-spherical particles are less predictive with conventional correlations. The prediction error with traditional correlations can be as high as 300%. Therefore a general drag model cannot be developed by traditional approach of curve fitting but it requires more sophisticated modelling such as machine learning models that can identify complex data patterns and accurately predict drag coefficients.

4 Machine Learning Model Development

Machine learning modelling requires a vectroization of the data with some features (inputs) that points to some unique output. These input features selections are the most important step in the model development, as it majorly decides the trained model's efficacy and usability. For the non-spherical particle drag modelling, it is clear that Reynolds number and sphericity are the two important feature as shown by Yen et al. [57]. However, their study has also shown very large errors for the low sphericity particles. Hence, it is important to identify additional features that could effectively demarcate the non-spherical particles. Another feature which was widely used by previous workers is a particle's aspect ratio (ratio of longest to shortest sides). To account for inertia, density ratio (solid to fluid) is also considered which indirectly accounts for the Stokes number. The Hölzer and Sommerfeld model [21] gave two new features, crosswise and lengthwise sphericity, to account for orientation. Crosswise and lengthwise sphericity are suitable features as these number depend on the particle projection area. Crosswise sphericity is the ratio of the projected area of the volume equivalent sphere to the projected area of the non-spherical particle perpendicular to the flow. Lengthwise sphericity is the ratio of the projected area of the volume equivalent sphere to the half non-spherical particle's total surface area minus the mean projected area of a non-spherical particle parallel to the flow. These two features by definition, requires real-time particle's orientation which makes it most challenging. Due to its limitations in the calculation, for this work, the crosswise and lengthwise sphericities are calculated for the settled particles. Hence, for a given particle's dimensions, all the sphericities would remain constant. The data collection and machine learning model development process is documented by Presa et al. [35] and readers are encouraged to refer it for full model details. Some of the key highlights of these articles are briefly discussed in the following subsections.

4.1 Particle Settling Orientation

The particle's settling orientation would require the total aerodynamic moment to be zero. This depends on a number of factor such as shape, density ratio, and Reynolds number. For a free particle, the Reynolds number remains low because it is carried by the fluid, resulting in a low relative velocity. The settled orientation calculation would requires flow resolving numerical model that would rotate a particle's body in the three dimensional space. However, this simulations can be time consuming and may not be suitable for large-scale

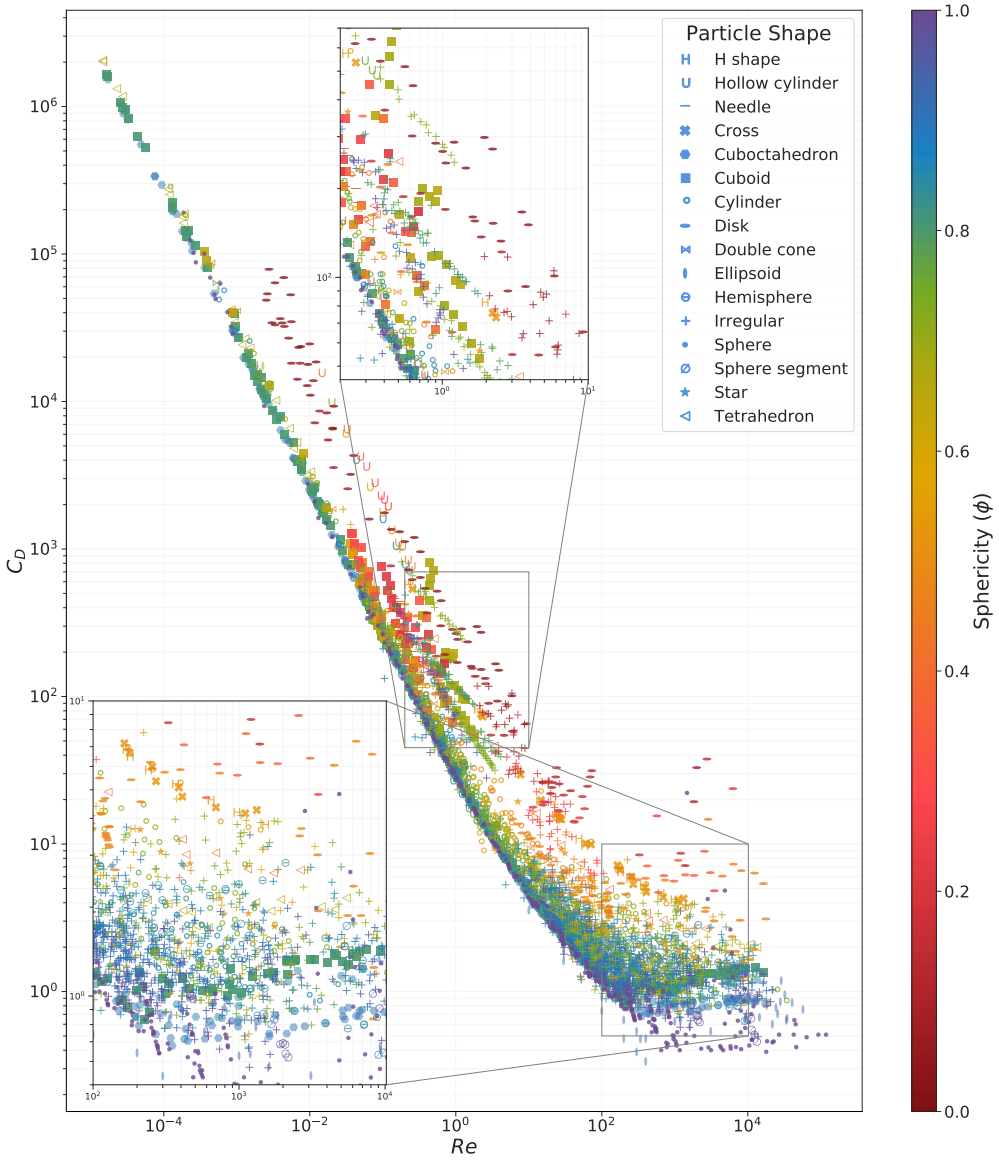
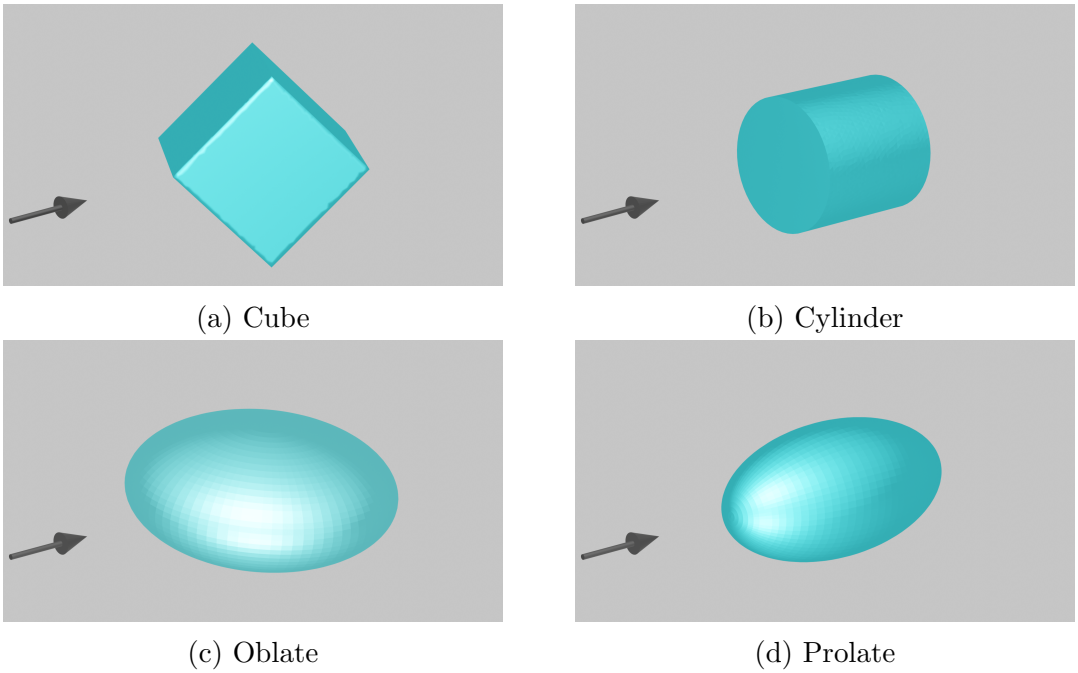


Figure 2: Compile drag data on the single chart for various regular non-spherical particles. Marker shape designating various regular particles shape and its color shows sphericity.[35]



(a) Cube (b) Cylinder

(c) Oblate (d) Prolate

Figure 3: Settling orientation of a regular particles calculated using the panel method, the black arrow is showing the relative fluid's velocity direction. [35]

database generation. Hence, a simpler approach that uses panel method type calculations to predict aerodynamics forces is employed. The particle position is kept fixed but it is allowed to rotate under the action of aerodynamic moments. The particle is subjected to very low Re flow which ensures flow is always attached to the particle surface. Flow velocity on each particle surface is estimated based on the surface normal towards or away from the flow. A diverging surface that pushes the fluid outward causes acceleration, whereas a converging surface that directs the flow inward causes deceleration. Additionally, the boundary layer behavior is assumed to follow that of a flat plate (Blasius solution), which allows direct estimation of the local shear stresses.

Figure-3 shows the sample result from the panel method for a regular particle's settling orientation. The cubic particle is oriented in a diagonal position, whereas the ideal cylinder is oriented lengthwise. The oblate particle settles in a diametric position whereas prolate settle in a lengthwise. This behavior of prolate and oblate matches with the prediction of Sheikh et al. [44]. The settles orientation projects a minimal area into the flow, which will minimize the drag coefficient. However, this method does not account for the wake effect, which will create wobbling and may not allow the particle to settle. The rate of response of a given non-spherical particle is a function of density ratio and a Reynolds number. This simple method provides a heuristic understanding of the non-spherical particle's behavior in a forced flow field such as a fluidized bed.

Table 1 summarizes all the data collected from various experimental and review papers on regular and irregular non-sphrical particles totaling more than 4000 data points. The table also shows important features such as sphericities and aspect ratio (AR) which will be discussed in the next section.

Table 1: Summary of the collected data.

No.	Source	Data Points	Shape	AR	Re	ϕ	R_ρ	C_D	$\phi_\perp$	$\phi_\parallel$
1	Baba & Komar, 1981* [1]	72	Irregular	1.5 - 6.1	0.04 - 2.01	0.58 - 1.00	2.0	17 - 790	1.4-10.5	1.1-4.0
2	Chen & Li, 2020 [6]	23	Regular	1.0	1472 - 11988	1.00	7.8	0.43 - 22.20	1	1
3	Smith & Cheung, 2003 [45]	998	Irregular	1.1 - 19.0	7.0 - 2645	0.34 - 1.00	2.6	0.7 - 10	1.1-32.2	1.0-5.6
4	Chhabra *et al.*, 1999 [8]	20	Regular	5.5 - 15.8	0.2 - 0.9	0.34 - 0.61	8.0 - 9.1	66 - 451	0.2-0.4	0.2-1.5
5	Corey, 1949* [11]	39	Irregular	1.0 - 2.7	270 - 2310	0.54 - 1.00	2.7	1.0 - 3.2	0.7-2.9	0.002-0.01
6	Dioguardi *et al.*, 2018 [13]	340	Irregular	1.1 - 3.2	0.03 - 9822	0.42 - 1.13	1.0 - 3.0	0.6 - 1860	0.7-3.5	0.6-1.9
7	Johnson *et al.*, 1987* [23]	62	Regular	1.0 - 48.1	0.04 -0.11	0.29 - 0.81	2.8	224 -1277	0.7-8.1	0.004-0.03
8	Kale, 1987 [24]	132	Irregular	1.0 - 2.1	0.11 - 4.05	0.71 - 0.81	1000 - 1666	32 - 999	0.9-1.1	0.5-0.6
9	Komar & Reimers, 1978 [25]	51	Irregular	1.6 - 19.4	0.14 - 2.59	0.8 - 1.24	2.7	28 - 427	1.5-132	1.2-22
10	Madhav & Chhabra, 1995 [28]	64	Regular	1.25 - 39.5	0.12 - 7.1	0.37 - 0.69	1.25 - 7.68	8.2 - 812	0.5-15	0.5-1
11	Malaika, 1949* [30]	132	Regular	1.0 - 4.0	0.0001 - 1254	0.49 - 1.00	8.9 - 9.5	0.5 - 280392	0.7-6.3	0.4-2.8
12	McKay *et al.*, 1988* [31]	103	Regular	1.0 - 4.9	9.6 - 5716	0.68 - 0.87	1.1 -2.2	0.95 - 10.95	1.0-3.8	0.7-1.1
13	Melkebeke *et al.*, 2020 [50]	140	Irregular	1.2 - 318	1.92 - 291	0.04 - 0.97	1.1 - 1.4	1.2 - 118	1.0-1299	0.7-7.6
14	Pettyjohn & Christiansen, 1948* [33]	577	Regular	1.0 - 1.4	0.00001 - 22630	0.67 - 1.00	1.3 - 12.8	0.41 - 2057343	0.85-1.0	0.76-1.0
15	Riazi & Türker, 2019 [37]	43	Irregular	1.0 -5.2	83 - 1814	0.73 - 1.00	1.6 -4.6	0.57 - 3.52	1-5.5	1-2.6
16	Rong *et al.*, 2015 [38]	28	Regular	2.0 - 2.5	1 - 400	0.89 - 0.91	1.6	0.34 -34	1.8-2.5	1.1-1.6
17	Schmiedel, 1928* [40]	57	Regular	1.0 - 136	0.05 - 73	0.1 - 1.00	2.2 - 17.4	3 - 768	1-5.3	1-3.3
18	Schulz, 1954* [41]	79	Irregular	1.26 - 3.8	3 - 216	0.60 - 1.00	2.57 - 2.7	1.03 - 16.22	1.1-4.1	0.9-2.1
19	Sharma & Chhabra, 1991 [42]	44	Regular	1.0 - 4.0	0.0002 - 1.4	0.69 - 1	1.2 - 5.6	27 - 190596	0.5-2.1	0.44-1
20	Sheaffer, 1987* [43]	24	Regular	16 -92	0.09 - 0.41	0.23 - 0.46	3.0	81 - 765	4.9-11.2	0.8-1.3
21	Song *et al.*, 2017 [46]	336	Regular	1.0 - 10.0	0.002 - 90	0.47 - 1.00	2.1 - 6.5	1.5 - 9326	0.9-3.8	0.7-1.4
22	Squires & Squires, 1937* [47]	72	Regular	158 - 1602	0.002 - 9.24	0.02 - 0.09	3.1 - 3.3	55 - 78831	0.01-0.04	0.01-0.04
23	Stringham *et al.*, 1969* [48]	244	Regular	1.0 - 10.1	6 - 118000	0.47 - 1.00	1.1 - 15.1	0.4 - 14.8	0.3-3.3	0.3-1.7
24	Tran-cong *et al.*, 2004 [49]	73	Regular	1.7 - 7.0	0.22 - 2625	0.41 - 0.52	2.0	1.7 - 579	-	-
25	Wang *et al.*, 2009 [52]	72	Regular	1.1 - 2.0	27 - 13215	0.84 - 0.99	1.5 - 1.6	0.46 - 5.8	-	-
26	Weidman & Lasso, 1986 [53]	50	Regular	1.0 - 8.0	0.014- 2.1	0.26 - 0.87	1.4	13 - 15622	-	-
27	Wilde, 1952* [54]	230	Regular	1.0 - 5.7	4.7 - 21642	0.6 - 1.0	2.4 - 3.3	0.49 - 9	1-7.9	0.8-3.2
28	Willmarth *et al.*, 1964 [55]	21	Regular	8 - 192	328 - 6261	0.08 - 0.53	1.0 - 21.2	3.7 - 37	0.04-0.3	0.04-0.3
29	Xie & Zhang, 2001 [56]	28	Regular	3.4 - 88	0.07 - 0.69	0.19 - 0.76	8.8 - 10	148 - 598	0.1-12	0.6-2.0

Total	
Regular Particles	2212
Irregular Particles	1990

*Particle shape and settling velocity are retrieved from Breakey *et al.*, 2018 [4]. Other parameters including Re and C_D are calculated by our team to ensure consistency with other data.

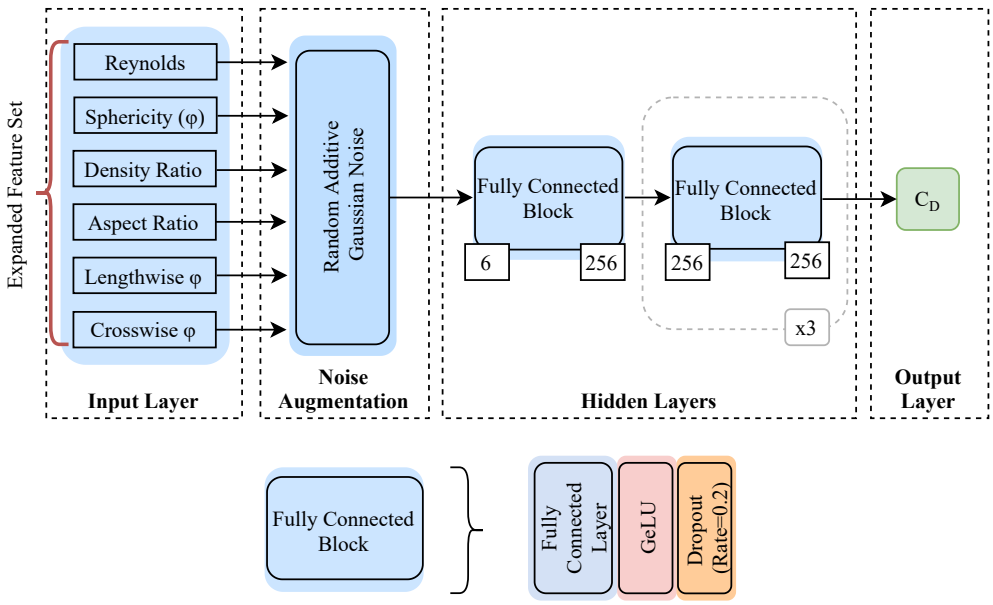


Figure 4: Architecture of the proposed single-model DNN. The input and output dimensions of each block are shown in the boxes. [35]

4.2 Deep Neural Networks

Neural networks are a class of machine learning models inspired by the biological mechanisms of the brain. They have received considerable attention in recent years due to their

outstanding performance in areas such as image classification, speech recognition, and natural language processing [34]. Deep learning, in particular, relies on hierarchical feature extraction, where multiple layers automatically learn representations from raw input data. Each layer transforms the data into a new representation, enabling increasingly linear separability of the features. Figure 4 presents the proposed DNN model designed to predict C_D from an expanded feature set. During training, a noise augmentation layer introduces Gaussian noise to the input features, improving robustness to measurement errors and uncertainties in the data. The network architecture consists of four fully connected blocks, each containing a dense layer, an activation function, and a dropout layer for regularization.

4.2.1 Activation Function

The activation function determines how each neuron transforms its input before passing it to subsequent layers. Our model employs the Gaussian Error Linear Unit (GELU), a relatively recent nonlinear activation function, that has shown superior empirical performance. Unlike the widely used Rectified Linear Unit (ReLU), which thresholds values based on their sign, GELU weights inputs by their magnitude. Prior studies demonstrate that GELU often outperforms ReLU and Exponential Linear Unit (ELU) across diverse datasets, making it a strong candidate for this work [20].

4.2.2 Loss Function

Training a neural network involves minimizing a loss function, which quantifies the discrepancy between predicted and true values. Loss functions not only measure model accuracy but also guide the optimization of trainable parameters. For the regression task considered here, three loss functions were evaluated for their robustness to noise and outliers: Mean Absolute Error (MAE), Log-Cosh, and Huber loss. While all three minimize the average deviation between predictions and targets, each exhibits strengths under different conditions, making them well-suited candidates for this problem.

4.3 Model Regularization

Regularization techniques aim to mitigate overfitting by improving the model's generalization ability without significantly increasing training error. In this work, we use two strategies to ensure the model learns the underlying relationships in the data rather than memorizing noise. Training on small or irregular datasets can lead to overfitting, as the model may capture spurious details rather than generalizable patterns. To address this, zero-centered Gaussian noise [36] is added to the input features during training, introducing small perturbations that improve robustness to experimental uncertainties. Additionally, a dropout layer is applied after each fully connected block (Figure 4). Dropout randomly removes neurons during training [34], forcing the network to rely on multiple pathways rather than memorizing specific activations. Together, noise augmentation and dropout provide complementary mechanisms to reduce variance, yielding a more generalized and reliable predictive model.

4.4 Drag Coefficient Correlation-aided Deep Neural Network (DCC-DNN)

Classical drag correlations in the literature are primarily derived for spherical particles and are often valid only within narrow ranges of shape and flow conditions. To overcome these limitations, we propose the Drag Coefficient Correlation-aided Deep Neural Network (DCC-DNN), which incorporates empirical correlations into a deep learning framework using meta-learning techniques. Specifically, four well-established correlations [18, 9, 59, 21] are integrated into the DNN to enhance prediction accuracy. Two meta-learning approaches are explored: Stack Generalization (SG) and Mixture of Experts (MoE). These methods allow the model to combine insights from both empirical correlations and the DNN, dynamically adapting to different flow regimes and particle shapes.

4.4.1 Stack Generalization

Stacked Generalization (SG), or stacking, is an ensemble learning approach that combines multiple predictive models to improve accuracy. The core idea is that aggregated predictions, which leverage the complementary strengths of individual models, often outperform any single predictor. As illustrated in Figure 5, SG consists of two hierarchical levels. The first level includes several base models, which in this study are the Deep Neural Networks (DNNs) and traditional correlation-based models. Their predictions $f(x)_i$ serve as inputs to the second level, known as the meta-model. The meta-model learns how to optimally weight the contributions of the base models by assigning constant coefficients w_i through linear regression, as expressed in Equation 1. To prevent overfitting, a separate data fold is used when training the meta-model. This methodology has been applied in related work, such as Rushd *et al.* [39], who explored stacking to integrate multiple classifiers for predicting terminal settling velocity. A limitation of SG, however, is that it only learns static weights, which may restrict its ability to generalize across diverse conditions.

$$H(x) = \sum_{i=1}^{N} w_i \cdot f(x)_i \tag{1}$$

4.4.2 Mixture of Experts

The Mixture-of-Experts (MoE) framework [27] extends the ensemble concept by allowing dynamic rather than static weighting of models. Unlike SG, which assigns fixed weights, MoE assumes that each predictor excels in specific regions of the input space. For example, while DNNs generally perform best across the full dataset, certain empirical correlations may yield superior accuracy at low Re, where their experimental basis lies. MoE employs a gating network to determine the relative importance of each expert for a given input. Both the raw features and the outputs of the base models are provided to this gating network, which produces a set of weights $G(x_i)$ corresponding to each expert $f(x_i)$. These weights are passed through a Softmax activation σ, ensuring that they sum to one and can be interpreted

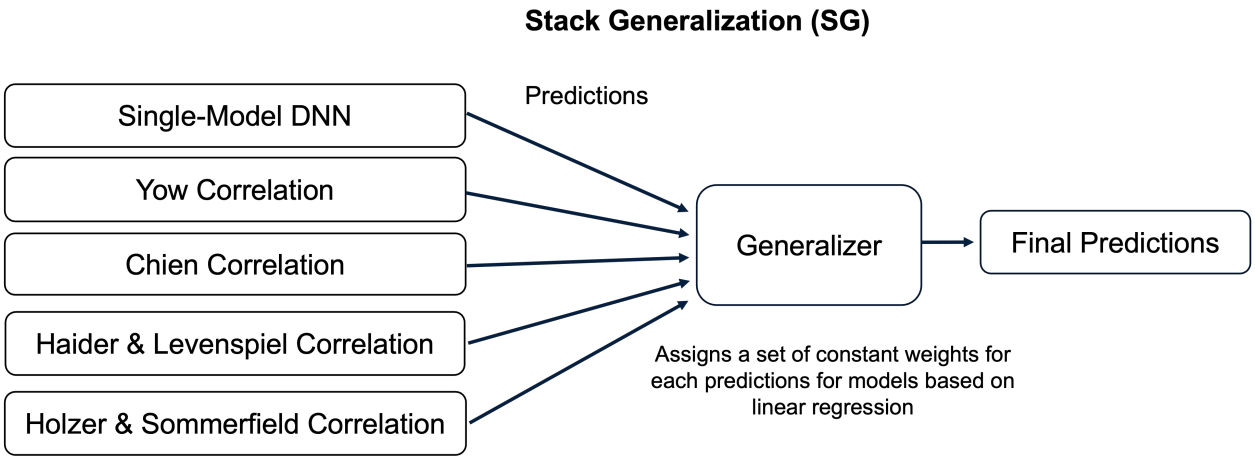


Figure 5: Architecture of Stacked Generalization: predictions from selected correlations and the single-model DNN are combined by the meta-model to produce the final output [35].

as probability-like confidence scores. The final prediction is then obtained as a weighted sum of all expert outputs, as shown in Figure 6 and defined in Equations 2 and 3.

$$H(x) = \sum_{i=1}^{N} \sigma(G(x_i)) \bullet f(x)_i \tag{2}$$

$$\sigma(G(x_i)) = \frac{e^{G(x_i)}}{\sum_{j=1}^{N} e^{G(x_j)}} \tag{3}$$

4.4.3 Data Preparation

Since both Re and C_D exhibit skewed distributions, a logarithmic transformation is applied to approximate normality before normalization. All features are standardized by removing the mean and scaling to unit variance using statistics computed from the training set. This ensures consistent feature scaling across training and evaluation [34]. To evaluate performance robustly, a 10-fold cross-validation scheme is employed. The dataset is partitioned into $k = 10$ subsets, with $(k-1)$ folds used for training and the remaining fold for validation. Importantly, folds are constructed to prevent overlap of experimental sources between splits, ensuring that validation evaluates true generalization across experiments. Missing geometric features (cross-wise and length-wise sphericity) are handled using iterative imputation [5]. Extra-Trees regression [10] predicts missing values based on correlated geometric attributes such as overall sphericity, aspect ratio, and principal-axis lengths. The procedure is iterative, updating imputed values across passes to refine estimates and improve feature completeness.

4.4.4 Performance Evaluation

Model performance is assessed using multiple regression metrics to provide a comprehensive evaluation and allow comparisons with both DNN-based and empirical models. For each metric, y_i denotes the true value, $\hat{y}_i$ the predicted value, and N the number of samples.

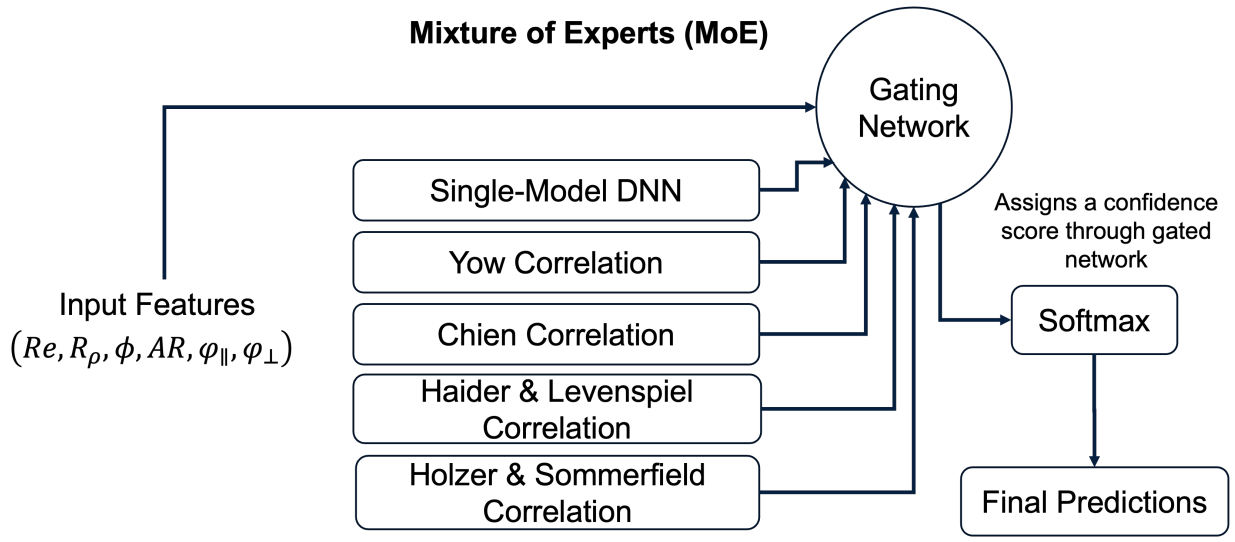


Figure 6: MoE architecture: Final prediction is the confidence score weighted prediction from each expert. [35]

- **Root Mean Squared Error (RMSE)**: Measures the standard deviation of residuals, indicating how closely predictions align with observed values.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{4}$$

- **Mean Relative Absolute Error (MRAE)**: Captures average relative error, though it is highly sensitive to small denominators and outliers.

$$MRAE = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i} \tag{5}$$

- **Normalized Residual Sum of Squares (NRSS)**: Quantifies residual variance relative to the target values, with smaller values indicating better fit.

$$NRSS = \sum_{i=1}^{N}\frac{(y_i - \hat{y}_i)^2}{y^2} \tag{6}$$

- **Sum of Squared Log Error (SSLE)**: Provides robustness to outliers by evaluating squared differences in logarithmic space.

$$SSLE = \sum_{i=1}^{N}\big(\log(y_i) - \log(\hat{y}_i)\big)^2 \tag{7}$$

- **Coefficient of Determination ($\mathbf{R^2}$)**: Measures the proportion of variance in the target explained by the model.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{8}$$

Table 2: Performance measures of predicting C_D using cross-validation among traditional correlations (TC), well-known machine learning (ML) methods, and different deep learning (DL) configurations including the proposed DCC-DNN, in which S : $< Re, \phi, \phi_\perp, \phi_\parallel, AR, R_\rho >$, represents the full feature set.

Type	Method	Input Features	RMSE	MRAE	NRSS	SSLE	R^2
TC	Haider & Levenspiel, 1989 [18]	$< Re, \phi >$	37.93±12.13	30.08±11.08	56.06±54.48	19.52±21.10	0.7146±0.38
	Chien, 1994 [9]	$< Re, \phi >$	49.46±12.29	38.59±9.06	92.21±90.81	26.14±29.03	0.6259±0.40
	Yow *et al.*, 2005 [59]	$< Re, \phi >$	200.91±92.01	164.10±85.42	2001.56±2686.58	31.47±34.76	-1.5375±7.44
	Holzer & Sommerfield, 2008 [21]	$< Re, \phi, \phi_\perp, \phi_\parallel >$	55.13±28.29	46.26±24.90	111.61±127.72	46.39±51.91	0.1171±1.71
ML	Random Forest [10]	S	48.52±12.73	34.70±9.08	121.77±190.37	19.85±20.62	0.5426±0.33
	Gradient Boosting [10]	S	45.12±14.10	33.01±8.21	108.76±152.78	18.23±20.01	0.5891±0.31
DL	Baseline	$< Re, \phi >$	36.38±9.72	28.59±6.16	62.14±87.10	11.84±13.33	0.7971±0.22
		S	37.09±8.92	27.05±5.18	72.66±116.09	72.66±116.09	0.7822±0.20
	Single-Model DNN	$< Re, \phi >$	31.54±12.89	23.60±9.41	42.62±49.90	8.31±9.76	0.7508±0.30
		S	26.63±10.63	17.43±6.29	40.83±62.95	6.96±10.50	0.8118±0.25
	DCC-DNN (SG)	S	44.66±12.47	33.92±7.29	76.41±81.48	13.22±12.52	0.8150±0.19
	DCC-DNN (MoE)	S	**25.98±10.18**	**17.05±6.03**	**38.00±58.17**	**6.76±10.02**	**0.8569±0.20**

4.4.5 Proposed Model Comparison

Table 2 presents the performance of different predictive approaches, including traditional correlations (TC), machine learning (ML), and deep learning (DL). For each method, the reported values correspond to the mean and standard deviation of the metrics defined in Section 4.4.4, obtained through 10-fold cross-validation. The table also specifies the input features used by each model. Comparisons with previously established correlations highlight the improvements achieved by the proposed method. Overall, DL-based models deliver superior accuracy, particularly when trained on the full feature set (S), underscoring the importance of richer input information for predicting C_D. Simply put, the more informative data the model is exposed to, the better its predictive performance. Unlike ML and DL models, TC methods are not trained on the dataset. Their results therefore reflect direct evaluation on the cross-validation test sets, ensuring a fair basis for comparison. Among ML techniques, Random Forest (RF) and Gradient Boosting (GB)—two widely used tree-based algorithms—show competitive performance, consistent with prior studies [10]. However, their limited generalization capacity becomes evident when applied to unseen data. The DL results include several configurations, beginning with the baseline model (a single hidden layer of 64 ReLU-activated neurons optimized using Mean Squared Error loss). Even this simple architecture outperforms RF and GB, though it shows reduced robustness when learning from the expanded feature set S. **On the performance of ensemble strategies:** The DCC-DNN model performs worse under Stack Generalization (SG). Although both SG and the Mixture of Experts (MoE) rely on meta-learners to combine predictions, their strategies differ. SG assigns static weights to each model, limiting adaptability across varying input spaces. By contrast, the MoE framework dynamically adjusts the contribu-

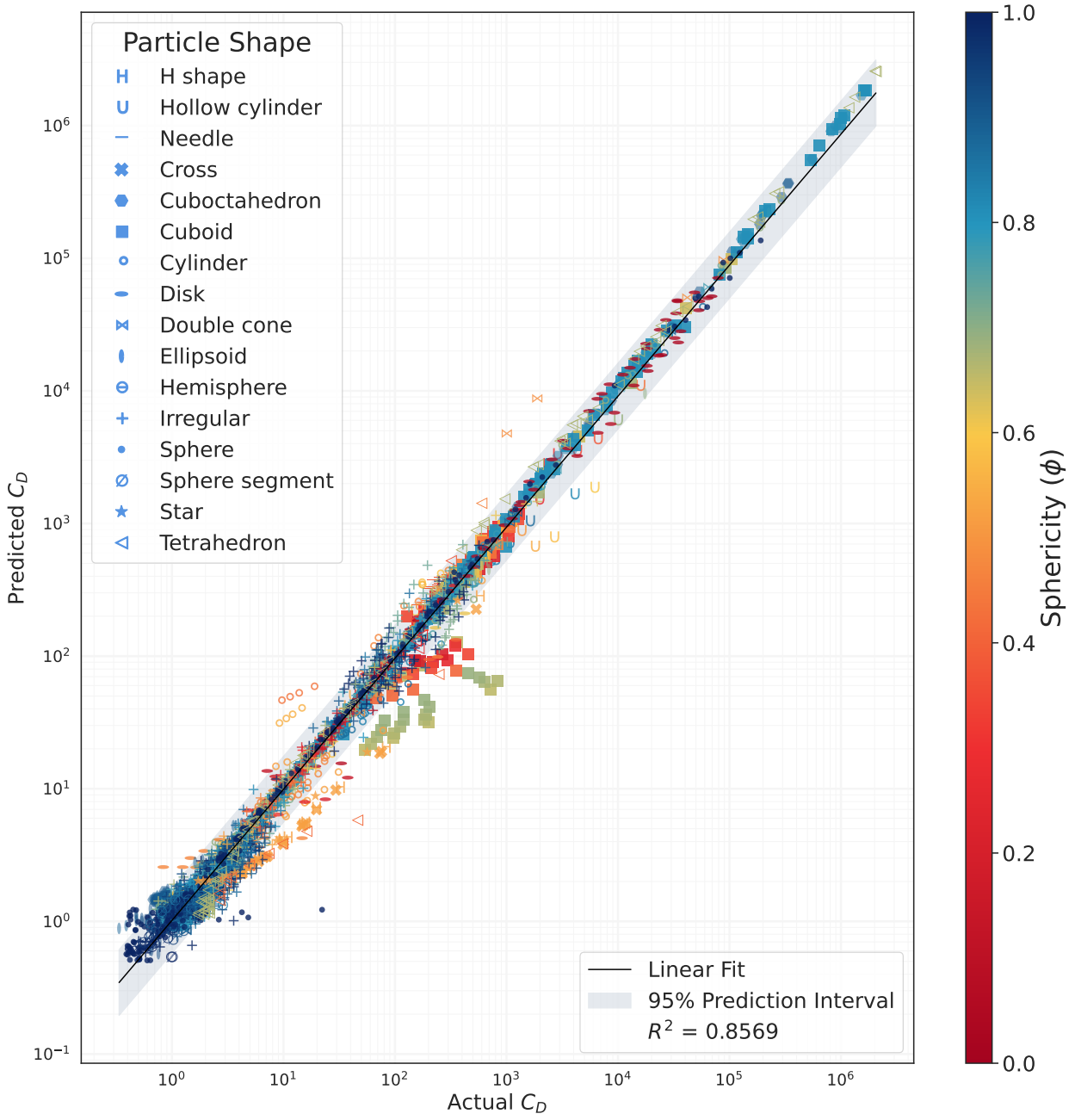


Figure 7: Illustration showing the predicted accuracy of the proposed model. The plot's marker style and color representing the shape and sphericity of the particle samples follow the format described in Figure 2.

tion of each expert, enabling better specialization across feature domains. This conditional weighting accounts for the superior performance of MoE compared to SG. **Prediction accuracy and uncertainty:** Figure 7 illustrates predicted versus actual C_D values. The clustering of points around the regression line demonstrates strong agreement between predictions and observations. The shaded 95% prediction interval conveys the expected range

of predicted values for a given true C_D. Nonetheless, the figure also reveals limitations in predicting certain particle geometries (e.g., cylinders, cuboids, and disks), which merit further investigation. The proposed DCC-DNN (MoE) model achieves the highest coefficient of determination, $R^2 = 0.8569$, explaining nearly 86% of the variance in the test data. At the same time, it yields error magnitudes that are substantially lower than those of all other tested methods, as summarized in Table 2.

5 CFD Study for Volume Fraction Data

The drag forces on non-spherical particles in a fluidized bed setup is influenced by neighboring particles. The close packing of particles can significantly increase flow resistance due to restricted flow passages which can amplify drag forces by several times. For this study, solid volume fraction or porosity, is used to quantify the local bed packing which can be used as an additional variable for drag modelling. Unfortunately, drag force information in such situations does not have sufficient experimental data. In addition, experimental information primarily reports average pressure drops and does not provide local drag forces due to measurement challenges. To address this, numerical simulations were used to generate drag force data in a packed bed arrangement with various solid volume fractions. In the first approach, spherical particles are arranged in a BCC lattice structure with at least 10 layers along the flow and 5 layers in the lateral directions as shown in Figure 8. The pressure drop along the fixed bed is compared against Ergun's correlations. Additionally, the velocity distribution in the bed is compared against nuclear measurement reasoning (NMR) measurements reported in the literature. The established methodology is then used for various non-spherical particles to generate data for a range of volume fractions. 350 total datapoints have been generated which are uniformly distributed along the Re and volume fraction vector. The result provided a reasonable pattern with the volume fraction for different shapes. The detailed numerical study is documented by Mahyawansi et al.[29]

Figure 9 shows the flow field in the fixed bed of non-spherical particles (spheroid and tetrahedron). The left and right image shows the impact of particle shape on the overall flow resistance offered by a bed. The tetrahedron particles, having sharp corners, results in stronger recirculations, which can give larger drag coefficients, whereas streamline particles such as spheroids, have low flow resistance. The flow resistance can increase or decrease depending on the packing arrangement. Figure 10 shows the compilation of all the non-spherical particles studied using this technique. The color pattern clearly shows a linear pattern in the log of drag coefficient verses the solid volume fraction for various particle sphericity.

A total 350 data points were generated using this method which aided in building an additional DNN model that could take input as single particle drag and estimate amplified drag coefficients for a given solid volume fraction. The volume fraction based DNN model was found to be highly accurate for the fluidized bed, however, it overpredicted drag forces for the unfluidized bed, i.e, for higher solid volume fractions. This error is probably due to low training data. The high turnaround time makes it practically challenging to generate a large number of data points with the current CFD methodology. Hence, another approach is designed where non-spherical particles are randomly arranged similar to the fluidized region

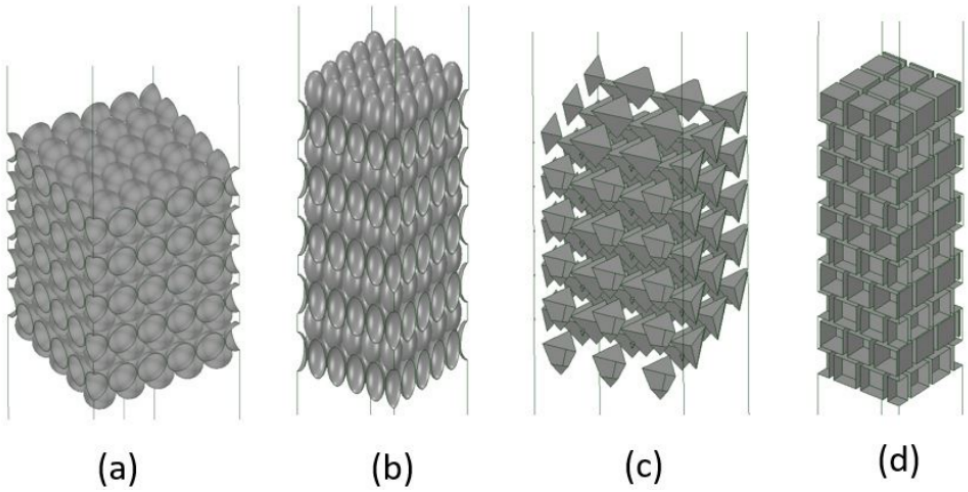


Figure 8: Non-spherical paraticles arranged in a BCC structure for numerical simulations. a) Spherical particles, b) Spheroid, c) Tetrahedron and d) Cube.

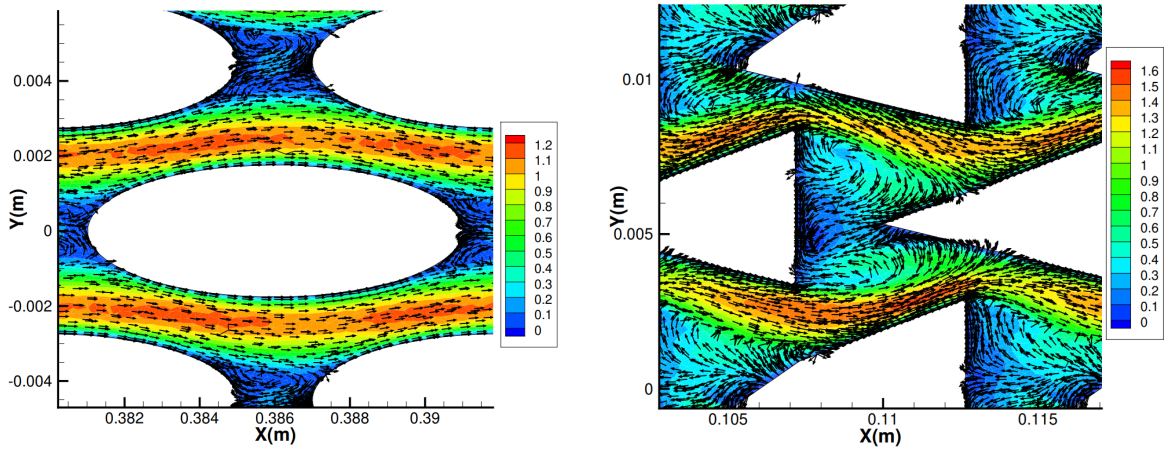


Figure 9: Resolved flow field in the bed of spheroid (left) and tetrahedron (right) particles.[29]

using the physics solver in the Blender animation tool.

5.1 Randomized Particle Arrangement

Figure 11 shows a sample of a random arrangement of cylinderical particles in a realistic situation similar to a fluidized bed. These randomly arranged particles can have a wide range of volume fraction for a given case. Hence, in a single simulation, multiple data points can be generated. Figure 12 shows the drag data generated using the randomized particle arrangement. Approximately 3000 new data sets were generated using this approach which is used for retraining the volume fraction-DNN.

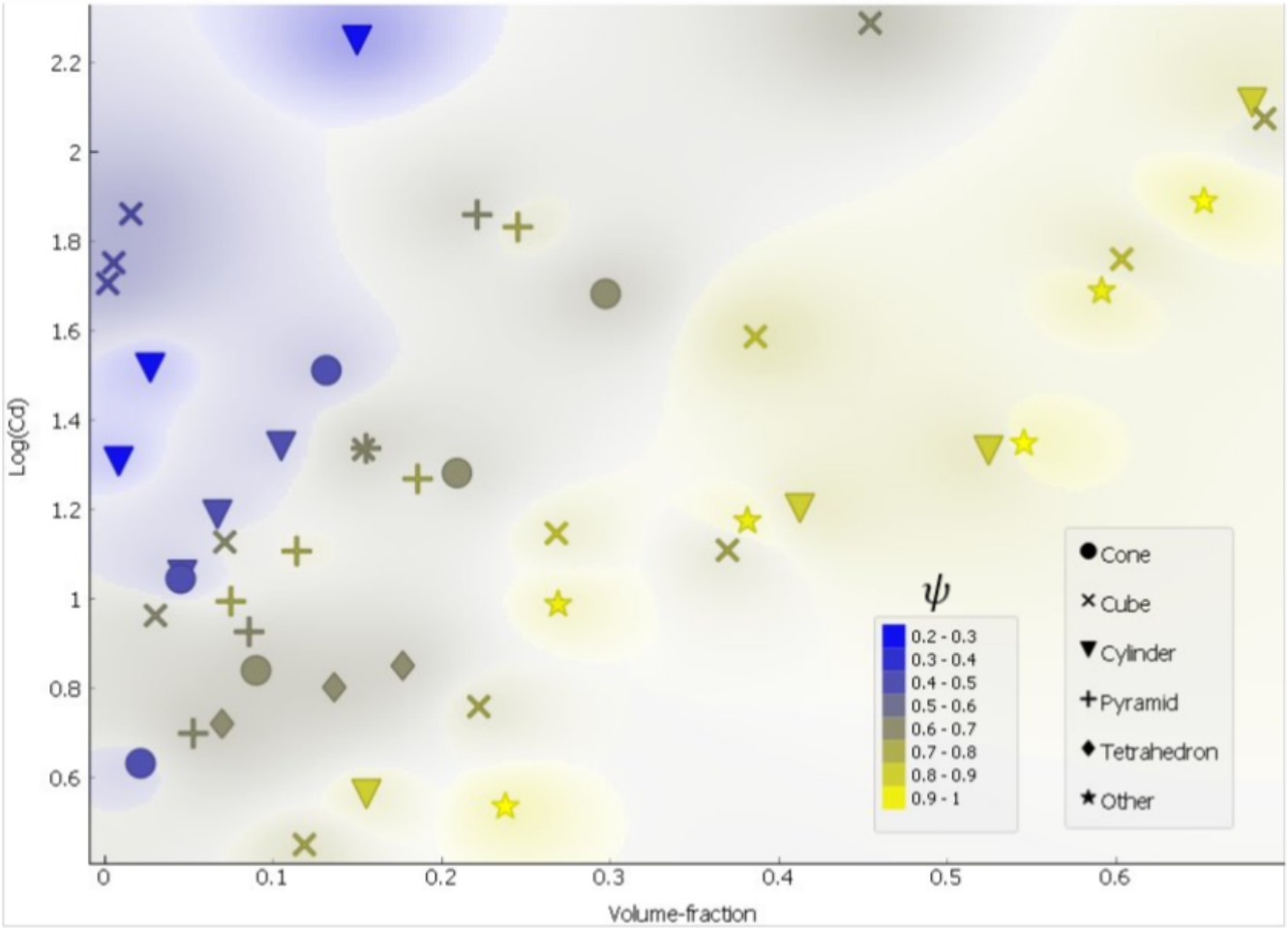


Figure 10: Compiled volume fraction data generated from the PR-DNS.

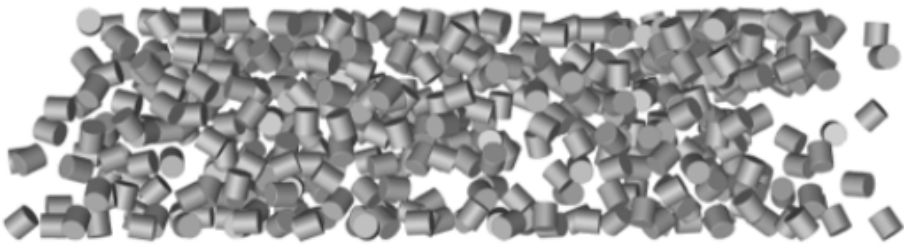


Figure 11: Randomly arranged cylindrical particle for the numerical simulations.

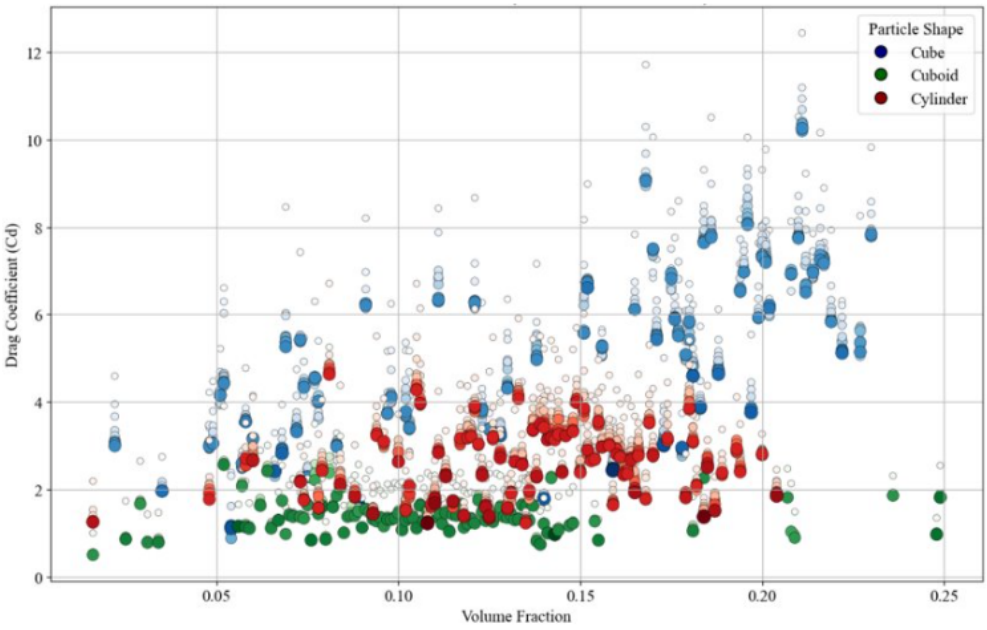


Figure 12: Randomly arranged cylindrical particle for the numerical simulations.

6 Integration with CFD

The trained drag model for CFD applications requires integration with a CFD package. The National Energy Technology Laboratory's (NETL) CFD tool, MFiX—developed primarily for particle flow simulations to meet energy industry requirements—is an open-source software package that allows users to modify or add functions for custom applications. The package is written in `FORTRAN`, whereas the trained drag model is implemented in Python, as most machine learning libraries are available in Python. This presents a major challenge in compiling the drag model with the CFD package. This section briefly discusses the integration method used in this study. The MFiX software package offers various particle simulation models, such as:

- **DEM (Discrete Element Model)**: Tracks all individual particles in the domain, assuming spherical shapes and including collision models.
- **CGP (Coarse-Grained Particle)**: Tracks groups or clusters of particles to reduce memory requirements, making it suitable for full-scale industrial problems.
- **PIC (Particle-in-Cell)**: Treats groups of particles as computational parcels, omitting some particle dynamics such as collisions.
- **GSM (Glued-Sphere Model)**: Similar to DEM, but allows for non-spherical particle shapes by gluing spherical particles together.
- **TFM (Two-Fluid Model)**: Treats the solid phase as a fluid, solved using an Eulerian approach.

To minimize modeling limitations, the DEM model is found to be the most suitable for this study.

6.1 Features Collection in the DEM Loop

In the DEM model, particles are explicitly tracked after each fluid time step. The particle time step is dynamically calculated to satisfy the collision model requirements. This time step is typically on the order of 1×10^{-6} s, whereas the fluid time step is approximately 1×10^{-4} s. Based on this, the drag model assumes that the drag coefficient remains constant during DEM time steps. Therefore, the drag model is invoked before the DEM loop to update the drag coefficient for all particles. Figure 13 shows the block diagram for drag model integration. Before every DEM loop, a user-defined subroutine is called to collect each particle's data, such as relative velocity and volume fraction. Other parameters, such as sphericity, crosswise sphericity, and lengthwise sphericity are defined during initialization via a text file and stored as common block (global) variables. The list of particle data required by the machine learning model is made available to the drag model subroutine. However, since the new drag model is written in Python, it is not directly readable in `FORTRAN`.

6.2 FTorch

The drag model requires the PyTorch Python library to correctly read and predict drag values. Recently, the Cambridge Institute of Computing and Climate Science (Cambridge-ICCS) developed the `FTorch` library—a Fortran-compatible version of PyTorch. This library must be compiled within the MFiX environment for use with the MFiX code. `FTorch` enables reading the new drag model stored in `pickle` format. As a result, the trained drag model can now be called within the MFiX environment to predict drag values. Once the drag values are computed, they are saved in a common block array. The `usr_drag` subroutine—the default user-defined drag subroutine invoked by both the DEM and flow solvers—can retrieve the updated drag values using each particle's ID.

7 CFD Validations

To establish the credibility of the proposed deep learning-based drag model, it is essential to validate its predictions against experimental observations using computational fluid dynamics (CFD). This section presents a systematic validation strategy in three stages. First, single-particle simulations are performed using the discrete element method (DEM) to confirm that the model can capture terminal velocities for particles of varying shapes and densities. Next, the drag model is tested in fluidized bed simulations, which represent a more complex multiparticle environment and serve as the primary application for the model. Finally, a retrained version of the drag model, enriched with additional volume fraction data, is assessed to address limitations observed in the earlier cases. Together, these validation steps provide a comprehensive assessment of the model's performance across increasingly challenging flow conditions.

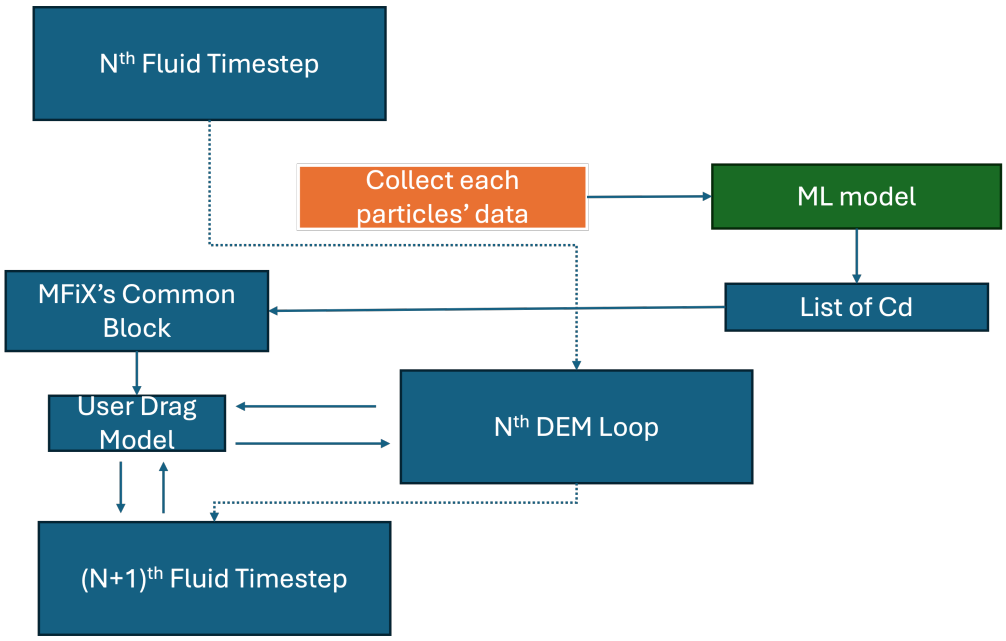


Figure 13: Drag model integration with MFiX.

7.1 Single Particle DEM

Song et al.[46] conducted experiments with non-spherical single particles of various materials to calculate the terminal speed for estimating drag coefficients. The MFiX CFD simulation is setup with the DEM approach to replicate this experimental observation for a single particle. Single particle drag does not depend on the volume fraction and hence its the simplest CFD problem to test the drag model.

Figure 14 compares numerically calculated DEM particles' terminal velocity with that of the experiments. DEM particle motion is influenced by gravity and the new drag force model. The first five cases represent spherical particles. Cases 6 to 12 are for cubic particles of different solid and fluid density. Similarly, cases 14 to 25 are for ideal cylinders with

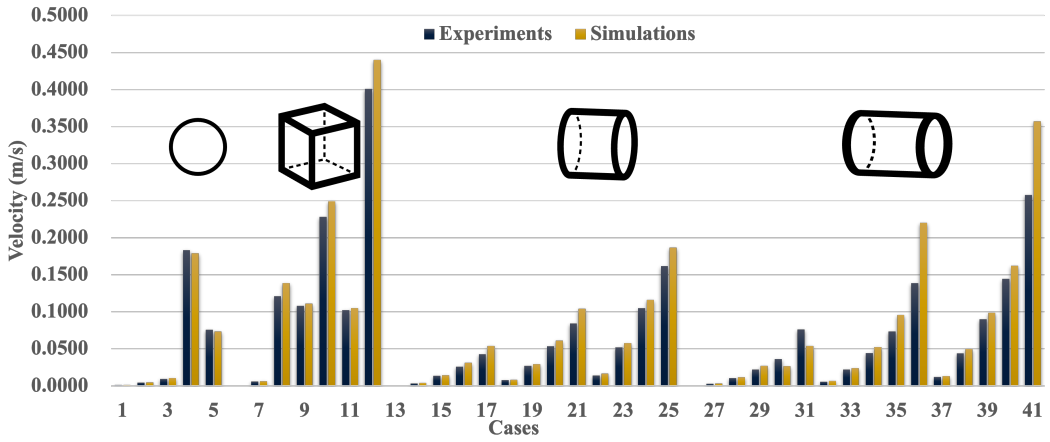
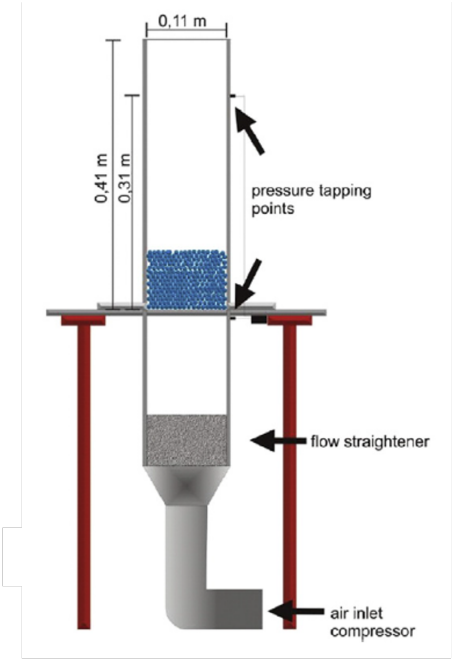
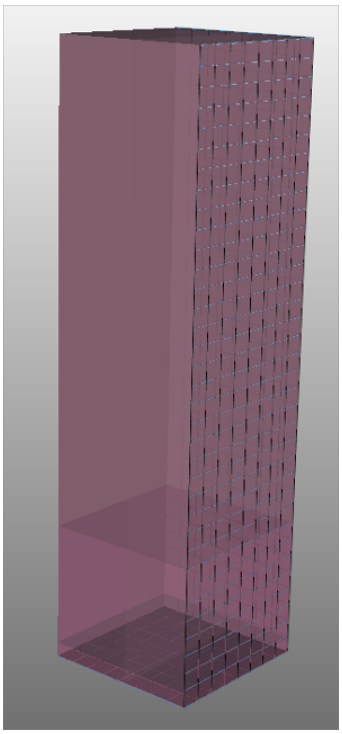


Figure 14: Single particle DEM study in MFiX compared against experiments



(a) Vollmari's experimental setup



(b) CFD domain used for DEM simulation

Figure 15: Fluidized bed experimental setup and CFD domain used for this study.

various density ratios. The remaining cases are for longer cylinders. The CFD values closely match with the experimental values with the error increasing with decrease in sphericity. The error is the highest for the lowest sphericity particle with the highest density (case 41). The results show the new drag model works as intended in the MFiX environment and closely predicts drag forces for the non-spherical particles.

7.2 Fludized Bed Simulations

The new drag model is primarily developed to model the fluidization of non-spherical particles' bed. For this, Vollamari et al. [51] experiments are most suitable data, as they are at lab scale, so there are fewer particles and it covers a wide range of particle shapes. Figure 15 shows the setup details such as the domain width and pressure probe locations. The CFD domain is designed based on these details, and probe points are placed at Z = 0 and 0.31 m. The domain height is kept higher than the experimental setup to prevent particles from escaping the domain. The domain's bottom boundary is treated as the velocity inlet, and its top boundary is open to the atmosphere (ambient pressure). All the remaining boundaries are treated as no-slip wall conditions. The bed is initialized with 0.531 kg of particle mass as per the experiments and subjected to the gas flow from the bottom. The air velocity, also known as superficial velocity for such system, is increased from 0.4 to 2.4 m/s with steps of 0.1 m/s for a parametric study. Figure 16 shows sample calculations of pressure drop for a given velocity condition. The fluidized bed is simulated for six seconds as the pressure

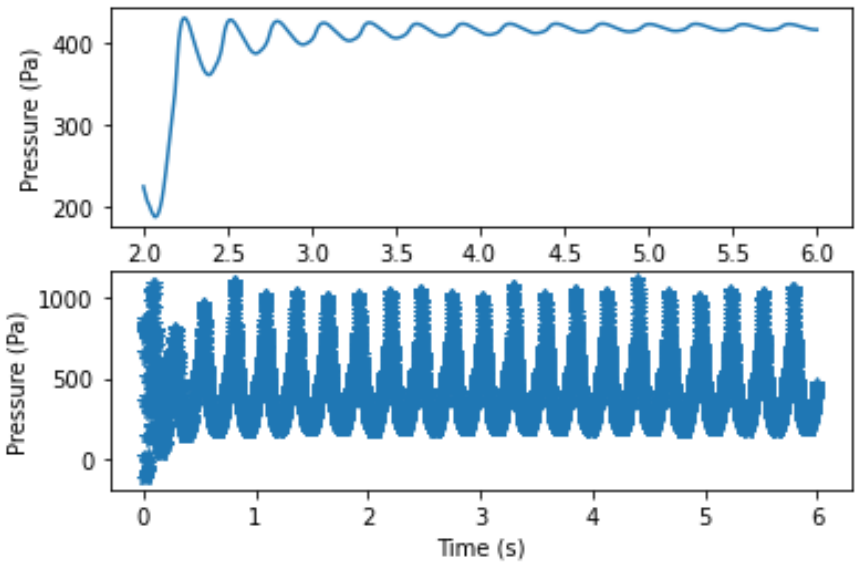


Figure 16: Statistical averaging of the pressure drop across the fluidized bed for a given velocity condition. The bottom plot shows an instantenous pressure drop with time whereas the top plot shows the average pressure drop with time.

drop pattern shows repeating pattern after one second and does not deviate for the next five seconds. The recorded pressure at the two probes is used to calculate the average drop across the domain. Due to high variation in the first second of the simulation, the average calculations starts after two seconds.

In the DEM simulations, the particles are treated as spheres due to simplicity, however the drag model predicts drag values for a given non-spherical particle to realize the dynamics of nonspherical particle beds. Figure 17 shows three conditions of the cube particle bed: 1) bed unfluidized, 2) bed beginning to fluidize, and 3) fluidized. These conditions match closely with the experimental observations. The bed is almost unresponsive to low superficial velocities from 0.4 to 0.8, after which it begins to fluidize. Figure 18 shows the pressure drop comparison against the test data. The pressure drop initially increases steeply with the superficial velocity during which the particle does not show any visible movement. The model underpredicts pressure drop for the unfluidized region and slightly overpredicts for the fluidized region. In the fluidized region, the current model shows reasonable agreement with the test data.

Figure 19 shows the three conditions for the fluidized simulations for the cylindrical particles. The fluidized bed height varies in a similar manner to that of cube particles. However, as shown in Figure 20 the pressure drop predictions closely follow the test data for the higher velocity conditions. For the lower velocity, the pressure drops are at least 20% over-predicted.

In the case of low-sphericity particles such as 7mm class elongated cuboid, the fluidized bed heights are relatively higher than the high-sphericity particles as shown in Figure 21. The low sphericity particles' drag can amplify significantly in the high volume fraction packing which could be the reason for the higher bed height. Figure 22 shows pressure drop

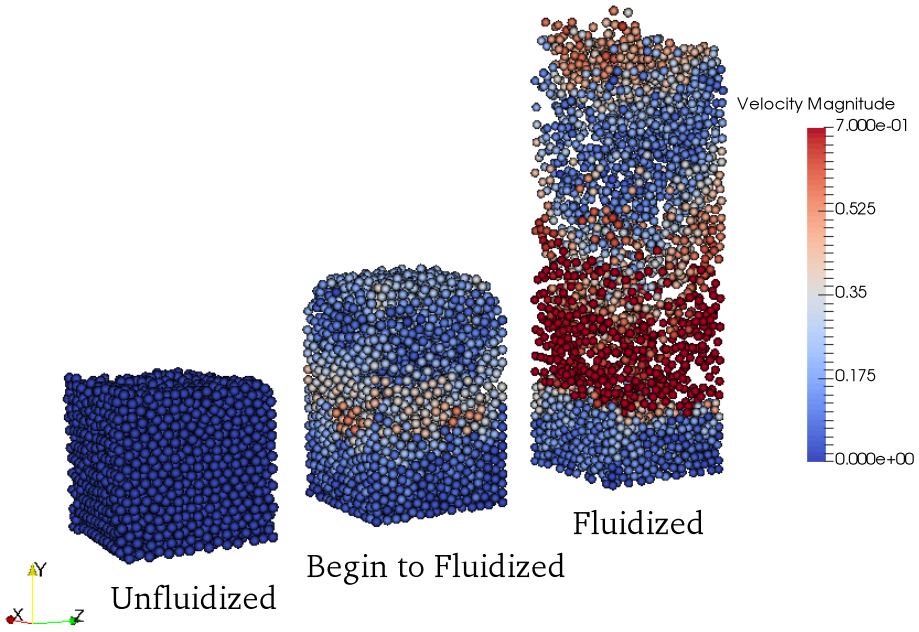


Figure 17: 7mm cube shaped particles in unfluidized, begin to fluidized and fluidized state for superficial velocity 0.4, 1.2 and 2.4 m/s respectively.

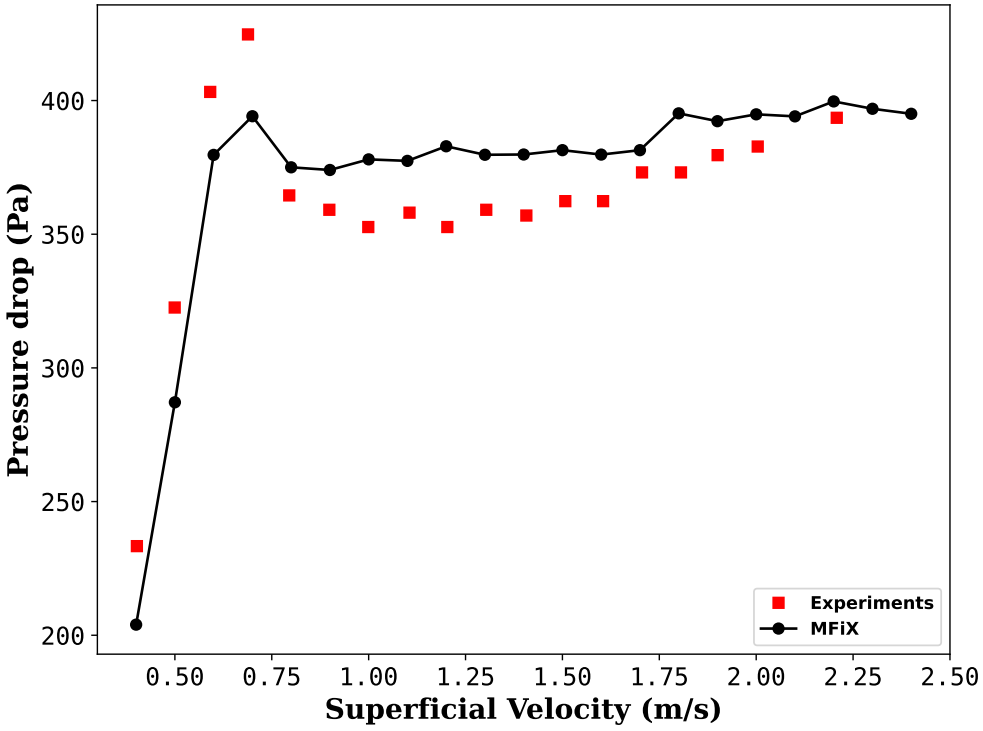


Figure 18: Pressure drop predictions for the 7mm class cube. Errors are high in the low velocity (unfluidized) region.

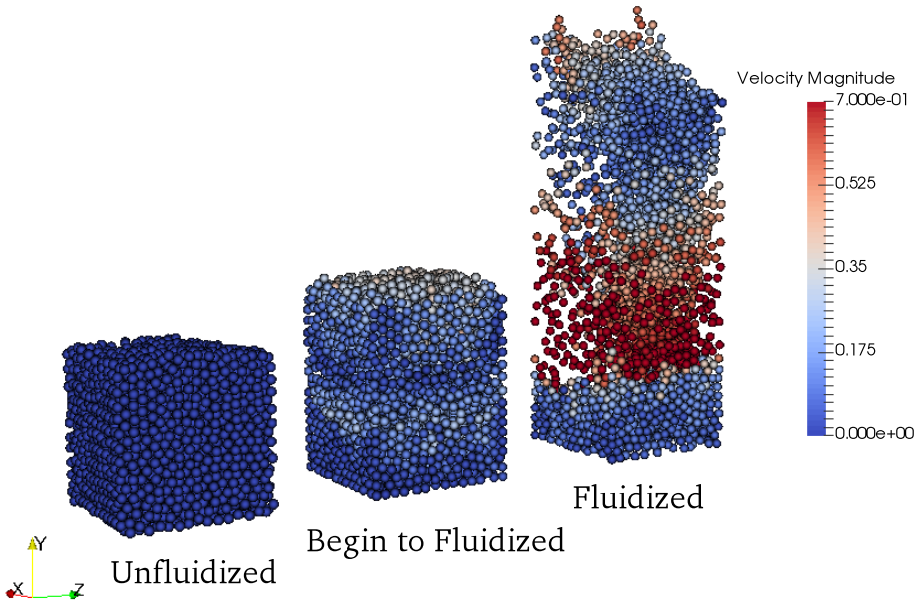


Figure 19: 7mm cylinder shaped particles in unfluidized, begin to fluidized and fluidized state for superficial velocity 0.4, 1.2 and 2.4 m/s respectively.

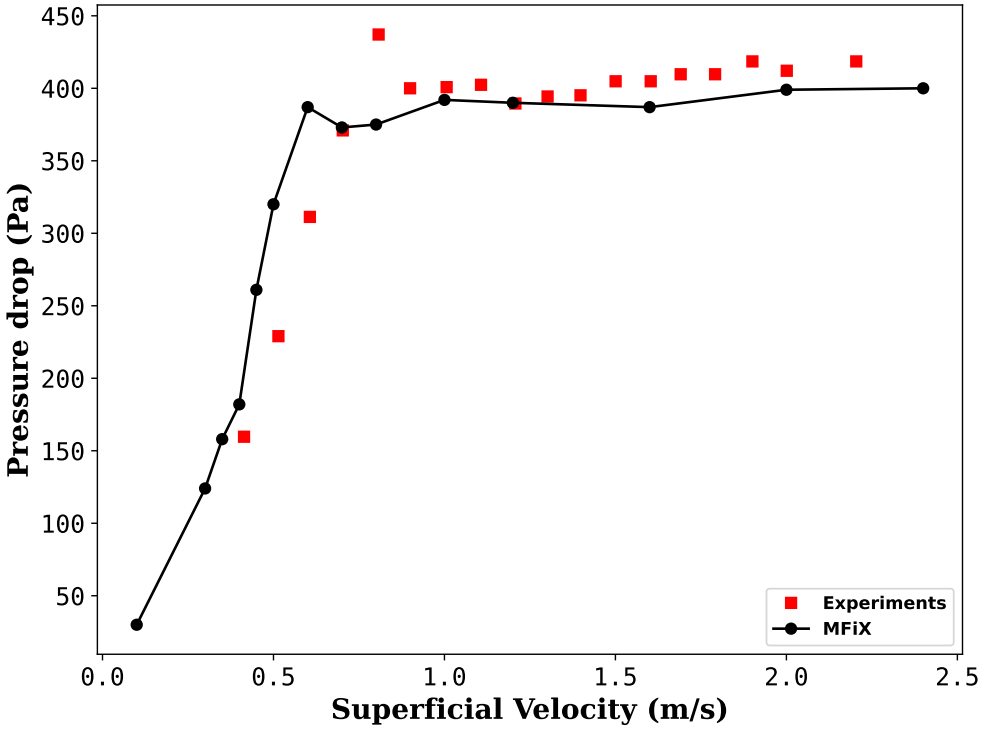


Figure 20: Pressure drop predictions for the 7mm class cylinder.

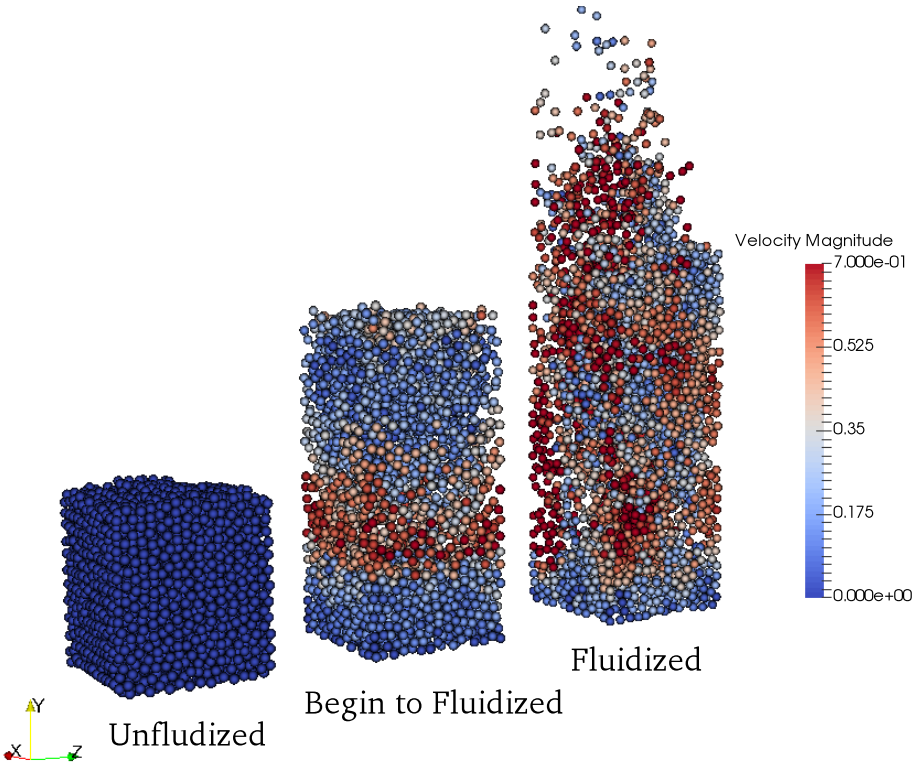


Figure 21: 7mm cuboid shaped particles in unfluidized, begin to fluidized and fluidized state for superficial velocity 0.4, 1.2 and 2.4 m/s respectively.

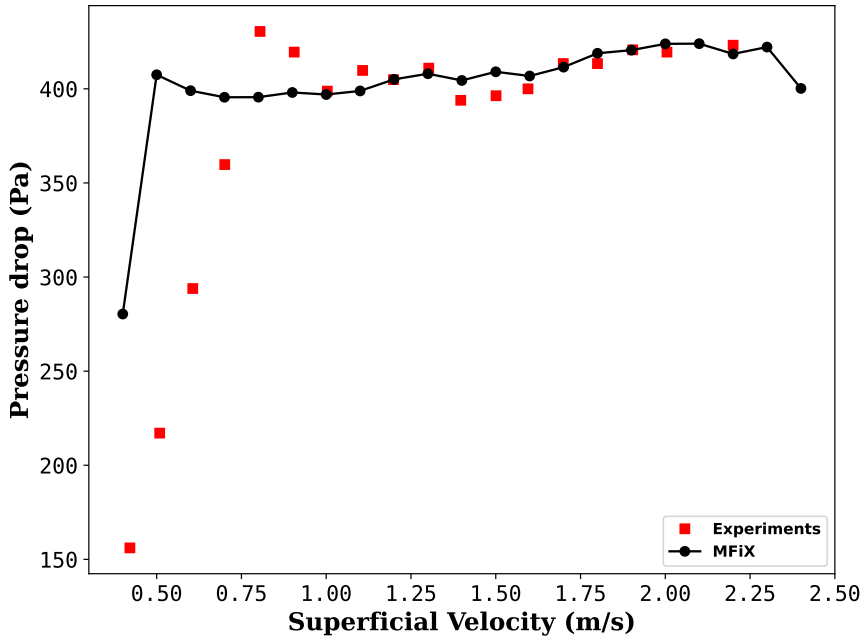


Figure 22: Pressure drop predictions for the 7mm class cuboid. Errors are high in the unfluidized region.

predictions across the cuboid particle's bed. The numerical results match closely with the test data for the fluidized conditions. For the low velocity region, pressure drop is over-predicted. One possible reason for this is the difference in the solid volume fraction for the spherical particles used by DEM and actual non-spherical particles in the test. This difference in packing arrangement can impact drag values.

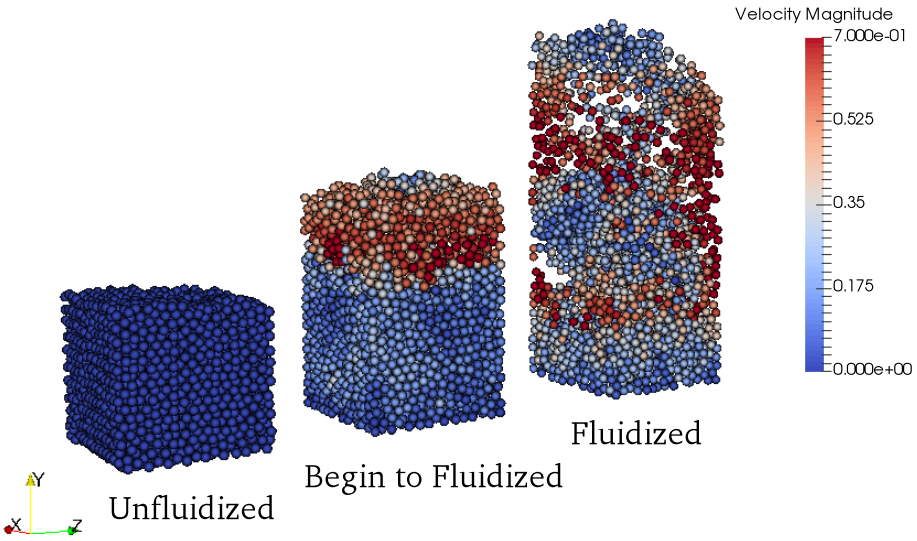


Figure 23: 7mm elongated shaped particles in unfluidized, begin to fluidized and fluidized state for superficial velocity 0.4, 1.2 and 2.4 m/s respectively

Figure 23 shows similar conditions for the elongated cylinder which is also a highly non-spherical particles. The fluidized height is lower in this case but higher for the intermediate velocity (1.2 m/s). Figure 24 compares pressure drop predictions with the test data that shows close agreement in the fluidized conditions but overpredicts pressure drop for the low velocity region. In summary, the new drag models performed well for the fluidized bed applications, however, the unfluidized conditions require more improvement to reduce the pressure drop error.

7.3 Retrained Model

The shortcoming of the drag models is addressed by generating additional volume fraction data which reinforced the DNN model. The new method of randomized particles as discussed in Section 5 generates a large number of data points. The retrained drag model with volume fraction is tested for two particles of the 5 mm class and discussed below.

Figure 25 shows the original drag model and retrained drag model performance with respect to the test data. The retrained model outperforms its predecessors in the low velocity conditions without comprising the high velocity conditions. Similar improvement is also seen in the highly non-spherical particles, elongated cube of the 5 mm class as shown in Figure 26. For this particle, the pressure drop matches closely with the test data except for the fluidized

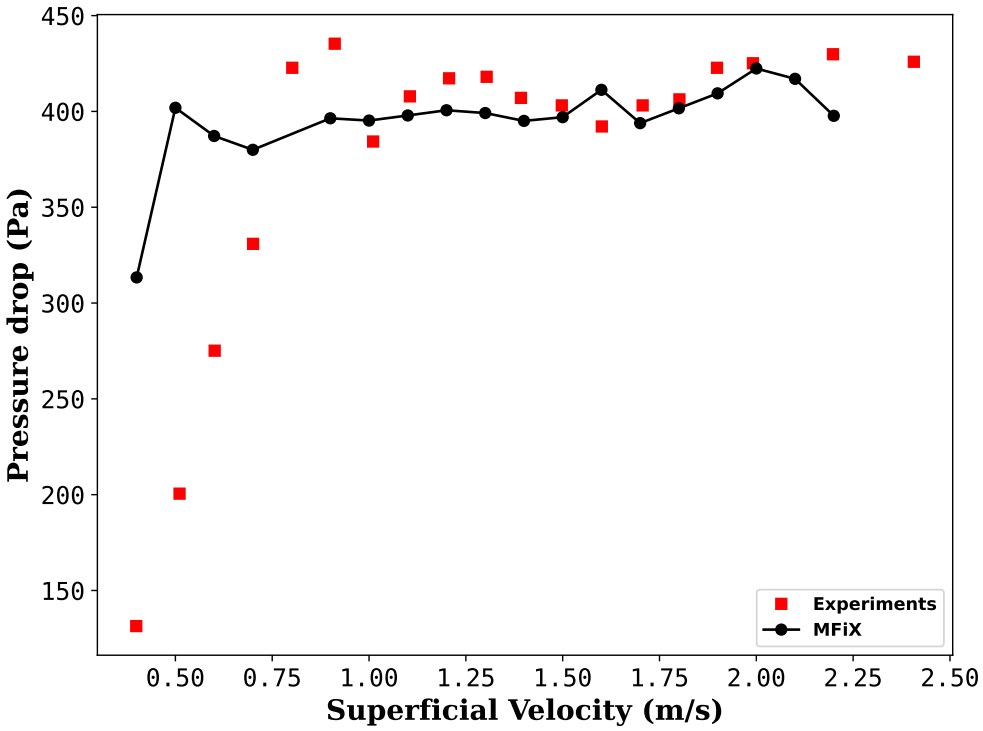


Figure 24: Pressure drop predictions for the 7mm class elongated cylinder. Errors are high in the unfluidized region.

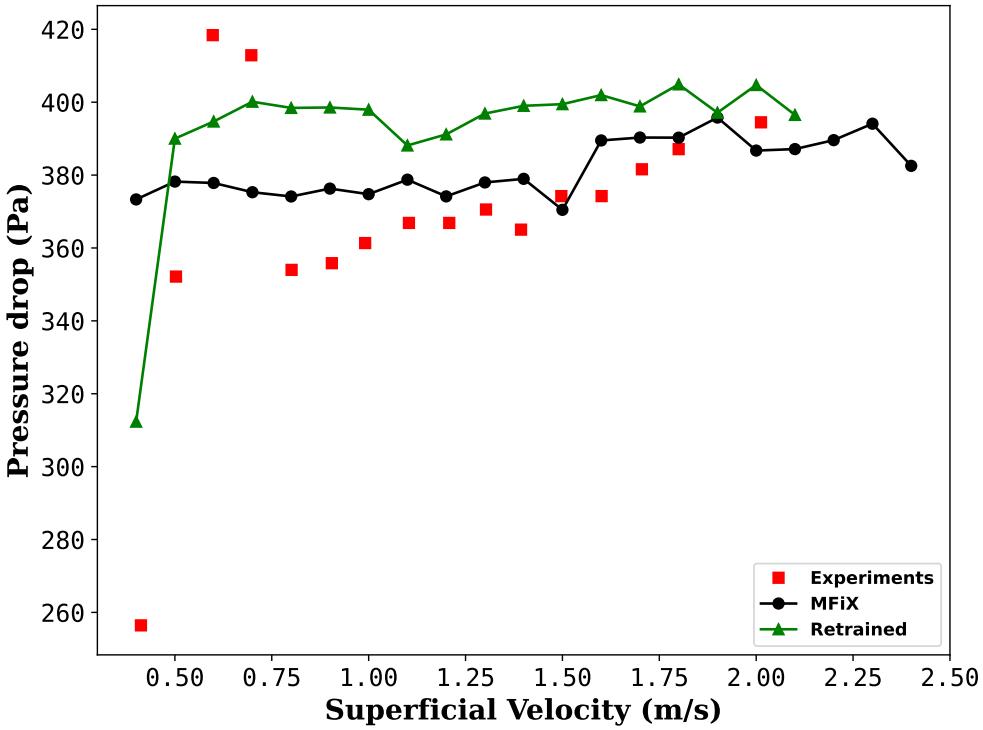


Figure 25: Pressure drop predictions for the 5mm class cube with retrained model.

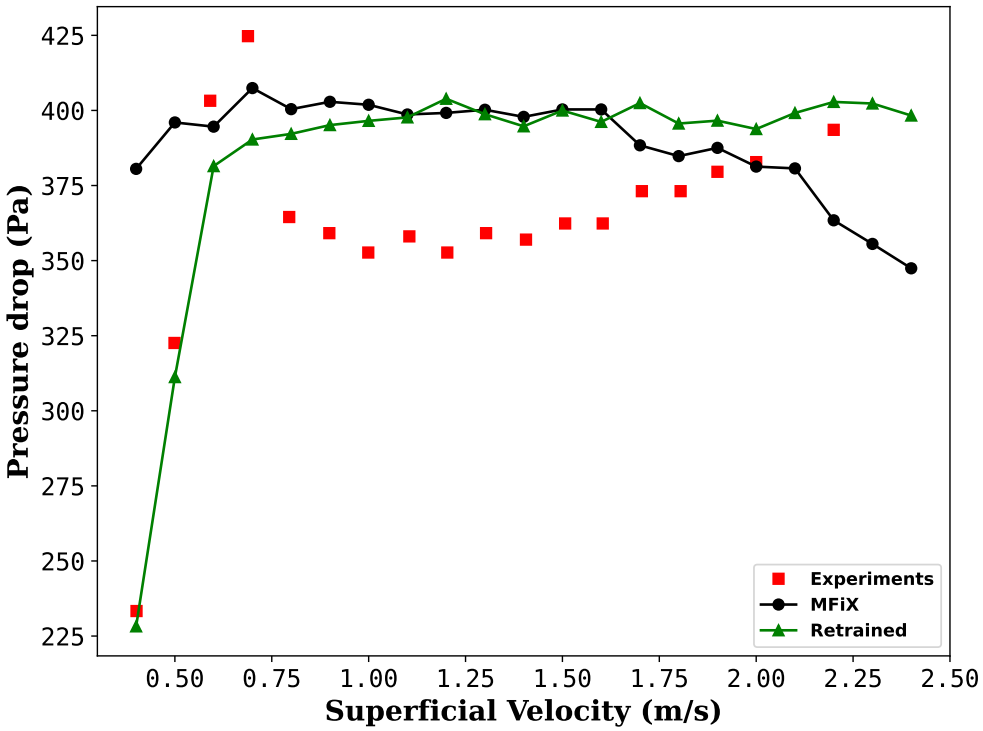


Figure 26: Pressure drop predictions for the 5mm class elongated cube with retrained model.

region, which still have large errors. The retraining exercise shows the model improves with a larger and diverse database. Hence, the model can be continuously improved with the additional availability of the volume fraction data. The new drag model used for fluidization of 7mm and 5mm class particles shows the efficacy of the model with different particle sizes and shapes. The drag model can now be implemented for the Eulerian models where the solid phase is treated as a fluid to avoid a large number of particle tracking. The domain sizes in industrial applications are much higher than the lab-scale test. The number of particles can be in the millions which can be computationally challenging. However, with the advancement of GPU processing and MFiX high-fidelity modeling, DEM modelling of such large number of particles has been demonstrated using a laptop computer [14]. The new drag model can complement these advancements and continue to improve modelling capabilities of gas-solid flows.

8 Conclusion

This study developed a data-driven framework for predicting drag forces on spherical and non-spherical particles in gas–solid multiphase flows. A novel DCC-DNN architecture, aided by empirical correlations and ensemble learning strategies, was trained on an extensive dataset that included experimentally measured and numerically generated drag coefficients. Comparative analysis demonstrated that deep learning models consistently outperformed traditional correlations and machine learning baselines, particularly when leveraging a full feature set of particle and flow descriptors.

The research extended beyond single-particle predictions by addressing the role of particle packing and local solid volume fraction. Using high-fidelity CFD simulations with randomized particle arrangements, additional training data were generated to capture drag amplification in dense regimes. Retrained models incorporating this information showed notable improvements, especially under fluidized conditions, where predictions aligned closely with laboratory-scale experiments.

Successful integration of the drag model into the MFiX solver via the FTorch library further highlights its practical utility. CFD-DEM validations confirmed the model's applicability for simulating both individual particle settling and complex fluidized beds, establishing a foundation for broader adoption in energy and process engineering applications.

Despite these advances, challenges remain. Prediction errors persist under unfluidized, high-volume-fraction conditions due to limited training data. Similarly, while the framework accounts for a wide range of shapes, extreme irregularities still pose uncertainties. Future work should focus on expanding the dataset through additional experiments and high-resolution simulations, improving orientation-resolved modeling, and exploring hybrid physics–ML approaches that embed conservation laws directly into the neural network. In summary, the project demonstrates that deep learning can significantly enhance drag modeling accuracy while reducing computational costs, offering a scalable pathway toward industrial-scale simulations of gas–solid multiphase systems. The combination of physics-based CFD, advanced neural architectures, and GPU-accelerated computation positions this methodology as a promising tool for future energy, environmental, and materials processing applications.

References

[1] J. Baba and P. Komar. Settling velocities of irregular grains at low reynolds numbers. *Journal of Sedimentary Research*, 51:121–128, 1981.

[2] S. Balachandar, W. C. Moore, G. Akiki, and K. Liu. Toward particle-resolved accuracy in Euler–Lagrange simulations of multiphase flow using machine learning and pairwise interaction extended point-particle (PIEP) approximation. *Theoretical and Computational Fluid Dynamics*, 34(4):401–428, 2020. ISSN 14322250. doi: 10.1007/s00162-020-00538-8. URL `https://doi.org/10.1007/s00162-020-00538-8`.

[3] B. Bhattacharya, R. K. Price, and D. P. Solomatine. Machine Learning Approach to Modeling Sediment Transport. *Journal of Hydraulic Engineering*, 133(4):440–450, 2007. ISSN 0733-9429. doi: 10.1061/(asce)0733-9429(2007)133:4(440).

[4] David E.S. Breakey, Farid Vaezi G., Jacob H. Masliyah, and R. Sean Sanders. Side-view-only determination of drag coefficient and settling velocity for non-spherical particles. *Powder Technology*, 339:182–191, 2018. ISSN 1873328X. doi: 10.1016/j.powtec.2018.07.056. URL `https://doi.org/10.1016/j.powtec.2018.07.056`.

[5] Samuel F Buck. A method of estimation of missing values in multivariate data suitable for use with an electronic computer. *Journal of the Royal Statistical Society: Series B (Methodological)*, 22(2):302–306, 1960.

[6] Jie Chen and Jingyin Li. Prediction of drag coefficient and ultimate settling velocity for high-density spherical particles in a cylindrical pipe. *Physics of Fluids*, 32(5), 2020. ISSN 10897666. doi: 10.1063/5.0003923.

[7] R. P. Chhabra, Andrew McKay, and Peter Wong. Drag on discs and square plates in pseudoplastic polymer solutions. *Chemical Engineering Science*, 51(24):5353–5356, 1996. ISSN 00092509. doi: 10.1016/S0009-2509(96)00369-7.

[8] R. P. Chhabra, L. Agarwal, and N. K. Sinha. Drag on non-spherical particles: An evaluation of available methods. *Powder Technology*, 101(3):288–295, 1999. ISSN 00325910. doi: 10.1016/S0032-5910(98)00178-8.

[9] Sze-Foo Chien. Settling velocity of irregularly shaped particles. *SPE Drilling & Completion*, 9(04):281–289, 1994.

[10] Linda A Clark and Daryl Pregibon. Tree-based models. In *Statistical models in S*, pages 377–419. Routledge, 2017.

[11] A. T. Corey. *Influence of shape on the fall velocity of sand grains*. PhD thesis, Colorado A & M College, 1949.

[12] Peter A Cundall and Otto DL Strack. A discrete numerical model for granular assemblies. *geotechnique*, 29(1):47–65, 1979.

[13] Fabio Dioguardi, Daniela Mele, and Pierfrancesco Dellino. A new one-equation model of fluid drag for irregularly shaped particles valid over a wide range of reynolds number: Aerodynamic drag of irregular particles. *Journal of Geophysical Research*, 123:144–156, 2018.

[14] M. Dosta, D. Andre, V. Angelidakis, R.A. Caulk, M.A. Celigueta, B. Chareyre, J.-F. Dietiker, J. Girardot, N. Govender, C. Hubert, R. Kobyłka, A.F. Moura, V. Skorych, D.K. Weatherley, and T. Weinhart. Comparing open-source dem frameworks for simulations of common bulk processes. *Computer Physics Communications*, 296: 109066, 2024. ISSN 0010-4655. doi: https://doi.org/10.1016/j.cpc.2023.109066. URL `https://www.sciencedirect.com/science/article/pii/S0010465523004113`.

[15] Gary H. Ganser. A rational approach to drag prediction of spherical and nonspherical particles. *Powder Technology*, 77(2):143–152, 1993. ISSN 0032-5910. doi: https://doi.org/10.1016/0032-5910(93)80051-B. URL `https://www.sciencedirect.com/science/article/pii/003259109380051B`.

[16] Dimitri Gidaspow and Bozorg Ettehadieh. Fluidization in two-dimensional beds with a jet. 2. hydrodynamic modeling. *Industrial & Engineering Chemistry Fundamentals*, 22 (2):193–201, 1983.

[17] Evan B. Goldstein and Giovanni Coco. A machine learning approach for the prediction of settling velocity. *Water Resources Research*, 50(4):3595–3601, 2014. ISSN 19447973. doi: 10.1002/2013WR015116.

[18] A Haider and O Levenspiel. Drag coefficient and terminal velocity of spherical and nonspherical particles. *Powder technology*, 58(1):63–70, 1989.

[19] Long He and Danesh K Tafti. A supervised machine learning approach for predicting variable drag forces on spherical particles in suspension. *Powder technology*, 345:379–389, 2019.

[20] Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

[21] Andreas Hölzer and Martin Sommerfeld. New simple correlation formula for the drag coefficient of non-spherical particles. *Powder Technology*, 184(3):361–365, 2008.

[22] Soohwan Hwang, Jianhua Pan, and Liang Shih Fan. A machine learning-based interaction force model for non-spherical and irregular particles in low Reynolds number incompressible flows. *Powder Technology*, 392:632–638, 2021. ISSN 1873328X. doi: 10.1016/j.powtec.2021.07.050. URL `https://doi.org/10.1016/j.powtec.2021.07.050`.

[23] David L Johnson, David Leith, and Parker C Reist. Drag on non-spherical, orthotropic aerosol particles. *Journal of Aerosol Science*, 18(1):87–97, 1987.

[24] SR Kale. Characterization of aerodynamic drag force on single particles. Technical report, West Virginia Univ., Morgantown (USA). Dept. of Mechanical and Aerospace, 1987.

[25] P. D. Komar and C. E. Reimers. Grain Shape Effects on Settling Rates. *The Journal of Geology*, 86(2):193–209, 1978. ISSN 0022-1376. doi: 10.1086/649674.

[26] Kun Luo, Dong Wang, Tai Jin, Shuai Wang, Zhuo Wang, Junhua Tan, and Jianren Fan. Analysis and development of novel data-driven drag models based on direct numerical simulations of fluidized beds. *Chemical Engineering Science*, 231:116245, 2021. ISSN 00092509. doi: 10.1016/j.ces.2020.116245. URL `https://doi.org/10.1016/j.ces.2020.116245`.

[27] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1930–1939, 2018.

[28] G. Venu Madhav and R. P. Chhabra. Drag on non-spherical particles in viscous fluids. *International Journal of Mineral Processing*, 43(1-2):15–29, 1995. ISSN 03017516. doi: 10.1016/0301-7516(94)00038-2.

[29] Pratik Mahyawansi and Cheng-Xian Lin. An investigation of the effects of volume fraction on drag coefficient of non-spherical particles using pr-dns. In *Fluids Engineering Division Summer Meeting*, volume 85284, page V001T02A024. American Society of Mechanical Engineers, 2021.

[30] J. W. Malaika. M.s. thesis. Master's thesis, University of Iowa, 1949.

[31] Gordon Mckay, WR Murphy, and M Hillis. Settling characteristics of discs and cylinders. *Chem. Eng. Res. Des.*, 1988.

[32] Francois Oehler, Giovanni Coco, Malcolm O. Green, and Karin R. Bryan. A data-driven approach to predict suspended-sediment reference concentration under non-breaking waves. *Continental Shelf Research*, 46:96–106, 2012. ISSN 02784343. doi: 10.1016/j.csr.2011.01.015. URL `http://dx.doi.org/10.1016/j.csr.2011.01.015`.

[33] E.S. Pettyjohn and E.B. Christiansen. Effect of particle shape on free-settling rates of isometric particles. *Chem. Eng. Prog.*, 44:157–172, 01 1948.

[34] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.

[35] Maria Presa-Reyes, Pratik Mahyawansi, Beichao Hu, Dwayne McDaniel, and Shu-Ching Chen. DCC-DNN: A deep neural network model to predict the drag coefficients of spherical and non-spherical particles aided by empirical correlations. *Powder Technology*, 435: 119388, 2024.

[36] Maria E Presa-Reyes and Shu-Ching Chen. Weakly-Supervised Damaged Building Localization and Assessment with Noise Regularization. In *The 4th IEEE International*

Conference on Multimedia Information Processing and Retrieval, pages 8–14, Tokyo, Japan, 2021.

[37] Amin Riazi and Umut Türker. The drag coefficient and settling velocity of natural sediment particles. *Computational Particle Mechanics*, 6(3):427–437, 2019. ISSN 21964386. doi: 10.1007/s40571-019-00223-6. URL https://doi.org/10.1007/s40571-019-00223-6.

[38] LW Rong, ZY Zhou, and AB Yu. Lattice–boltzmann simulation of fluid flow through packed beds of uniform ellipsoids. *Powder Technology*, 285:146–156, 2015.

[39] Sayeed Rushd, Mohammad Tanvir Parvez, Majdi Adel Al-Faiad, and Mohammed Monirul Islam. Towards optimal machine learning model for terminal settling velocity. *Powder Technology*, 387:95–107, 2021. ISSN 1873328X. doi: 10.1016/j.powtec.2021.04.011. URL https://doi.org/10.1016/j.powtec.2021.04.011.

[40] Johannes Schmiedel. Experimentelle untersuchungen uber die fallbewegung von kugeln and scheiben in reibenden flussigkeiten. *Physik. Z. Bd.*, 29:593–609, 1928.

[41] Edmund F Schulz, Robert Harris Wilde, and Maurice L Albertson. *Influence of shape on the fall velocity of sedimentary particles*. PhD thesis, Colorado State University. Libraries, 1954.

[42] M. K. Sharma and R. P. Chhabra. An experimental study of free fall of cones in Newtonian and Non-Newtonian media: drag coefficient and wall effects. *Chemical Engineering and Processing*, 30(2):61–67, 1991. ISSN 02552701. doi: 10.1016/0255-2701(91)80012-E.

[43] Alan W Sheaffer. Drag on modified rectangular prisms. *Journal of Aerosol Science*, 18 (1):11–16, 1987.

[44] Muhammad Zubair Sheikh, Kristian Gustavsson, Diego Lopez, Emmanuel Lévêque, Bernhard Mehlig, Alain Pumir, and Aurore Naso. Importance of fluid inertia for the orientation of spheroids settling in turbulent flow. *Journal of Fluid Mechanics*, 886:A9, 2020.

[45] David A. Smith and Kwok Fai Cheung. Settling Characteristics of Calcareous Sand. *Journal of Hydraulic Engineering*, 129(6):479–483, 2003. ISSN 0733-9429. doi: 10.1061/(asce)0733-9429(2003)129:6(479).

[46] Xianzhi Song, Zhengming Xu, Gensheng Li, Zhaoyu Pang, and Zhaopeng Zhu. A new model for predicting drag coefficient and settling velocity of spherical and non-spherical particle in Newtonian fluid. *Powder Technology*, 321:242–250, 2017. ISSN 1873328X. doi: 10.1016/j.powtec.2017.08.017.

[47] Walter Squires. *The sedimentation of thin discs*. PhD thesis, Massachusetts Institute of Technology, Department of Chemical Engineering, 1936.

[48] Glen E Stringham, Daryl B Simons, and Harold P Guy. *The behavior of large particles falling in quiescent liquids*. US Government Printing Office, 1969.

[49] Sabine Tran-Cong, Michael Gay, and Efstathios E. Michaelides. Drag coefficients of irregularly shaped particles. *Powder Technology*, 139(1):21–32, 2004. ISSN 00325910. doi: 10.1016/j.powtec.2003.10.002.

[50] Michiel Van Melkebeke, Colin Janssen, and Steven De Meester. Characteristics and Sinking Behavior of Typical Microplastics including the Potential Effect of Biofouling: Implications for Remediation. *Environmental Science and Technology*, 54(14):8668–8680, 2020. ISSN 15205851. doi: 10.1021/acs.est.9b07378.

[51] Kevin Vollmari, Raimondas Jasevivcius, and Harald Kruggel-Emden. Experimental and numerical study of fluidization and pressure drop of spherical and non-spherical particles in a model scale fluidized bed. *Powder Technology*, 291:506–521, 2016.

[52] Jinsheng Wang, Haiying Qi, and Changfu You. Experimental study of sedimentation characteristics of spheroidal particles. *Particuology*, 7(4):264–268, 2009. ISSN 16742001. doi: 10.1016/j.partic.2009.03.008.

[53] P. D. Weidman and I. A. Lasso. Stokes drag on hollow cylinders and conglomerates. *Phys. Fluids*, 29(12 , Dec. 1986):3921–3934, 1986. ISSN 00319171. doi: 10.1063/1.865732.

[54] Robert Harris Wilde. *Effect of Shape on the Fall Velocity of Gravel Sized Particles*. PhD thesis, Colorado Agricultural and Mechanical College, 1952.

[55] William W. Willmarth, Norman E. Hawk, and Robert L. Harvey. Steady and unsteady motions and wakes of freely falling disks. *Physics of Fluids*, 7(2):197–208, 1964. ISSN 10706631. doi: 10.1063/1.1711133.

[56] H. Y. Xie and D. W. Zhang. Stokes shape factor and its application in the measurement of sphericity of non-spherical particles. *Powder Technology*, 114(1-3):102–105, 2001. ISSN 00325910. doi: 10.1016/S0032-5910(00)00269-2.

[57] Sheng Nan Yan, Tian Yu Wang, Tian Qi Tang, An Xing Ren, and Yu Rong He. Simulation on hydrodynamics of non-spherical particulate system using a drag coefficient correlation based on artificial neural network. *Petroleum Science*, 17(2):537–555, 2020. ISSN 19958226. doi: 10.1007/s12182-019-00411-2. URL `https://doi.org/10.1007/s12182-019-00411-2`.

[58] Hyun Doug Yoon, Daniel T. Cox, and Munki Kim. Prediction of time-dependent sediment suspension in the surf zone using artificial neural network. *Coastal Engineering*, 71:78–86, 2013. ISSN 03783839. doi: 10.1016/j.coastaleng.2012.08.005. URL `http://dx.doi.org/10.1016/j.coastaleng.2012.08.005`.

[59] H. N. Yow, M. J. Pitt, and A. D. Salman. Drag correlations for particles of regular shape. *Advanced Powder Technology*, 16(4):363–372, 2005. ISSN 09218831. doi: 10.1163/1568552054194221.

[60] Li Tao Zhu, Bo Ouyang, He Lei, and Zheng Hong Luo. Conventional and data-driven modeling of filtered drag, heat transfer, and reaction rate in gas–particle flows. *AIChE Journal*, 67(8):1–13, 2021. ISSN 15475905. doi: 10.1002/aic.17299.