

Point-particle drag, lift, and torque closure models using machine learning: Hierarchical approach and interpretability

B. Siddani^{*} and S. Balachandar[†]

Department of Mechanical & Aerospace Engineering, University of Florida, Gainesville, Florida 32611, USA



(Received 18 July 2022; accepted 23 December 2022; published 17 January 2023)

Developing deterministic neighborhood-informed point-particle closure models using machine learning has garnered interest recently from the dispersed multiphase flow community. The robustness of neural models for this complex multibody problem is hindered by the availability of particle-resolved data. The present work addresses this unavoidable limitation of data paucity by implementing two strategies: (1) by using a rotation and reflection equivariant neural network and (2) by pursuing a physics-based hierarchical machine learning approach. The resulting machine-learned models are observed to achieve a maximum accuracy of 85% and 96% in the prediction of neighbor-induced force and torque fluctuations, respectively, for a wide range of Reynolds number and volume fraction conditions considered. Furthermore, we pursue force and torque network architectures that provide universal prediction spanning a wide range of Reynolds number ($0.25 \leq \text{Re} \leq 250$) and particle volume fraction ($0 \leq \phi \leq 0.4$). The hierarchical nature of the approach enables improved prediction of quantities such as streamwise torque, by going beyond binary interactions to include trinary interactions.

DOI: [10.1103/PhysRevFluids.8.014303](https://doi.org/10.1103/PhysRevFluids.8.014303)

I. INTRODUCTION

Dispersed multiphase flows can be described as flow systems that contain discrete elements scattered within a continuous phase. In particle-laden flows, which is one such multiphase flow system, the particles are dispersed in a carrier fluid [1,2]. Particle-laden flows of practical interest often involve billions of particles and more. Particle-resolved (PR) simulations of such large particle-laden systems is out of current computational reach. However, only by resolving the interface between the particles and the surrounding fluid, we can obtain the force and torque that couple the two phases. The two widely used alternate approaches are the Euler-Lagrange (EL) and Euler-Euler (EE) methodologies. In both these approaches, the microscale details of the flow on the scale of the particle diameter and interparticle separation distance are not resolved. The advantage is that by sacrificing the information at the small scales, computational attention could be focused on resolving the larger scales and their dynamics. However, the challenge is that first-principle-based access to hydrodynamic force and torque on the particles is lost. EL and EE approaches must therefore rely upon empirical force and torque closure relations.

In the present work, we will mainly focus attention on the EL approach where all the particles within the system are tracked and therefore the averaging is only over the small scales of the fluid phase. How well the dynamics of the flow and the particles in an EL simulation duplicate the true dynamics of a PR simulation is influenced by the accuracy of the force and torque closure models employed in the EL simulation [3]. Furthermore, we will restrict attention to a steady

^{*}siddanib@ufl.edu

[†]bala1s@ufl.edu

static system, where all the particles are held stationary within the system and the Reynolds number is sufficiently small (smaller than a few hundred) that the flow remains time-independent. While there are some direct applications of this restricted configuration in porous media flows and slow-moving inertia-dominated particle-laden flows, our intent is to use the static system as a starting point. Successful demonstration of a machine learning-based framework for the accurate prediction of force and torque in the static system can be extended to more relevant nonstatic conditions.

Consider a steady static system consisting of $N + 1$ particles within a triply periodic domain with a one-dimensional mean flow of Reynolds number Re (based on particle diameter and mean fluid velocity within the domain) driven by a steady uniform streamwise pressure gradient. Particle-resolved simulations of this static configuration with $N \sim O(10^2)$ to $O(10^3)$ have been performed by a number of research groups [4–10] with the primary purpose of understanding the complex three-dimensional flow at the microscale and the modeling of force and torque on the particles.

The modeling of force (and similarly torque) on a particle, which will henceforth be called the *reference particle*, has been approached at two different levels. In the traditional *mesoscale approach*, starting from the pioneering work of Richardson and Zaki [11], the nondimensional force is modeled as a function of the mesoscale Reynolds number Re and particle volume fraction ϕ . Here Re serves to account for the intensity of the flow and ϕ accounts for the number density of neighbors as seen by the reference particle. Both Re and ϕ are averaged over a large enough volume that particle-scale variations are erased, while retaining mesoscale variation in flow velocity and particle volume fraction. The advantage of the mesoscale approach to modeling is its simplicity: force and torque depend only on the two parameters Re and ϕ [4–7]. But simplicity comes at a price. An ensemble of reference particles, all having the same Re and ϕ , will experience a distribution of forces and torques, and the mesoscale model can only predict the average. Particle-to-particle variation of force can be stochastically accounted for in the mesoscale approach with additional knowledge of the force variance as a function of Re and ϕ [10,12–16].

The alternate is the *microscale approach*, where force and torque are modeled in terms of not only the mesoscale variables Re and ϕ , but also taking into account where the neighbors are located. The advantage of the microscale approach is that we can deterministically account for particle-to-particle variation in force and torque, without just relying on statistical variation. This is accomplished with the additional burden of making the model dependent on the relative location of the neighbors. The number of neighbors M being considered in the microscale model is an important parameter. The complexity of the model as well as its predictive capability will increase with the number of neighbors being considered.

The microscale modeling approach must properly recognize the multibody nature of the problem. Even in the Stokes regime, the perturbing influence of any one neighbor on the reference particle is affected by the presence of all other neighbors. At finite Re , the interaction between the particles is further complicated by nonlinear effects. Thus, in the microscale approach, the perturbing influence of all the M neighbors must ideally be considered together as an $(M + 1)$ -body problem. Accounting for such complete multibody interaction is often hard to accomplish in a model. Following the hierarchical approach pursued in classical physics and molecular dynamics [17,18], here we hierarchically account for the effect of the M neighbors by (1) considering the perturbing effect of each neighbor individually, (2) considering the perturbing effect of each pair of neighbors (there are ${}_MC_2$ pairs of neighbors), (3) considering the perturbing effect of three neighbors at a time (there are ${}_MC_3$ combinations of neighbors), and so on. Since at each level we are concerned with the interaction of one, two, three, or more neighbors with the reference particle, these levels will be termed “binary,” “trinary,” “quaternary,” and so on.

First and foremost, in microscale modeling, force or torque are expressed as functions of Re , ϕ , and M neighbor locations, i.e., depends on $3M + 2$ variables. Simple curve fits of force as a function of Re and ϕ were sufficient in the case of mesoscale models. The number of independent variables rapidly increases with M and here we resort to modern machine learning techniques to extract the proper statistical relation between force and torque, and the independent variables. There

has been considerable interest in the deterministic prediction of force and torque using data-driven and deep-learning methods.

Akiki *et al.* [19] presented pairwise interaction extended point-particle (PIEP) model that approximates the undisturbed flow of a reference particle using superposition of perturbation flow fields induced by the M neighbors taken one at a time. As the name suggests, the PIEP model accounts for only the binary interactions, because it considers each neighbor individually. The undisturbed fields obtained with the PIEP model are then utilized to evaluate particle force using the Faxén form of the force relation. The force and torque models under this framework are presented in Akiki *et al.* [20]. The performance of this model decreases considerably at high particle volume fractions. This shortcoming was addressed in Moore *et al.* [21] through a hybrid model that combined physical PIEP method with a data-driven approach, which relies on PR simulations and nonlinear regression. Seyed-Ahmadi and Wachs [10] also presented a binary interaction approach called microstructure-informed probability-driven point-particle (MPP) model that leverages neighbors' conditional probability density function (PDF) to approximate force and torque fluctuations.

He and Tafti [22] used a multilayer perceptron (MLP) that takes Re , ϕ , and the relative position of 15 nearest neighbors together as inputs [(15+1)-body model] to predict the drag force acting on a reference particle. Muralidhar *et al.* [23] showed that the inclusion of physical intermediates such as pressure, velocity, shear and pressure forces, and physics-guided loss terms significantly improves network generalizability for this (15+1)-body problem. More recently, Seyed-Ahmadi and Wachs [24] presented a neural network model that is similar in approach to the PIEP model [19]. The neighbors are considered individually instead of the multibody approach. However, they developed their model to have neighbor-order dependency. The method ensured generalizability in capturing force and torque fluctuations for data paucity applications, unlike the multibody model. Ashwin *et al.* [25] showed that deep learning can also be successfully applied to predict drag forces on nonspherical particles.

Moore and Balachandar [26] defined the perturbation flow induced by particles at a given Re and ϕ to be the *superposable wake*. The flow field around any reference particle can be recreated by adding its own superposable wake and those of each neighbor. Siddani *et al.* [27] pursued this flow prediction task using conditional generative adversarial network (GAN) [28,29]. The volumetric representation of the M neighbors [($M + 1$)-body input] and the convolutional neural network (CNN) architecture enabled the approach in reconstructing the complex flow through the random bed of particles far more accurately than the superposable wake approach. The higher accuracy can be attributed to the multibody nature of GAN prediction over the binary approximation of the PIEP model. This suggests the potential advantage of going beyond binary interactions and including trinary and higher-order interactions among the particles. Similarly, Muralidhar *et al.* [30] presented a physics-guided architecture to predict the flow fields. The task of achieving particle force and torque upon the availability of surface resolved flow field is a linear process. Therefore, an additional neural network (NN) was used to evaluate drag force using these flow fields. The results indicate an improvement over previous methods [22,23].

The following two additional factors pertaining to machine learning must be considered in the microscale modeling of force and torque. First, the NN's architecture is determined by many attributes, including the number of hidden layers (i.e., the depth), the width of the hidden layers, the form of nonlinearity enforced, and other hyperparameters used in the training of the NN. Collectively we refer to all these details as the *neural network details*. Second, for any chosen architecture, the neural network must be trained appropriately with ground truth. In the present case of steady stationary particle-laden flow, the ground truth for each reference particle consists of its Re , ϕ , and local information of its M nearest neighbors as input and the hydrodynamic force and torque as the output. The number of reference particles P used in the training process becomes an important parameter. Only with a very large training data set, the network can be properly trained without overfitting. However, since the training data come from either PR simulations or high-quality experimental measurements, P is typically limited.

In summary, the performance of a deterministic NN-based microscale model, in terms of predicting the force and torque on individual particles, can be expected to be substantially better than the mesoscale models. Nevertheless, based on above discussion, we can identify four reasons why the predicted force or torque by a microscale model may not perfectly yield the corresponding exact values as obtained in a PR simulation. (1) *Neighbor-truncation error*: In the present geometry of a triply periodic box, each reference particle's neighborhood at the microscale is fully determined by the location of its N neighbors. If the NN model is developed based on only the M nearest neighbors, there will be an error associated with the truncation $M < N$. (2) *Interaction-truncation error*: The NN model based on M nearest neighbors may not attempt to solve the $(M + 1)$ -body problem. The model truncation at binary, trinary, or quaternary levels will contribute to additional error. (3) *Network error*: The third source of error comes from any limitations of the network architecture and optimization. We admit that the network architecture and the chosen hyperparameters (including the optimization algorithm) that are being used for training may not be perfect, and the difference contributes to network error. (4) *Generalization error*: Even if all the other errors are eliminated by choosing $M = N$, an $(N + 1)$ -body interaction, and a perfect NN, the resulting trained model will approach perfection only if there is enormous amount of training data. With a finite training set of P realizations, there will be generalization error.

The above errors are interconnected. For example, as the complexity of the model increases, either with increasing number of neighbors M or with increasing level of multibody interaction, one needs a more complex network architecture in order to be able to appropriately map the force and torque on the expanding set of input parameters. As the depth, width, and complexity of the NN increases, the amount of training data P required to train the expanding number of trainable weights increases as well. Thus, with the value of P dictated by the amount of available ground truth, one may have to limit M , restrict attention to only binary and trinary interactions, and consider simpler NN architectures. As a result, the predicted model will not be perfect and all four errors will contribute to the overall error.

The present work aims to add to the existing body of knowledge on the development of robust force and torque coupling models appropriate for particle-laden flows using machine learning with the following contributions:

(1) The deterministic microscale model, when interpreted as a mapping between the input features consisting of Re , ϕ , and the relative position $\mathbf{r}_j$ of the $j = 1, 2, \dots, M$ neighbors, must obey the following fundamental symmetries: Galelian and translation invariance, and rotation and reflection equivariance. By formulating the problem in the frame attached to the reference particle, Galilean and translation invariance are automatically satisfied. Rotation and reflection equivariance imply that as the configuration of the M neighbors rotate or reflect about the center of the reference particle, the corresponding predicted force and torque must equivalently rotate or reflect as well. Past machine learning models of force and torque have not enforced rotation and reflection equivariance. In the present work, with the use of an equivariant NN framework [31], we will strictly enforce these symmetries. As demonstrated in the context of flow prediction around a random distribution of particles [32], imposition of rotation and reflection equivariance not only enforces these fundamental symmetries, but has the great advantage of amplifying the effectiveness of available training data. Thus, enforcing rotation and reflection symmetries in the present problem is of fundamental importance due to the limited number of training samples for the complex modeling task at hand.

(2) Past binary interaction modeling efforts have been at fixed values of Re and ϕ [21,24]. Models developed at varying discrete combinations of Re and ϕ are then interpolated to make predictions at any intermediate values of Re and ϕ . The advantage of the machine learning approach is that Re and ϕ can be used as parameters in order to develop a universal model that spans a wide range of Reynolds number and volume fraction. The only requirement is that the training data span the required range of Re and ϕ .

(3) Following the theoretical PIEP modeling effort of Akiki *et al.* [19], the machine learning modeling effort of Seyed-Ahmadi and Wachs [24] has clearly demonstrated that force modeling as

a complete M -body problem is not possible due to the limited availability of training data from PR simulations. They further showed the viability of force modeling at the level of binary interaction with available data. Here we will develop a rigorous hierarchical multibody force and torque modeling framework, following a similar approach pursued in molecular dynamics [17,18], and use the framework to go beyond binary interaction to consider trinary interactions. We quantitatively illustrate the importance of trinary interactions, especially through the prediction of streamwise torque component, which cannot be captured using a binary model.

(4) The present work will also analyze the limitations of the deterministic microscale modeling arising from limited training data availability. In particular, we will expound on the four different sources of error. Attention will also be paid to establishing the adequacy of network architecture.

The rest of the paper is arranged as follows: Sec. II explains mathematical details regarding the strict imposition of rotational and reflectional symmetries in the utilized neural networks. Section III entails a discussion regarding the PR simulation data and its preparation for model development. The proposed hierarchical machine learning framework is thoroughly detailed in Sec. IV, and the associated neural network details are mentioned in Sec. V. The performance and analysis of developed models is carried out in Sec. VI. Finally, the interpretability attained through the pursued systematic method is demonstrated in Sec. VII.

II. EQUIVARIANT NEURAL NETWORKS: SYMMETRY-PRESERVING MODELS

It is well understood that physical systems, and governing equations describing them satisfy certain symmetries. A closure model that is used to approximate any physical phenomenon is also expected to satisfy these underlying symmetries. Equivariant neural networks are a particular type of NN that by construction preserve certain transformations, for example, rotational and/or reflectional symmetries [33–35]. As mentioned earlier, the imposition of these symmetries into neural closure models significantly improves the generalizability through innate data amplification, especially for three-dimensional (3D) spaces. Many such implementations across different applications achieved superior performance [32,36–39].

The current work leverages basic MLP architecture in developing the particle force and torque models. Furthermore, the input information for the problem at hand is only scalars (Re and ϕ) and 3D vectors (relative position of neighbors and streamwise direction). Nonlinearity can be applied without any restrictions to scalars as they are rotation and reflection invariant. On the other hand, rotational and reflectional equivariance for vectors (or pseudovectors) reduces to ensuring that nonlinearity is applied only to their magnitude and not applied separately to each component. The following constitutive block, shown in Eq. (1), details how m input vectors denoted as $\mathbf{a}$ are nonlinearly combined into n output vectors ($\mathbf{b}$) using the so-called norm-based activation [31]

$$\mathbf{b}_n = \frac{\mathcal{F}(|w_{nm}\mathbf{a}_m| + d_n)}{|w_{nm}\mathbf{a}_m|} w_{nm}\mathbf{a}_m, \quad (1)$$

where $\mathbf{w}$, $\mathbf{d}$, and $\mathcal{F}$ are weights, bias, and activation function, respectively. The NN architectures utilized in this work are based on the above idea and hence preserve rotational and reflectional symmetries.

III. PARTICLE-RESOLVED SIMULATION DATA AND CURATION

Particle-resolved simulation data used in the present work is gathered from two different sources that generate the random particle distributions and simulate the flow using different computational approaches. The first six data sets presented in Table I are produced using NekIBM solver [40,41]. The remaining 10 data sets are obtained from Seyed-Ahmadi and Wachs [24]. The incompressible Navier-Stokes equations with no-slip and no-penetration boundary conditions on particle surface are solved to a steady state in a triply-periodic box. The reader is directed to Seyed-Ahmadi and

TABLE I. Assimilated data sets used in the current study to develop robust universal point-particle closure models.

$\langle \text{Re} \rangle$	$\langle \phi \rangle$	$ \langle \mathbf{F} \rangle $	σ_{Drag}	σ_{Lift}	$\sigma_{\text{Torque}, \parallel}$	$\sigma_{\text{Torque}, \perp}$	N
9.86	0.10	2.76	0.7823	0.4661	0.2407	0.9253	1000
121.36	0.10	9.55	3.1110	1.4986	0.5043	1.3689	1000
6.95	0.21	4.03	0.9581	0.7454	0.4195	1.2987	1000
73.40	0.21	9.51	2.7987	1.6178	0.6441	1.7103	1000
27.81	0.40	12.68	2.0613	1.5028	0.6734	2.0589	1000
73.42	0.40	19.14	3.6087	2.4135	1.0285	2.5955	1000
2.20	0.10	2.63	0.5425	0.3973	0.2375	0.9241	2984
10.92	0.10	3.28	0.7351	0.4839	0.2405	0.9838	3000
165.96	0.10	9.48	2.4191	1.5662	0.4339	1.1972	2984
0.25	0.20	4.42	0.7971	0.6737	0.4295	1.4509	3055
2.48	0.20	4.47	0.8116	0.6707	0.4217	1.4465	3000
49.78	0.20	7.72	1.8194	1.2608	0.5078	1.6549	3055
187.25	0.20	14.24	3.6989	2.6873	0.8446	2.0219	3055
3.25	0.40	11.75	2.7535	2.1329	1.0256	2.4789	2578
64.35	0.40	19.67	5.0963	3.6885	1.3063	2.9899	2578
245.65	0.40	36.62	9.5692	7.1215	2.0385	3.9776	2578

Wachs [10], Yang and Balachandar [40], and references therein for a thorough explanation of the considered numerical simulations and setups.

These PR simulations are performed in a large computational domain of size $O(10d_p)$, where d_p is particle diameter, in order to consider $O(10^3)$ particles within the domain. Due to the triply periodic boundary condition, all the N particles within the computational domain are statistically equivalent. Thus, the first step of the curation process is to make each of the N particles within the domain to be the reference particle and consider a *local volume* around it for the purposes of microscale modeling. The local volume is typically much smaller than the entire computational box and determines the number of M closest neighbors that are within which will be considered in the microscale modeling [10,27,32]. The location of the M neighbors within the local volume defines the microscope environment of the reference particle, while averages within the local volume are used to define the mesoscale Re and ϕ .¹ Thus, from each steady-state solution of the PR simulation, N curated samples centered around each particle is extracted.

For the PR simulations of Seyed-Ahmadi and Wachs [24] a larger sphere of diameter $4d_p$ for $\phi = 0.1, 0.2$, and $3d_p$ for $\phi = 0.4$ was chosen as the local volume to evaluate volume-averaged fluid velocity around each reference particle. For the NekIBM simulations, the local volume is a cube with size L_A , where $L_A = [\frac{(M+1)\pi}{6\phi}]^{1/3}d_p$. This local cube size L_A was evaluated using $M = 26$ for NekIBM simulated data sets presented in Table I. The chosen value of $M = 26$ is within the range of influential neighbors that make a significant contribution for the considered data sets as mentioned in previous studies [10,21,24]. In a random distribution of particles, the number of nearest neighbors M can be chosen without restriction. Whereas in simple cubic (SC) and body-centered cubic (BCC) arrangements, to preserve the spatial symmetries, only special values of M are allowed. $M = 26$ satisfies this requirement for the first, second, and third nearest neighbors in SC and BCC arrangements. The streamwise direction in all these PR simulations is along $+x$ axis. Therefore, the local volume-averaged fluid velocity in every data set is also approximately along

¹The mesoscale variables (Re , ϕ) are therefore somewhat dependent on the local volume size. However, the dependency is weak with a reasonable choice of local volume size.

+x with minor deviations in direction and magnitude due to the unique particulate microstructure occurring in each individual local volume.

The gathered data is further curated in the following manner before it is used in the data-driven approach. Particle diameter is chosen as the length scale. The local volume-averaged fluid velocity, $\bar{\mathbf{u}}_f$, is nondimensionalized using $\mu/(\rho d_p)$, where μ is dynamic viscosity, and ρ is fluid density, and, therefore, the nondimensional velocity magnitude is the Reynolds number, Re. Particle forces are scaled by the Stokes drag, $3\pi\mu d_p|\bar{\mathbf{u}}_f|$, and particle torques are nondimensionalized using $\mu d_p^2|\bar{\mathbf{u}}_f|$.

Generally, the quantities of interest (QoI), such as Re, ϕ , and nondimensional forces and torques, are reported by averaging over the entire computational domain. In contrast, the current work evaluates these QoI based on local volume averages for microscale modeling. Hence, the statistics presented in this work are based on local volume averages. Thus, every reference particle in each PR simulation has its own unique set of QoI: Re_i , ϕ_i , $\mathbf{F}_i$, and $\mathbf{T}_i$ are, respectively, the Reynolds number, particle volume fraction, nondimensional force, and nondimensional torque of the i th reference particle evaluated based on its local volume-averaged fluid velocity. For a given data set, an ensemble average, $\langle \rangle$, is defined as an average over all the N particles as follows:

$$\begin{aligned} \langle \text{Re} \rangle &= \frac{1}{N} \sum_i^N \text{Re}_i, & \langle \phi \rangle &= \frac{1}{N} \sum_i^N \phi_i, \\ \langle \mathbf{F} \rangle &= \frac{1}{N} \sum_i^N \mathbf{F}_i, & \langle \mathbf{T} \rangle &= \frac{1}{N} \sum_i^N \mathbf{T}_i. \end{aligned} \quad (2)$$

Note that $\langle \text{Re} \rangle$ and $\langle \phi \rangle$ will be slightly different from the Reynolds number and volume fraction defined based on the entire computational volume. For the considered data sets, the largest variation between the ensemble average and the entire domain average is 3.5% for the Reynolds number and 2.4% for particle volume fraction. This is due to the nonuniform sampling of the entire domain using local volumes around the randomly distributed particles.

Particle ensemble-averaged quantities for every data set are presented in Table I.² The statistics along the mean flow direction is different from that along the transverse directions. The standard deviation of nondimensional force along the streamwise direction, denoted by σ_{Drag} , measures the deviation of the individual particle's nondimensional drag force from the ensemble average. Also shown in the table are the corresponding standard deviation of force along the transverse direction, represented by σ_{Lift} , and the mean transverse force is zero. Similarly, the standard deviation of nondimensional torque along the streamwise and transverse directions are notated as $\sigma_{\text{Torque},\parallel}$ and $\sigma_{\text{Torque},\perp}$ respectively, with their corresponding mean values approaching zero.

IV. HIERARCHICAL MODELING FRAMEWORK

The important takeaway from previous deep learning studies of particulate force modeling is that an enormous amount of training data from PR simulations is required to achieve a generalizable multibody model. Thus, a physics-guided hybrid machine learning approach has the best chance of success, since we are faced with the unavoidable situation of modeling with a limited amount of training data. References [21,24] presented machine learning models that consider only binary interactions among the particles and thereby incorporated some simplifying multiphase flow physics in order to cope with the limited data availability. The current effort builds upon these physics-guided approaches and presents a hierarchical methodology that systematically considers progressively higher-order interactions between the particles in a multibody system. Furthermore, the present work ensures that the neural network models to be developed strictly preserve Galilean,

²For all data sets, the Reynolds number reported in the table is based on average interstitial velocity. Note that Seyed-Ahmadi and Wachs [10,24] reported their cases using Reynolds number based on superficial velocity.

scale, and translational invariance, and rotational and reflectional equivariance. By doing so the present models make the optimal use of the limited available data.

A. Hierarchical approach

We construct the hierarchical approach for a large homogeneous distribution of particles of uniform macroscale volume fraction $\langle\phi\rangle$ subjected to a steady uniform ambient flow of Reynolds number $\langle\text{Re}\rangle$. The nondimensional hydrodynamic force acting on the i th reference particle considering all the N neighbors.³ It can be represented by the following series expansion:

$$\begin{aligned} & \mathbf{F}_i(\langle\text{Re}\rangle, \langle\phi\rangle, \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\}) \\ &= \mathbf{F}_{1i}(\langle\text{Re}\rangle, \langle\phi\rangle) + \sum_{j=1}^N \mathbf{F}_{2i}(\langle\text{Re}\rangle, \langle\phi\rangle, \mathbf{r}_j) + \sum_{j=1}^{N-1} \sum_{k=j+1}^N \mathbf{F}_{3i}(\langle\text{Re}\rangle, \langle\phi\rangle, \mathbf{r}_j, \mathbf{r}_k) + \dots \end{aligned} \quad (3)$$

A general formulation that takes neighbors' relative motion into account for more general flows is presented in Appendix A. The first term on the right-hand side, $\mathbf{F}_{1i}(\langle\text{Re}\rangle, \langle\phi\rangle)$, will be referred to as the *unary term*. It accounts for the effect of the neighboring particles only in a collective sense through the volume fraction $\langle\phi\rangle$. Furthermore, the predicted force of the unary model is identically the same for all particles in a homogeneous system, and hence is the same as the best possible mesoscale model.

The second term, $\sum_{j=1}^N \mathbf{F}_{2i}(\langle\text{Re}\rangle, \langle\phi\rangle, \mathbf{r}_j)$, is the *binary interaction term* as it considers the perturbing influence of one neighbor at a time. In the static monodispersed configuration of present interest, the j th neighbor is taken into account with the distance $\mathbf{r}_j$ from the reference particle. In a more general context, the j th neighbor's information will also include its size, velocity, and acceleration. The *trinary interaction term*, $\sum_{j=1}^{N-1} \sum_{k=j+1}^N \mathbf{F}_{3i}(\langle\text{Re}\rangle, \langle\phi\rangle, \mathbf{r}_j, \mathbf{r}_k)$, accounts for the influence of two neighbor pairs at a time. In the binary and trinary terms, the sums account for all two and three-particle combinations that include the reference particle. Likewise, other higher-order terms until the $(N+1)$ -body interaction term, $\mathbf{F}_{(N+1)i}(\langle\text{Re}\rangle, \langle\phi\rangle, \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\})$, which considers all the N neighbors simultaneously, are considered as part of the hierarchical approach.

The above representation is complete in the sense that it considers all the N neighbors and accounts for the complete $N+1$ body interaction among the particles. If we obtain the functional dependence of all the unary, binary, trinary, and higher-order terms perfectly, then the resulting model will make perfect prediction and the deterministic estimation $\mathbf{F}_i(\langle\text{Re}\rangle, \langle\phi\rangle, \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\})$ given in the left-hand side will indeed be the force on the i th particle. We call this perfect model as the *ideal model*, since perfection can only be achieved in the idealized sense. In practice, the deterministic prediction will be imperfect due to the four sources of errors discussed in the introduction. Although we provide no formal proof for the existence of the ideal model based on machine learning, we point out that the PR simulations that use conventional CFD can be thought of as the ideal model, since it is capable of extracting the forces accurately given $\langle\text{Re}\rangle$, $\langle\phi\rangle$, $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\}$.

B. Probabilistic interpretation

We obtain further insight into the above hierarchical model with the following consideration. Let the distribution of neighbors around the reference particle be characterized by the probability $P_N(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$, where $P_N(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)d\mathbf{r}_1 d\mathbf{r}_2, \dots, d\mathbf{r}_N$ represents the probability of finding the N neighbors in a volume $d\mathbf{r}_1$ around $\mathbf{r}_1$, $d\mathbf{r}_2$ around $\mathbf{r}_2$, and so on. The unary term can then be

³In a triply periodic domain being considered in the present work, $N+1$ represents the total number of particles (including the reference particle) within the domain and it is finite. In an unbounded homogeneous system $N \rightarrow \infty$.

expressed as

$$\mathbf{F}_{1i}(\langle \mathbf{Re} \rangle, \langle \phi \rangle) = \int \cdots \int \mathbf{F}_i(\langle \mathbf{Re} \rangle, \langle \phi \rangle, \{\mathbf{r}_1, \dots, \mathbf{r}_N\}) P_N(\mathbf{r}_1, \dots, \mathbf{r}_N) d\mathbf{r}_1 \cdots d\mathbf{r}_N, \quad (4)$$

where there are N integrals and each extends over the entire domain to cover all possible locations of the neighboring particles. The above equation defines the unary term to be the ensemble-averaged force when particles are uniformly distributed according to the stated probability. The above expression was derived by multiplying Eq. (3) with P_N and integrating over the position of all the particles. It also requires the following conditions be satisfied:

$$\begin{aligned} \int \mathbf{F}_{2i}(\langle \mathbf{Re} \rangle, \langle \phi \rangle, \mathbf{r}_j) P_1(\mathbf{r}_j) d\mathbf{r}_j &= 0, \\ \iint \mathbf{F}_{3i}(\langle \mathbf{Re} \rangle, \langle \phi \rangle, \mathbf{r}_j, \mathbf{r}_k) P_2(\mathbf{r}_j, \mathbf{r}_k) d\mathbf{r}_j d\mathbf{r}_k &= 0, \\ &\dots, \end{aligned} \quad (5)$$

where $P_1(\mathbf{r}_j)$ is the marginal probability of one particle being at $\mathbf{r}_j$, while others can be anywhere, and similarly $P_2(\mathbf{r}_j, \mathbf{r}_k)$ is the marginal probability of two particles being at $\mathbf{r}_j$ and $\mathbf{r}_k$, while others can be anywhere. These two marginal probabilities are expressed as

$$\begin{aligned} P_1(\mathbf{r}_j) &= \int \cdots \int P(\mathbf{r}_1 = \mathbf{r}_j, \mathbf{r}_2, \dots, \mathbf{r}_N) d\mathbf{r}_2 \cdots d\mathbf{r}_N, \\ P_2(\mathbf{r}_j, \mathbf{r}_k) &= \int \cdots \int P(\mathbf{r}_1 = \mathbf{r}_j, \mathbf{r}_2 = \mathbf{r}_k, \mathbf{r}_3, \dots, \mathbf{r}_N) d\mathbf{r}_3 \cdots d\mathbf{r}_N. \end{aligned} \quad (6)$$

In order to obtain a similar probabilistic expression for the binary term, we first define the following probability:

$$P_{N-1}(\mathbf{r}_2, \dots, \mathbf{r}_N | \mathbf{r}_1 = \mathbf{r}_j) = \frac{P(\mathbf{r}_1 = \mathbf{r}_j, \mathbf{r}_2, \dots, \mathbf{r}_N)}{P_1(\mathbf{r}_j)}, \quad (7)$$

where the subscript “ $N - 1$ ” in P_{N-1} indicates that the position of one of the neighbors of the reference particle has been fixed at $\mathbf{r}_j$ and the probability is over the position of the other $N - 1$ neighbors. In the above, the denominator on the right-hand side serves to normalize so that integral of P_{N-1} over the domain of all its arguments becomes unity. The binary term can then be expressed as

$$\begin{aligned} \mathbf{F}_{2i}(\langle \mathbf{Re} \rangle, \langle \phi \rangle, \mathbf{r}_j) &= \int \cdots \int \mathbf{F}_i(\langle \mathbf{Re} \rangle, \langle \phi \rangle, \{\mathbf{r}_1 = \mathbf{r}_j, \dots, \mathbf{r}_N\}) \\ &\quad P_{N-1}(\mathbf{r}_2, \dots, \mathbf{r}_N | \mathbf{r}_1 = \mathbf{r}_j) d\mathbf{r}_2 \cdots d\mathbf{r}_N - \mathbf{F}_{1i}(\langle \mathbf{Re} \rangle, \langle \phi \rangle). \end{aligned} \quad (8)$$

This offers the interpretation that $\mathbf{F}_{2i}(\langle \mathbf{Re} \rangle, \langle \phi \rangle, \mathbf{r}_j)$ is the ensemble-averaged force on the reference particle with one of the neighbor being at $\mathbf{r}_j$ and the other particles being anywhere else within the domain according to stated probability minus the unary term obtained by averaging over all possible positions of all the particles. Thus, $\mathbf{F}_{2i}(\langle \mathbf{Re} \rangle, \langle \phi \rangle, \mathbf{r}_j)$ corresponds to the added effect of fixing one of the neighbor's position to be at the fixed location of $\mathbf{r}_j$. The summation in the second term of Eq. (3) thus accounts for the added perturbation effect of each neighbor evaluated from its known position, but taken one at a time.

For the trinary term, we first fix the position of two neighbors at $\mathbf{r}_j$ and $\mathbf{r}_k$ and define the probability P_{N-2} of finding all the other neighbors at small volumes around $\mathbf{r}_3, \mathbf{r}_4$, and so on:

$$P_{N-2}(\mathbf{r}_3, \dots, \mathbf{r}_N | \mathbf{r}_1 = \mathbf{r}_j, \mathbf{r}_2 = \mathbf{r}_k) = \frac{P(\mathbf{r}_1 = \mathbf{r}_j, \mathbf{r}_2 = \mathbf{r}_k, \mathbf{r}_3, \dots, \mathbf{r}_N)}{P_2(\mathbf{r}_j, \mathbf{r}_k)}. \quad (9)$$

We then multiply Eq. (3) with the two fixed-neighbor probability P_{N-2} and integrate over the position of all the other $N - 2$ neighbors, which yields the relation

$$\begin{aligned} F_{3i}(\langle \text{Re} \rangle, \langle \phi \rangle, \mathbf{r}_j, \mathbf{r}_k) &= \int \cdots \int F_i(\langle \text{Re} \rangle, \langle \phi \rangle, \{\mathbf{r}_1 = \mathbf{r}_j, \mathbf{r}_2 = \mathbf{r}_k, \dots, \mathbf{r}_N\}) \\ &\quad P_{N-2}(\mathbf{r}_3, \dots, \mathbf{r}_N | \mathbf{r}_1 = \mathbf{r}_j, \mathbf{r}_2 = \mathbf{r}_k) d\mathbf{r}_3 \cdots d\mathbf{r}_N - F_{1i}(\langle \text{Re} \rangle, \langle \phi \rangle) \\ &\quad - F_{2i}(\langle \text{Re} \rangle, \langle \phi \rangle, \mathbf{r}_j) - F_{2i}(\langle \text{Re} \rangle, \langle \phi \rangle, \mathbf{r}_k). \end{aligned} \quad (10)$$

The trinary term can be interpreted as the ensemble-averaged drag on the reference particle when two neighbors are fixed at $\mathbf{r}_j$ and $\mathbf{r}_k$, while the position of all others is varied according to stated probability minus the unary and binary contributions. In the subtraction, the unary contribution is independent of the known neighbor positions, while the binary contribution is based on the two known neighbor positions.

From the above discussion, it is clear that each term of the hierarchical approach systematically adds a layer of additional information that was not included in the previous terms. Thus, the binary term adds the effect of binary interaction between the reference and a neighbor, taken one neighbor at a time. Similarly, the trinary term adds the effect of trinary interaction between the reference and two neighbors, taken one neighbor pair at a time, and so on. It is interesting to note that if the expansion given in Eq. (3) were to be continued, the last term of the expansion will be $F_{(N+1)i}(\langle \text{Re} \rangle, \langle \phi \rangle, \mathbf{r}_j, \mathbf{r}_k, \dots)$, without any summation. Functionally it is exactly as complicated as the force F_i on the left-hand side that we are trying to expand on the right-hand side. However, the informational content of this last term of the expansion will likely be negligible, since all the prior terms have accounted for much of the physics. Thus, the real advantage of the hierarchical expansion is that the complexity of multiparticle interaction is progressively increased, so that depending on the situation it can be truncated at any level.

C. Truncation and other approximations

The ideal model presented in Eq. (3) is neither necessary nor practical. First and foremost, as we will see below in the Results section, the amount of PR simulation data available for model training is sufficient for the binary term and inadequate even for the trinary term. Therefore, additional terms in the expansion can be included in the force model only upon obtaining vast amount of additional information. The results to be discussed below show diminishing returns with the inclusion of each additional term and thus for any desired level of accuracy going beyond the first few terms may not be necessary.

Based on these considerations, we introduce the following truncated hierarchical expansion, which will be used for machine learning model:

$$F_i[M_2, M_3, 3] = \underbrace{F_{1i}(\text{Re}_i, \phi_i) + \sum_{j=1}^{M_2} F_{2i}(\text{Re}_i, \phi_i, \mathbf{r}_j)}_{F_i[M_2, 2]} + \sum_{j=1}^{M_3-1} \sum_{k=j+1}^{M_3} F_{3i}(\text{Re}_i, \phi_i, \mathbf{r}_j, \mathbf{r}_k), \quad (11)$$

where on the left-hand side we have dropped presenting the dependence on $(\langle \text{Re} \rangle, \langle \phi \rangle, \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\})$ explicitly. The notation $F_i[M_2, M_3, 3]$ indicates that it is a trinary model where the binary term is based on M_2 neighbors and the trinary term is based on M_3 neighbors. Here we recognize the fact that each term of the hierarchical expansion can be based on a different set of nearest neighbors. It can be noted that if the trinary term is neglected then the binary model can be denoted as $F_i[M_2, 2]$.

Clearly, $F_i[M_2, 2]$ and $F_i[M_2, M_3, 3]$ are increasing level of approximations to the ideal model F_i . When compared to the ideal model, the above summation involves two truncations: (1) instead of all the N neighbors, the binary and trinary terms are based on only subsets of M_2 and M_3 neighbors, and (2) the expansion itself is truncated, since the quaternary and higher-order terms are neglected. These two truncations introduce error, and they can be separated as shown below for the trinary

model:

$$\underbrace{F_i[M_2, M_3, 3] - F_i}_{\text{Truncation error}} = \underbrace{F_i[N, N, 3] - F_i}_{\text{Interaction-truncation}} + \underbrace{F_i[M_2, M_3, 3] - F_i[N, N, 3]}_{\text{Neighbor-truncation}}. \quad (12)$$

It can be seen that the interaction-truncation error is due to limiting the expansion to only the first three terms and the neighbor-truncation error is due to limiting the number of neighbors in the binary and trinary terms. As discussed in the introduction, when developing machine learning models of the binary and trinary terms two additional errors, namely, network and generalization errors, arise, which we shall consider in the following sections.

One additional important modification has been made in Eq. (11) compared to the ideal model given in Eq. (3). On the right-hand side, $\langle \text{Re} \rangle$ has been replaced by Re_i and $\langle \phi \rangle$ has been replaced by ϕ_i . The rationale for this change is as follows. The Reynolds number Re_i and volume fraction ϕ_i evaluated within the averaging volume around the reference particle will differ from the ensemble-averaged values. The difference depends on the size of the averaging volume chosen around the reference particle. With increasing averaging volume, the difference decreases. This dependence of Re_i and ϕ_i on averaging volume has been discussed by Siddani *et al.* [27]. The resulting particle-to-particle variation in Re_i and ϕ_i in some sense accounts for the interaction of the reference particle with its neighbors in a backhanded way. As a result, even the unary model F_{1i} with its dependence on Re_i and ϕ_i has in it some aspects of the effect of the specific neighbor interactions with the reference particle. This replacement is clearly *ad hoc* and has been made mainly as a result of extensive experimentation which consistently showed a slight improvement in predictive capability.

It should also be pointed out that the dependence of the binary term can be expanded to include Re_j and ϕ_j as input variables and the dependence of the trinary term can be expanded to include Re_j , ϕ_j , Re_k , and ϕ_k . We tried this modeling option and it did not result in observable improvement in predictive capability with the considered data sets. This lack of improvement is partly due to the fact that variation in Re , ϕ in the immediate neighborhood of each particle is small for the considered uniformly random distributions, and partly due to limited training data availability to capture such subtle dependencies. Therefore, we restrict attention to the model given in Eq. (11). It must also be emphasized that the local volume considered in evaluating Re_i and ϕ_i is independent of the number of nearest neighbors used in the modeling of the binary term (M_2) and the trinary term (M_3).

The above force model is duplicated to obtain the corresponding torque model, which is given as

$$T_i[M_2, M_3, 3] = \underbrace{T_{1i}(\text{Re}_i, \phi_i)}_{T_{i[M_2, 2]}} + \sum_{j=1}^{M_2} T_{2i}(\text{Re}_i, \phi_i, \mathbf{r}_j) + \sum_{j=1}^{M_3-1} \sum_{k=j+1}^{M_3} T_{3i}(\text{Re}_i, \phi_i, \mathbf{r}_j, \mathbf{r}_k). \quad (13)$$

D. Unary model

The current work utilizes the mesoscale force model developed by Tenneti *et al.* [5] as the unary force model, which can be expressed as

$$F_{1i}(\text{Re}_i, \phi_i) = \left[\frac{1 + 0.15 \text{Re}_{m,i}^{0.687}}{(1 - \phi_i)^2} + \frac{5.81 \phi_i}{(1 - \phi_i)^2} + \frac{0.48 \phi_i^{1/3}}{(1 - \phi_i)^3} + \phi_i^3 \text{Re}_{m,i} \left(0.95(1 - \phi_i) + \frac{0.61 \phi_i^3}{(1 - \phi)} \right) \right] \hat{\mathbf{e}}_i, \quad (14)$$

where $\text{Re}_{m,i} = \text{Re}_i(1 - \phi_i)$, and $\hat{\mathbf{e}}_i$ denotes unit vector along the mean flow direction for the i th reference particle. The unary torque model is trivial because the PR simulations performed in a triply periodic domain are not subject to any external moments. As a result, the mean zero torque has not been presented in Table I. The unary force model used in this work is thus an empirical fit and does not involve any machine learning.

For the considered data sets, a machine-learned unary model that has better agreement with particle-resolved data than Tenneti *et al.* [5] model can be developed. However, when considering

a binary model, there is flexibility in defining the unary model, since, during the training process, the binary interaction portion will adjust to the prescribed unary model. In other words, if we take the unary model to be the empirical fit of Tenneti *et al.* [5], then the binary contribution averaged over all the particles of the considered data sets, will not be zero. The nonzero average of the binary contribution, when added to the unary contribution, would correspond to the earlier stated machine-learned unary model.

An important point about the unary model must be discussed. The above unary model by Tenneti *et al.* [5] is based on a certain N -particle uniform probability distribution P_N . Although it is not explicitly known, we assume that there exists an N -particle distribution that corresponds to the manner in which the particles were distributed in their PR simulations. If we call this N -particle distribution to be $P_{N,Tn}$, then only for this distribution and its marginal one- and two-particle distributions the relations (4) and (5) apply; i.e., only for this particular uniform distribution of particles that obey the probability $P_{N,Tn}$ the ensemble averaged-drag will be the same as the unary model given above. For any other distribution of particles, uniform or otherwise, whose $P_N \neq P_{N,Tn}$, the relation (5) does not apply and the ensemble-averaged drag will receive contributions from binary, trinary, and higher-order interactions, and thus will differ from the above unary model. It should be noted that there exists a discrepancy among different unary models presented in the literature, especially at high Reynolds numbers. The discrepancy does not necessarily stem from differences in the manner particles are distributed. The reasons for the discrepancy may be multiple and not well analyzed and understood.

V. NEURAL NETWORKS

A *sequential learning process* is adopted to obtain the binary and trinary terms. This means we first develop the best possible binary term before moving onto a similar strategy in search of the best trinary term. This process along with other network-related aspects used to produce the described results are detailed in this section.

It was stated earlier that a single global microscale model that can accurately predict forces and torques in the presence of substantial mesoscale variation will be pursued. The Reynolds number of the training data set varies between [0.25, 245.65] and the volume fraction extends over the range [0, 0.4]. The mean drag as represented by the unary model increases with increasing Re and ϕ . Particle-to-particle force variation, measured as rms drag, has been observed to scale as the mean force [10,13,27]. Furthermore, the number of training samples available for each combination of Reynolds number and volume fraction varies. Therefore, we choose the following force loss function ($\mathcal{L}_{\text{Force}}$) for training the global force models:

$$\mathcal{L}_{\text{Force}} = \sum_{\text{datasets}} \frac{\sum_{\text{samples}} \frac{|\mathbf{F}_{i,\text{PR}} - \mathbf{F}_{i,\text{NN}}|^2}{|\langle \mathbf{F} \rangle|^2}}{N}, \quad (15)$$

where $\mathbf{F}_{i,\text{PR}}$ denotes the curated ground truth of the i th reference particle obtained from PR simulations and $\mathbf{F}_{i,\text{NN}}$ is a corresponding neural network prediction, using the unary, binary, and trinary truncation.

As part of the sequential learning process we first set out to achieve the best binary interaction force term, $\mathbf{F}_{2i}$. Hence, we substitute $\mathbf{F}_i[M_2, 2]$ for $\mathbf{F}_{i,\text{NN}}$ in Eq. (15) during the training stage of the binary model. Once a satisfactory binary term is produced, we set out to achieve the best trinary interaction force term, $\mathbf{F}_{3i}$ by substituting $\mathbf{F}_i[M_2, M_3, 3]$ for $\mathbf{F}_{i,\text{NN}}$ in Eq. (15). In the modeling of the trinary term, only the machine learning model for $\mathbf{F}_{3i}$ is optimized, while the already optimized binary model of the previous step is utilized for the binary term.

A similar strategy is employed for the torque models as well. However, the torque loss function ($\mathcal{L}_{\text{Torque}}$) is slightly different. Torque standard deviation along transverse directions, $\sigma_{\text{Torque},\perp}$, which

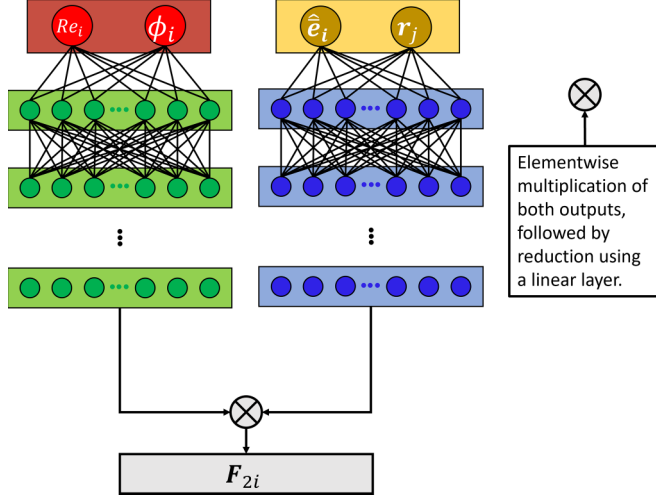


FIG. 1. Binary-interaction force model network architecture.

is the larger component is used for normalization:

$$\mathcal{L}_{\text{Torque}} = \sum_{\text{datasets}} \frac{\sum_{\text{samples}} \frac{|\mathbf{T}_{i,\text{PR}} - \mathbf{T}_{i,\text{NN}}|^2}{\sigma_{\text{Torque},\perp}^2}}{N}. \quad (16)$$

A. Binary force network details

The architectures used in the present work are inspired by Lu *et al.* [42] for their ability to accurately predict multiscale bubble growth dynamics [43,44]. The architecture of the binary force model is shown in Fig. 1. The mesoscale information (Re_i , ϕ_i) is passed to the left subnetwork. The unit vector along the mean flow direction ($\hat{e}_i$) and the neighbor's relative location (r_j) are passed to the right subnetwork. The appropriate mixing of these subnetworks' outputs occurs only at the final stage. As the left subnetwork deals only with scalars no special effort is required to preserve rotational and reflectional symmetries. An unbounded activation function, rectified linear unit (ReLU), is used in the left subnetwork based on the reasoning that the mean and standard deviations of force are directly proportional to Reynolds number and particle volume fraction. On the contrary, the right subnetwork takes in the two vectors ($\hat{e}_i$, r_j) as input. Thus, the constitutive block given by Eq. (1) is implemented to preserve equivariance. Hyperbolic tangent is used as the activation function because of its proven success in achieving accurate results at fixed mesoscale variables [24]. The final outputs of left and right subnetworks are scalars and vectors respectively. However, we need only a single force contribution vector, F_{2i} , as the final binary force model output. Hence, we create equal number of output scalars and vectors from both subnetworks to perform elementwise multiplication [42]. This leaves us with only vectors that are finally combined into a single vector using a linear (vector summation) layer, which is also optimizable.

The following training details are same for every model considered in this work. All models are implemented using PyTorch [45] and e3nn libraries [31] for rotational and reflectional equivariance. A minibatch size of 100 and ADAM optimizer [46] with an initial learning rate of 0.001 is used. Learning rate scheduler of type ReduceLROnPlateau, which reduces the learning rate by a factor of 0.1 if the total training loss does not reduce in 10 consecutive epochs, is also availed. Default weight initialization of PyTorch and e3nn is used for the network layers, and these computations were carried out on a single NVIDIA A100 GPU. The process of k -fold cross-validation [47], with $k = 5$, was adopted to produce the model test performance. In a single round of the k -fold cross-validation

TABLE II. Grid search hyperparameter tuning for F_{2i} with $M_2 = 26$: Force loss ($\mathcal{L}_{\text{Force, test}}$) evaluated for the test data using different network architectures.

Nonlinear layers (L_2)	Width of each layer (W_2)	
	50	100
2	0.6447	0.5688
3	0.5614	0.5285
4	0.5330	0.5134
5	0.5165	0.5066
6	0.5164	0.5104

procedure, 20% of every data set is used in creating the total *test samples*. Out of the remaining 80% in each data set 64% is used in training a network and the remaining 16% is used as part of *validation samples*. Thus, for each $\langle \text{Re} \rangle$ and $\langle \phi \rangle$ combination given in Table I, the total number of data samples N was divided into training, validation, and testing data samples (i.e., into N_{trn} , N_{val} , and N_{test}). In the fivefold cross-validation, by using different 20% as test samples in each fold, the entire data set is included in the testing process. Every network is trained for a minimum of 100 epochs and the maximum number of epochs is limited to 1000. The training process is stopped (early stopping) if the total validation loss does not reduce further in 50 consecutive epochs.

Table II shows the performance of different binary force networks on the test samples measured in terms of the force loss. The reported values are a collective representation of five independent versions of the network, which is a consequence of the ($k = 5$)-fold cross-validation procedure. The rows indicate the number of nonlinear layers (L_2) used in each subnetwork and the columns correspond to the width (W_2) of the nonlinear layer. For a layer of $W_2 = 50$ the left subnetwork (Fig. 1) has 50 scalars, and the right subnetwork has 50 vectors. From the performed grid search hyperparameter tuning, $(L_2, W_2) = (6, 50)$ was chosen for the binary force model as its performance is comparable with that of $(5, 100)$ and $(6, 100)$. The difference in $\mathcal{L}_{\text{Force, test}}$ is less than 0.01. Furthermore, the selected model has significantly fewer parameters. For instance, the $(6, 50)$ network has 25850 trainable parameters whereas $(5, 100)$ network has 81500 trainable parameters. These tests were performed by including 26 neighbors in the binary force model (i.e., $M_2 = 26$).

B. Trinary force network details

The training, validation, and testing partitioning of the data set for the trinary model is the same as that used in the binary force model. The architecture of the trinary force model (Fig. 2) is similar to its binary counterpart. With an intention to have a trinary network that has equal or higher expressiveness, the hyperparameter tuning, shown in Table III, is started with the number of layers and their width to be $(L_3, W_3) = (6, 50)$. Moreover, the hyperparameter tuning is performed using

TABLE III. Grid search hyperparameter tuning for F_{3i} with $M_3 = 10$: Test sample force loss ($\mathcal{L}_{\text{Force, test}}$) values for different architectures.

Nonlinear layers (L_3)	Width of each layer (W_3)	
	50	100
6	0.4891	0.4868
10	0.4767	0.4965
15	0.4858	0.4961

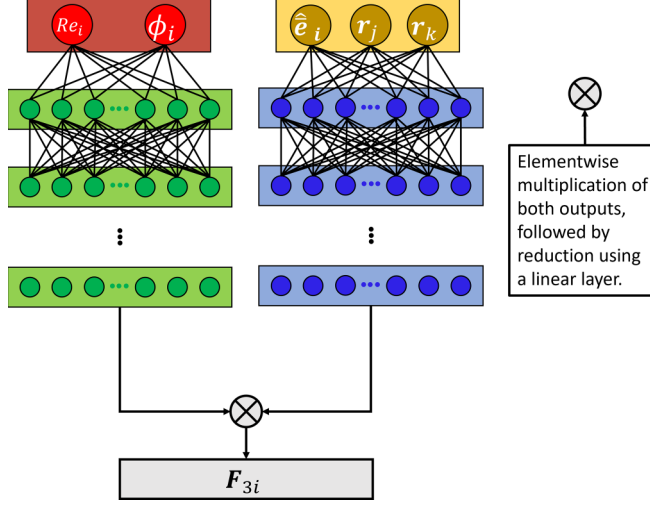


FIG. 2. Trinary-interaction force model network architecture.

only 10 nearest neighbors (i.e., $M_3 = 10$), which yields ${}_{10}C_2 = 45$ pairs of neighbor combinations. This decision has been taken to substantially reduce the computational cost, since including 26 neighbors as in the binary model would result in ${}_{26}C_2 = 325$ neighbor pair combinations. Finally, a network with $(L_3, W_3) = (10, 50)$ is chosen as the optimal trinary force model for the given training data.

C. Torque network details

While the input Reynolds number, volume fraction, and relative location of neighbor information comprise only scalars and vectors, the torque on the reference particle as one of the desired outputs is a pseudovector. Therefore, special manipulation is necessary to preserve rotational and reflectional equivariance. Binary torque architecture presented in Fig. 3 illustrates the methodology used in the creation of a rotation and reflection equivariant pseudovector output. The left subnetwork takes Re_i and ϕ_i as inputs and it uses ReLU as the activation function. The right subnetwork takes the position of a neighbor as input and uses a hyperbolic tangent for nonlinear manipulation of intermediate vector magnitudes. Similar to the force models, the subnetworks' outputs are combined at the final stage to create a single vector. A cross product between two vectors results in a pseudovector. This operation also preserves rotational and reflectional symmetries. Therefore, the mean flow direction unit vector, $\hat{e}_i$, is involved in a cross product with the output vector to create binary torque contribution, T_{2i} . Based on hyperparameter tuning tabulated in Table IV, a network with $(L_2, W_2) = (5, 100)$ is chosen as the final binary torque model.

A binary torque model will not accurately predict the streamwise component of torque [20,24] due to the two-body flow symmetry. This signifies the importance of including trinary interactions in the torque model. The architecture of the trinary torque model considered in this work is illustrated in Fig. 4. The network contains three different subnetworks. The left subnetwork takes mesoscale information (scalars) as inputs and produces scalars as outputs. The activation function used to apply nonlinearity is ReLU. The middle and right subnetworks are identical in terms of their architecture. The inputs are the vectors $\hat{e}_i$, r_j , and r_k and hyperbolic tangent is used as the activation function. Elementwise cross product is applied among the final output vectors of the middle and right subnetworks, thereby creating pseudovectors. The resulting pseudovectors are involved in an elementwise multiplication with the scalar outputs of the left subnetwork in the succeeding step. These final pseudovectors are then combined together with the aid of a linear (pseudovector

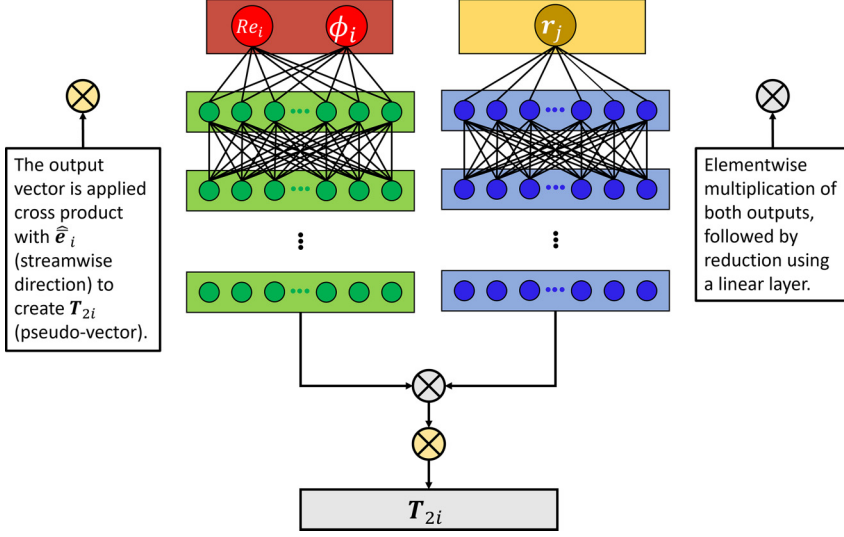


FIG. 3. Binary-interaction torque model network architecture.

summation) layer to produce the trinary torque contribution, T_{3i} . Similar to trinary force model, the hyperparameter tuning was only performed using $M_3 = 10$ neighbors. Based on results presented in Table V, a network with $(L_3, W_3) = (5, 150)$ was selected as the trinary torque model.

D. Errors in machine learning process

In order to quantify the network and generalization errors incurred during the training and deployment of the binary and trinary machine learning models, we consider $F[M_2, 2]$ and $F[M_2, M_3, 3]$ to be the best-trained binary and trinary terms that result in zero network and generalization errors. In other words, $F[M_2, 2]$ and $F[M_2, M_3, 3]$ are the best-trained models that result from perfect neural networks that were trained with unlimited training data. However, as discussed in the above subsections, actual implementations are limited both in terms of the network architecture and the amount of training data. The number of layers and their width (i.e., L and W) are the network architecture limitations and the number of data samples N is the training data limitation. As a result of these limitations, the resulting binary and trinary models will involve additional errors and will be distinguished from the best-trained models. We now introduce the notation $\tilde{F}[M_2, 2; L_2, W_2; N]$ and $\tilde{F}[M_2, M_3, 3; L_2, W_2, L_3, W_3; N]$, where the tilde denotes the fact that the force prediction is based on a finite training data set and network architecture and thus

 TABLE IV. Grid search hyperparameter tuning for T_{2i} with $M_2 = 26$: Test sample torque loss ($\mathcal{L}_{\text{Torque, test}}$) values for different networks.

Nonlinear layers (L_2)	Width of each layer (W_2)			
	50	100	150	200
2	13.4651	12.9478	12.9741	13.1131
3	12.9238	12.3638	12.4767	12.4124
4	13.5391	12.3857	12.4419	12.4785
5	14.7822	12.3424	12.3664	12.4092
6	15.3737	12.4346	12.6441	12.6697

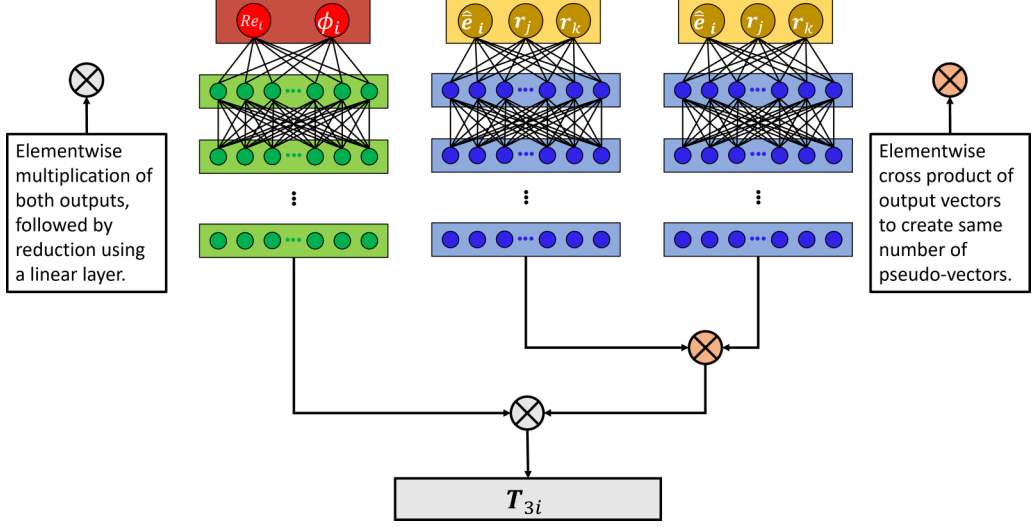


FIG. 4. Trinary-interaction torque model network architecture.

involves additional errors. Furthermore, the details of the network and the number of data samples used in training, validation, and testing is also included.

With this notation, the network and generalization errors of the trinary model can be expressed as

$$\underbrace{F[M_2, M_3, 3] - \tilde{F}[M_2, M_3, 3; L_2, W_2, L_3, W_3, \infty]}_{\text{Network error}}, \quad (17)$$

$$\underbrace{\tilde{F}[M_2, M_3, 3; L_2, W_2, L_3, W_3, \infty] - \tilde{F}[M_2, M_3, 3; L_2, W_2, L_3, W_3, N]}_{\text{Generalization error}}.$$

According to this notation, $F[M_2, M_3, 3]$ can be interpreted as $\tilde{F}$ in the limit $N \rightarrow \infty$, and with infinitely deep and wide (expressive) network layer(s). We now conclude this section with the following remarks. The above notations and errors also apply to the binary and trinary torque models. It should also be noted that the network and the generalization errors are connected. For example, with the limited availability of training data, it does not make sense to employ a very complicated network architecture. Henceforth in our discussion of the results, we will not use the complete notation presented above. We will simply denote the force and torque resulting from the binary model to be $\tilde{F}[2]$ and $\tilde{T}[2]$ and the force and torque resulting from the trinary model to be $\tilde{F}[3]$ and $\tilde{T}[3]$.

 TABLE V. Grid search hyperparameter tuning for T_{3i} with $M_3 = 10$: Test sample torque loss ($\mathcal{L}_{\text{Torque, test}}$) values for different network architectures.

Nonlinear layers (L_3)	Width of each layer (W_3)			
	50	100	150	200
5	6.4323	6.4714	6.4048	6.6055
10	6.5684	6.6677	6.6741	6.7255
15	6.5978	6.8225	6.7167	6.8780

VI. RESULTS AND DISCUSSION

The performance of neural network architectures described in the previous section will be examined below. Several decisions in terms of the number of neighbors to be included in the binary and trinary models and the depth and width of the network layers must be optimally made. These decisions will be made based on a compromise between computational difficulty and error minimization. Particular attention will be focused on the four different sources of errors. Neighbor-truncation error will determine the values of M_2 and M_3 . The choice of $M_2 = 26$ and $M_3 = 10$ will be discussed in the Appendix. The interaction-truncation error will be evaluated by comparing the performance of unary, binary, and trinary models of force and torque, which will be presented in the following sections. The network error has already been considered in the previous section where we have determined the optimal values of the layer depth and width for the different neural networks.

A. Accurate universal predictions

In a typical EL simulation, even under nominally homogeneous conditions, the mesoscale quantities Re_i and ϕ_i evaluated at the particles will show substantial variation across the computational domain. This mesoscale variation is in addition to variations seen in the random arrangement of neighbors at the microscale. Therefore, here we have developed machine learning-based deterministic models that can effectively handle such micro- and mesoscale variations. When deep learning closure models are developed only at fixed (Re, ϕ) there exists an additional task of switching between different networks and interpolating between the different models. A robust universal model that spans a wide range of Re and ϕ will mitigate this issue.

A summary of force and torque predictions for the test samples for all combinations of $\langle \text{Re} \rangle$ and $\langle \phi \rangle$ for the unary, binary, and trinary truncations are shown in Figs. 5 and 6, respectively. The three columns of these figures correspond to x , y , and z components respectively. It can be observed that the characteristics of the y and z components in these figures are similar but they are distinct from those of the x component, which is the direction of the mean flow. The different rows of Fig. 5 represent different hierarchical force models with increasing complexity from top to bottom. In each frame, the ground truth obtained from the PR simulations is plotted along the horizontal coordinate and the corresponding model prediction is plotted along the vertical coordinate. A perfect model will lead to a scatter plot that aligns with the diagonal dash line.

The first row illustrates the predictions of the unary force model. As expected, a unary model that depends only on mesoscale information is able to capture the mean drag of the different data sets, but it is not able to capture the particle-to-particle variation within the different data sets. This explains the horizontal slablike predictions for all three components. A closer look at the slabs indicates that they are positioned around the mean forces, $|\langle \mathbf{F} \rangle|$, presented in Table I. In interpreting the y and z components of forces it should be noted that the local volume-averaged fluid velocity of the particles does not perfectly align along the x axis, and therefore y and z forces cannot be equated to the lift force of the particle, as they may contain a small component of the drag force (i.e., force along the direction of local average flow). The second row shows predictions obtained with a binary force model with $M_2 = 26$. As the deterministic information of closest neighbors is being passed as part of the input feature, the neighbor influence in inducing higher and lower than average forces are predicted quite well. It can be seen from the last row of plots that the predictions are closer to the ground truth with the inclusion of trinary interactions, however, the improvement is not as large as that obtained from the inclusion of the binary interactions. As will be discussed below, this may be due to the limited availability of data, restricting $M_3 = 10$, and the complexity of the trinary network architecture.

Similarly, the first row of Fig. 6 shows the performance of the binary torque model with $M_2 = 26$ for the test samples of all the data sets. The second row corresponds to predictions achieved with the inclusion of trinary interactions with $M_3 = 10$. In the case of torque, significantly more accurate

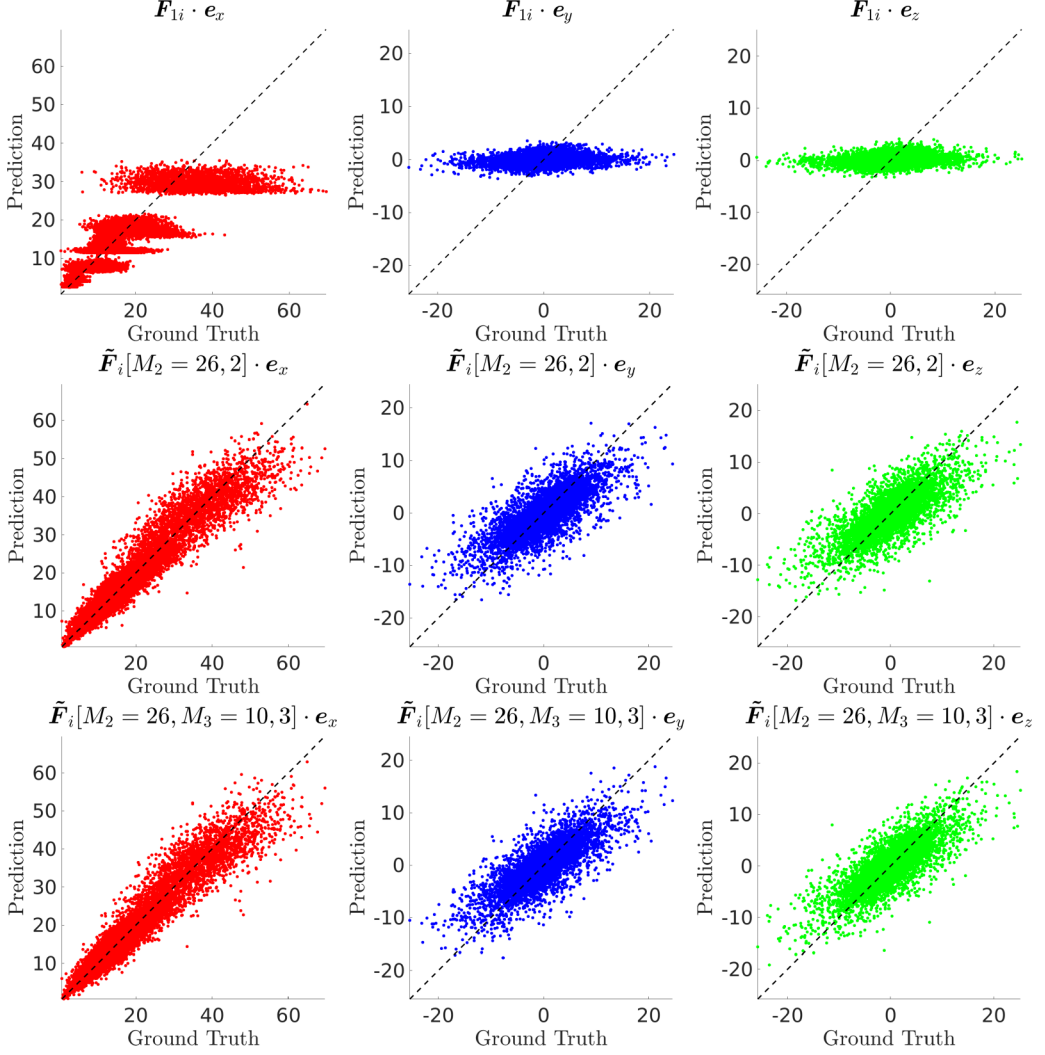


FIG. 5. Universal force predictions using the presented hierarchical approach. The first, second, and third rows corresponds to predictions obtained using $\mathbf{F}_{1i}$, $\tilde{\mathbf{F}}_i[M_2 = 26, 2]$ and $\tilde{\mathbf{F}}_i[M_2 = 26, M_3 = 10, 3]$ for test samples from all data sets.

predictions can be witnessed with the trinary interactions for all three components. Especially, the streamwise component of torque on the particles can be captured only with the trinary model, although the predictive capability of the trinary model is only modest. The quantitative analysis of these predictions and the importance of binary and trinary models will be discussed in the succeeding subsection.

B. Neighbor-induced perturbation capture and interaction-truncation error

The ability of current models in capturing the neighbor-induced fluctuations of each individual data set is quantified using the R^2 metric defined below. Although the flow averaged over the local volumes is not perfectly aligned along the $+x$ axis, the alignment is quite close. Hence, streamwise force predictions are quantified as R_{Drag}^2 and the transverse force predictions are measured as R_{Lift}^2 .

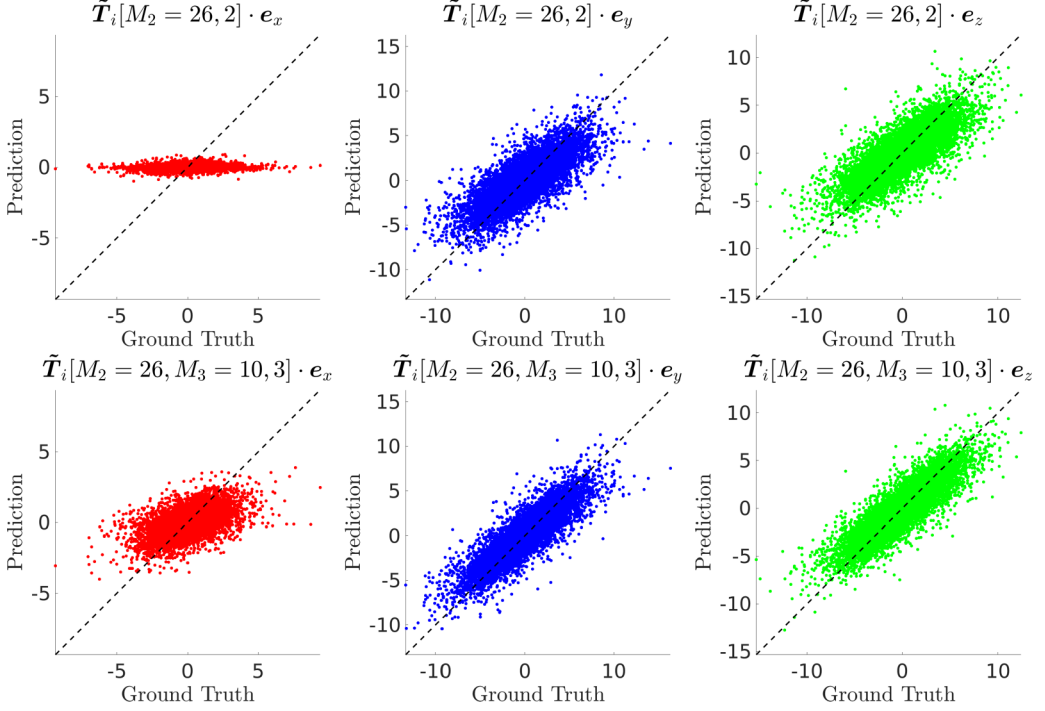


FIG. 6. Universal torque predictions for test samples from all considered data sets. The first row depicts predictions obtained using $\tilde{T}_i[M_2 = 26, 2]$, and the second row corresponds to predictions using $\tilde{T}_i[M_2 = 26, M_3 = 10, 3]$.

Similarly, $R_{\text{Torque}, \parallel}^2$ and $R_{\text{Torque}, \perp}^2$ measure streamwise and transverse torque predictions respectively. A value of unity for R^2 denotes a perfect model that is able to deterministically capture particle-to-particle variations. A value of zero indicates that the model is not capturing any variation. The value can also be negative if a model's predictions are opposite in sign to the ground truth:

$$R_{\text{Drag}}^2 = 1 - \frac{\sum_{i=1}^N (F_{i,(x,\text{PR})} - F_{i,(x,\text{NN})})^2}{\sum_{i=1}^N (F_{i,(x,\text{PR})} - \langle F_{i,(x,\text{PR})} \rangle)^2}, \quad (18)$$

$$R_{\text{Lift}}^2 = 1 - \frac{\sum_{i=1}^N (F_{i,(y,\text{PR})} - F_{i,(y,\text{NN})})^2 + (F_{i,(z,\text{PR})} - F_{i,(z,\text{NN})})^2}{\sum_{i=1}^N (F_{i,(y,\text{PR})} - \langle F_{i,(y,\text{PR})} \rangle)^2 + (F_{i,(z,\text{PR})} - \langle F_{i,(z,\text{PR})} \rangle)^2}, \quad (19)$$

$$R_{\text{Torque}, \parallel}^2 = 1 - \frac{\sum_{i=1}^N (T_{i,(x,\text{PR})} - T_{i,(x,\text{NN})})^2}{\sum_{i=1}^N (T_{i,(x,\text{PR})} - \langle T_{i,(x,\text{PR})} \rangle)^2}, \quad (20)$$

$$R_{\text{Torque}, \perp}^2 = 1 - \frac{\sum_{i=1}^N (T_{i,(y,\text{PR})} - T_{i,(y,\text{NN})})^2 + (T_{i,(z,\text{PR})} - T_{i,(z,\text{NN})})^2}{\sum_{i=1}^N (T_{i,(y,\text{PR})} - \langle T_{i,(y,\text{PR})} \rangle)^2 + (T_{i,(z,\text{PR})} - \langle T_{i,(z,\text{PR})} \rangle)^2}. \quad (21)$$

Binary force predictions with $M_2 = 26$ is reported in Table VI. It can be seen that the learned binary force model is highly generalizable based on the small difference in training and testing performance of all data sets. The binary model indicates negligible underfitting (variation in third decimal) of transverse fluctuations for the data sets $(\langle \text{Re} \rangle, \langle \phi \rangle) = (165.96, 0.1)$, $(49.78, 0.20)$, and $(187.25, 0.20)$. This is probably due to us seeking a global minimum that results in the optimization process underfitting a few data sets at the expense of overfitting the other data sets. For the data sets obtained from Seyed-Ahmadi and Wachs [24], the performance of the present model is either similar or better than those reported in their work. A consistent improvement of 6% to 11% and

TABLE VI. Individual data set R^2 performance using $\tilde{F}_i[M_2 = 26, 2]$ and $\tilde{F}_i[M_2 = 26, M_3 = 10, 3]$ force models for training and testing samples.

$\langle \text{Re} \rangle$	$\langle \phi \rangle$	$\tilde{F}_i[M_2 = 26, 2]$				$\tilde{F}_i[M_2 = 26, M_3 = 10, 3]$			
		Drag		Lift		Drag		Lift	
		Train	Test	Train	Test	Train	Test	Train	Test
9.86	0.10	0.775	0.740	0.797	0.776	0.842	0.762	0.862	0.803
121.36	0.10	0.685	0.645	0.707	0.696	0.787	0.650	0.805	0.701
6.95	0.21	0.816	0.809	0.812	0.800	0.853	0.817	0.862	0.827
73.40	0.21	0.757	0.728	0.746	0.739	0.818	0.745	0.833	0.772
27.81	0.40	0.807	0.788	0.795	0.784	0.837	0.809	0.841	0.823
73.42	0.40	0.759	0.738	0.754	0.738	0.807	0.769	0.817	0.786
2.20	0.10	0.837	0.833	0.803	0.800	0.854	0.838	0.842	0.826
10.92	0.10	0.810	0.807	0.812	0.808	0.826	0.815	0.861	0.851
165.96	0.10	0.624	0.611	0.715	0.718	0.679	0.615	0.759	0.732
0.25	0.20	0.756	0.747	0.757	0.753	0.769	0.758	0.773	0.764
2.48	0.20	0.780	0.773	0.764	0.758	0.791	0.782	0.791	0.784
49.78	0.20	0.747	0.743	0.716	0.719	0.772	0.754	0.795	0.787
187.25	0.20	0.662	0.656	0.668	0.671	0.698	0.661	0.731	0.711
3.25	0.40	0.698	0.672	0.592	0.567	0.722	0.679	0.625	0.584
64.35	0.40	0.744	0.727	0.655	0.643	0.764	0.739	0.717	0.696
245.65	0.40	0.661	0.632	0.574	0.558	0.698	0.649	0.654	0.611

3% to 6% for streamwise and transverse components, respectively, is observed at higher particle volume fraction of $\langle \phi \rangle = 0.4$. This additional improvement is attributed to the strict enforcement of rotational and reflectional symmetries in the present binary model.

The improvement achieved by the inclusion of the trinary term with $M_3 = 10$ can also be inferred from Table VI. It can be observed that the trinary force model consistently improves upon the corresponding binary prediction for all combinations of $\langle \text{Re} \rangle$ and $\langle \phi \rangle$. Furthermore, the difference between training and testing performance for both streamwise and transverse components is larger than the corresponding difference when only a binary model is used. This is in agreement with our understanding that far more training data is required to create a generalizable trinary interaction model than a binary interaction model. This also suggests that the performance of the trinary model can be further improved with the availability of more training data and thus the present models are by no means final.

The performance of the binary torque model with $M_2 = 26$ is presented in Table VII. Only the results for the transverse torque component are presented, due to the fundamental limitation of binary interactions in capturing the streamwise torque component. The learned binary torque model is sufficiently generalizable as indicated by the consistent minor difference between training and testing performance. Surprisingly, the data sets that exhibit slight underfitting in binary force model also show such nature for the binary torque model. The model's training performance indicates that it has difficulty in learning the particular data set $(\langle \text{Re} \rangle, \langle \phi \rangle) = (121.36, 0.10)$. For the data sets obtained from Seyed-Ahmadi and Wachs [24] the performance of the current binary torque model is comparable to that reported by them. The current universal model displays affinity to low Reynolds number cases with a maximum improvement of 15% for $(\langle \text{Re} \rangle, \langle \phi \rangle) = (0.25, 0.20)$. However, this comes at the expense of performance loss at high $\langle \text{Re} \rangle$ with the largest reduction in performance of 9% for the data set $(165.96, 0.10)$. It should be noted that the performance reported in Seyed-Ahmadi and Wachs [24] is based on training and testing carried out on each individual data set, i.e., fixed $(\langle \text{Re} \rangle, \langle \phi \rangle)$, whereas the present model is trained universally to cover a range of $\langle \text{Re} \rangle$ and $\langle \phi \rangle$.

TABLE VII. Individual data set R^2 performance using $\tilde{T}_i[M_2 = 26, 2]$ and $\tilde{T}_i[M_2 = 26, M_3 = 10, 3]$ torque models for training and testing samples.

$\langle \text{Re} \rangle$	$\langle \phi \rangle$	$\tilde{T}_i[M_2 = 26, 2]$		$\tilde{T}_i[M_2 = 26, M_3 = 10, 3]$			
		Torque, $\perp$		Torque, $\parallel$		Torque, $\perp$	
		Train	Test	Train	Test	Train	Test
9.86	0.10	0.752	0.751	0.602	0.469	0.933	0.899
121.36	0.10	0.308	0.285	0.467	0.011	0.835	0.618
6.95	0.21	0.827	0.825	0.699	0.610	0.951	0.929
73.40	0.21	0.496	0.486	0.458	0.234	0.868	0.774
27.81	0.40	0.657	0.647	0.658	0.544	0.937	0.897
73.42	0.40	0.435	0.434	0.571	0.437	0.864	0.804
2.20	0.10	0.899	0.898	0.678	0.630	0.969	0.962
10.92	0.10	0.798	0.794	0.583	0.549	0.928	0.918
165.96	0.10	0.533	0.541	0.181	0.047	0.793	0.741
0.25	0.20	0.896	0.895	0.673	0.647	0.954	0.949
2.48	0.20	0.898	0.896	0.665	0.653	0.958	0.954
49.78	0.20	0.713	0.717	0.415	0.363	0.876	0.864
187.25	0.20	0.595	0.595	0.192	0.120	0.793	0.758
3.25	0.40	0.820	0.813	0.661	0.598	0.916	0.886
64.35	0.40	0.694	0.685	0.451	0.408	0.806	0.777
245.65	0.40	0.565	0.564	0.305	0.208	0.768	0.715

The added benefit of including the trinary term in the torque model is also quantified in Table VII. The trinary model offers a significant improvement over the binary model for the transverse torque component. It can also be observed that the learning deficiency of the binary model for certain data sets, particularly $(\langle \text{Re} \rangle, \langle \phi \rangle) = (121.36, 0.10)$, is addressed by the trinary model. As mentioned earlier the trinary torque model shows promise in capturing the streamwise torque component. This component is learned and predicted better at low $\langle \text{Re} \rangle$ and it perhaps requires more training data to achieve robust predictions at higher $\langle \text{Re} \rangle$. Furthermore, even higher-order interactions may be needed for the prediction of streamwise torque.

C. Statistical analysis of predictions

The ability of presented force and torque models in capturing important statistical quantities (shown in Table I) is evaluated here. The force statistics presented in Table VIII illustrate the disparity between mean drag ($|\langle \mathbf{F} \rangle|$) obtained in the PR simulation and that predicted by the unary force model. As explained earlier, this difference can be partly attributed to the fact that the uniform distribution of N particles in the present data sets was not strictly the same as those employed in the generation of the unary model; i.e., the uniform probability distribution P_N of the current data sets differs from $P_{N,Tn}$. The difference is also partly due to limited samples used for calculating the force average both in the present work as well in the construction of the unary model by Tenneti *et al.* [5].

The binary-interaction model ($\tilde{\mathbf{F}}_{2i}$) ensures that the mean drag is brought substantially close to the actual value for every considered data set. It is important to realize that the binary model achieves this trait for random particle distributions generated in different ways. As the mean drag values are already well captured by the binary force model, the added benefit of the trinary-interaction model ($\tilde{\mathbf{F}}_{3i}$) is generally quite small.

By definition, the unary model predicts the same force for all the particles, and therefore its standard deviation is identically zero. It can be observed that the standard deviation of both the drag and lift forces predicted by the binary model are noticeably smaller than the PR simulation results.

TABLE VIII. Statistical analysis of test sample force predictions obtained using $\mathbf{F}_{1i}$, $\tilde{\mathbf{F}}_i[M_2 = 26, 2]$ and $\tilde{\mathbf{F}}_i[M_2 = 26, M_3 = 10, 3]$ models and their comparison with respective particle-resolved (PR) simulation data.

(Re)	$\langle\phi\rangle$	$ \langle\mathbf{F}\rangle $				σ_{Drag}			σ_{Lift}		
		PR	$\mathbf{F}_{1i}$	$\tilde{\mathbf{F}}_i[2]$	$\tilde{\mathbf{F}}_i[3]$	PR	$\tilde{\mathbf{F}}_i[2]$	$\tilde{\mathbf{F}}_i[3]$	PR	$\tilde{\mathbf{F}}_i[2]$	$\tilde{\mathbf{F}}_i[3]$
9.86	0.10	2.76	3.14	2.84	2.81	0.7823	0.7510	0.7520	0.4661	0.4368	0.4690
121.36	0.10	9.55	7.08	9.54	9.53	3.1110	2.4820	2.5262	1.4986	1.3442	1.4512
6.95	0.21	4.03	4.84	4.02	4.00	0.9581	0.8774	0.8798	0.7454	0.6616	0.6803
73.40	0.21	9.51	8.30	9.54	9.47	2.7987	2.4118	2.4994	1.6178	1.4297	1.4733
27.81	0.40	12.68	14.51	12.69	12.67	2.0613	1.8689	1.9283	1.5028	1.3872	1.4330
73.42	0.40	19.14	18.35	18.99	18.90	3.6087	3.3962	3.5509	2.4135	2.3316	2.4221
2.20	0.10	2.63	2.55	2.63	2.63	0.5425	0.4941	0.4997	0.3973	0.3474	0.3611
10.92	0.10	3.28	3.15	3.19	3.19	0.7351	0.6504	0.6597	0.4839	0.4057	0.4374
165.96	0.10	9.48	8.15	9.53	9.49	2.4191	2.0396	2.0913	1.5662	1.2302	1.3251
0.25	0.20	4.42	4.00	4.42	4.41	0.7971	0.6819	0.6848	0.6737	0.5908	0.6005
2.48	0.20	4.47	4.31	4.47	4.46	0.8116	0.7243	0.7285	0.6707	0.5798	0.5951
49.78	0.20	7.72	7.11	7.75	7.75	1.8194	1.5955	1.5994	1.2608	1.0695	1.1245
187.25	0.20	14.24	12.16	14.21	14.19	3.6989	2.9954	3.0572	2.6873	2.1671	2.2788
3.25	0.40	11.75	11.60	11.72	11.70	2.7535	2.2873	2.3119	2.1329	1.5490	1.5951
64.35	0.40	19.67	17.47	19.73	19.66	5.0963	3.9178	4.0322	3.6885	2.8854	2.9399
245.65	0.40	36.62	29.69	36.47	36.24	9.5692	7.1343	7.4556	7.1215	5.3349	5.4929

As a result, including trinary interactions does lead to improvement in the prediction of standard deviations of streamwise and transverse force components. This property of standard deviation improvement using trinary-interaction torque model is also observed for torque statistics evaluated in Table IX. Despite the observed improvement, there still exists a substantial deficit in the standard deviation when compared to the ground truth. We associate this deficit to several factors including

TABLE IX. Statistical analysis of test sample torque predictions obtained using $\tilde{\mathbf{T}}_i[M_2 = 26, 2]$ and $\tilde{\mathbf{T}}_i[M_2 = 26, M_3 = 10, 3]$ and their comparison with corresponding particle-resolved (PR) simulation data.

(Re)	$\langle\phi\rangle$	$\sigma_{\text{Torque}, \parallel}$			$\sigma_{\text{Torque}, \perp}$		
		PR	$\tilde{\mathbf{T}}_i[2]$	$\tilde{\mathbf{T}}_i[3]$	PR	$\tilde{\mathbf{T}}_i[2]$	$\tilde{\mathbf{T}}_i[3]$
9.86	0.10	0.2407	0.0525	0.2090	0.9253	0.9389	0.9848
121.36	0.10	0.5043	0.0253	0.3621	1.3689	0.8834	1.2477
6.95	0.21	0.4195	0.0991	0.3270	1.2987	1.1596	1.2687
73.40	0.21	0.6441	0.0677	0.4365	1.7103	1.2492	1.6479
27.81	0.40	0.6734	0.0808	0.5222	2.0589	1.6960	2.0102
73.42	0.40	1.0285	0.0885	0.7665	2.5955	1.9596	2.4156
2.20	0.10	0.2375	0.0493	0.1897	0.9241	0.8779	0.9239
10.92	0.10	0.2405	0.0471	0.1654	0.9838	0.7693	0.8692
165.96	0.10	0.4339	0.0208	0.2269	1.1972	0.7821	1.0430
0.25	0.20	0.4295	0.0743	0.3514	1.4509	1.3778	1.4472
2.48	0.20	0.4217	0.0726	0.3324	1.4465	1.3860	1.4377
49.78	0.20	0.5078	0.0575	0.3520	1.6549	1.3675	1.5794
187.25	0.20	0.8446	0.0390	0.4093	2.0219	1.4511	1.7918
3.25	0.40	1.0256	0.0826	0.8007	2.4789	2.2178	2.3772
64.35	0.40	1.3063	0.1911	0.7421	2.9899	2.0702	2.3461
245.65	0.40	2.0385	0.1246	1.0671	3.9776	3.1240	3.2752

not sufficient training of the trinary interactions and the neglect of even higher-order interactions. Furthermore, the intertwined errors elaborated in the Introduction and the lower emphasis placed by the loss function on extreme or rarely occurring configuration samples in the data sets may contribute to the difference.

The force perturbation induced by the binary-interaction model for a sample is given by $\tilde{F}_i[M_2, 2] - F_{1i}$, and similarly the added perturbation induced by the trinary-interaction model can be expressed as $\tilde{F}_i[M_2, M_3, 3] - \tilde{F}_i[M_2, 2]$. A statistical measure of these force perturbations learned through $\tilde{F}_{2i}$ and $\tilde{F}_{3i}$ of each data set is obtained as their rms value. These rms values in turn denote how much the binary and trinary terms contribute to the overall force and torque values. Thus, the relative importance of the trinary term compared to the binary term can be measured from the ratio of rms of trinary interaction to rms of binary interaction. This ratio for the streamwise force of the different cases is in the range [0.13, 0.26], and the corresponding range for the transverse force is [0.16, 0.38]. Thus, it can be concluded that the trinary force term contributes to only 10% to 30% of the binary term. The rms ratio of the transverse torque component is observed to be in the range [0.26, 1.13]. The comparatively larger range for torque is mainly due to the binary torque model underperforming for a data set (see Table VII). The corresponding ratio for the streamwise torque component is not computed due to the limitation that the binary term cannot account for this quantity.

VII. INTERPRETATION OF THE MACHINE-LEARNED MODELS

This section will explore physical interpretations of the machine-learned binary and trinary models of force and torque. The advantage of the hierarchical approach over a direct approach to force and torque modeling based on M nearest neighbors is that we can clearly interpret the binary influence of a neighbor, and the trinary influence of each pair of neighbors, and so on in a systematic manner.

A. Binary force and torque maps

The low dimensionality of binary interaction models means that they can be easily visualized and interpreted. The physics of flow over a two-body system can be essentially visualized using a plane that is formed by the mean-flow direction (streamwise direction) and the line connecting the centers of both particles. Binary-interaction maps for streamwise force (drag), transverse force (lift) and transverse torque components are plotted in Figs. 7–9, respectively. The maps shown in these figures are averaged over 5 different versions of the respective models that are a product of ($k = 5$)-fold cross-validation. These maps are to be interpreted in the following manner. The black-filled circle of unit nondimensional diameter at the center represents the reference particle on the streamwise-transverse plane. The white wider circle indicates the region where a neighboring particle's center cannot exist due to nonoverlapping condition. The value at any other location in a map denotes the influence a neighbor, which has its center at that position, would have on the reference particle for the quantity of interest. Perturbation maps for $Re_i = \{1, 10, 100, 250\}$ at $\phi_i = \{0.10, 0.20, 0.40\}$ with Re_i increasing along rows and ϕ_i along columns are shown in these figures. Among the considered data sets the farthest $M_2 = 26$ th neighbor for $\langle\phi\rangle = \{0.10, 0.20, 0.40\}$ is at $\{3.93, 3.17, 2.41\}$ respectively. Dashed circles with the above-mentioned radius at respective volume fractions are drawn on the maps. The binary maps have been properly learned only in regions within these dashed circles.

The force perturbation induced by a neighbor in a two-body system exists only in the plane formed by the streamwise direction and the line joining particle centers. The universal nature of the models presented in this work enables one to easily plot maps for Re_i and ϕ_i combinations that do not precisely exist in training data sets. The force maps depict smooth contours closer to the reference particle where there is a higher probability for the considered $M_2 = 26$ neighbors to appear. However, the region further away from the reference particle, where the influence

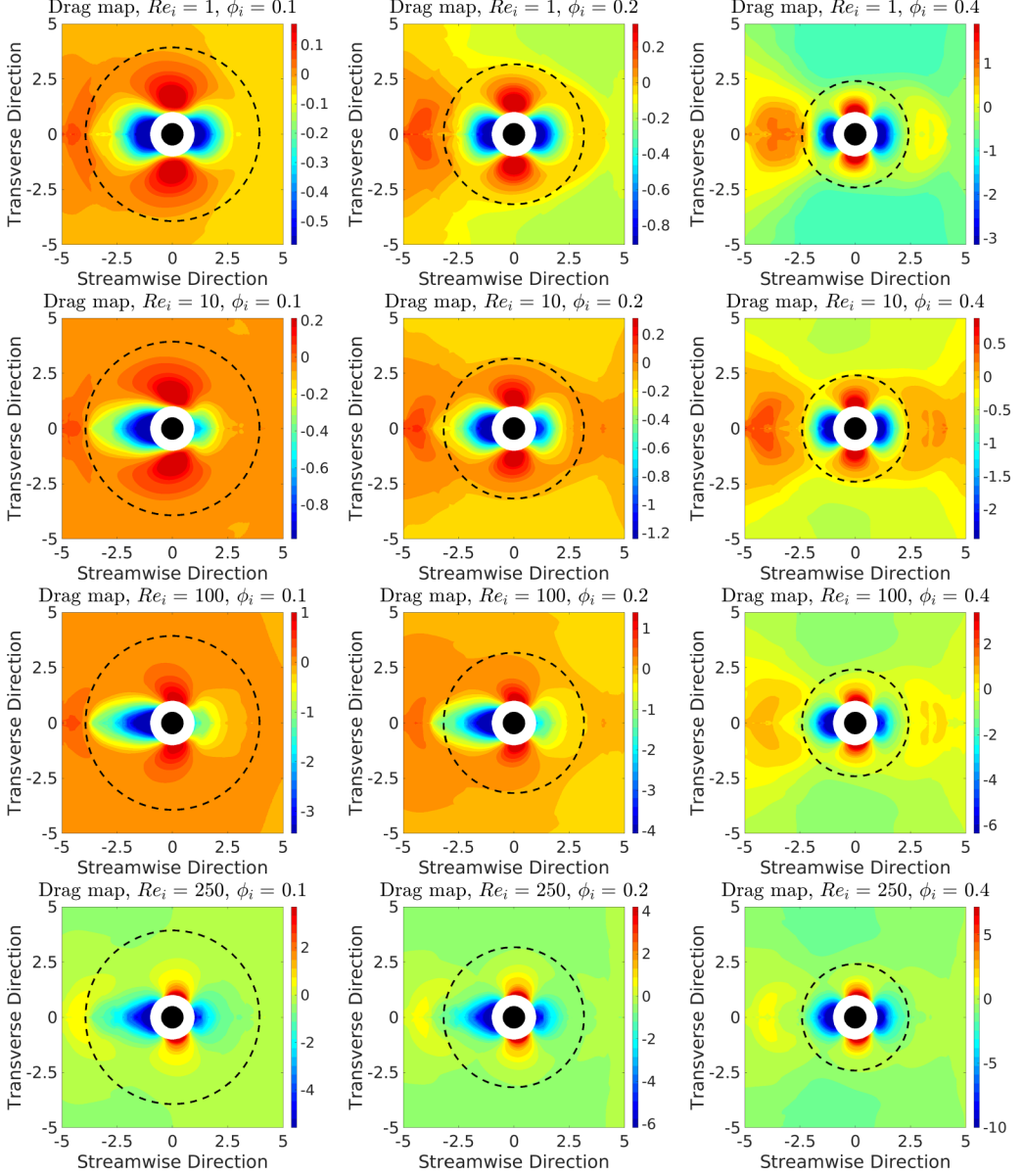


FIG. 7. Binary-interaction drag (streamwise force component) maps obtained using $\tilde{\mathbf{F}}_{2i}$ with $M_2 = 26$.

is traditionally expected to become negligible, has unphysical values due to either rare or no occurrence of neighbors in the training data. This provides a clear idea on the range of applicability of the binary maps. It can also be observed that the drag and lift maps at a fixed Reynolds number have a similar shape for different volume fractions. The blue regions upstream and downstream of the reference particle (i.e., to the left and right of the black circle) represent the negative binary influence of the neighbor on the drag of the reference particle. Thus, the binary influence of an upstream or downstream neighbor contributes to drag reduction, whereas the binary influence of a neighbor to the side contributes to drag increase. In the lift map, only off-axis neighbors have an influence on the lift force. An off-axis neighbor located upstream of the reference particle results

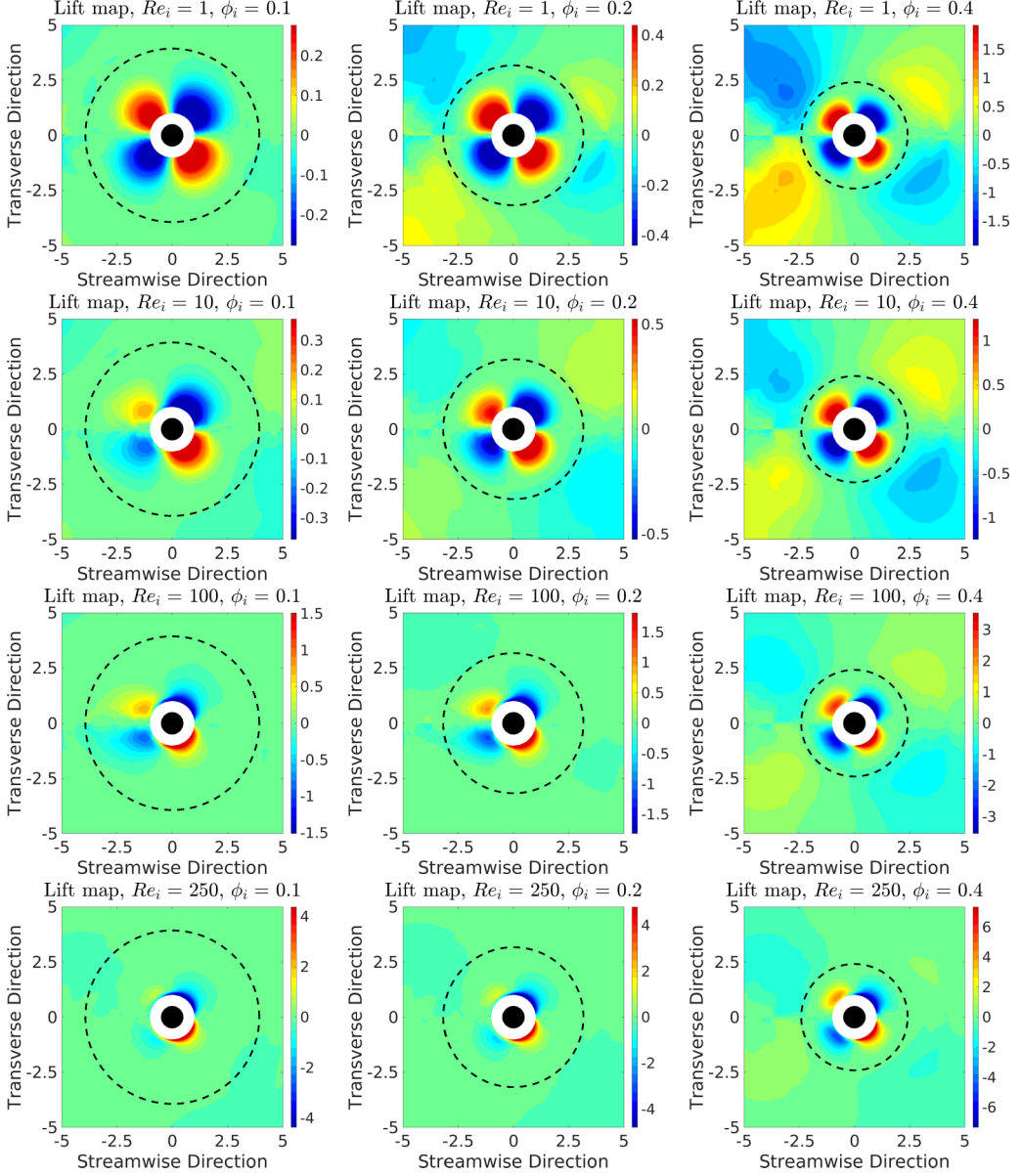
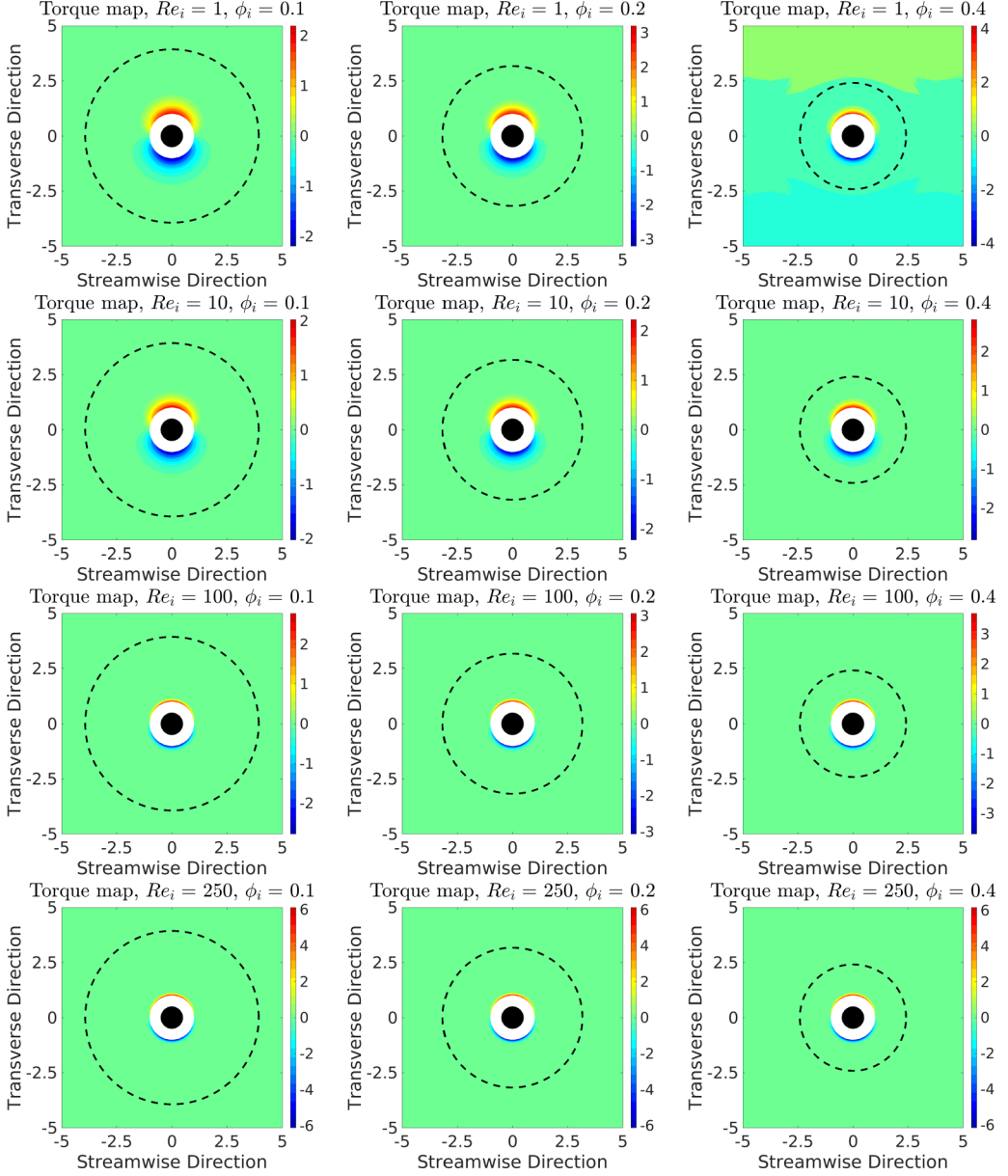


FIG. 8. Binary-interaction lift (transverse force component) maps obtained using $\tilde{\mathbf{F}}_{2i}$ with $M_2 = 26$.

in a binary influence, whose transverse force is directed toward the neighbor. Whereas, an off-axis neighbor located downstream results in a transverse force that is directed away from the neighbor. It can be observed that at low Reynolds numbers, upstream and downstream neighbors have nearly similar drag and lift influence, indicating fore-aft symmetry. This symmetry is broken at higher Reynolds numbers, where an upstream neighbor has a stronger effect on drag, while a downstream neighbor has a stronger lift influence. These binary maps are similar to those obtained with the PIEP model Akiki *et al.* [19] as well to those obtained by Seyed-Ahmadi and Wachs [24].

Torque perturbation induced by a neighbor in the two-body setup is directed perpendicular to the considered plane. Interestingly the range of influence of torque on the reference particle at a given (Re_i, ϕ_i) is considerably smaller than the corresponding influence range for force. Therefore,


 FIG. 9. Binary-interaction torque maps obtained using $\tilde{T}_{2i}$ with $M_2 = 26$.

satisfactory torque predictions can be obtained using fewer neighbors compared to force predictions. A detailed quantitative analysis of neighbor-truncation error for force and torque predictions is presented in Appendix B.

It can be observed that binary maps for conditions outside of training data are also included in Figs. 7–9. For example, the maps at $(Re_i, \phi_i) = (250, 0.10)$, and $(250, 0.20)$ can be perceived as an extrapolation from the case $(250, 0.40)$ that was included in the training data set. The maps appear to follow the respective characteristic trends observed in the other maps. As the volume fraction decreases at a fixed Reynolds number, the force maps display an increase in the radius of influence and a decrease in perturbation magnitude. On the other hand, it is evident from $(\phi_i = 0.40)$

maps that the models produce unphysical results even slightly outside of the dashed circle, i.e., for $|\mathbf{r}_j| > 2.41$. This suggests that the machine-learned models have difficulty in extrapolating trends from seen to unseen (lower to higher) volume fractions at intermediate particle separations that are just beyond the training region. Based on this discussion we believe that the models have a higher generalizability for unseen Reynolds numbers than unseen particle volume fractions.

B. Trinary force and torque maps

As an example, the trinary interaction maps are shown for $(\text{Re}_i, \phi_i) = (50, 0.20)$. The three-body interaction problem must be visualized in 3D by fixing the position of one neighbor (called the *first neighbor* along with the reference particle, while varying the position of the *second neighbor*. The reference particle will be represented using red and the first neighbor will be shown using white. We illustrate for three different locations of the first neighbor. The chosen positions of the first neighbor relative to the reference particle are $(-1.1, 0, 0)$, $(1.1, 0, 0)$, and $(0, 1.1, 0)$, where the mean-flow (streamwise) direction is along $+x$. The reference and first neighbor particles are visualized at twice their actual size, to hide regions where the third particle's center cannot exist (nonoverlapping condition). This explains the intersection of the red and white spheres in the figure. Similar to the binary maps, the trinary maps are also averaged over five independent versions of the model as a result of $(k = 5)$ -fold cross-validation. Furthermore, the maps only depict spatial regions where the models have been trained, i.e., interior of the farthest location of neighbor from $M_3 = 10$ at $\langle \phi \rangle = 0.20$. The value at a given spatial location indicates the influence of the second neighbor with its center at that point along with the first neighbor would have on the reference particle.

The force perturbations induced by the trinary-interaction term $\tilde{\mathbf{F}}_{3i}$ with $M_3 = 10$ for the three configurations are shown in Figs. 10–12. It can be clearly seen that the presented maps preserve symmetries perfectly. It can also be observed that the physics learned by the trinary-interaction model for all three configurations is mainly close to the reference particle on the upstream side. Interestingly this is true even when the first neighbor is located downstream. Interpretation of the trinary maps requires some care. The total perturbing influence of the first and the second neighbors (whose positions relative to the reference particle are known) can be obtained by adding the binary influence of the two neighbors taken one at a time and the trinary influence of them acting together. Thus, the trinary map must be properly interpreted as a correction to the sum of perturbations induced by the two neighbors as predicted by the binary maps.

The trinary maps are noisier than the binary maps even in the region of training. It is quite reasonable to expect the trinary maps, even when fully converged, to appear far more complex than the binary maps. However, well-trained maps will not suffer from “noise.” The noise observed in the trinary maps is partly due to the reason that many combinations of trinary interactions in terms of neighbor locations have not been explored with the limited training data. Hence, the trinary maps in these unexplored regions are not well-trained. Nevertheless, the trinary model obtained in the present work clearly attempts to capture key aspects of the underlying physics, as indicated by the consistent improvement seen in the R^2 values with the inclusion of the trinary term. Further improvement of the trinary term is possible with the availability of additional training data.

The torque fluctuation maps $\tilde{\mathbf{T}}_{3i}$ for $M_3 = 10$ are shown in Figs. 13–15. As the flow direction and the line of separation to the first neighbor are parallel in both the upstream (Fig. 13) and downstream (Fig. 14) configurations, the trinary model by construction should not produce any torque fluctuation along the streamwise direction for any positioning of the second neighbor, which can be observed to be satisfied in the torque maps. Unlike the trinary force maps, the trinary torque maps exhibit more coherent physics in regions that are in close proximity to both the reference particle and the first neighbor.

VIII. CONCLUSIONS AND FUTURE SCOPE

Euler-Lagrange (EL) methodology is an accurate and scalable approach that can be utilized to simulate particle-laden flow systems of many natural and industrial applications. The method

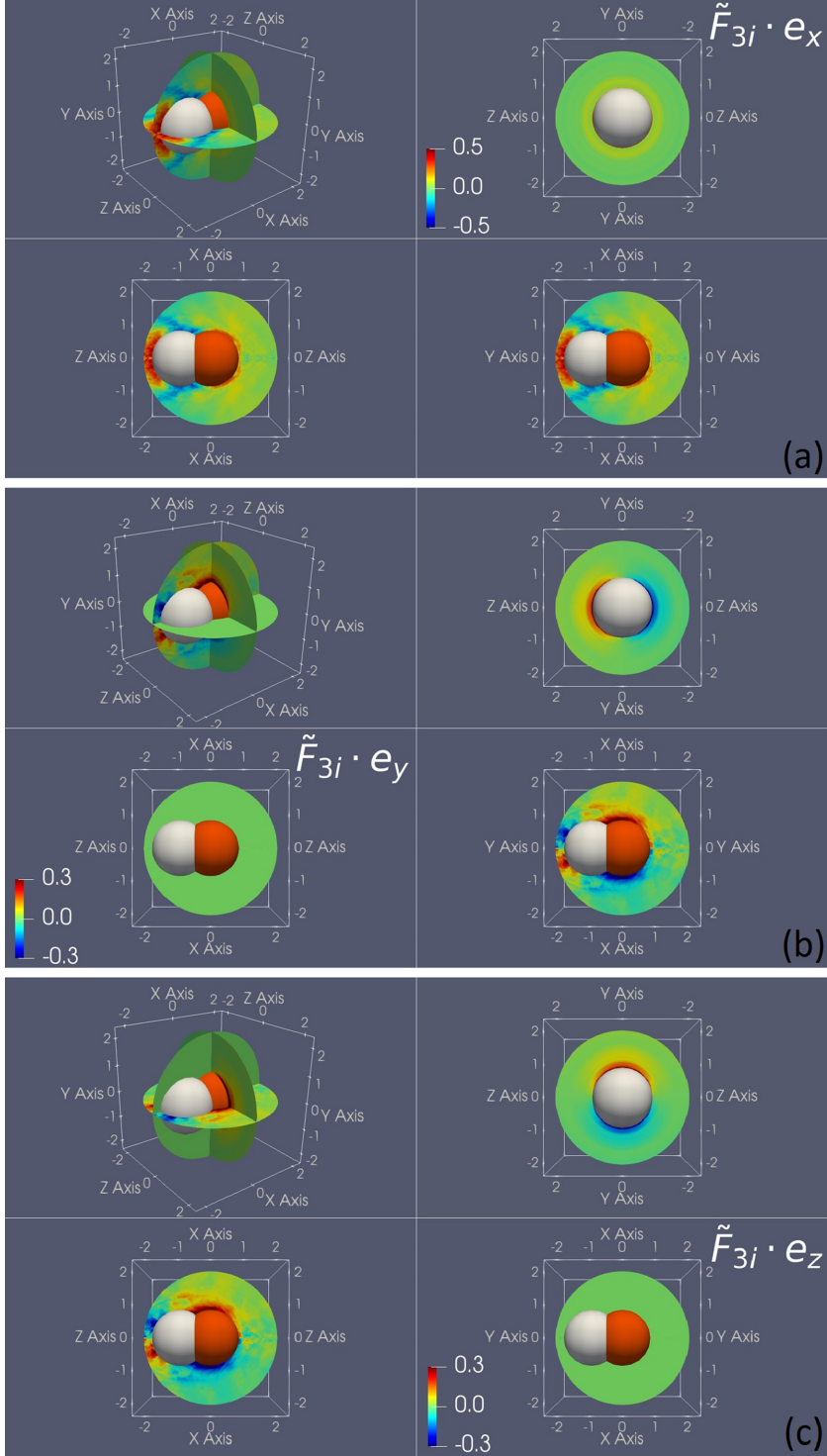


FIG. 10. Trinary-interaction force influence with upstream first neighbor (denoted with white) at $(-1.1, 0, 0)$ relative to reference particle (represented with red). (a) Force perturbation along x ($\tilde{\mathbf{F}}_{3i} \cdot \mathbf{e}_x$), (b) force perturbation along y ($\tilde{\mathbf{F}}_{3i} \cdot \mathbf{e}_y$), and (c) force perturbation along z ($\tilde{\mathbf{F}}_{3i} \cdot \mathbf{e}_z$).

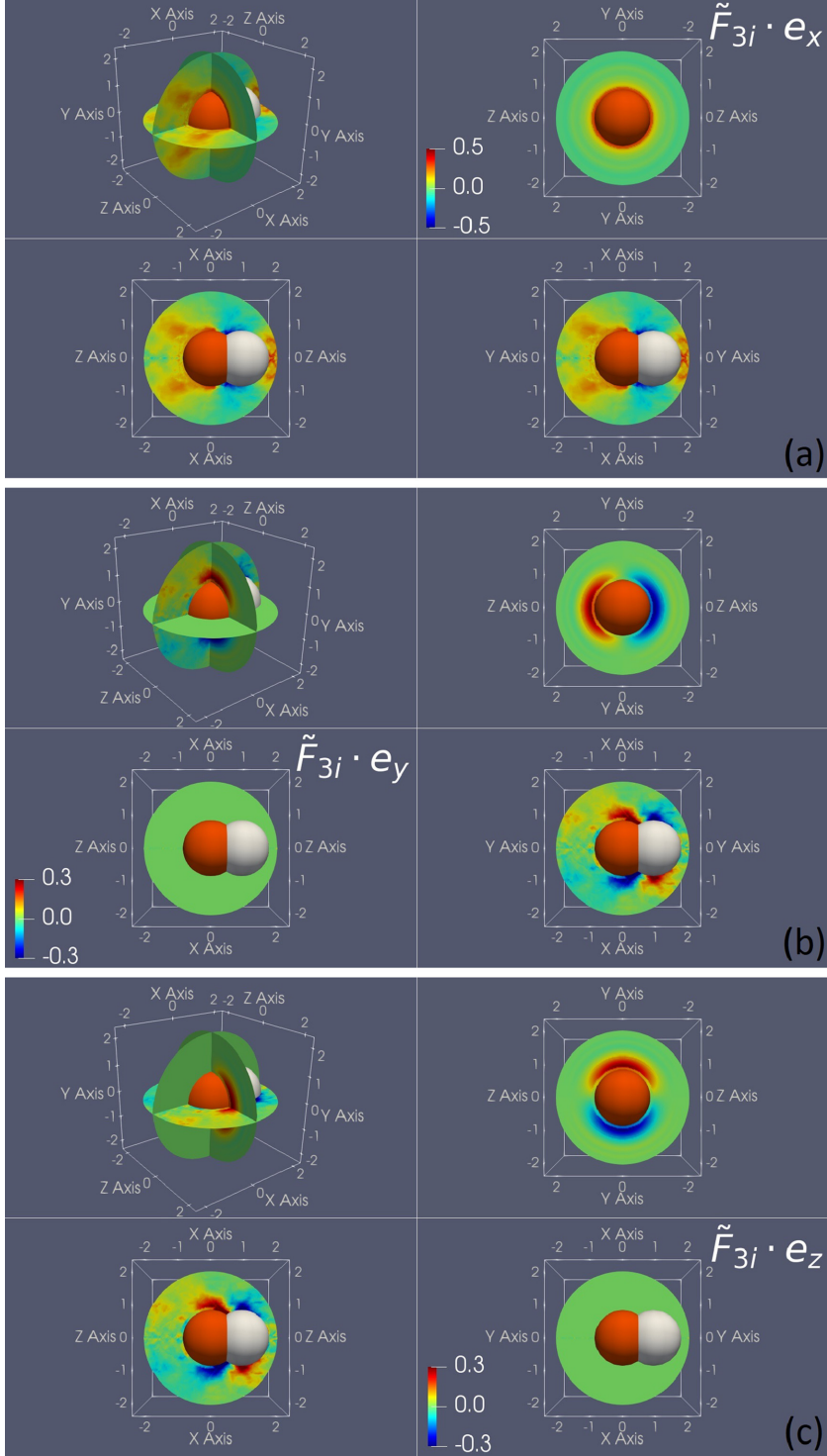


FIG. 11. Trinary-interaction force influence with downstream first neighbor (denoted with white) at (1.1, 0, 0) relative to reference particle (represented with red). (a) Force perturbation along x ($\tilde{F}_{3i} \cdot e_x$), (b) force perturbation along y ($\tilde{F}_{3i} \cdot e_y$), and (c) force perturbation along z ($\tilde{F}_{3i} \cdot e_z$).

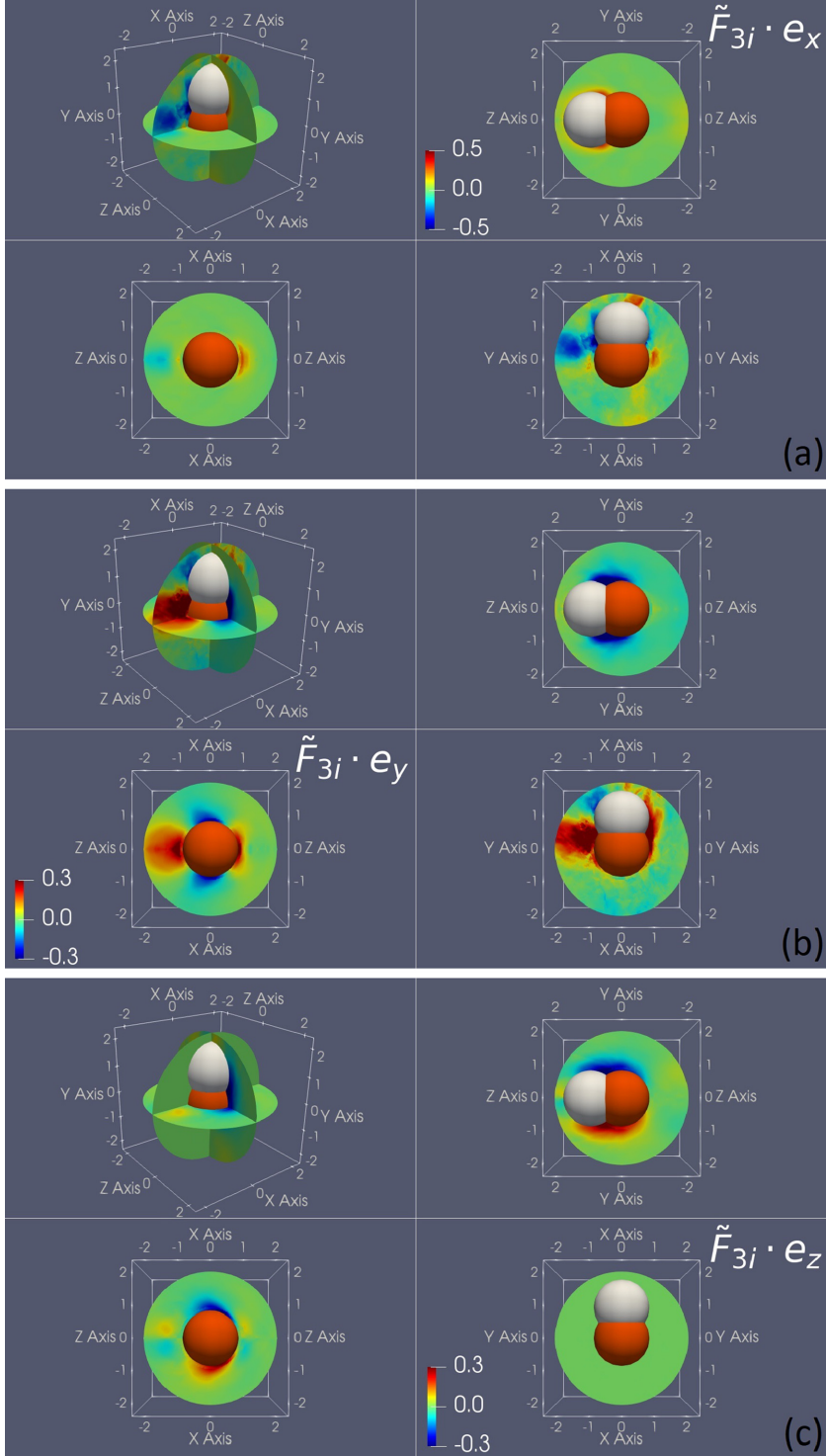


FIG. 12. Trinary-interaction force influence with transverse first neighbor (denoted with white) at $(0, 1.1, 0)$ relative to reference particle (represented with red). (a) Force perturbation along x ($\tilde{\mathbf{F}}_{3i} \cdot \mathbf{e}_x$), (b) force perturbation along y ($\tilde{\mathbf{F}}_{3i} \cdot \mathbf{e}_y$), and (c) force perturbation along z ($\tilde{\mathbf{F}}_{3i} \cdot \mathbf{e}_z$).

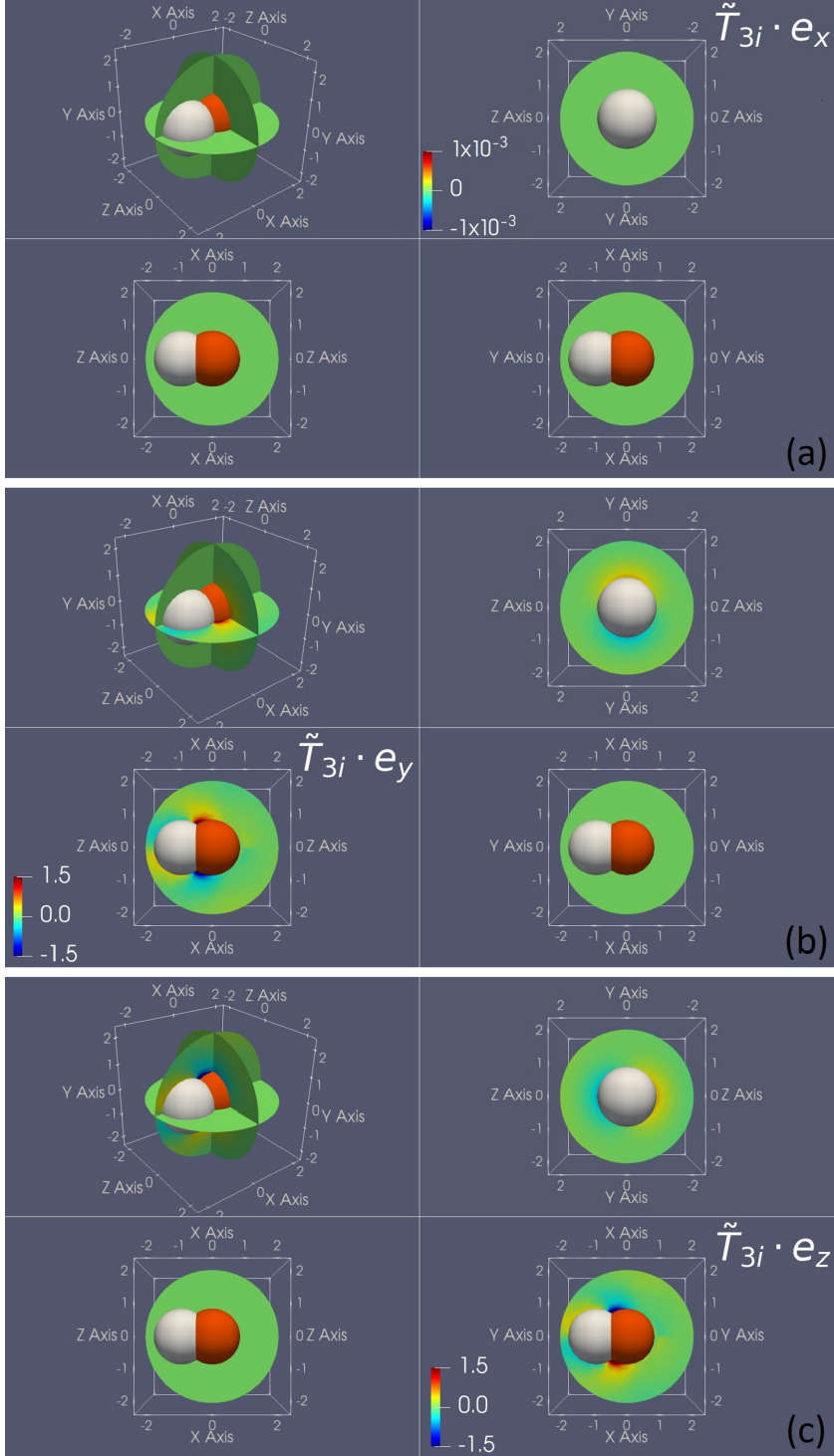


FIG. 13. Trinary-interaction torque influence with upstream first neighbor (denoted with white) at $(-1.1, 0, 0)$ relative to reference particle (represented with red). (a) Torque perturbation along x ($\tilde{T}_{3i} \cdot e_x$), (b) torque perturbation along y ($\tilde{T}_{3i} \cdot e_y$), and (c) torque perturbation along z ($\tilde{T}_{3i} \cdot e_z$).

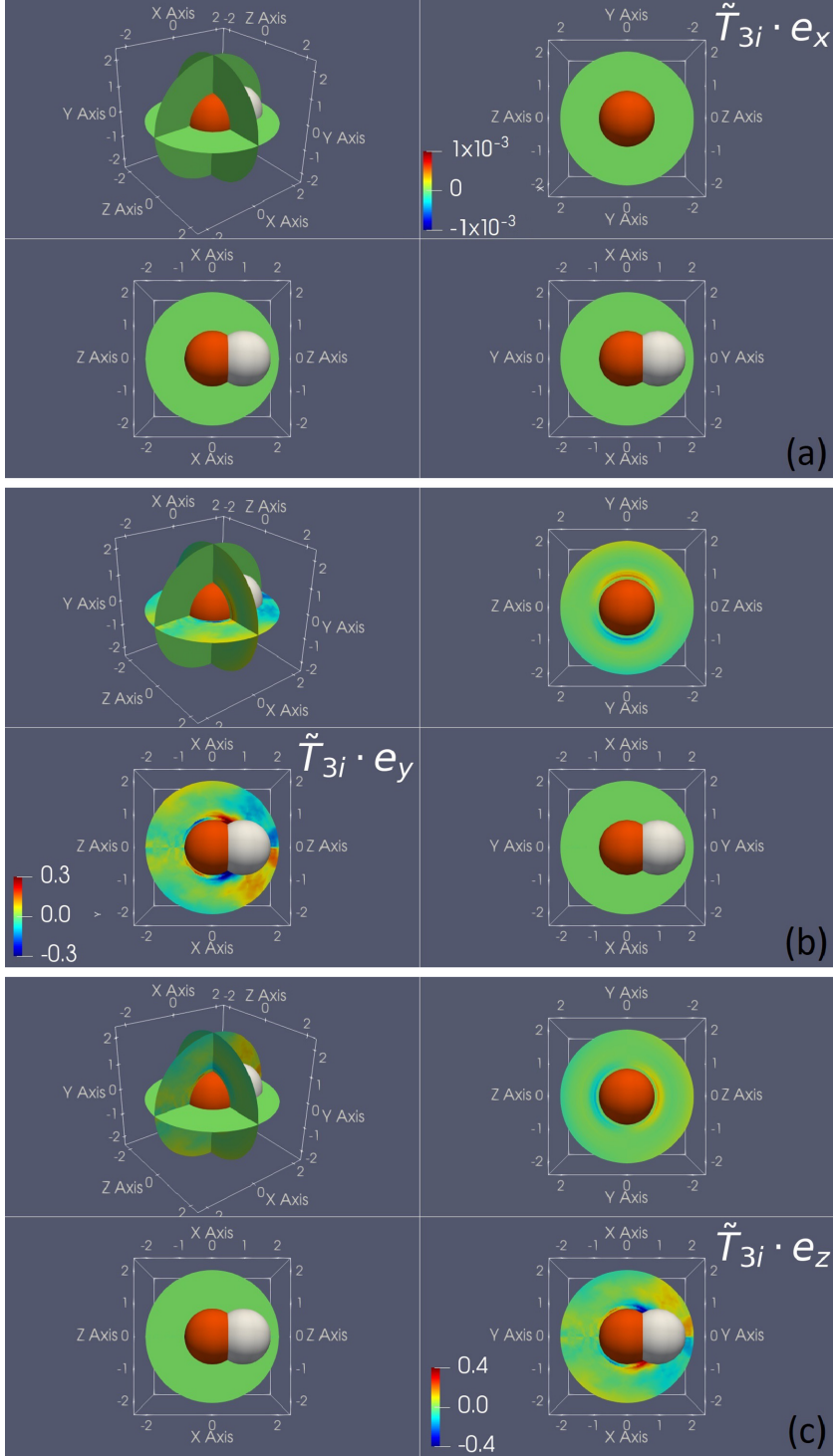


FIG. 14. Trinary-interaction torque influence with downstream first neighbor (denoted with white) at (1.1, 0, 0) relative to reference particle (represented with red). (a) Torque perturbation along x ($\tilde{T}_{3i} \cdot e_x$), (b) torque perturbation along y ($\tilde{T}_{3i} \cdot e_y$), and (c) torque perturbation along z ($\tilde{T}_{3i} \cdot e_z$).

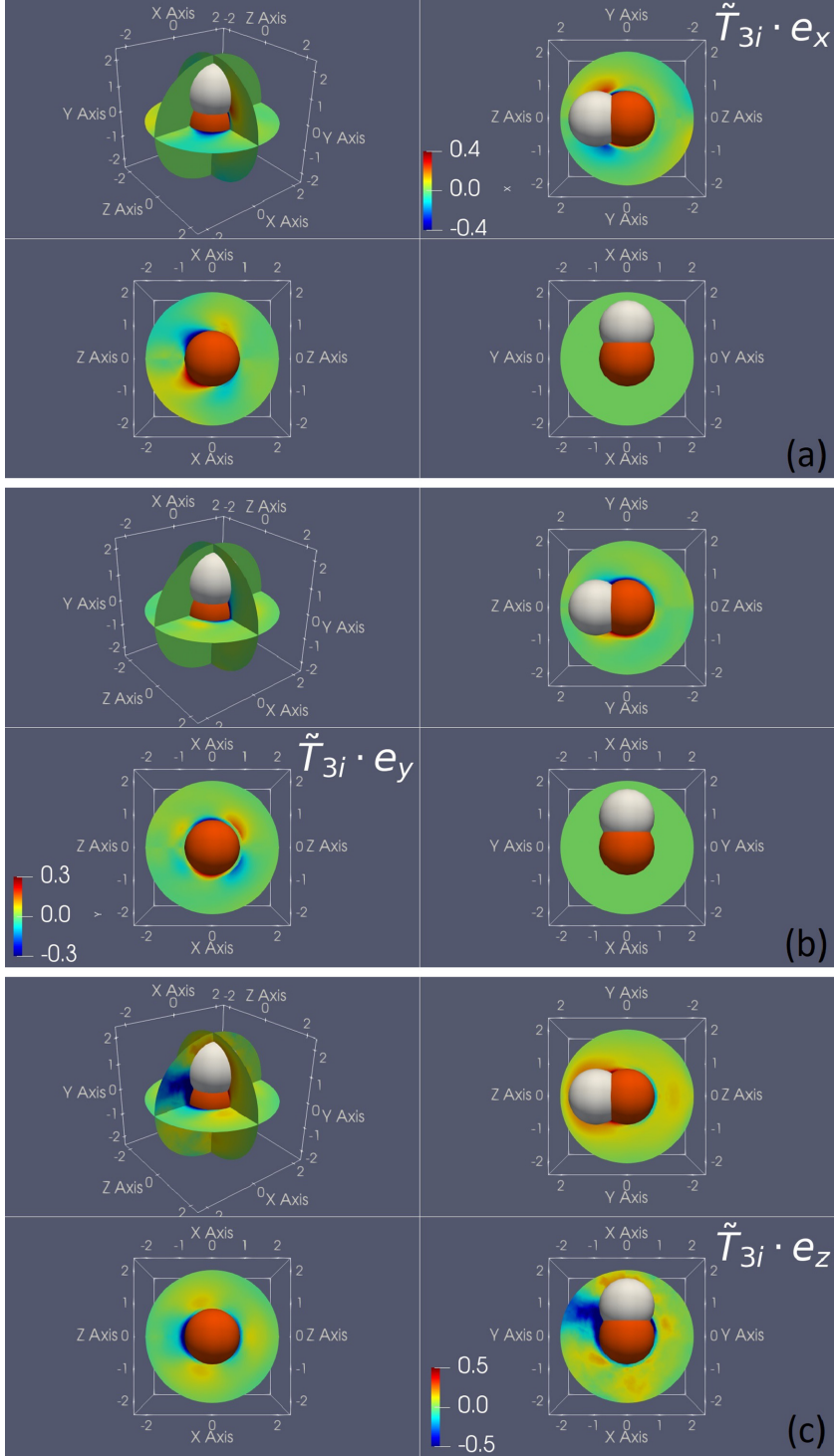


FIG. 15. Trinary-interaction torque influence with transverse first neighbor (denoted with white) at $(0, 1, 1, 0)$ relative to reference particle (represented with red). (a) Torque perturbation along x ($\tilde{T}_{3i} \cdot e_x$), (b) torque perturbation along y ($\tilde{T}_{3i} \cdot e_y$), and (c) torque perturbation along z ($\tilde{T}_{3i} \cdot e_z$).

achieves a desirable balance of accuracy and computational cost, primarily by considering the particles of the system as point-particles. The point-particle assumption comes with the requirement of closure models for particle forces and torques. The accuracy of point-particle closures determines the ability of an EL simulation to replicate actual flow dynamics. Standard EL approaches only consider mesoscale information of Reynolds number (Re) and particle volume fraction (ϕ), and do not consider the deterministic influence induced by nearby particles (neighbors) in approximating force and torque of a particle of interest (reference particle). However, recent studies have elucidated the importance of neighbor-induced perturbations and several data-driven modeling efforts with aid of particle-resolved (PR) simulations have developed deterministic force and torque models that take into account the neighbor interactions. A consistent conclusion of these studies has been that to achieve robustness it is important to integrate underlying physics in the prediction process in order to overcome the paucity of PR simulation data.

Including the effect of neighbors in a deterministic way leads to a multibody interaction problem. Past physics-based efforts have limited the interaction to only pairwise leading to a binary approximation. The current work presents a hierarchical approach that systematically considers terms of increasing complexity, starting from unary, to binary, to trinary interaction model. We also present neural network architectures that implement these hierarchical interactions to provide accurate predictions for a large range of mesoscale variables. While Galilean, translational and scale invariances are preserved by the theoretical formulation, the neural networks have been designed to preserve rotational and reflectional symmetries. These unique characteristics ensure that the systematic approach achieves robustness even with limited training data.

The current method demonstrates excellent universal predictions for $0.25 \leq Re \leq 250$ and $0 \leq \phi \leq 0.4$. The systematic approach of considering different levels of interactions separately has provided interpretability, and thereby yielding a thorough understanding of the models' applicability. The influence of each individual neighbor (binary interaction), and the influence of a pair of neighbors (trinary interaction) on a reference particle are visualized to understand the physics of neighbor perturbation and exemplify the interpretability of the hierarchical approach.

The success of the current approach in achieving accurate predictions for particle distributions obtained from two different sources provides promise for the inclusion of other homogeneous particle distributions such as structured or ordered particle arrays in training and testing the current networks. The generalizable binary-interaction models achieved in this work might suggest that reasonably accurate binary-interaction models can be produced from $O(10^3)$ particle systems for each Reynolds number and particle volume fraction combination. Performing enhanced EL simulations using these rich models and evaluating their performance is an important future direction. Also, developing stochastic Euler-Euler (EE) models based on these improved EL simulations is also an exciting idea.

The data and code that support the findings of this study are available in the GitHub repository [48]. The data presented in this repository are a curated version of particle-resolved simulation data from different sources [10,24,40,49].

ACKNOWLEDGMENTS

This work was supported by the Office of Naval Research (ONR) as part of the Multidisciplinary University Research Initiatives (MURI) Program, under Grant No. N00014-16-1-2617. This work also partly benefited from the U.S. Department of Energy, National Nuclear Security Administration, Advanced Simulation and Computing Program, as a Cooperative Agreement to the University of Florida under the Predictive Science Academic Alliance Program, under Contract No. DE-NA0002378. B.S. would like to thank Majid Allahyari for generously sharing particle-resolved simulation data generated using NekIBM. We also thank Arman Seyed-Ahmadi and Anthony Wachs for generously providing access to their data.

APPENDIX A: GENERAL MODEL FORMS OF HIERARCHICAL APPROACH

Typical applications of particle-laden flows involve moving particles that exhibit relative motion among themselves. In these flows, the hydrodynamic force experienced by a particle of interest depends also on each neighbor's relative motion in addition to the neighbor's position. Therefore, a general formulation of the hierarchical framework that takes neighbors' relative translational and rotational motion into account is presented here. Two different Reynolds numbers can be defined based on translational and angular relative velocity as

$$\text{Re}_i = \frac{|\bar{\mathbf{u}}_{f,i} - \mathbf{V}_i| \rho d_p}{\mu}, \quad \text{Re}_{\Omega,i} = \frac{|(\bar{\boldsymbol{\omega}}_{f,i}/2) - \boldsymbol{\Omega}_i| \rho d_p^2}{\mu}, \quad (\text{A1})$$

where $\mathbf{V}_i$ and $\boldsymbol{\Omega}_i$ are the translational and angular velocities of the i th particle, and $\bar{\mathbf{u}}_{f,i}$ and $\bar{\boldsymbol{\omega}}_{f,i}$ are the locally averaged fluid velocity and vorticity.

The nondimensional hydrodynamic force acting on the i th particle considering all the N neighbors is represented by the following series expansion:

$$\begin{aligned} & \mathbf{F}_i(\text{Re}_i, \phi_i, \text{Re}_{\Omega,i}, \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\}, \{\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_N\}, \{\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \dots, \boldsymbol{\Omega}_N\}) \\ &= \mathbf{F}_{1i}(\text{Re}_i, \phi_i, \text{Re}_{\Omega,i}) + \sum_{j=1}^N \mathbf{F}_{2i}(\text{Re}_i, \phi_i, \text{Re}_{\Omega,i}, \mathbf{r}_j, \mathbf{V}_j, \boldsymbol{\Omega}_j) \\ &+ \sum_{j=1}^{N-1} \sum_{k=j+1}^N \mathbf{F}_{3i}(\text{Re}_i, \phi_i, \text{Re}_{\Omega,i}, \mathbf{r}_j, \mathbf{r}_k, \mathbf{V}_j, \mathbf{V}_k, \boldsymbol{\Omega}_j, \boldsymbol{\Omega}_k) + \dots, \end{aligned} \quad (\text{A2})$$

where $\{\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_N\}$ is the nondimensional translational velocity of the N neighbors relative to the i th particle, and $\{\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \dots, \boldsymbol{\Omega}_N\}$ is nondimensional relative angular velocity of neighbors. The translational velocities are normalized using $\mu/(\rho d_p)$, and the angular velocities are scaled with $\mu/(\rho d_p^2)$.

APPENDIX B: NEIGHBOR-TRUNCATION ERROR ANALYSIS

The influence of neighbors considered in binary and trinary interactions is expected to be different for each output quantity. Hence, it is important to quantify the improvement achieved by adding additional neighbors so that important deployment decisions can be made based on desired accuracy and computational cost associated with it. In this section we provide a neighbor-truncation analysis for binary and trinary force and torque models. The strategy of ($k = 5$)-fold cross-validation is employed to obtain all results presented below.

1. Binary models neighbor-truncation error

As part of neighbor-truncation error analysis for the binary term we investigate the performance obtained using $M_2 = \{10, 15, 20, 26\}$. The models are trained and tested with the same number of neighbors, i.e., M_2 is fixed. Training and testing performance of each individual data set, in terms of R^2 , achieved with different M_2 values for binary force model ($\tilde{\mathbf{F}}[2]$) are tabulated in Table X. It is clear that both training and testing performance increases with an increase in M_2 for every data set. This suggests that the learnability and generalizability increase with increasing number of neighbors. The average test performance over all data sets for drag and lift components is mentioned below, and the values are presented for an ascending order of M_2 . $\bar{R}_{\text{Drag, test}}^2 = \{0.601, 0.667, 0.701, 0.728\}$ and $\bar{R}_{\text{Lift, test}}^2 = \{0.623, 0.686, 0.705, 0.721\}$. It can be observed that the net improvement achieved between consecutive values of M_2 is decreasing, and the final increment only being around 2%–3% for both components. This is an indication that adding more neighbors ($M_2 > 26$) will yield decreasing improvement for the binary force model.

TABLE X. Neighbor-truncation error analysis of $\tilde{\mathbf{F}}[2]$ model presented for each data set in terms of R^2 .

$\langle \text{Re} \rangle$	$\langle \phi \rangle$	$\tilde{\mathbf{F}}_i[M_2 = 10, 2]$						$\tilde{\mathbf{F}}_i[M_2 = 15, 2]$						$\tilde{\mathbf{F}}_i[M_2 = 20, 2]$						$\tilde{\mathbf{F}}_i[M_2 = 26, 2]$					
		Drag		Lift				Drag						Drag						Drag					
9.86	0.10	0.669	0.611	0.730	0.708	0.712	0.681	0.770	0.750	0.743	0.708	0.783	0.763	0.775	0.740	0.797	0.776								
121.36	0.10	0.599	0.542	0.677	0.663	0.635	0.595	0.690	0.681	0.664	0.634	0.703	0.687	0.685	0.645	0.707	0.696								
6.95	0.21	0.663	0.633	0.716	0.697	0.754	0.736	0.768	0.750	0.792	0.781	0.794	0.782	0.816	0.809	0.812	0.800								
73.40	0.21	0.639	0.597	0.713	0.693	0.712	0.682	0.732	0.714	0.741	0.719	0.738	0.730	0.757	0.728	0.746	0.739								
27.81	0.40	0.655	0.625	0.663	0.654	0.734	0.711	0.744	0.731	0.772	0.755	0.775	0.764	0.807	0.788	0.795	0.784								
73.42	0.40	0.622	0.600	0.648	0.639	0.692	0.672	0.707	0.698	0.731	0.712	0.736	0.725	0.759	0.738	0.754	0.738								
2.20	0.10	0.705	0.695	0.669	0.663	0.771	0.761	0.738	0.734	0.809	0.802	0.775	0.773	0.837	0.833	0.803	0.800								
10.92	0.10	0.720	0.714	0.782	0.775	0.745	0.744	0.795	0.791	0.781	0.778	0.805	0.801	0.810	0.807	0.812	0.808								
165.96	0.10	0.555	0.541	0.712	0.710	0.584	0.568	0.711	0.713	0.604	0.592	0.706	0.708	0.624	0.611	0.715	0.718								
0.25	0.20	0.600	0.584	0.604	0.598	0.686	0.672	0.688	0.682	0.730	0.717	0.728	0.722	0.756	0.747	0.757	0.753								
2.48	0.20	0.630	0.623	0.629	0.620	0.702	0.698	0.708	0.701	0.744	0.737	0.739	0.733	0.780	0.773	0.764	0.758								
49.78	0.20	0.634	0.634	0.695	0.695	0.687	0.690	0.700	0.704	0.720	0.717	0.706	0.708	0.747	0.743	0.716	0.719								
187.25	0.20	0.564	0.555	0.660	0.659	0.614	0.605	0.659	0.660	0.634	0.630	0.664	0.668	0.662	0.656	0.668	0.671								
3.25	0.40	0.536	0.515	0.439	0.414	0.631	0.604	0.527	0.507	0.667	0.633	0.567	0.543	0.698	0.672	0.592	0.567								
64.35	0.40	0.624	0.609	0.596	0.584	0.701	0.685	0.635	0.626	0.725	0.705	0.644	0.629	0.744	0.727	0.655	0.643								
245.65	0.40	0.556	0.541	0.527	0.513	0.600	0.574	0.550	0.538	0.630	0.596	0.564	0.548	0.661	0.632	0.574	0.558								

TABLE XI. Neighbor-truncation error analysis of $\tilde{T}[2]$ model for each data set in terms of R^2 .

(Re)	$\langle\phi\rangle$	$\tilde{T}_i[M_2 = 10, 2]$		$\tilde{T}_i[M_2 = 15, 2]$		$\tilde{T}_i[M_2 = 20, 2]$		$\tilde{T}_i[M_2 = 26, 2]$	
		Torque, $\perp$		Torque, $\perp$		Torque, $\perp$		Torque, $\perp$	
		Train	Test	Train	Test	Train	Test	Train	Test
9.86	0.10	0.745	0.747	0.745	0.748	0.751	0.748	0.752	0.751
121.36	0.10	0.306	0.281	0.298	0.275	0.307	0.285	0.308	0.285
6.95	0.21	0.825	0.821	0.824	0.824	0.827	0.826	0.827	0.825
73.40	0.21	0.494	0.484	0.493	0.481	0.496	0.484	0.496	0.486
27.81	0.40	0.657	0.642	0.652	0.641	0.654	0.643	0.657	0.647
73.42	0.40	0.431	0.431	0.428	0.429	0.434	0.432	0.435	0.434
2.20	0.10	0.893	0.893	0.896	0.896	0.898	0.898	0.899	0.898
10.92	0.10	0.791	0.786	0.792	0.789	0.796	0.793	0.798	0.794
165.96	0.10	0.527	0.532	0.510	0.518	0.532	0.542	0.533	0.541
0.25	0.20	0.892	0.891	0.892	0.892	0.896	0.895	0.896	0.895
2.48	0.20	0.894	0.892	0.894	0.892	0.897	0.895	0.898	0.896
49.78	0.20	0.711	0.715	0.713	0.717	0.709	0.711	0.713	0.717
187.25	0.20	0.586	0.584	0.571	0.571	0.593	0.593	0.595	0.595
3.25	0.40	0.817	0.813	0.819	0.812	0.816	0.809	0.820	0.813
64.35	0.40	0.690	0.688	0.680	0.674	0.694	0.689	0.694	0.685
245.65	0.40	0.555	0.554	0.528	0.524	0.561	0.561	0.565	0.564

It was earlier noticed in binary torque maps that the radius of influence is substantially smaller for all volume fractions when compared with corresponding binary force maps. Neighbor-truncation analysis of $\tilde{T}[2]$ presented in Table XI confirms this observation. Test performance averaged over

TABLE XII. Neighbor-truncation error analysis for $\tilde{F}[3]$ model in terms of R^2 .

(Re)	$\langle\phi\rangle$	$\tilde{F}_i[M_2 = 26, M_3 = 5, 3]$				$\tilde{F}_i[M_2 = 26, M_3 = 10, 3]$			
		Drag		Lift		Drag		Lift	
		Train	Test	Train	Test	Train	Test	Train	Test
9.86	0.10	0.816	0.732	0.851	0.797	0.842	0.762	0.862	0.803
121.36	0.10	0.766	0.641	0.789	0.701	0.787	0.650	0.805	0.701
6.95	0.21	0.832	0.808	0.841	0.813	0.853	0.817	0.862	0.827
73.40	0.21	0.797	0.729	0.809	0.764	0.818	0.745	0.833	0.772
27.81	0.40	0.818	0.794	0.816	0.799	0.837	0.809	0.841	0.823
73.42	0.40	0.787	0.749	0.779	0.751	0.807	0.769	0.817	0.786
2.20	0.10	0.846	0.834	0.824	0.811	0.854	0.838	0.842	0.826
10.92	0.10	0.821	0.810	0.849	0.838	0.826	0.815	0.861	0.851
165.96	0.10	0.660	0.605	0.757	0.732	0.679	0.615	0.759	0.732
0.25	0.20	0.760	0.749	0.762	0.754	0.769	0.758	0.773	0.764
2.48	0.20	0.786	0.777	0.773	0.765	0.791	0.782	0.791	0.784
49.78	0.20	0.757	0.747	0.771	0.768	0.772	0.754	0.795	0.787
187.25	0.20	0.681	0.660	0.715	0.701	0.698	0.661	0.731	0.711
3.25	0.40	0.702	0.672	0.598	0.570	0.722	0.679	0.625	0.584
64.35	0.40	0.748	0.728	0.688	0.670	0.764	0.739	0.717	0.696
245.65	0.40	0.677	0.636	0.619	0.594	0.698	0.649	0.654	0.611

TABLE XIII. Neighbor-truncation error analysis for $\tilde{T}[3]$ model in terms of R^2 .

(Re)	$\langle\phi\rangle$	$\tilde{T}_i[M_2 = 26, M_3 = 5, 3]$				$\tilde{T}_i[M_2 = 26, M_3 = 10, 3]$			
		Torque, $\parallel$		Torque, $\perp$		Torque, $\parallel$		Torque, $\perp$	
		Train	Test	Train	Test	Train	Test	Train	Test
9.86	0.10	0.475	0.326	0.906	0.862	0.602	0.469	0.933	0.899
121.36	0.10	0.368	-0.049	0.798	0.556	0.467	0.011	0.835	0.618
6.95	0.21	0.523	0.371	0.923	0.893	0.699	0.610	0.951	0.929
73.40	0.21	0.310	0.070	0.816	0.723	0.458	0.234	0.868	0.774
27.81	0.40	0.409	0.244	0.870	0.812	0.658	0.544	0.937	0.897
73.42	0.40	0.324	0.221	0.772	0.709	0.571	0.437	0.864	0.804
2.20	0.10	0.500	0.449	0.953	0.943	0.678	0.630	0.969	0.962
10.92	0.10	0.443	0.395	0.901	0.891	0.583	0.549	0.928	0.918
165.96	0.10	0.104	-0.013	0.757	0.702	0.181	0.047	0.793	0.741
0.25	0.20	0.452	0.435	0.938	0.932	0.673	0.647	0.954	0.949
2.48	0.20	0.455	0.437	0.937	0.933	0.665	0.653	0.958	0.954
49.78	0.20	0.280	0.245	0.842	0.832	0.415	0.363	0.876	0.864
187.25	0.20	0.136	0.080	0.754	0.729	0.192	0.120	0.793	0.758
3.25	0.40	0.384	0.340	0.879	0.859	0.661	0.598	0.916	0.886
64.35	0.40	0.263	0.242	0.758	0.739	0.451	0.408	0.806	0.777
245.65	0.40	0.168	0.122	0.696	0.660	0.305	0.208	0.768	0.715

all data sets of transverse torque components yields $\bar{R}_{\text{Torque},\perp,\text{test}}^2 = \{0.672, 0.668, 0.675, 0.677\}$ for $M_2 = \{10, 15, 20, 26\}$, indicating that improvement beyond $M_2 = 10$ is almost negligible.

2. Ternary models neighbor-truncation error

The neighbor-truncation analysis of ternary models is carried out by fixing $M_2 = 26$. Only two different values of $M_3 = \{5, 10\}$ are considered to reduce the computational cost associated with higher values of M_3 . The performance of $\tilde{F}[3]$ model is evaluated in Table XII. The data-set-averaged test performance for drag and lift are $\bar{R}_{\text{Drag},\text{test}}^2 = \{0.729, 0.740\}$ and $\bar{R}_{\text{Lift},\text{test}}^2 = \{0.739, 0.754\}$. Although the improvement appears to be only around 1%–2% when M_3 is increased from 5 to 10, it must be realized that this improvement is only under conditions of limited training data. The data paucity can be observed from the large difference between training and testing performance at both values of M_3 for all data sets.

As mentioned earlier ternary-interaction torque model is necessary for streamwise torque component prediction. Hence, it can be seen from Table XIII that $\tilde{T}[3]$ has high sensitivity with respect to M_3 . Data-set-averaged test performance for streamwise and transverse torque components of $\tilde{T}[3]$ is $\bar{R}_{\text{Torque},\parallel,\text{test}}^2 = \{0.245, 0.408\}$ and $\bar{R}_{\text{Torque},\perp,\text{test}}^2 = \{0.798, 0.840\}$. 16% improvement in prediction is noticed for streamwise component when M_3 is increased from 5 to 10. However, the substantial difference between training and testing performance indicate data paucity.

-
- [1] S. Balachandar and J. K. Eaton, Turbulent dispersed multiphase flow, *Annu. Rev. Fluid Mech.* **42**, 111 (2010).
 - [2] L. Brandt and F. Coletti, Particle-laden turbulence: Progress and perspectives, *Annu. Rev. Fluid Mech.* **54**, 159 (2022).
 - [3] J. Capecehatro and O. Desjardins, An Euler–Lagrange strategy for simulating particle-laden flows, *J. Comput. Phys.* **238**, 1 (2013).

- [4] R. Beetstra, M. A. van der Hoef, and J. A. M. Kuipers, Drag force of intermediate Reynolds number flow past mono- and bidisperse arrays of spheres, *AIChE J.* **53**, 489 (2007).
- [5] S. Tenneti, R. Garg, and S. Subramaniam, Drag law for monodisperse gas–solid systems using particle-resolved direct numerical simulation of flow past fixed assemblies of spheres, *Int. J. Multiphase Flow* **37**, 1072 (2011).
- [6] S. Bogner, S. Mohanty, and U. Rüde, Drag correlation for dilute and moderately dense fluid-particle systems using the lattice Boltzmann method, *Int. J. Multiphase Flow* **68**, 71 (2015).
- [7] Y. (Yali) Tang, E. A. J. F. Peters, J. A. M. Kuipers, S. H. L. Kriebitzsch, and M. A. van der Hoef, A new drag correlation from fully resolved simulations of flow past monodisperse static arrays of spheres, *AIChE J.* **61**, 688 (2015).
- [8] G. Akiki and S. Balachandar, Immersed boundary method with non-uniform distribution of lagrangian markers for a non-uniform Eulerian mesh, *J. Comput. Phys.* **307**, 34 (2016).
- [9] L. He, D. K. Tafti, and K. Nagendra, Evaluation of drag correlations using particle resolved simulations of spheres and ellipsoids in assembly, *Powder Technol.* **313**, 332 (2017).
- [10] A. Seyed-Ahmadi and A. Wachs, Microstructure-informed probability-driven point-particle model for hydrodynamic forces and torques in particle-laden flows, *J. Fluid Mech.* **900**, A21 (2020).
- [11] J. Richardson and W. Zaki, The sedimentation of a suspension of uniform spheres under conditions of viscous flow, *Chem. Eng. Sci.* **3**, 65 (1954).
- [12] Andrews, P. N. Loezos, and S. Sundaresan, Coarse-grid simulation of gas-particle flows in vertical risers, *Ind. Eng. Chem. Res.* **44**, 6022 (2005).
- [13] G. Akiki, T. L. Jackson, and S. Balachandar, Force variation within arrays of monodisperse spherical particles, *Phys. Rev. Fluids* **1**, 044202 (2016).
- [14] A. Esteghamatian, F. Euzenat, A. Hammouti, M. Lance, and A. Wachs, A stochastic formulation for the drag force based on multiscale numerical simulation of fluidized beds, *Int. J. Multiphase Flow* **99**, 363 (2018).
- [15] A. M. Lattanzi, V. Tavanashad, S. Subramaniam, and J. Capecelatro, Stochastic model for the hydrodynamic force in Euler–Lagrange simulations of particle-laden flows, *Phys. Rev. Fluids* **7**, 014301 (2022).
- [16] A. M. Lattanzi, V. Tavanashad, S. Subramaniam, and J. Capecelatro, Fluid-mediated sources of granular temperature at finite Reynolds numbers, *J. Fluid Mech.* **942**, A7 (2022).
- [17] G. A. Cisneros, K. T. Wikfeldt, L. Ojamäe, J. Lu, Y. Xu, H. Torabifard, A. P. Bartók, G. Csányi, V. Molinero, and F. Paesani, Modeling molecular interactions in water: From pairwise to many-body potential energy functions, *Chem. Rev.* **116**, 7501 (2016).
- [18] F. Paesani, Getting the right answers for the right reasons: Toward predictive molecular simulations of water with many-body potential energy functions, *Acc. Chem. Res.* **49**, 1844 (2016).
- [19] G. Akiki, T. L. Jackson, and S. Balachandar, Pairwise interaction extended point-particle model for a random array of monodisperse spheres, *J. Fluid Mech.* **813**, 882 (2017).
- [20] G. Akiki, W. Moore, and S. Balachandar, Pairwise-interaction extended point-particle model for particle-laden flows, *J. Comput. Phys.* **351**, 329 (2017).
- [21] W. Moore, S. Balachandar, and G. Akiki, A hybrid point-particle force model that combines physical and data-driven approaches, *J. Comput. Phys.* **385**, 187 (2019).
- [22] L. He and D. K. Tafti, A supervised machine learning approach for predicting variable drag forces on spherical particles in suspension, *Powder Technol.* **345**, 379 (2019).
- [23] N. Muralidhar, J. Bu, Z. Cao, L. He, N. Ramakrishnan, D. Tafti, and A. Karpatne, Phynet: Physics guided neural networks for particle drag force prediction in assembly, in *Proceedings of the 2020 SIAM International Conference on Data Mining (SDM)* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2020), pp. 559–567.
- [24] A. Seyed-Ahmadi and A. Wachs, Physics-inspired architecture for neural network modeling of forces and torques in particle-laden flows, *Comput. Fluids* **238**, 105379 (2022).
- [25] N. R. Ashwin, Z. Cao, N. Muralidhar, D. Tafti, and A. Karpatne, Deep learning methods for predicting fluid forces in dense particle suspensions, *Powder Technol.* **401**, 117303 (2022).
- [26] W. C. Moore and S. Balachandar, Lagrangian investigation of pseudo-turbulence in multiphase flow using superposable wakes, *Phys. Rev. Fluids* **4**, 114301 (2019).

- [27] B. Siddani, S. Balachandar, W. C. Moore, Y. Yang, and R. Fang, Machine learning for physics-informed generation of dispersed multiphase flow using generative adversarial networks, *Theor. Comput. Fluid Dyn.* **35**, 807 (2021).
- [28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial networks, *Commun. ACM* **63**, 139 (2020).
- [29] M. Mirza and S. Osindero, Conditional generative adversarial nets, [arXiv:1411.1784](https://arxiv.org/abs/1411.1784).
- [30] N. Muralidhar, J. Bu, Z. Cao, N. Raj, N. Ramakrishnan, D. Tafti, and A. Karpatne, Phyflow: Physics-guided deep learning for generating interpretable 3D flow fields, in *2021 IEEE International Conference on Data Mining (ICDM)* (2021), pp. 1246–1251.
- [31] M. Geiger, T. Smidt, A. Musaelian, B. K. Miller, W. Boomsma, B. Dice, K. Lapchevskiy, M. Weiler, M. Tyszkiewicz, S. Batzner, D. Madisetti, M. Uhrin, J. Frellsen, N. Jung, S. Sanborn, M. Wen, J. Rackers, M. Rød, and M. Bailey, Euclidean neural networks: e3nn (2022), doi:[10.5281/zenodo.6459381](https://doi.org/10.5281/zenodo.6459381).
- [32] B. Siddani, S. Balachandar, and R. Fang, Rotational and reflectional equivariant convolutional neural network for data-limited applications: Multiphase flow demonstration, *Phys. Fluids* **33**, 103323 (2021).
- [33] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds [arXiv:1802.08219](https://arxiv.org/abs/1802.08219).
- [34] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. S. Cohen, 3d steerable cnns: Learning rotationally equivariant features in volumetric data, *Adv. Neural Inf. Proc. Syst.* **31** (2018).
- [35] R. Kondor, Z. Lin, and S. Trivedi, Clebsch–Gordan nets: A fully Fourier space spherical convolutional neural network, in *Advances in Neural Information Processing Systems*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Inc., Montreal, Canada, 2018), Vol. 31, pp. 10117–10126.
- [36] R. Wang, R. Walters, and R. Yu, Incorporating symmetry into deep dynamics models for improved generalization, [arXiv:2002.03061](https://arxiv.org/abs/2002.03061).
- [37] O. Unke, M. Bogojeski, M. Gastegger, M. Geiger, T. Smidt, and K.-R. Müller, Se(3)-equivariant prediction of molecular wavefunctions and electronic densities, in *Advances in Neural Information Processing Systems*, edited by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (Curran Associates, 2021), Vol. 34, pp. 14434–14447.
- [38] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, *Nat. Commun.* **13**, 2453 (2022).
- [39] M. Lino, S. Fotiadis, A. A. Bharath, and C. Cantwell, REMuS-GNN: A rotation-equivariant model for simulating continuum dynamics, in *Proceedings of Topological, Algebraic, and Geometric Learning Workshops*, edited by A. Cloninger, T. Doster, T. Emerson, M. Kaul, I. Ktena, H. Kvinge, N. Miolane, B. Rieck, S. Tymochko, and G. Wolf (PMLR, 2022), Vol. 196, pp. 226–236.
- [40] Y. Yang and S. Balachandar, A scalable parallel algorithm for direct-forcing immersed boundary method for multiphase flow simulation on spectral elements, *J. Supercomput.* **77**, 2897 (2021).
- [41] M. Wang, Y. Yang, D. Z. Zhang, and S. Balachandar, Numerical calculation of the particle–fluid–particle stress in random arrays of fixed particles, *Phys. Rev. Fluids* **6**, 104306 (2021).
- [42] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators, *Nat. Mach. Intell.* **3**, 218 (2021).
- [43] C. Lin, Z. Li, L. Lu, S. Cai, M. Maxey, and G. E. Karniadakis, Operator learning for predicting multiscale bubble growth dynamics, *J. Chem. Phys.* **154**, 104118 (2021).
- [44] C. Lin, M. Maxey, Z. Li, and G. E. Karniadakis, A seamless multiscale operator neural network for inferring bubble dynamics, *J. Fluid Mech.* **929**, A18 (2021).
- [45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., Vancouver, Canada, 2019), Vol. 32, pp. 8024–8035.

- [46] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, edited by Y. Bengio and Y. LeCun (2015).
- [47] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning* (Springer, New York, NY, 2013), Vol. 112.
- [48] B. Siddani and S. Balachandar, Point-particle_closures_hierarchical_approach: v1.0.1 (2023), doi:[10.5281/zenodo.7502457](https://doi.org/10.5281/zenodo.7502457).
- [49] M. Allahyari, W. Moore, B. S. Siddani, and S. Balachandar, A deep neural network-based approach for the force predictions of particulate multiphase flows, in *APS Division of Fluid Dynamics Meeting Abstracts* (2020), pp. J12–011.