Modeling Droplet Dynamics in Emulsions with Graph Neural Networks

Giulio Ortali [1] Federico Toschi [1] Jan-Willem van de Meent [2]

Abstract

We develop a neural model for dynamics in stabilized dense emulsions, trained on data from first-principles simulations of biphase flows. By using a hybrid approach where a graph neural network predicts corrections to a hand-crafted baseline model, we achieve accurate and stable predictions. This method addresses the instability issues found in naive GNN implementations, paving the way for a deeper understanding of emulsion physics and more efficient numerical simulations.

1. Introduction

The study of emergent phenomena from low-level descriptions of physical systems has always been a key focus of scientific research (Anderson, 1972). Examples include self-organization in ecosystems (Levin, 2005), turbulence in fluid dynamics (Frisch, 1995) and the development of consciousness from neural activity in the brain (Crick & Koch, 1990). The interest in emergent behaviors is twofold: reaching a deeper understanding of the physical systems, and the development of cost-effective reduced methods for their numerical simulation. Recently, there has been a significant rise in the use of data-driven approaches alongside traditional methods to study emergent behaviors, employing machine learning, statistical models, and large-scale data analysis to find patterns and predict dynamics in complex systems (Duraisamy et al., 2018; Van Veen et al., 1997).

To this end, in this work we aim to study the emergent behavior of droplets in emulsions, modelling their effective dynamics. Emulsions are mixtures of two immiscible fluids where one fluid is dispersed in the form of droplets within the other (see Fig. 1). Emulsions are employed in a wide range of applications, including pharmaceuticals, food processing, cosmetics, and materials science, due to their ability to combine the properties of the two immiscible fluids in a stable mixture (Friberg et al., 2003; Barkat et al., 2011). The numerical simulation of emulsions poses significant challenges due to the complex interfaces, multiscale interactions, and the effects of surface tension.

Traditional approaches to the numerical simulation of emulsions rely on fully modeling the biphase flow, requiring expensive simulation settings (Wörner, 2012; Kumar et al., 2020a). On the other hand, developing simplified, effective models for droplets in emulsions, which aim to represent droplets dynamics without fully simulating the biphase flow, is challenging due to the difficulty in accurately capturing the nuanced behaviors in densely packed systems, involving non-Newtonian behavior with both elastic and plastic effects (Kumar et al., 2023; 2020b).

This paper explores the potential of graph neural networks (GNNs) (Scarselli et al., 2009; Bruna et al., 2014) and specifically message-passing variants of such networks (Gilmer et al., 2017; Battaglia et al., 2018) for modeling the effective dynamics of droplets in emulsions. GNNs have previously been used to learn neural surrogates for dynamics of Newtonian fluids, as well as of granular and Non-Newtonian systems (Sanchez-Gonzalez et al., 2020), and more generally in the simulation of many-body systems and molecular dynamics (Satorras et al., 2021; Park et al., 2021).Our work addresses the unique challenges posed by densely packed droplets and their complex interactions. We tackle stability issues through a hybrid design and multi-step training strategies, detailed in the following sections of this paper.

We begin by introducing a hand-crafted effective model, which we refer to as the elasto-viscous (EV) model, inspired by basic physical insights into the behavior of droplets within emulsions. We compare this model to a GNN, which is trained end-to-end to predict droplets displacements. We show that the GNN yields accurate one-step predictions but highly unstable long-time dynamics. To address this, we develop a hybrid approach, termed EV+GNN, that combines both methods. Our results demonstrate that this hybrid approach provides both accurate and stable predictions, leveraging the strengths of both models (Fig. 1)

[1] Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands [2] University of Amsterdam, 1012 WP Amsterdam, The Netherlands. Correspondence to: Giulio Ortali <g.ortali@tue.nl>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

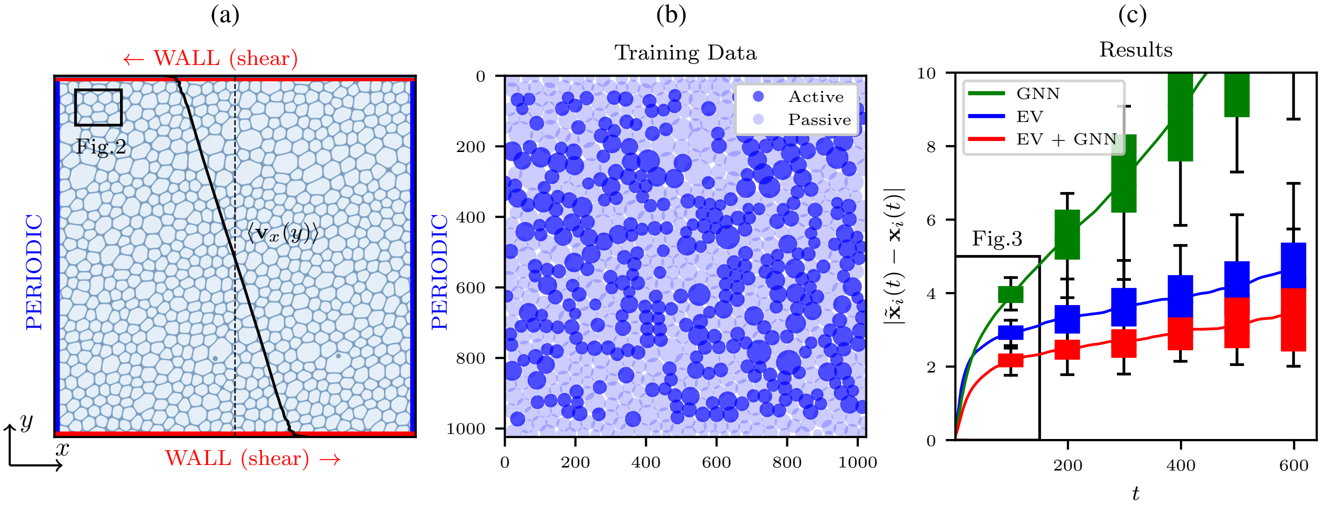

Figure 1. **(a)** Lattice Boltzmann simulations of a 2D Couette biphase flow. The colormap represents fluid density, highlighting the biphase nature of the fluid. The left and right boundaries represent periodic boundary conditions, while the top and bottom walls are walls moving in opposite direction with non-slip boundary condition, generating a shear in the flow $\mathbf{v}_x(y)$, shown as a black line. The small box in the top-left corner represents the portion of domain considered in Fig. 2. **(b)** Training data. Results of Lattice Boltzmann simulations are pre-processed with standard computer vision tools to obtain sizes and positions of droplets, which we approximate as spherical. During training, only a fraction of the droplets, dubbed "active", are evolved according to the learned model, while the "passive" droplets evolve according to their positions in the simulations. **(c)** Mean absolute error between the predicted trajectories $\tilde{\mathbf{x}}_i(t)$ and the ground truth trajectory $\mathbf{x}_i(t)$ in time for three different effective models: an end-to-end Graph Neural Network predicting droplet displacement (GNN), a hand-crafted physical model (EV) and a hybrid method combining the two (EV+GNN). Figure 3 reports a zoom in of the error on the first 150 timesteps. We observe that a naive application of a GNN is unstable and results in fast divergence, whereas using the GNN to predict a correction relative to the EV baseline leads to greater stability and improved accuracy.

1.1. Related Work

Neural Models for Particle-based Dynamics. There is a long line of work on neural methods for particle-based or Lagrangian dynamics in fluids and related systems (Li et al., 2019; Sanchez-Gonzalez et al., 2020; Ummenhofer et al., 2020; Prantl et al., 2022). Many such approaches generate training data using established particle-based methods for fluids, such as smoothed-particle hydrodynamics (SPH; Gingold & Monaghan (1977)) and dissipative particle dynamics (DPD; Hoogerbrugge & Koelman (1992)), or more generally from particle-based descriptions of solids, such as position-based dynamics (PBD; Müller et al. (2007)) and the material point method (MPM; Sulsky et al. (1995)). Note here that SPH and DPD in particular also inspire the design of our non-neural elasto-viscous model in Section 3.2.

This work is most directly related to the work by Sanchez-Gonzalez et al. (2020) on graph network-based simulators (GNS), which develops GNN-based surrogates for simulation of Newtonian fluids, non-Newtonian fluids, and granular systems. This work differs somewhat from the work we consider here in that it trains a GNN to approximate dynamics of existing particle-based methods, with an eye to obtaining a differentiable surrogate that can be used to, e.g., solve inverse problems. Here we are interested in a setting where no existing particle-based simulation is available. Moreover, the application to dense emulsions that we consider here involves specific challenges that arise from the crowded environment in such systems.

Other Neural Methods for Fluid Mechanics. More broadly, there is an extensive body of work on machine learning and deep learning for fluid mechanics (Ladický et al., 2015; Yang et al., 2016; Chu & Thuerey, 2017; Tompson et al., 2017; Wiewel et al., 2019; Wang et al., 2020a; Pfaff et al., 2021). Many of these studies aim to accelerate or substitute numerical simulations by taking an Eulerian point of view. A more recent development in this space are equivariant networks, which incorporate layers that respect translational and rotational symmetries by construction (Wang et al., 2020b; 2022; Lino et al., 2022; Ruhe et al., 2024; Zhdanov et al., 2024). The GNN in this paper does not incorporate rotational symmetry, which is broken in the 2D Couette geometry that we consider, but messages are invariant to translations and Galilean transformations, since they are computed using relative positions and displacements.

2. Preliminaries

2.1. Emulsion Dynamics in Sheared Systems

We consider flows of dense stabilized emulsions in a 2-dimensional Couette geometry (Fig. 1a). This geometry assumes non-slip boundary conditions on two moving walls at the top and bottom of a cell, and period boundary on the left and right walls. We simulate this system using a variant of Lattice Boltzmann method (Krueger et al., 2016), a class of grid-based methods that simulate populations of particles with discretized positions and velocities. The specific variant that we employ has been designed to simulate the biphasic flows in emulsions (Kumar et al., 2020a).

At the level of LBM simulations, droplet formation is an emergent phenomenon. As such, the simulation results, which are essentially images, do not provide a description at the level of droplets. Our goal in this work is to learn an effective model for the dynamics of droplet interactions. To train this model, we generate a dataset by pre-processing LBM simulation results to detect and track droplets (Fig. 2, see Sec. 4.1 for details).

In the following, we will assume that data of the form of N droplet trajectories $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ each comprising $t = 1, \ldots, T$ time points, where we use $\mathbf{x}_i(t) \in \mathbb{R}^2$ to refer to the position a droplet at a particular time point. For each droplet trajectory, we additionally define:

- The "velocity", i.e. the displacement relative to the preceding time point,

$$\mathbf{v}_i(t) = \mathbf{x}_i(t) - \mathbf{x}_i(t-1).$$

- The radius $r_i \in \mathbb{R}$, which does not vary in time.
- A set of neighbors, which is defined as the indices of particles for that are separated by no more than a tolerance $\epsilon_{\mathcal{N}} = 20$:

$$\mathcal{N}_i(t) = \{j : \|\mathbf{x}_j(t) - \mathbf{x}_i(t)\| \leq r_i + r_j + \epsilon_{\mathcal{N}}\}$$

- The overlap relative to other droplets j,

$$o_{ij} = r_i + r_j - ||\mathbf{x}_j - \mathbf{x}_i||. \quad (1)$$

- An indicator variable $b_i \in \{0, 1\}$ that is equal to 1 when a droplet resides within distance $\epsilon_{\text{wall}} = 20$ of the top or bottom wall at any point in the trajectory. In both training and testing, border droplets are never evolved according to the effective models, and always follow the ground truth dynamics. This is done to simplify the setting since border droplets, in contact with walls, experience different physical phenomena with respect to bulk droplets.

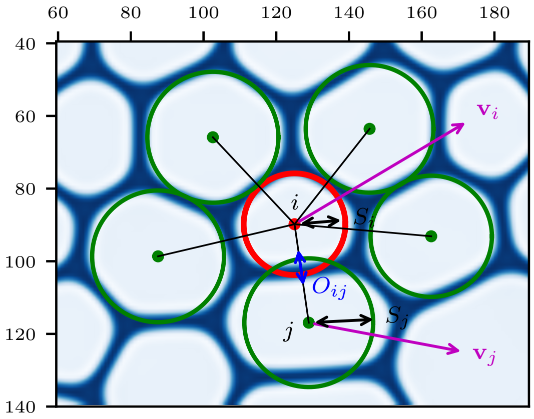

Figure 2. Representation of different quantities computed from simulation data, used to define the effective motion of droplet i, considering the interactions with neighbour droplets $j \in \mathcal{N}_i(t)$.

2.2. Graph Neural Networks

Message passing graph Neural Networks are a class of neural networks designed to process and analyze data structured as graphs (Gilmer et al., 2017). GNNs operate through an iterative process of message passing, where node representations are updated based on their neighbors' features.

Consider a graph $G = (V, E)$ with $\{i \in V\}$ a set of nodes and $\{e_{ij}\}$ a set of edges. Let h_i^k represent the feature vector of node i at layer k. One iteration of message passing can be represented as:

$$\begin{aligned} m_{ij}^k &= \Phi_1(h_i^k, h_j^k, e_{ij}; \theta_1), && \text{(msg. evaluation)} \\ m_i^k &= \text{AGG}(m_{ij}^k : j \in \mathcal{N}_i), && \text{(msg. aggregation)} \\ h_i^k &= \Phi_2(h_i^k, m_i^k; \theta_2), && \text{(node update)} \end{aligned}$$

with Φ_1 and Φ_2 two Neural Networks with weights θ_1 and θ_2, AGG an aggregation function (for example, mean), and $\mathcal{N}_i$ the set of neighbor nodes of i.

Unlike traditional neural networks that operate on fixed-sized input data like images or sequences, GNNs can directly handle the variable-sized and complex relationships inherent in graph data. This flexibility makes GNNs particularly powerful for tasks involving physical systems, social networks and molecular structure where interactions between entities are naturally represented as graphs.

3. Methods

3.1. Learning Problem

We formulate the learning problem as follows: define a model $\mathcal{M}$, operating locally on each droplet i, taking as input quantities for the droplet i and its neighbor droplets

$j \in \mathcal{N}_i(t)$ at time t and returning an approximation to the velocity (displacement) at time $t+1$, in formulas:

$$\mathbf{v}_i(t+1) \approx \mathcal{M}(r_i(t), \mathbf{v}_i(t), \\ \{(\mathbf{x}_j - \mathbf{x}_i)(t), (\mathbf{v}_j - \mathbf{v}_i)(t), \\ r_j(t),\ \forall j \in \mathcal{N}_i(t)\}). \quad (2)$$

In order to use the model $\mathcal{M}$, we then consider an initial condition $\{\mathbf{x}_i(0), r_i(0),\ i = 1 \dots N\}$ and we apply it to the droplets in the domain, evolving the system in time.

The formulation we consider is Markovian (the prediction at timestep $t+1$ only depends on the quantities at the previous timestep t) and local, implying that for each timestep interactions between droplets only happen between neighbors. Additionally, the neighbors droplets' features are only provided to the model through relative quantities, and the position of droplet i is not given to the model, in order to avoid the model learning information regarding its location in the domain, possibly hindering generalization.

During training, for each epoch and for each sample, we randomly flag a fraction (half) of the non boundary droplets as *active* and the remaining droplets are *passive* (see Fig. 1b for an illustration). Starting from the initial condition $t = 0$, we evolve the active droplets according to the models, yielding predicted displacements $\tilde{\mathbf{v}}_i(t)$, while the passive droplets follow the simulated displacements $\mathbf{v}_i(t)$. Simultaneously, for each timestep we accumulate the loss, computed as MSE on predicted velocity for the active droplets

$$\mathcal{L} = \sum_{t,i} \left\| \tilde{\mathbf{v}}_i(t) - \mathbf{v}_i(t) \right\|^2. \quad (3)$$

Some considerations on the training strategy. The need for the multi-step training strategy here described, as opposed to a one-step strategy with a fixed dataset of input and output couples, comes from the importance of properly learning the interactions between two or more droplets. In fact, we found that models trained with the one-step strategy, despite reaching a higher one-step accuracy, were more unstable when used as effective models, diverging in few timesteps. The decision of only selecting a fraction of the droplets to be evolved according to the models is done in order to reduce the computational and memory requirements, and to help with stabilizing the dynamics in the first epochs. Finally, the droplet radius is assumed, both here and in testing, to remain approximately constant during the evolution.

3.2. Elasto-Viscous model (EV)

In this section, we define a hand-crafted effective model that we refer to as the *elasto-viscous* model or **EV** model. This model will be used as a baseline with respect to the data-driven models.

The action of the EV model is defined as:

$$\bar{\mathbf{v}}_i = \beta \mathbf{v}_i(t) + (1-\beta) \frac{1}{|\mathcal{N}_i(t)|} \sum_{j \in \mathcal{N}_i(t)} \mathbf{v}_j(t) \quad (4)$$

$$\mathbf{F}_{ij} = \kappa \ \max\left(o_{ij} + \epsilon_{EV}, 0\right)^{\alpha} \frac{\mathbf{x}_j(t) - \mathbf{x}_i(t)}{||\mathbf{x}_j(t) - \mathbf{x}_i(t)||}, \quad (5)$$

$$\mathbf{v}_i^{EV}(t+1) = \bar{\mathbf{v}}_i - \sum_{j \in \mathcal{N}_i(t)} \mathbf{F}_{ij} \quad (6)$$

$$\tilde{\mathbf{x}}_i(t+1) = \tilde{\mathbf{x}}_i(t) + \mathbf{v}_i^{EV}(t+1) \quad (7)$$

where $\beta = 0.5$, $\kappa = 0.1$, $\epsilon_{EV} = 5$ and $\alpha = 0.5$ are model's hyperparameters, here tuned by hand, o_{ij} is the overlap between droplet i and j (Eq. 2.1), and $\mathbf{F}_{ij}$ is the elastic force acting from droplet j on droplet i.

As the name suggests, the EV model acts with two main physical mechanisms: a viscous step (Eq. 4), where we compute a linear combination between the velocity at the previous timestep and the average velocity of the neighbors; next an elastic step, consisting of computing, for each neighbor, an elastic force $\mathbf{F}_{ij}$ considering the overlap between the two droplets (Eq. 5). All the force contribution are then summed, and the final velocity is updated (Eq. 6) (conceptually assuming mass and timestep both equal to 1). Finally, the droplet position $\mathbf{x}_i(t)$ is updated using the obtained velocity $\mathbf{v}_i^{EV}(t+1)$.

3.3. Graph Neural Network Model (GNN)

We begin by considering naive GNN-based implementention, in the form of a single-layer Graph Neural Network returning the predicted displacement $\mathbf{v}_i(t+1)$. The GNN model is composed of two MLPs Φ_1 (layer sizes $[8, 512, 128, 64]$, leaky-ReLU activation function) and Φ_2 (layer sizes $[67, 64, 2]$, leaky-ReLU activation function), which perform a single round of message-passing updates:

$$\mathbf{m}_{ij} = \Phi_1(\mathbf{v}_i, r_i, \mathbf{x}_j - \mathbf{x}_i, \mathbf{v}_j - \mathbf{v}_i, r_j; \theta_1), \quad (8)$$

$$\mathbf{v}_i^{GNN}(t+1) = \Phi_2(\mathbf{v}_i, r_i, \sum_{j \in \mathcal{N}_i(t)} \mathbf{m}_{ij}; \theta_2), \quad (9)$$

$$\tilde{\mathbf{x}}_i(t+1) = \tilde{\mathbf{x}}_i(t) + \mathbf{v}_i^{GNN}(t+1). \quad (10)$$

The proposed architecture was designed through analogy with the action of the EV model (Sec. 3.2), and is a combination between a generic message passing Graph Neural Network, as defined in (Gilmer et al., 2017) and the Equivariant GraphNN, introduced in (Satorras et al., 2021).

3.4. Hybrid Model (EV+GNN)

The second data-driven model used in this work is a combination between the EV model and the GNN model, which we will label **EV+GNN**. The Graph Neural Network used in this model is identical to the one used for the GNN, described in Sec. 3.3. However, the output of the GNN is now used as a correction to the displacements predicted by the EV model, in formulas:

$$\mathbf{m}_{ij} = \Phi_1(\mathbf{v}_i, r_i, \mathbf{x}_j - \mathbf{x}_i, \mathbf{v}_j - \mathbf{v}_i, r_j; \theta_1), \quad (11)$$

$$\tilde{\mathbf{v}}_i = \Phi_2(\mathbf{v}_i, r_i, \sum_{j\in\mathcal{N}_i(t)} \mathbf{m}_{ij}; \theta_2), \quad (12)$$

$$\mathbf{v}_i^{EV+GNN}(t+1) = \mathbf{v}_i^{EV}(t+1) + \tilde{\mathbf{v}}_i, \quad (13)$$

$$\tilde{\mathbf{x}}_i(t+1) = \tilde{\mathbf{x}}_i(t) + \mathbf{v}_i^{EV+GNN}(t+1). \quad (14)$$

4. Results

4.1. Data Generation

The training and testing datasets are generated using a multi-component Lattice Boltzmann Method (LBM) simulation (Krueger et al., 2016). The numerical approximation is performed on a computational grid of size $[1024, 1024]$, fully resolving the biphase flow. For more details on the physical setting and discretization strategy, we refer to (Kumar et al., 2020a).

Starting from simulation data, we then perform some preprocessing steps in order to extract quantities to be used to define the effective dynamics of droplets.

We define an (arbitrary) effective sampling timestep $\Delta t_{\text{eff}} = 100 \times \Delta t$, where Δt is the simulation timestep. This sampling time must be fine enough to capture the continuous dynamics of the droplets, but coarse enough to provide a computational speedup for the effective models.

As to space discretization, as already anticipated in Sec. 2.1, we make the (strong) assumption of approximating droplets with circles, described by a droplet center $\mathbf{x}_i \in \mathbb{R}^2$ and a radius (or size) r_i. This assumption is used to represent each droplet with a small number of degrees of freedom, and is approximately verified for small droplets, whereas big droplets are typically more deformed.

4.2. Training Procedure

The training dataset is composed of $M_{\text{train}} = 900$ samples, where each sample consists of the full time evolution for all the droplets i in the domain, for $T = 100$ timesteps, specified as droplet position, velocity and radius: $\{\mathbf{x}_i(t), \mathbf{v}_i(t), r_i(t),\ i = 1, \ldots N,\ t = 0 \ldots T\}$. We train the models for 20 epochs, using an Adam optimizer with initial learning rate $\lambda = 10^{-3}$.

A roll-out terminates when the trajectories for active particles have been fully evolved up to T, or when one of the predicted positions $\tilde{\mathbf{x}}_i(t)$ diverges too far from the LBM simulation trajectory $\mathbf{x}_i(t)$, i.e. if the maximum absolute displacement is larger than $\epsilon_{\text{train}} = 15$.

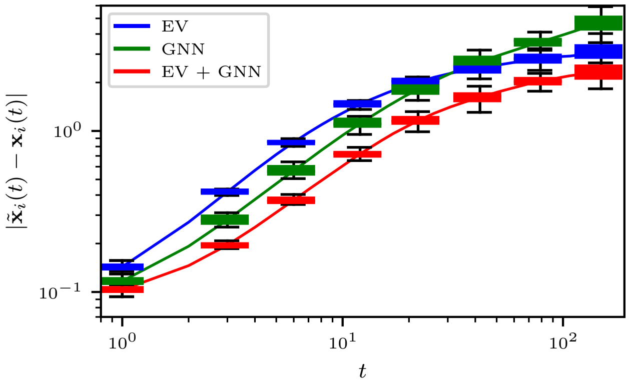

Figure 3. Mean absolute error between ground truth droplet position $\mathbf{x}_i(t)$ and predicted droplet position $\tilde{\mathbf{x}}_i(t)$, for the three effective models EV, GNN and EV+GNN, for the first 150 timesteps.

4.3. Testing procedure

For testing, we consider a setting analogous to the one used during training, with $M_{\text{test}} = 20$ test roll-outs consisting of the full time evolution for all the droplets i in the domain, for $T = 10^3$ timesteps, $\{\mathbf{x}_i(t), \mathbf{v}_i(t), r_i(t),\ i = 1, \ldots N,\ t = 0 \ldots T\}$. Starting from the initial condition $\{\mathbf{x}_i(0), \mathbf{v}_i(0), r_i(0)\}$, we evolve part or all the droplets in the domain according to the effective models, obtaining the predicted trajectories $\tilde{\mathbf{x}}_i(t)$.

4.4. Conditional Simulation Results

We begin by considering a test-time procedure that is analogous to the one used during training (Fig. 1 panel (b)), in which half of the particles are tagged as active and evolved according to the effective models, and the remaining evolved according to the ground truth dynamics. We refer to this setting as conditional simulation, in reference to the analogous case of conditional generation in generative AI.

In Fig.1, panel (c), we report the absolute error on droplet trajectories over time for different effective models $\tilde{\mathbf{x}}_i(t)$ compared to the ground truth simulation data $\mathbf{x}_i(t)$. A detailed view of the first 150 timesteps is shown in Fig.3. We observe that, in terms of one-step prediction error, both data-driven models are more accurate than the EV model. However, the GNN model demonstrates unstable dynamics, with errors diverging after a few hundred timesteps. In contrast, the EV and EV+GNN models maintain stable dynamics. Furthermore, the EV+GNN model consistently outperforms the baseline EV model in terms of accuracy, maintaining lower error levels throughout all timesteps.

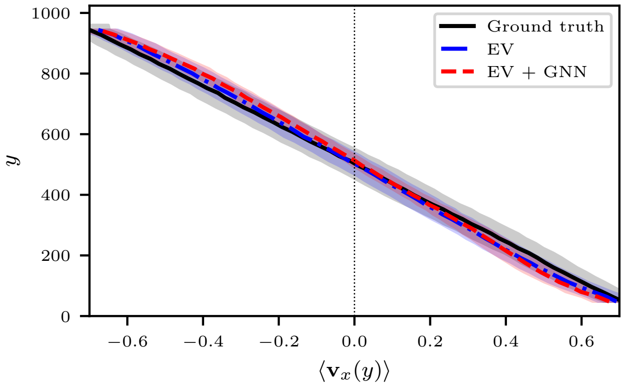

Figure 4. Average shear velocity profile $\langle \mathbf{v}_x(y) \rangle$, for ground truth data, EV and EV+GNN

4.5. Full Simulation Results

Next, we consider the setting where all the non-boundary particles are considered active and evolved according to the effective models. In this setting, we observed some instabilities that where not present in the active-passive setup. To address these instabilities, we implemented a modification in the effective models where, at each timestep, the average displacement in the x and y directions is subtracted from each prediction, in formulas:

$$\mathbf{v}_i(t+1)^{EV+GNN} - \frac{1}{N}\sum_{j=1}^{N} \mathbf{v}_j(t+1)^{EV+GNN}$$

This adjustment, combined with the multi-step training strategy and the hybrid model, proved sufficient to stabilize the effective models, although further study is needed for full understanding.

Given the chaotic nature of the system—where any perturbation in the dynamics leads to an exponential divergence from the ground-truth trajectories—the subsequent results are based on statistics computed from long-running simulations. These simulations start from the initial condition and evolve all the non-boundary droplets using the two stable effective models, EV and EV+GNN.

We report in Fig. 4 the average x-velocity profile as a function of y, for the ground truth data and the two effective models. This figure confirms that both the EV and the EV+GNN are able to capture the average profile within error-bars.

In Fig. 5, we report the Probability Distribution Function (PDF) for the x and y component of the velocity field, for ground truth data and the two effective models. Notably, we can see that the EV+GNN model recovers a PDF of $\mathbf{v}_y$ more closely resembling the ground truth data. Here the main difference is found in the tails of the PDF, corresponding to events with higher velocity value, less represented in the EV model. These events can be associated to plastic events happening in the droplet configuration, in which the topology of the droplet configuration drastically changes in a very short time window (Kumar et al., 2023). In the PDF of $\mathbf{v}_x$, instead, we can see some oscillations and asymmetries in both the effective models behavior, to be further investigated.

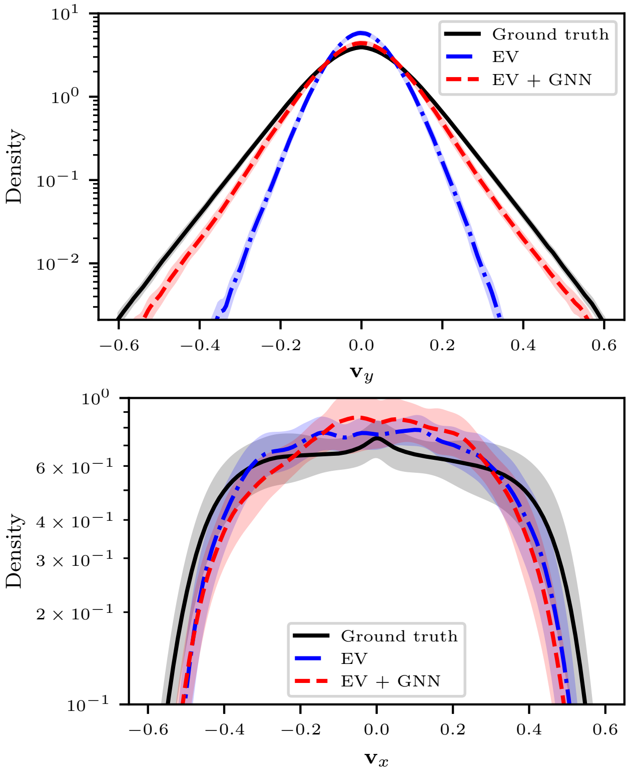

Figure 5. PDF of the y and x-component of the droplet velocity $\mathbf{v}$, for ground truth data, EV and EV+GNN.

5. Discussion

In this work, we developed a neural model for simulating the dynamics of droplets in stabilized dense emulsions, aiming to capture their emergent behavior. We proposed a hybrid approach that combines a Graph neural network with a handcrafted baseline model addressing the challenges associated with modeling the complex interactions in densely packed systems. The hybrid model, termed EV+GNN, demonstrated significant improvements over standalone GNN and EV models, particularly in maintaining long-term stability and accuracy in predicting droplet displacements.

5.1. Limitations

While the initial results from our work are encouraging, there are substantial limitations to the results that we report.

Spherical Droplets. One limitation is our approximation of droplets as spherical with constant size (Sec 3.1). We have verified that the assumption of constant droplet size holds to good approximation for the emulsion settings considered, which is consistent with the fact that both the fluids composing the emulsion are incompressible. However, the spherical approximation does not account for actual deformations of the droplet shape, possibly limiting the information available to the models. The GNN that we employ has access to the relative distances to all neighbors and does not assume rotational symmetry when computing messages, so it is possible that the GNN is able to implicitly represent time-varying droplet shapes. With that said, the implications of the spherical approximation require further investigation.

Scalability and Generalization. A second limitation of the current results is that we consider a single geometry (2D Couette Flow) and system size. There are in principle no fundamental limitations in terms of computational scalability; the number of neighbors per droplet is constant (approximately 10 in our experiments), so computation will scale approximately linearly with the number of droplets. However the stability of roll-outs has not been evaluated in larger systems sizes. We have also not considered more complex systems, such as polydisperse emulsions, emulsions in the presence of surfactants, or active emulsions. While the fundamental design indicates potential adaptability to these scenarios, additional research is necessary to validate and optimize scalability in these more involved contexts.

Baselines. Finally, this work lacks a comparison to baseline methods on a variety of datasets, as we are accustomed to in ML research. Because the elasto-viscous model is quite specific to the setting of emulsions that we consider in this paper, it not clear whether evaluations of the proposed model on different datasets would be informative.

5.2. Outlook

Very recent work by Toshev et al. (2024) develops Neural SPH, which uses SPH-based relaxation methods to reduce errors in learned neural solver (such as GNS), both at training and at inference time. It may be possible to identify similar relaxations that will be applicable in the context of droplet dynamics in emulsions.

More generally, future research will explore more advanced GNN architectures and validate the model across various types of emulsions and conditions. This work aims to contribute to a deeper understanding of emulsion physics and enhances the efficiency of numerical simulations, aligning with the broader objective of developing scalable models for emerging behavior in complex systems.

References

Anderson, P. W. More is different. *Science*, 177(4047): 393–396, August 1972.

Barkat, A., Khan, B., Naveed, A., Muhammad, H., Khan, H. M. s., Waseem, K., Mahmood, T., Rasul, A., and Iqbal, M. Basics of pharmaceutical emulsions: A review. *African journal of pharmacy and pharmacology*, 525:2715–2725, 12 2011.

Battaglia, P., Hamrick, J. B. C., Bapst, V., Sanchez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G. E., Vaswani, A., Allen, K., Nash, C., Langston, V. J., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., and Pascanu, R. Relational inductive biases, deep learning, and graph networks. *arXiv*, 2018.

Bruna, J., Zaremba, W., Szlam, A., and Lecun, Y. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014.

Chu, M. and Thuerey, N. Data-driven synthesis of smoke flows with cnn-based feature descriptors. *ACM Trans. Graph.*, 36(4), jul 2017. ISSN 0730-0301.

Crick, F. and Koch, C. Toward a neurobiological theory of consciousness. *Seminars in the Neurosciences*, 2:263–275, 1990.

Duraisamy, K., Iaccarino, G., and Xiao, H. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 2018.

Friberg, S., Larsson, K., and Sjoblom, J. (eds.). *Food Emulsions*. Food Science and Technology. CRC Press, Boca Raton, FL, 4 edition, November 2003.

Frisch, U. *Turbulence: The Legacy of AN Kolmogorov*. Cambridge University Press, 1995.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 06–11 Aug 2017.

Gingold, R. A. and Monaghan, J. J. Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389, December 1977. ISSN 0035-8711.

Hoogerbrugge, P. J. and Koelman, J. M. V. A. Simulating Microscopic Hydrodynamic Phenomena with Dissipative Particle Dynamics. *Europhysics Letters*, 19(3):155, June 1992. ISSN 0295-5075.

Krueger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G., and Viggen, E. *The Lattice Boltzmann Method: Principles and Practice*. Graduate Texts in Physics. Springer, 2016. ISBN 978-3-319-44647-9.

Kumar, P., Benzi, R., Trampert, J., and Toschi, F. A multi-component lattice boltzmann approach to study the causality of plastic events. *Philos. Trans. A Math. Phys. Eng. Sci.*, 378(2175):20190403, July 2020a.

Kumar, P., Korkolis, E., Benzi, R., Denisov, D., Niemeijer, A., Schall, P., Toschi, F., and Trampert, J. On interevent time distributions of avalanche dynamics. *Sci. Rep.*, 10 (1):626, January 2020b.

Kumar, P., Benzi, R., Trampert, J., and Toschi, F. Direct observations of causal links in plastic events and relevance to earthquake seismology. *Phys. Rev. Res.*, 5:033211, Sep 2023.

Ladický, L., Jeong, S., Solenthaler, B., Pollefeys, M., and Gross, M. Data-driven fluid simulations using regression forests. *ACM Transactions on Graphics*, 34(6):199:1–199:9, November 2015. ISSN 0730-0301.

Levin, S. A. Self-organization and the Emergence of Complexity in Ecological Systems. *BioScience*, 55(12):1075–1079, 12 2005. ISSN 0006-3568.

Li, Y., Wu, J., Tedrake, R., Tenenbaum, J. B., and Torralba, A. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *International Conference on Learning Representations*, 2019.

Lino, M., Fotiadis, S., Bharath, A. A., and Cantwell, C. D. Multi-scale rotation-equivariant graph neural networks for unsteady Eulerian fluid dynamics. *Physics of Fluids*, 34(8):087110, August 2022. ISSN 1070-6631.

Müller, M., Heidelberger, B., Hennix, M., and Ratcliff, J. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, April 2007. ISSN 1047-3203.

Park, C. W., Kornbluth, M., Vandermause, J., Wolverton, C., Kozinsky, B., and Mailoa, J. P. Accurate and scalable graph neural network force field and molecular dynamics with direct force architecture. *Npj Comput. Mater.*, 7(1), May 2021.

Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., and Battaglia, P. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021.

Prantl, L., Ummenhofer, B., Koltun, V., and Thuerey, N. Guaranteed Conservation of Momentum for Learning Particle-based Fluid Dynamics. *Advances in Neural Information Processing Systems*, 35:6901–6913, December 2022.

Ruhe, D., Brandstetter, J., and Forré, P. Clifford group equivariant neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., and Battaglia, P. Learning to Simulate Complex Physics with Graph Networks. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 8459–8468. PMLR, November 2020.

Satorras, V. G., Hoogeboom, E., and Welling, M. E(n) equivariant graph neural networks. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9323–9332. PMLR, 18–24 Jul 2021.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80, January 2009. ISSN 1941-0093.

Sulsky, D., Zhou, S.-J., and Schreyer, H. L. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, 87(1):236–252, May 1995. ISSN 0010-4655.

Tompson, J., Schlachter, K., Sprechmann, P., and Perlin, K. Accelerating Eulerian Fluid Simulation With Convolutional Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3424–3433. PMLR, July 2017.

Toshev, A. P., Erbesdobler, J. A., Adams, N. A., and Brandstetter, J. Neural SPH: Improved Neural Modeling of Lagrangian Fluid Dynamics. (arXiv:2402.06275), February 2024.

Ummenhofer, B., Prantl, L., Thuerey, N., and Koltun, V. Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations*, 2020.

Van Veen, B., van Drongelen, W., Yuchtman, M., and Suzuki, A. Localization of brain electrical activity via linearly constrained minimum variance spatial filtering. *IEEE transactions on bio-medical engineering*, 44:867–80, 10 1997.

Wang, R., Kashinath, K., Mustafa, M., Albert, A., and Yu, R. Towards Physics-informed Deep Learning for Turbulent Flow Prediction. In *Proceedings of the 26th*

ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20, pp. 1457–1466, New York, NY, USA, August 2020a. Association for Computing Machinery. ISBN 978-1-4503-7998-4.

Wang, R., Walters, R., and Yu, R. Incorporating Symmetry into Deep Dynamics Models for Improved Generalization. In *International Conference on Learning Representations*, October 2020b.

Wang, R., Walters, R., and Yu, R. Approximately Equivariant Networks for Imperfectly Symmetric Dynamics. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 23078–23091. PMLR, June 2022.

Wiewel, S., Becher, M., and Thuerey, N. Latent Space Physics: Towards Learning the Temporal Evolution of Fluid Flow. *Computer Graphics Forum*, 38(2):71–82, 2019. ISSN 1467-8659.

Wörner, M. Numerical modeling of multiphase flows in microfluidics and micro process engineering: a review of methods and applications. *Microfluid. Nanofluidics*, 12 (6):841–886, May 2012.

Yang, C., Yang, X., and Xiao, X. Data-driven projection method in fluid simulation. *Computer Animation and Virtual Worlds*, 27(3-4):415–424, 2016. ISSN 1546-427X.

Zhdanov, M., Ruhe, D., Weiler, M., Lucic, A., Brandstetter, J., and Forré, P. Clifford-Steerable Convolutional Neural Networks, February 2024.