Bubble Column Simulation with EGNN

Luca Mainardi (2014602) Rachna Ramesh (1956477)

November 8, 2024

1 Context

Bubble columns are crucial components in various industrial processes, including chemical processing, wastewater treatment, and gas-liquid mass transfer systems [2]. Their utility lies in enhancing mixing and mass transfer between the gas and liquid phases, making them essential for efficient industrial operations. However, modeling the complex dynamics within these systems poses significant challenges due to intricate interactions between bubbles, the surrounding fluid, and external forces such as gravity.

Bubble motion is influenced by nonlinear forces such as buoyancy, drag, and surface tension, as well as interactions with neighboring bubbles. These interactions can lead to complex phenomena, including bubble deformation, coalescence, and breakup. Accurate modeling of bubble columns requires high-fidelity simulations, such as computational fluid dynamics (CFD) or methods based on spherical harmonics, to capture behaviors like bubble deformation and rising patterns. While these methods provide precise results, they are computationally intensive and impractical for large-scale or real-time applications necessary for tasks like optimization, uncertainty quantification, or industrial control systems.

The complexity of bubble behavior, compounded by the presence of asymmetric forces like gravity, necessitates the development of efficient surrogate models. These models aim to approximate the behavior of high-fidelity simulations while significantly reducing computational costs. Graph-based machine learning techniques offer a promising avenue for achieving this goal. By representing the bubble system as a graph, where nodes correspond to individual bubbles and edges encode their relative interactions, we can simulate the system's dynamics efficiently while respecting physical symmetries.

In this work, we develop an E(n) Equivariant Graph Neural Network (EGNN) [4] to model bubble dynamics within a column, focusing on key physical symmetries such as translation, rotation, and reflection. By representing bubbles as nodes in a graph and using spherical harmonics to capture deformations, the EGNN can predict system dynamics while maintaining physical consistency. We incorporate a novel update equation to handle spherical harmonic coefficients, in order to enable the model to account for bubble deformations.

2 Problem Formulation

The objective of this work is to develop a surrogate model that efficiently predicts the dynamics of N interacting bubbles within a bubble column, capturing both their translational motion and shape deformations due to interactions and external forces. Each bubble, potentially of different sizes, is subject to forces such as surface tension, buoyancy, and gravity, which affect individual bubble motion and interactions with neighbors. The inclusion of gravity introduces asymmetry into the system's dynamics.

We aim to model the evolution of the system over time by predicting the future states of the bubbles, including their positions $\mathbf{x}_i \in \mathbb{R}^3$, velocities $\mathbf{v}_i \in \mathbb{R}^3$, and shape deformations represented by spherical harmonic coefficients $\mathbf{orb}_i \in \mathbb{R}^k$, where $i = 1, \ldots, N$.

Formally, given the current state of the system at time t, represented by $(\mathbf{x}_i^t, \mathbf{v}_i^t, \mathbf{orb}i^t)$, we aim to predict the state at the next time step $t+1$: $(\mathbf{x}_i^{t+1},\ \mathbf{v}_i^{t+1},\ \mathbf{otb}_i^{t+1})$.

2.1 Dataset

The dataset used in this study contains six simulations, in which the trajectories and shape deformations of 32 interacting bubbles within a cubic domain over time are recorded. For each bubble at each time step, the dataset records the bubble's position $\mathbf{x}_i \in \mathbb{R}^3$, velocity $\mathbf{v}_i \in \mathbb{R}^3$, and spherical harmonic coefficients $\mathbf{orb}_i \in \mathbb{R}^{225}$, along with the time step t. Each trajectory (simulation) in the dataset also includes two parameters: the initial diameter d_i for all bubbles in that trajectory, and the gas hold-up ϵ, representing the volumetric fraction occupied by gas within the bubble column.

The initial bubble diameter d_i influences buoyancy and drag forces, which shape bubble interactions within the column.

The gas hold-up ϵ directly influences gas distribution and bubble formation, affecting the fluid-dynamic interactions within the column [1].

The bubble system is confined within a cubic domain with periodic boundary conditions applied in all three spatial dimensions. The box size L_{box} is determined by:

$$L_{\text{box}} = \left(\frac{N \times \frac{4}{3}\pi r_{\text{bub}}^3}{\epsilon} \right)^{1/3}$$

where r_{bub} is the bubble radius, and $N = 32$ is the number of bubbles. The formula applies uniformly to the width, depth, and height of the box.

To create a continuous simulation environment, periodic boundary conditions are applied in each spatial dimension, meaning that when a bubble exits one side of the box, it re-enters from the opposite side. However, bubble positions in the dataset are recorded relative to a fixed origin (0, 0, 0), which can lead to coordinates outside the intended bounds of the cubic domain. As part of preprocessing, we applied these periodic boundary conditions to adjust any out-of-bound coordinates, ensuring all bubble positions remain within the box range $[0, L_{\text{box}}]$ in each dimension.

After applying periodic boundary conditions, we scaled each position by the box size L_{box}. This scaling step standardizes the position data, bringing every trajectory into a consistent range regardless of the initial box size. We further normalized the velocities by rescaling them based on the minimum and maximum velocity values observed in the training data, which helps to maintain a uniform velocity range across trajectories and facilitates model training.

3 Approach

3.1 Graph Representation of the System

The dynamic system of bubbles is represented as a graph, where each bubble corresponds to a node, and interactions between bubbles are captured as edges. In this representation, nodes contain attributes describing the physical state of each bubble, while edges represent relationships between bubbles based on their relative positions and velocities.

Each node is associated with attributes that reflect both the motion and shape characteristics of the bubble at each time step. These attributes include the 3D position coordinates (x, y, z) and the velocity components (v_x, v_y, v_z). To represent the shape of each bubble, we use spherical harmonic coefficients, specifically selecting the first 16 coefficients (up to l=3) from the available 225. This choice provides a simplified yet effective description of the bubble's shape, capturing key deformations while minimizing computational complexity. These lower-order harmonics focus on primary shape features, which are generally sufficient for modeling bubble deformations in this system. A more detailed explanation of this choice is presented in Section 3.2. Additionally, each node stores the gas hold-up value ϵ and the norm of the velocity vector, which further characterizes each bubble's dynamics.

Edges between nodes capture the spatial and motion-based relationships among bubbles. For each pair of interacting bubbles, edges store the relative positions $(\Delta x, \Delta y, \Delta z)$ and relative velocities $(\Delta v_x, \Delta v_y, \Delta v_z)$. These relative attributes enable the model to learn interaction patterns, as the dynamics of bubble interactions depend on both the distance between bubbles and their velocity differences.

We omit an explicit encoding of bubble size in the graph since the first spherical harmonic coefficient, known as the monopole term, inherently represents the bubble's overall size. This coefficient is proportional to the bubble's volume or radius, so variations in bubble size are already captured through

this feature. By relying on this harmonic term, we avoid redundancy, as it implicitly incorporates size information through the harmonic representation.

3.2 Spherical Harmonic Coefficients

The full set of 225 spherical harmonic coefficients corresponds to a maximum order $l_{\max} = 14$, which would provide a highly detailed description of the bubble's surface. However, in this study, we use spherical harmonics up to order $l = 3$, reducing the number of coefficients from 225 to 16. This approach balances computational efficiency with accuracy in describing essential bubble deformations. Each harmonic order from $l = 0$ to $l = 3$ captures progressively more complex aspects of the bubble's shape, as illustrated in Figure 1.

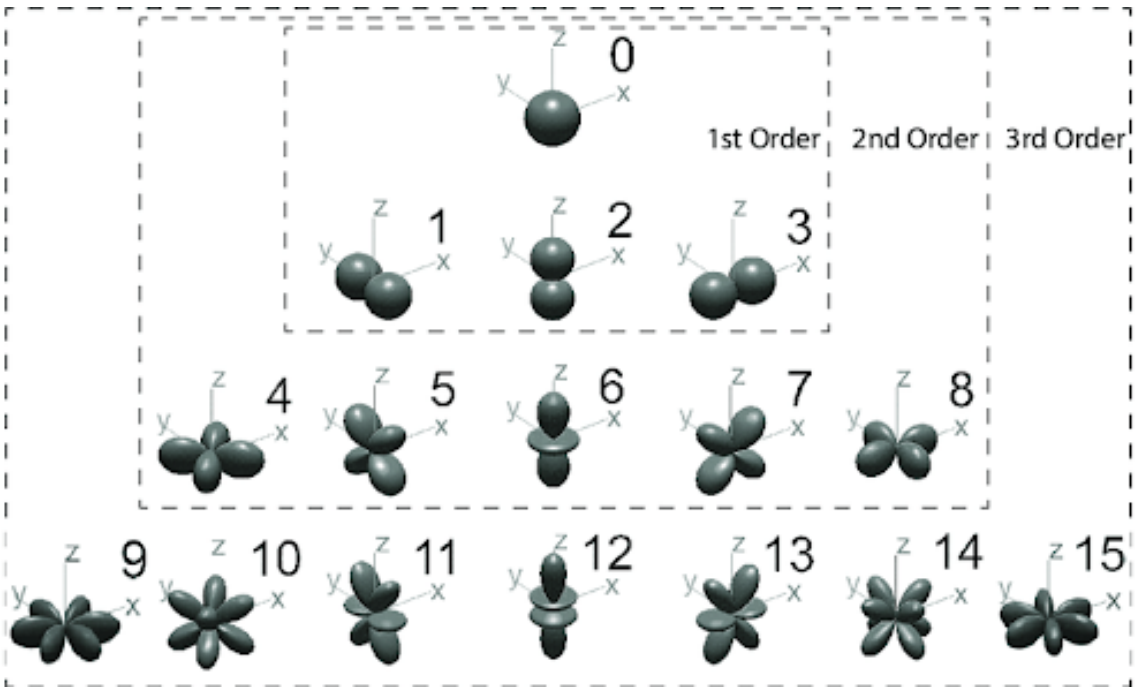

Figure 1: Spherical Harmonic Coefficients up to order $l = 3$. Figure taken from [3] Page 3.

The monopole term (order $l = 0$) describes the perfectly spherical shape of the bubble, which represents its natural configuration in the absence of external forces. Within a water column, forces such as buoyancy and surface tension tend to maintain this spherical form, as they act uniformly on all sides of the bubble.

The dipole term (order $l = 1$) captures the first asymmetry, in which the bubble may elongate or shift slightly due to drag forces as it rises. This elongation reflects the displacement of the bubble's center of mass, which becomes prominent under buoyancy as the drag from water resists its upward motion.

The quadrupole term (order $l = 2$) represents ellipsoidal or flattened deformations that occur when a bubble is subjected to uneven forces, such as pressure variations or boundary interactions. These anisotropic forces cause the bubble to elongate or flatten in a symmetric but non-spherical way. Surface tension, however, acts to restore the bubble to a more spherical shape following such deformations, particularly when it is near container walls or within an unstable fluid flow.

The octupole term (order $l = 3$) captures localized and complex shape distortions due to turbulence, neighboring bubbles, or other fluid fluctuations. These higher-order deformations represent undulations or wobbles on the bubble surface, which arise under dynamic or turbulent conditions. In these cases, the octupole term can model the detailed surface dynamics that occur from non-uniform pressures or high-shear flow environments.

By focusing on harmonics up to order $l = 3$, we aim to approximate the bubbles' general shape without introducing unnecessary complexity. Expanding the model to higher orders remains an option if further detail is needed to describe minor irregularities or subtle surface features.

3.3 Symmetries in our system

The simulation of bubbles in a column requires the model to account for several physical symmetries, such as translations, rotations, and reflections. Our system must respect both invariance and equivariance under these transformations.

Translation symmetry requires that the equations governing bubble dynamics remain consistent under spatial translations. If the entire bubble column is shifted by a vector $\mathbf{t}$, the physical relationships among bubbles should remain unchanged. Although the positions of individual bubbles are translated as $\mathbf{x}_i\prime = \mathbf{x}_i + \mathbf{t}$, their relative distances $|\mathbf{x}_i - \mathbf{x}_j|$ are unaffected. Consequently, our model must be equivariant to translations, meaning that if the input positions are translated, the predicted positions should translate by the same vector $\mathbf{t}$.

Rotation and reflection symmetry require that the model be equivariant to both rotations and reflections. If the bubble system undergoes a rotation or reflection represented by a transformation $\mathbf{R} \in O(3)$, where $O(3)$ includes both rotations and reflections in three dimensions, the positions, velocities, and deformations of the bubbles should transform accordingly. For a bubble's position $\mathbf{x}_i$, a rotation or reflection would apply as $\mathbf{x}_i\prime = \mathbf{R}\mathbf{x}_i$. In practical terms, this means the model's predictions must transform consistently: when the bubble configuration rotates or reflects, the output positions and velocities should undergo the same transformation.

Velocity symmetry applies a similar principle to velocities, as they are vector quantities and must respect equivariance to rotations and invariance to translations. Translation invariance implies that when the entire system is shifted, the velocities remain unchanged: $\mathbf{v}_i\prime = \mathbf{v}_i$. However, if the system is rotated or reflected by a matrix $\mathbf{R}$, the velocities should transform as $\mathbf{v}_i\prime = \mathbf{R}\mathbf{v}_i$. Thus, our model must respect equivariance to rotations and reflections for accurate velocity predictions.

Spherical harmonics symmetry is inherent to the spherical harmonic functions $Y_{lm}(\theta, \phi)$, which form an orthonormal basis on the sphere and describe bubble surface deformations. Each spherical harmonic is indexed by two integers, l and m , that determine its complexity: l represents the degree of the harmonic, which defines the detail level of the surface deformation, while m , ranging from $-l$ to $+l$, controls the function's variation with the azimuthal angle ϕ. Together, these indices determine the harmonic's nodal pattern and oscillations across the sphere.

Spherical harmonics are equivariant to rotations. When a rotation $\mathbf{R} \in SO(3)$ is applied to the system, the harmonics transform into a linear combination of harmonics of the same degree l. In terms of the spherical harmonics coefficients orb_{lm}, which capture the bubble's surface deformation, the transformation under a rotation is expressed as

$$orb_{lm\prime} = \sum_{m\prime=-l}^{l} D^{(l)} mm\prime(\mathbf{R})\, orblm\prime$$

where $D^{(l)}_{mm\prime}(\mathbf{R})$ are elements of the Wigner rotation matrix for degree l. These matrices define the transformations of spherical harmonics under rotations and are part of the irreducible representations of the rotation group $SO(3)$. To ensure that bubble deformations remain consistent under rotations and reflections, our model must update these coefficients in an equivariant manner.

3.4 E(n) Equivariant Graph Neural Network (EGNN)

We chose the E(n) Equivariant Graph Neural Network (EGNN) proposed by Satorras et al. as the starting point for the development of our model, as it has proven effective for complex dynamic tasks similar to ours. In particular, EGNN has shown excellent performance in predicting the position and velocity of interacting particles in three-dimensional systems, while respecting physical symmetries such as rotations and translations [4].

However, we made some modifications to the original model to correctly handle spherical harmonics and their properties. This was necessary to update the harmonic coefficients in an equivariant manner with respect to rotational and reflective transformations, ensuring that the predicted bubble deformations are treated in a physically consistent way within the model.

The EGNN is based on a message passing mechanism that leverages the relative distances between nodes and preserves the equivariant properties of the system. Its architecture is defined by the following key equations, which are used to update velocities, positions, and node embeddings. Figure 2 shows how a message passing layer works.

1. **Message computation:**

   For each pair of nodes i and j, the message m_{ij} is computed as:

   $$m_{ij} = \phi_e(h_i^l, h_j^l, \|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

   where h_i^l and h_j^l are the embeddings of nodes i and j at layer l, and $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ is the squared distance between their positions. The function ϕ_e is a neural network that uses this information to compute the message, which will then be used to update both velocity and position.

2. **Velocity update:**

   The velocities of the nodes are updated by considering the messages received from neighboring nodes. The velocity $\mathbf{v}_i^{l+1}$ of node i is updated using the following equation:

   $$\mathbf{v}_i^{l+1} = \phi_v \mathbf{v}_i^l + C \cdot \sum_{j \neq i} (\mathbf{x}_i - \mathbf{x}_j) \phi_x(m_{ij})$$

   where C is a normalization factor, $\mathbf{v}_i^l$ is the velocity of node i at layer l, and $\phi_x(m_{ij})$ is a function that depends on the message m_{ij}. The velocity update is influenced by the relative positions of the nodes and the messages received.

3. **Position update:**

   The positions $\mathbf{x}_i^{l+1}$ of the nodes are then updated using the new velocities:

   $$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \mathbf{v}_i^{l+1}$$

   This simple equation updates each node's position based on its updated velocity, ensuring that the movement of the nodes is consistent with the system's geometric information.

4. **Message aggregation:**

   The messages calculated from neighboring nodes are aggregated to update each node's embedding. The aggregation is performed as follows:

   $$m_i = \sum_{j \neq i} m_{ij}$$

   where m_i is the aggregated message for node i, obtained by summing the messages received from all neighboring nodes j.

5. **Node embedding update:**

   Finally, the node embeddings are updated using the current features and the aggregated messages:

   $$h_i^{l+1} = \phi_h(h_i^l, m_i)$$

   where h_i^{l+1} is the new embedding of node i, computed as a function of the current features h_i^l and the aggregated messages m_i. The function ϕ_h is a neural network that learns how to combine this information to produce meaningful new embeddings.

The EGNN maintains equivariance with respect to rotations and translations by using only the relative distances $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ in the message computations and updates positions and velocities consistently. This ensures that the model respects the physical symmetries required for a three-dimensional dynamic system, without the need for higher-order representations, such as tensors or spherical harmonics, in intermediate features.

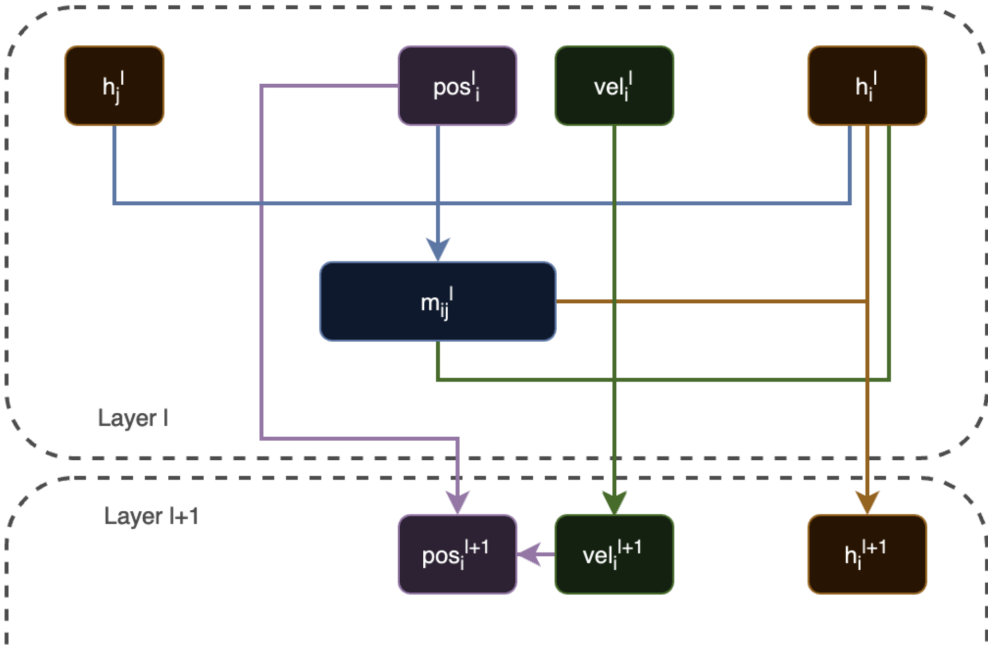

Figure 2: Flow diagram of an EGNN message passing layer

3.5 Update Equation for Spherical Harmonics Coefficients

In order to handle spherical harmonics, we extended the EGNN by introducing an update equation for the harmonic coefficients, which is designed to respect the system's physical symmetries, specifically invariance to translations and equivariance to rotations, while capturing the interactions between the bubbles (as shown in Figure 3):

$$\text{orb}_i^{l+1} = \phi_{\text{orb}}(h_i^l) \cdot \text{orbs}_i^l + C \cdot \sum_{j \neq i} \|\mathbf{x}_i - \mathbf{x}_j\|_{PBC}^2 \cdot \phi_{\text{xorb}}(m_{ij}) \cdot (\text{orb}_i^l - \text{orb}_j^l)$$

The first term of the equation, $\phi_{\text{orb}}(h_i^l) \cdot \text{orb}_i^l$, models the intrinsic dynamics of each bubble based on its current state. Here, $\phi_{\text{orb}}(h_i^l)$ is a neural network implemented as a Multi Layer Perceptron (MLP) with SiLU activation, designed to produce an invariant scalar from the node embedding h_i^l of bubble i. Since h_i^l is a vector of invariant features (independent of spatial direction), the MLP, composed of fully connected layers and non-linear activations such as ReLU, preserves this invariance by producing an output scalar that is also invariant. When this scalar is multiplied by orb_i^l, the spherical harmonic coefficients, the result is a term that remains equivariant to rotations. This ensures that the update of the bubble's harmonic coefficients is consistent with rotational transformations, preserving the spherical shape structure regardless of external interactions.

The second term, $C \cdot \sum_{j \neq i} \|\mathbf{x}_i - \mathbf{x}_j\|_{ORB}^2 \cdot \phi_{\text{xorb}}(m_{ij}) \cdot (\text{orb}_i^l - \text{orb}_j^l)$, introduces the interaction between neighboring bubbles. The distance term $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ is invariant under both translations and rotations, ensuring that the interaction strength depends on the distance between the bubbles. The function $\phi_{\text{xorb}}(m_{ij})$, also a neural network implemented as an MLP with SiLU activation, produces an invariant scalar from the message m_{ij}. For $\phi_{\text{xorb}}(m_{ij})$ to respect the required invariance, it is important that the message m_{ij} consists of invariant features, which is ensured by the EGNN architecture.

The difference $(\text{orb}_i^l - \text{orb}_j^l)$ between the harmonic coefficients of bubbles i and j is equivariant to rotations, ensuring that the influence of one bubble's shape on another transforms correctly under rotational transformations. This term captures how neighboring bubbles influence each other's deformations based on their relative distances and respective shapes.

The relative distance $\|\mathbf{x}_\text{i} - \mathbf{x}_\text{j}\|_{\text{PBC}}$ represents the distance adjusted for Periodic Boundary Con-

ditions, selecting the shortest separation across boundaries. This adjustment is applied consistently throughout the model, ensuring that all distances respect the periodic nature of the simulation domain. By updating the distances in the message-passing framework, we ensure that interactions correctly scale across boundaries, preserving the required symmetries.

Invariance to translations is maintained because the only positional dependency in the equation comes from the relative distances between the bubbles, $\|\mathbf{x}_i - \mathbf{x}_j\|^2$, which remain invariant when the system is translated. Equivariance to rotations is ensured by the way the spherical harmonic coefficients transform under rotation, following the Wigner D-matrices, and by the fact that operations such as differences and sums of equivariant vectors preserve equivariance.

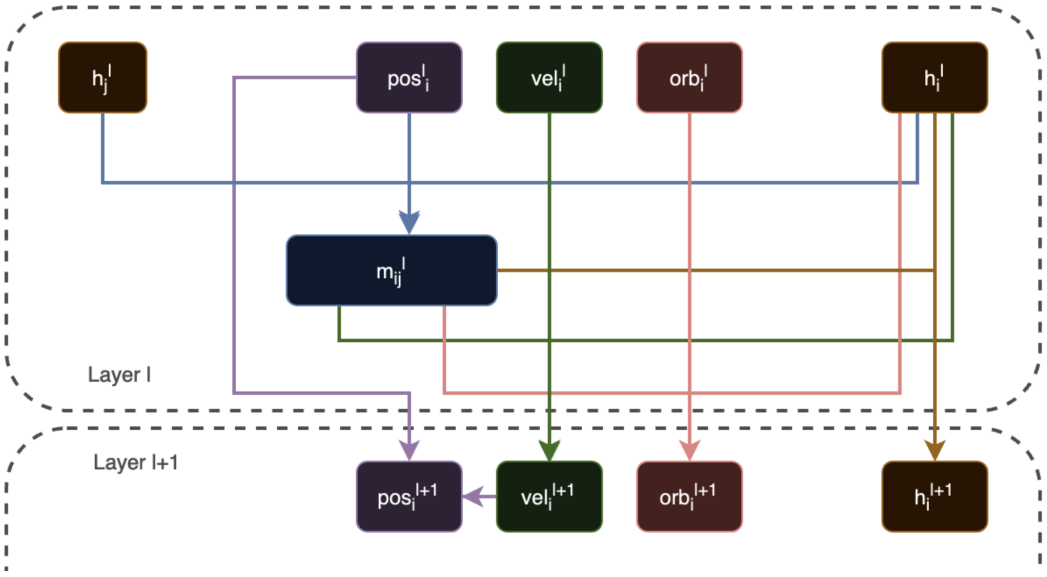

Figure 3: Flow diagram of an EGNN message passing layer, with the added term for spherical harmonic coefficients

3.6 Initial node embedding

Each node in the graph is initialized with an embedding h_0 that captures essential features for spatial, velocity, and shape information. To compute this initial embedding, we use a multi-layer perceptron (MLP), denoted $\phi_{\text{in\_emb}}$, which consists of a configurable number of linear layers, each followed by a SiLU activation. The input to $\phi_{\text{in\_emb}}$ includes the norm of the velocity vector and the gas hold-up value ϵ.

This initial embedding h_0 is computed as:

$$h_0 = \phi_{\text{in\_emb}}(\|\mathbf{v}\|, \epsilon)$$

where $\|\mathbf{v}\|$ is the velocity norm. This transformation maps the input features into a hidden space that is compatible with the model's equivariant updates. The resulting h_0 embeddings are then passed to the EGNN's message-passing layers.

A diagram of the complete architecture of our network is shown in Figure 4.

4 Experiments

The trajectories in the dataset include two initial diameter values, 4 mm and 6 mm, and three gas hold-up (ϵ) values: 15, 25, and 35. Given the limited number of trajectories, we split the data to maximize diversity across the training, validation, and test sets. The validation set includes a trajectory with a 6 mm diameter and the highest ϵ value of 35, while the test set contains a trajectory with a 4 mm

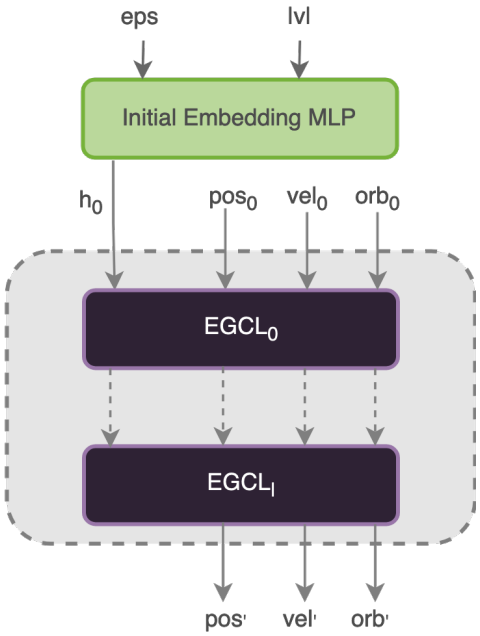

Figure 4: Diagram of the developed EGNN.

diameter and the lowest ϵ value of 15. This setup provides distinct conditions for evaluation, allowing the model to be tested on previously unseen combinations. The training set includes the four remaining trajectories, covering a range of bubble diameters and gas hold-up values.

We conducted two types of experiments to assess the model's predictive capabilities. In the first experiment, we calculated the mean squared error (MSE) for each individual data point in the test set, evaluating the model's accuracy in predicting the immediate next state of each bubble.

In the second experiment, we split the test trajectory, which spans 250 steps, into smaller segments of 10 consecutive time steps each. Each segment is treated as a temporary, individual trajectory, enabling us to compute the MSE in position, velocity, and spherical harmonic coefficients for each segment independently. Within each segment, predictions are generated autoregressively: after an initial input, the model predicts each subsequent time step based on its own previous predictions, rather than relying on the ground truth at each step.

After evaluating all 10-step segments in this autoregressive manner, we compute the MSE between the model's predictions and the ground truth values for each segment and then calculate the mean MSE across all segments.

4.1 Setup

This section introduces the experimental setup, detailing the baseline model used for comparison with the EGNN, along with the loss function and model hyperparameters. The baseline model serves as a reference to evaluate the performance improvements achieved by the EGNN approach. Additionally, the loss function is designed to assess the discrepancies between predicted and actual bubble velocities and shape deformations, providing a reliable metric for model optimization. The chosen model hyperparameters are also described.

4.1.1 Baseline Model: Message Passing Neural Network (MPNN)

The baseline model is a Message Passing Neural Network (MPNN) tailored for capturing bubble dynamics in graph-structured data. This MPNN architecture is composed of several key components: an input embedding layer, multiple message-passing layers, and a final output layer.

The model starts with an input embedding layer that combines initial node features (positions, velocities, and orbital coefficients) with node attributes (gas hold-up ϵ and velocity norm) into a higher-dimensional feature space. This layer, implemented as a Multi-Layer Perceptron (MLP), consists of two

linear layers with ReLU activations. After embedding, the model applies a series of message-passing layers designed to iteratively update each node's representation through interaction with neighboring nodes. Each layer computes messages based on node and edge features, which are then aggregated to refine the node embeddings. This iterative message-passing process allows the model to learn spatial and relational structures within the graph, with each layer progressively capturing deeper contextual information about local neighborhood interactions. Finally, the output layer uses the refined node embeddings from the last message-passing layer to predict the positions, velocities, and spherical harmonic coefficients of the bubbles.

The primary difference between the MPNN and the EGNN model is the treatment of symmetries and geometric invariances. The EGNN explicitly incorporates equivariant transformations for maintaining rotational and translational symmetry in spatial dimensions, allowing it to capture geometric structures directly. This is achieved through the use of equivariant message functions that consider the relative position between nodes. The MPNN, by contrast, does not enforce equivariance and treats the spatial features without any specific symmetry constraints.

4.1.2 Loss Function

We used a weighted Mean Squared Error (MSE) as the loss function to capture discrepancies between the predicted and actual values for both bubble velocities and shape deformations. The MSE penalizes larger errors, helping the model focus on accurately reconstructing both velocity and shape aspects of bubble dynamics. This approach is particularly suitable for our task, as it promotes balanced learning between positional data (velocities) and shape features (harmonic coefficients), reducing errors across different aspects of bubble behavior.

To prioritize the most prominent shape features, we applied exponentially decreasing weights to the harmonic components. Spherical harmonics capture shape deformation details that grow more complex with each harmonic order, with lower-order terms representing larger, more significant shape features and higher-order terms capturing finer, localized details. By assigning higher weights to the lower-order harmonics, the model is encouraged to focus on the most prominent shape characteristics while still learning finer details when necessary. The decay rate of these weights can be adjusted, allowing the model to flexibly emphasize different scales of deformation based on the specific needs of the task.

The loss function is defined as follows:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left((\mathbf{v}_{\text{pred}} - \mathbf{v}_{\text{true}})^2 + \sum_{i=0}^{15} (\mathbf{H}_{\text{pred},i} \cdot w_i \cdot \text{orb\_weight} - \mathbf{H}_{\text{true},i} \cdot w_i \cdot \text{orb\_weight})^2 \right)$$

where $\mathbf{v}_{\text{pred}}$ and $\mathbf{v}_{\text{true}}$ represent the predicted and actual velocity vectors, and $\mathbf{H}_{\text{pred},i}$ and $\mathbf{H}_{\text{true},i}$ are the predicted and actual values of the i-th spherical harmonic coefficient, ranging from 0 to 15. The weighting term $w_i = \exp(-\text{decay\_rate} \cdot i)$ applies an exponentially decreasing weight to each harmonic, placing greater emphasis on lower-order terms by assigning them higher weights. The parameter orb_weight serves as a scaling factor to control the overall importance of the harmonic terms relative to the velocity terms, and N represents the batch size.

Empirically, an orbital weight of 200 was chosen, as it provided a balanced emphasis on shape deformations while preserving model performance on dynamic attributes. The decay rate is set as 0.5. Subsequent analysis and experiments are conducted with these values.

4.1.3 EGNN Model Parameters

Table 1 gives the hyperparameters used to train the EGNN and the MPNN (baseline model described in Section 4.1.1.

The in_emb_depth refers to the number of layers in the MLP that computes the initial node embeddings. The model_depth parameter indicates the depth of the internal models within each message-passing layer—specifically, the coord_model, vel_model, and orb_model for the EGNN, and the message_mlp and update_mlp for the baseline. n_layers is the total number of message-passing layers. The SiLU activation function was selected because it was demonstrated to be the most effective choice in the experiments conducted by Satorras et al. with the original EGNN implementation [4].

Models	Hyperparameters						Trainable Params
	in_emb_depth	model_depth	n_layers	hidden_ft	dropout	act_fn	
Baseline	2	1	2	64	0.5	ReLU	49,430
EGNN	2	2	2	32	0.2	SiLU	35,654

Table 1: Hyperparameters used for EGNN and Baseline)

4.2 Results

Table 2 presents the MSE values for both the single-step and 10-step trajectory experiments. The results indicate that the EGNN outperforms the baseline model in terms of MSE for position and orbital predictions in both experiments, suggesting that it is more capable of capturing complex bubble dynamics and shape. While the EGNN's accuracy in predicting positions is only slightly better than the baseline, it demonstrates a notable improvement in predicting the spherical harmonics. The performance for velocity prediction on the 10 steps trajectory is comparable to that of the baseline.

Models	Single Step		10 Steps Trajectory		
	velocity MSE	orbitals MSE	position MSE	velocity MSE	orbitals MSE
Baseline	0.01289	7.177e-07	2.284e-05	0.002689	6.456e-07
EGNN	0.006717	3.064e-09	1.893e-05	0.002390	3.742e-09

Table 2: MSE values for the 2 experiments - EGNN and Baseline

The results of the bubble trajectory prediction are presented through visual comparisons of the predicted trajectories from both the EGNN and the baseline MPNN models. Figure 5 illustrates the predicted bubble trajectories using the EGNN model. The first positions of the bubbles are generally predicted with reasonable accuracy. However, as the trajectory evolves and interactions become more intricate, the model's predictions show a noticeable decrease in accuracy. This suggests that while the EGNN is capable of recognizing the presence of interactions between bubbles, its performance could be further improved with more training data. The model demonstrates potential for refinement, and with additional fine-tuning, it may provide more precise trajectory predictions.

In contrast, Figure 6 presents the predicted trajectories from the baseline model, the MPNN. The predicted bubble positions are mostly concentrated along a single line with nearly constant xy-coordinates. This result indicates that the baseline struggles to accurately model the complex interactions between the bubbles. The lack of variation in the predicted positions highlights the model's inability to fully capture the dynamic, spatial relationships within the bubble system.

Figure 7 shows a comparison between the bubble shapes predicted by the EGNN and the target shapes. While the predicted shapes initially appear to follow the target shapes, suggesting that the EGNN may capture some early-stage dynamics and interactions, this alignment is short-lived. As the trajectory progresses, the model's predictions increasingly revert to a simplified spherical shape, failing to replicate the more complex deformations observed in the target bubbles. This pattern indicates that the EGNN struggles to maintain shape accuracy over longer time steps, likely due to difficulties in modeling sustained interactions and intricate deformations over time.

This observation is further supported by the graphs of the spherical harmonic coefficients shown in Figures 8 and 9, where only the first few coefficients are presented for clarity. Figure 8, which represents the monopole term, reveals that the model exhibits greater variability in this coefficient, which does not stabilize within the initial steps as other coefficients do. This behavior may be attributed to the monopole term's larger magnitude, which gives it more weight in the loss function, especially due to the exponential decay applied to higher-order terms. In contrast, Figure 9 shows that, after approximately five steps, the other coefficients cease to vary, leading to a stabilization of the bubble shape. This suggests that the model tends toward a static shape over time, limiting its ability to capture shape deformations.

5 Future Work

Several directions exist for enhancing the current model's performance and applicability in simulating bubble dynamics.

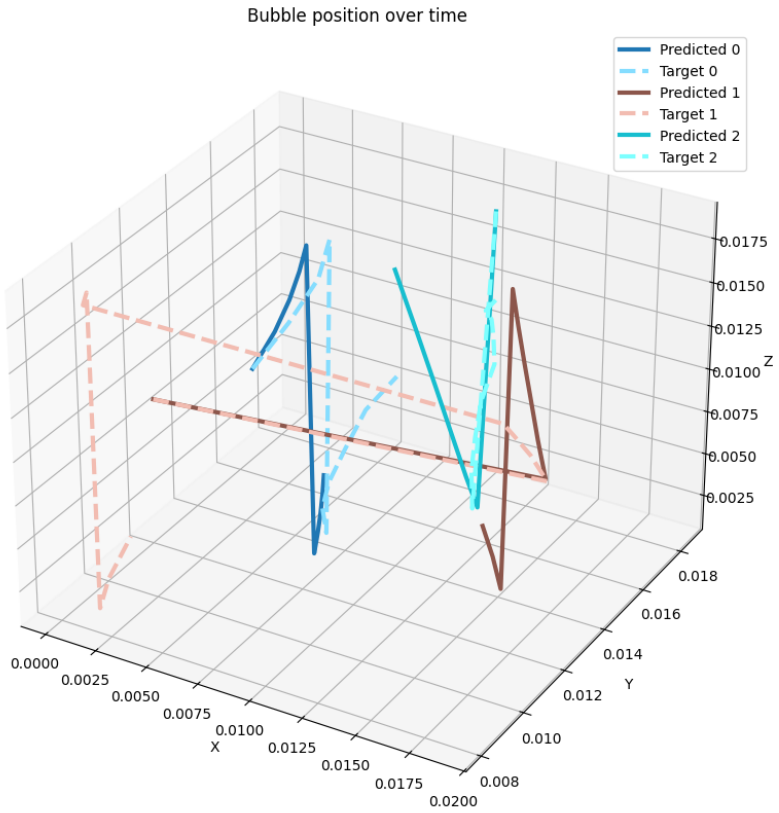

Figure 5: Predicted Trajectories for three bubbles - EGNN

Firstly, the current training approach relies on single-step predictions, which can lead to the accumulation of errors when the model is applied autoregressively over longer time horizons. To mitigate this issue, integrating a multi-step loss function is a promising direction. By training the model to minimize errors over multiple future timesteps, the model can learn temporal dependencies more effectively, reducing error accumulation and improving stability in long-term forecasting of bubble dynamics. This approach would enhance the model's utility for simulations where accurate long-term predictions are critical.

Secondly, to better capture the variability and uncertainty in bubble shape deformations and interactions, we propose integrating a Variational Autoencoder (VAE) into the EGNN framework. By embedding a probabilistic component within the model, the VAE can more effectively model the latent space of bubble dynamics. Specifically, the EGNN would serve as both the encoder and decoder in an autoencoder architecture, with a probabilistic latent variable introduced between them. In the encoding stage, the EGNN processes the input graph data—bubble positions, velocities, and spherical harmonic coefficients—to produce parameters of a latent distribution. Then, we sample latent variables from this distribution. In the decoding stage, the sampled latent variables are input into another EGNN to reconstruct the future states of the bubbles, including positions, velocities, and shape deformations. This setup enables the model to generate multiple plausible future states by sampling from the latent distribution, capturing the uncertainty and variability in bubble dynamics.

Thirdly, an improvement would be the incorporation of orbital velocities. In the current version of the model, we update the spherical harmonic coefficients directly without considering their inertia or dynamic behavior. Introducing orbital velocities—time derivatives of the spherical harmonic coefficients—and updating the coefficients based on these velocities, similar to how positions are updated based on linear velocities, could better capture the dynamics of bubble deformations. This modification would allow the model to account for the temporal evolution of bubble shapes more accurately.

Additionally, developing an effective normalization strategy for the orbital weights could enhance the model's performance. In the current implementation, we manually assign a weight to all the spherical harmonic coefficients to increase their importance in the loss function, making their magnitudes

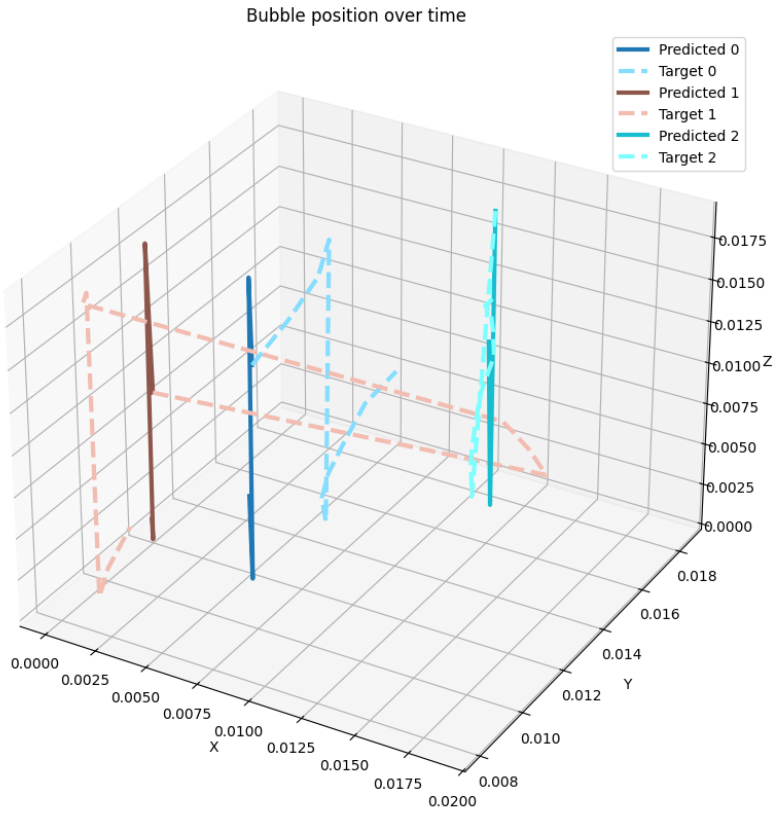

Figure 6: Predicted Trajectories for three bubbles - Baseline (MPNN)

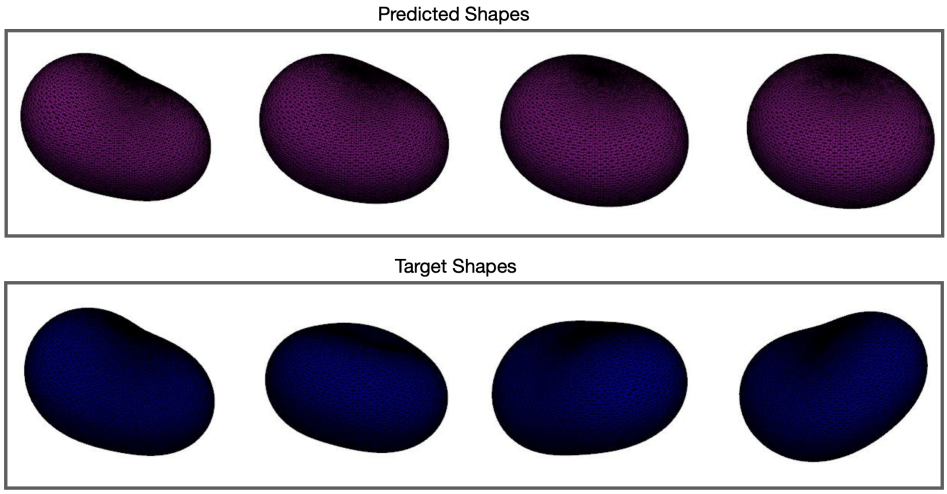

Figure 7: Comparison between Predicted (EGNN) and Target Bubble Shapes

comparable to that of the velocity terms in the Mean Squared Error (MSE). However, finding a normalization technique that automatically balances the contributions of the orbital coefficients and velocity could eliminate the need for manual weighting, leading to more accurate predictions, and simplifying the training process.

Furthermore, the limited size of the dataset, consisting of only six simulations, poses a challenge for the model's ability to generalize and capture the full range of bubble dynamics. Acquiring more data would significantly benefit the training process by providing a broader spectrum of scenarios and

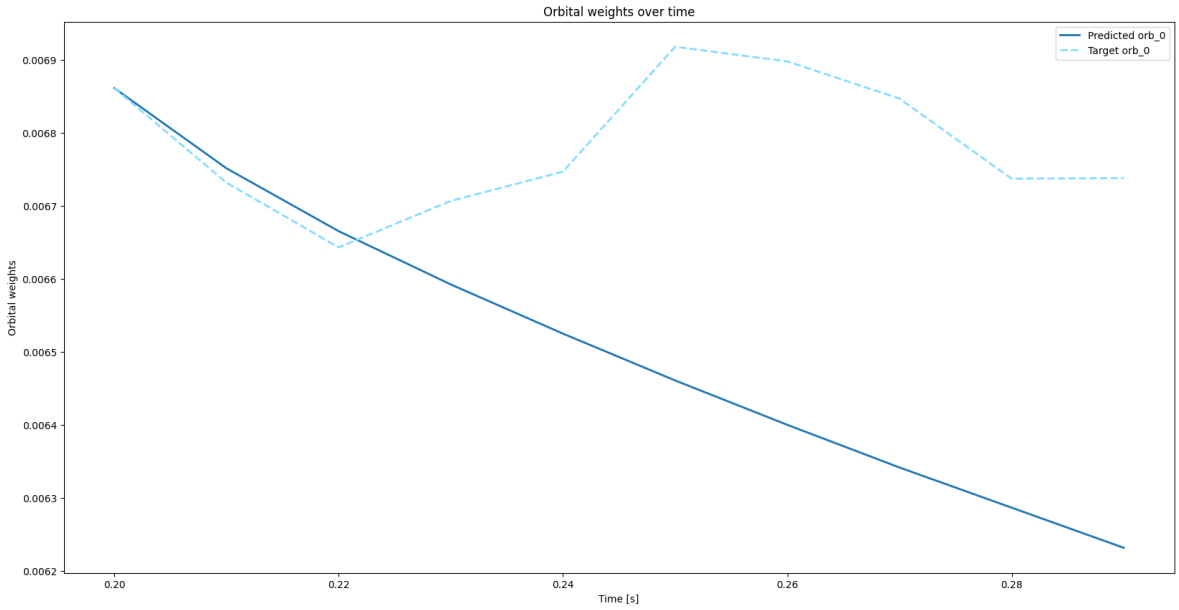

Figure 8: Predicted Spherical Harmonic Coefficient 0 - EGNN

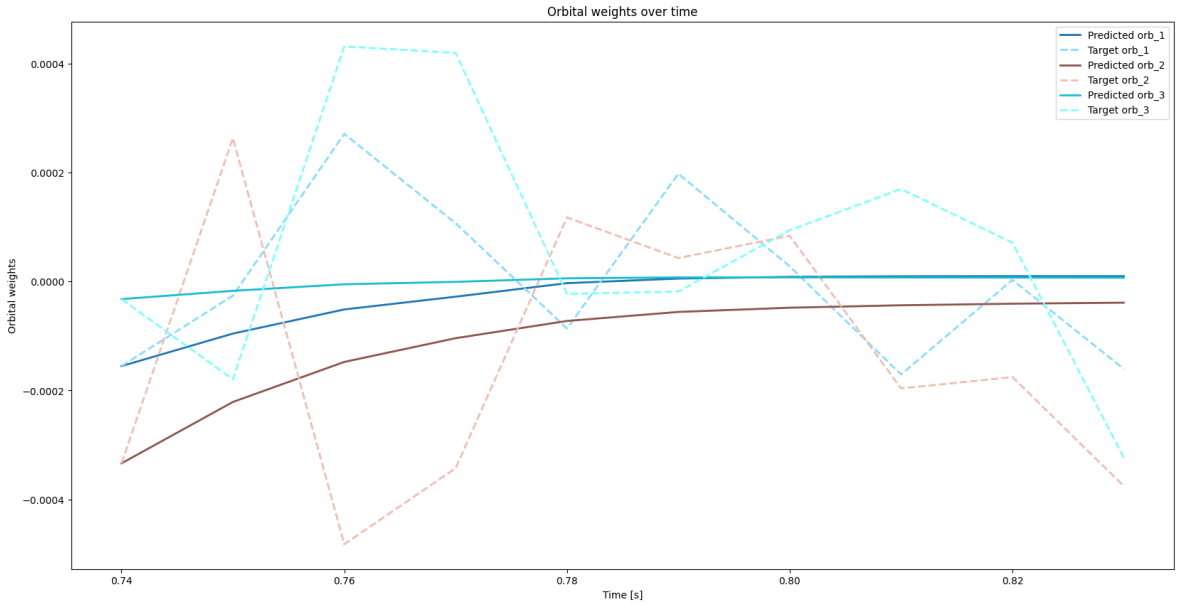

Figure 9: Prediction of coefficients 1, 2, and 3 - EGNN

interactions for the model to learn from. If obtaining additional data is not feasible, employing cross-validation techniques could help maximize the utility of the existing dataset. By systematically rotating the roles of training, validation, and test sets, cross-validation ensures that the model is evaluated and trained across all available data, potentially improving its robustness and reducing overfitting.

6 Conclusion

In this work, we developed an E(n) Equivariant Graph Neural Network (EGNN) to simulate the dynamics of interacting bubbles within a bubble column. By representing the bubble system as a graph and incorporating spherical harmonic coefficients to capture bubble deformations, our model respects key physical symmetries such as translation, rotation, and reflection. We introduced a novel

update equation for the spherical harmonic coefficients, enabling the EGNN to account for both the translational motion and shape deformations of bubbles due to interactions and external forces.

Our experiments demonstrate that the EGNN outperforms a baseline Message Passing Neural Network (MPNN) in predicting bubble trajectories and shape deformations. The EGNN showed lower mean squared errors (MSE) in both single-step predictions and autoregressive 10-step trajectory predictions.

However, with a qualitative analysis of the predictions, we observed that the EGNN performs poorly when used to predict trajectories in an autoregressive manner, especially for bubble shapes (spherical harmonics). The model always tends to revert to predicting more spherical shapes, indicating a limitation in capturing complex deformations sustained over long periods.

Future work will focus on addressing the identified limitations by integrating multi-step loss functions, incorporating probabilistic modeling techniques such as variational autoencoders, and improving the handling of spherical harmonic coefficients. Additionally, expanding the dataset or employing cross-validation techniques could enhance the model's generalizability and robustness.

References

[1] H Hikita, S Asai, K Tanigawa, K Segawa, and M Kitao. Gas hold-up in bubble columns. *The Chemical Engineering Journal*, 20(1):59–67, 1980.

[2] C. Leonard, J.-H. Ferrasse, O. Boutin, S. Lefevre, and A. Viand. Bubble column reactors for high pressures and high temperatures operation. *Chemical Engineering Research and Design*, 100:391–421, 2015.

[3] Miroslaw Narbutt, Jan Skoglund, Andrew Allen, Michael Chinen, Dan Barry, and Andrew Hines. Ambiqual: Towards a quality metric for headphone rendered compressed ambisonic spatial audio. *Applied Sciences (Switzerland)*, 10(9), May 2020. Publisher Copyright: © 2020 by the authors.

[4] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. *ArXiv*, abs/2102.09844, 2021.