Proceedings of the ASME 2026 45th International Conference on Ocean, Offshore and Arctic Engineering
OMAE2026
June 7-12, 2026, Tokyo, Japan

OMAE2026-181906

A DATA-DRIVEN HYBRID APPROACH COMBINING MPS METHOD WITH GRAPH ATTENTION NETWORKS FOR FREE SURFACE FLOWS

Yujia Liu
Computational Marine Hydrodynamic Lab (CMHL), School of Ocean And Civil Engineering, Shanghai Jiao Tong University, Shanghai, China

Maokun Ye
Computational Marine Hydrodynamic Lab (CMHL), School of Ocean And Civil Engineering, Shanghai Jiao Tong University, Shanghai, China

Fengze Xie
Computational Marine Hydrodynamic Lab (CMHL), School of Ocean And Civil Engineering, Shanghai Jiao Tong University, Shanghai, China

Decheng Wan*
Computational Marine Hydrodynamic Lab (CMHL), School of Ocean And Civil Engineering, Shanghai Jiao Tong University, Shanghai, China

ABSTRACT

The Moving Particle Semi-implicit (MPS) method is a widely used Lagrangian Particle method for simulating free-surface flows. A major computational bottleneck in conventional MPS simulations is solving the pressure Poisson equation (PPE) during the pressure-projection step. To address this issue, we propose a data-driven hybrid approach that integrates the MPS method with the Graph Attention Network (GAT), denoted as MPS-GAT. In this approach, GAT serves as a surrogate model to replace the traditional PPE solver for pressure estimation in MPS. The effectiveness of the proposed method is evaluated through two benchmark cases: dam breaking and tank sloshing. The accuracy is verified by comparing the pressure predictions from MPS-GAT with numerical results obtained from solving the PPE. Furthermore, the proposed hybrid method significantly improves computational efficiency. The speed-up ratio increases further with larger particle counts, demonstrating clear advantages for large-scale free-surface flow simulations.

Keywords: MPS; PPE; GAT; GNN; dam breaking; tank sloshing

1. INTRODUCTION

The Moving Particle Semi-Implicit (MPS) method has been developed as a meshless particle-based approach, serving as an extension of the Smoothed Particle Hydrodynamics (SPH) method. It was first introduced to address the challenges of incompressible flow simulation[1]. It was applied to investigate breaking wave phenomena and analyze various breaking wave patterns [2]. To date, the MPS method has been extensively used for tank sloshing studies. Simulations of 2D rectangular tanks revealed that severe oscillations occur at resonance frequencies [3]. Researchers subsequently examined 3D LNG tank sloshing, analyzing fill level and excitation frequency effects to determine critical operating ranges [4]. The improved MPS method was used to assess dynamic characteristics such as base shear, overturning moment, and impact pressures [5]. Researchers then developed GPU-accelerated MPS solvers, enabling 3D simulations under various conditions with significantly reduced computational costs [6]. The MPS method employs a fractional step approach, achieving better pressure accuracy and stability than explicit particle methods like SPH [7]. The enhanced accuracy requires greater computational resources, particularly for solving the pressure Poisson equation (PPE) with increasing particle counts [8]. Additionally, these iterative approaches face GPU parallelization limitations, restricting large-scale MPS simulations without CPU clusters. To address this, researchers developed an explicit MPS variant that computes particle pressures independently, enabling efficient GPU parallelization and lower computational costs [9]. However, the explicit method demonstrates higher time step sensitivity and lower stability compared to implicit approaches.

Recent advances in fluid dynamics have seen widespread application of machine learning (ML) techniques, particularly for accelerating numerical computations by replacing computationally intensive traditional procedures with trained ML models. In Lagrangian methods, neural network-based deep learning has been applied to PPE solution acceleration - notably, one data-driven model improved MPS efficiency while

preserving accuracy through significant speedup [10]. These methods significantly lower computational costs while maintaining realistic fluid behavior, enabling more practical data-driven fluid modeling.

However, a key challenge arises because most neural networks are designed for structured data, while Lagrangian particle methods generate inherently unstructured data due to the irregular motion of particles. This limitation has driven the adoption of Graph Neural Networks (GNNs), originally developed for analyzing unstructured relational data like social networks [11]. Specifically for Lagrangian particle methods, GNN applications in fluid dynamics have shown promising results. Researchers successfully integrated a GNN model with the ISPH method, significantly accelerating the PPE solution process [12]. The study showed that a GNN model trained exclusively on wave-only case data could effectively handle wave-floater interaction scenarios [13].

Inspired by Zhang's work [12], we adapt this framework to the MPS method, presenting a novel data-driven approach that combines MPS with a Graph Attention Network (GAT) to accelerate computations, reduce resource requirements, and deliver reliable engineering solutions. This work represents the first successful integration of GNN models within an MPS framework for free-surface flow simulations.

The paper is organized as follows: Chapter 2 introduces the mathematical framework of the MPS method and GAT model, their hybrid implementation strategy, and the database construction process. Chapter 3 validates the model's accuracy through comparing numerical simulation results with the prediction model results. Chapter 4 gives the conclusions of this paper and proposes future research directions.

2. MATERIALS AND METHODS

2.1 The Moving Particle Semi-Implicit (MPS) method

The MPS method is formulated within a Lagrangian framework, in which the continuous flow field is discretized into a collection of particles that carry physical quantities such as mass, momentum, energy, and other relevant properties. Interactions among particles are described through the integration of a kernel function, which enables the evaluation of inter-particle forces and the tracking of particle motion, thereby capturing the overall flow dynamics.

In the MPS method, the attribute value of a target particle i within the flow field is determined as the weighted average of the corresponding attribute values of its neighboring particles, as expressed in Eq. (1).

$$\langle f_i \rangle = \frac{\sum_{j \neq i} f_j W(|r_j - r_i|)}{\sum_{j \neq i} W(|r_j - r_i|)} \tag{1}$$

In the formula, r_i and r_j denote the coordinates of the target particle i and the neighboring particle j, respectively, while f_i and f_j represent the corresponding attribute values at particles i and j. The function $W(r_j - r_i)$ is the weight function.

The MPS method is developed for viscous incompressible fluids, and its governing equations consist of the continuity equation and the Navier-Stokes equations, as given in Eq. (2) and Eq. (3).

$$\frac{1}{\rho}\frac{D\rho}{Dt} = -\nabla \cdot V = 0 \tag{2}$$

$$\frac{DV}{Dt} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 V + f \tag{3}$$

In the formula, $\frac{D}{Dt}$ denotes the material derivative, ρ is the fluid density, p represents the pressure, V is the velocity vector, υ is the kinematic viscosity of the fluid, and f denotes the body force, which is generally gravity.

In the MPS method, interactions among particles are realized through a kernel function, which serves as the weighting function in the discretization of the governing equations. Consequently, the kernel function constitutes the most fundamental model in the MPS method and forms the basis for other particle interaction models.

In the traditional MPS method, a widely used kernel function was proposed by Koshizuka et al. (1998). Its specific form is given as follows:

$$W(r) = \begin{cases} \frac{r_e}{r} - 1 & 0 < r < r_e \\ 0 & r_e \leq r \end{cases} \tag{4}$$

In the formula, r denotes the distance between two particles, and r_e is the radius of the particle interaction domain, which is a prescribed constant. As the inter-particle distance r approaches zero, the kernel function $W(r)$ tends to infinity. When r becomes sufficiently small, the derivative of $W(r)$ increases sharply, such that even a slight variation in r can lead to a drastic change in the kernel value. Consequently, under conditions involving strong flow variations or relatively large time steps, this behavior often gives rise to non-physical pressure oscillations in the flow field, thereby degrading the numerical stability of the simulation. To mitigate these non-physical high-frequency oscillations, a more suitable kernel function is adopted in the MLparticle-SJTU solver, as shown in Eq. (5).

$$W(r) = \begin{cases} \frac{r_e}{0.85r + 0.15r_e} - 1 & 0 \leq r \leq r_e \\ 0 & r_e \leq r \end{cases} \tag{5}$$

The kernel function given in Eq. (5) is free of singularities, thereby avoiding high-frequency oscillations caused by excessively large derivatives of the kernel function.

In the MPS method, the gradient model is employed to discretize first-order derivatives and is also used to approximate

the pressure gradient term in the Navier-Stokes equations. Essentially, the gradient model represents a weighted average of the relative position vectors among particles within the interaction domain. Accordingly, the pressure gradient at particle iii can be expressed as follows:

$$< \nabla P >_i = \frac{D}{n^0} \sum_{j \neq i} \frac{P_i + P_j}{|r_j - r_i|^2} (r_j - r_i) \cdot W(|r_j - r_i|) \quad (6)$$

The Laplacian model is used to discretize second-order derivatives. Both the evaluation of the viscous term $\nabla^2 V$ in the Navier-Stokes equations and the solution of the pressure Poisson equation (PPE) require the spatial discretization of the Laplacian operator ∇^2. The Laplacian model proposed by Koshizuka et al. (1996) is given as follows:

$$< \nabla^2 \phi >_i = \frac{2D}{n^0 \lambda} \sum_{j \neq i} (\phi_j - \phi_i) \cdot W(|r_j - r_i|) \quad (7)$$

where D denotes the spatial dimension and n^0 is the initial particle number density. The parameter λ is introduced to ensure that the numerical results are consistent with the analytical solution of the diffusion equation, and its specific expression is given as follows:

$$\lambda = \frac{\sum_{j \neq i} W(|r_j - r_i|) \cdot |r_j - r_i|^2}{\sum_{j \neq i} W(|r_j - r_i|)} \quad (8)$$

The divergence model has a form similar to that of the gradient model. For the velocity vector V, the discrete expression of the velocity divergence is given as follows:

$$\nabla \cdot V = \frac{D}{n^0} \sum_{j \neq i} \frac{V_j - V_i}{|r_j - r_i|^2} (r_j - r_i) \cdot W(|r_j - r_i|) \quad (9)$$

Under conditions of severe tank sloshing, traditional free-surface identification methods may encounter difficulties. When neighboring particles in the flow field become relatively sparse, particles may be incorrectly classified as free-surface particles due to the reduced local particle number density. To address this issue, the MLparticle-SJTU solver determines free-surface particles based on the asymmetry of the neighboring particle distribution. The degree of asymmetry, denoted by F, is defined as follows:

$$\langle F \rangle_i = \frac{D}{n^0} \sum_{j \neq i} \frac{1}{|r_i - r_j|} (r_i - r_j) W(r_{ij}) \quad (10)$$

When the condition $< |F| >_i > \alpha$ is satisfied for particle i, it is identified as a free-surface particle. In this expression, $|F|$ denotes the magnitude of the asymmetry measure F, and α is a threshold parameter defined as $0.9|F_0|$, where $|F_0|$ represents the value of $|F|$ for a free-surface particle at the initial time.

Based on the mixed source term method (Tanaka and Masunaga, 2010; Khayyer et al., 2011), the PPE was rephrased as follows [15]:

$$< \nabla^2 P^{n+1} >_i = (1 - \gamma) \frac{\rho}{\Delta t} \nabla \cdot V_i^* - \gamma \frac{\rho}{\Delta t^2} \frac{< n^* >_i - n^0}{n^0} \quad (11)$$

In the formula, the parameter γ represents the proportion of particle number density in the source term. A smaller value of γ corresponds to a lower particle number density proportion and a higher velocity divergence contribution, resulting in a smoother pressure field. However, excessively small values of γ can lead to non-conservation of fluid volume. This occurs because, during the computation, errors in the velocity divergence accumulate over successive iterations, causing deviations in the total fluid volume. Therefore, a reasonable value of γ must be chosen to balance stability and volume conservation. In this study, γ is set to 0.01.

2.2 Graph Attention Networks (GAT) model

Graph Neural Networks (GNNs) have emerged as a powerful framework for processing data that are inherently structured as graphs. The seminal work introduced the concept of iterative message passing, in which each node's representation is updated by aggregating information from its neighboring nodes [11]. A major breakthrough in this field was the introduction of Graph Convolutional Networks (GCNs), which extended the concept of convolution from regular Euclidean domains to graph-structured data [16]. Inspired by the success of Convolutional Neural Networks (CNNs) in processing grid-structured data, GCNs extend the convolutional operation to graph-structured data by performing localized aggregations of node features. However, the standard GCN approach uses a fixed or uniformly normalized weighting scheme, implicitly assuming that all neighboring nodes contribute equally to the updated representation. This uniform treatment can be limiting in heterogeneous graphs, where the influence of individual neighbors may vary significantly, potentially leading to suboptimal performance [17].

Graph Attention Networks (GATs) was built on the strengths of both GCNs and attention mechanisms [18]. GATs incorporate a self-attention mechanism directly into the graph convolution process to dynamically assign weights to the features of neighboring nodes during aggregation. For each node in the graph, a raw attention score is computed for each neighbor by applying a shared learnable function to the concatenated features of the node and its neighbor. These raw scores are then normalized using a softmax function to produce attention coefficients that reflect the relative importance of each neighboring node. The final node representation is obtained as the weighted sum of its neighbors' transformed features. This dynamic weighting allows GATs to capture the heterogeneous influence of neighbors more accurately than the uniform

aggregation employed by GCNs [18,19], thus significant advantages over traditional GCNs are offered by GATs.

In particle-based simulations, such as those performed using the MPS method, fluid behavior is governed by local interactions among particles. In this context, interactions can be effectively modeled using Graph Attention Networks (GATs), leading to improved predictions of key physical quantities, such as pressure and velocity. The localized attention mechanism in GATs focuses only on immediate neighbors, thereby avoiding the computational overhead associated with global feature extraction. As a result, computational efficiency is enhanced and memory usage is reduced compared with the traditional MPS method, which is especially important for simulations involving thousands or even millions of particles. Consequently, the dynamic and localized weighting provided by GATs enables a more nuanced and robust representation of particle interactions. These observations indicate that integrating GATs into the MPS framework holds significant promise for improving computational efficiency while maintaining accuracy comparable to that of the traditional method.

FIGURE 1: DIAGRAM OF THE GRAPH ATTENTION NETWORK ARCHITECTURE

The GAT model used in this study (Figure 1) comprises three main components: the encoder, the message-passing module, and the decoder [20]. Within the GAT model, graph convolutions serve as a message-passing mechanism, enabling information to be directly transmitted between nodes through the edges. The encoder is responsible for transforming input features into node embeddings, which then serve as inputs for the subsequent message-passing stage. Before encoding, the input features are normalized using batch normalization to ensure stable and efficient training. These normalized embeddings are then used for graph convolutions during the message-passing step. Specifically, for a given particle i in Figure 1, the input feature vector f_i^0 is first normalized to g_i^0 and then encoded into a node embedding h_i^0 through a learnable encoder function $encode()$, which is implemented as a multilayer perceptron (MLP) with n layers. Mathematically, the encoding process can be expressed as:

$$encode(g_i^l) = \sigma\left(w_l g_i^{l-1} + b_l\right), l = 1,2,\dots,n \tag{12}$$

$$h_i^0 = encode(g_i^0) \tag{13}$$

In the message-passing block, node embeddings are iteratively updated through a recursive scheme. At this stage, the influence of a neighboring node j on node i is first quantified, denoted as $m_{j\to i}$. Since $m_{j\to i}$ inherently depends on the relative positions of the nodes, an appropriate weight is introduced to account for spatial relationships. Following the approach proposed by Li and Farimani [21], the neighboring influence in the l-th layer convolution is defined as follows:

$$m_{j\to i}^l = h_j^{l-1} W(r_{ij}, r_0) \tag{14}$$

Here, h_j^{l-1} denotes the node embedding of the neighboring node j in the $(l-1)$-th convolution layer. The weight function is defined as follows, with a lower bound introduced to prevent the weight from approaching infinity when the inter-particle distance becomes very small, thereby avoiding spurious pressure oscillations.

$$W(r_{ij}, r_0) = \begin{cases} \frac{r_0}{0.85 r_{ij} + 0.15 r_0} - 1, & 0 < r_{ij} \le r_0, \\ 0, & r_{ij} > r_0, \end{cases} \tag{15}$$

In the formula, r_0 denotes the radius of the influence domain, which is set to 4.01 dx based on a series of numerical experiments conducted by Koshizuka. Here, dx represents the initial particle spacing, which is set to 0.025 in this study. The aggregated influence from all neighboring nodes is then given by:

$$M_i^l = \frac{\sum_{j=1}^N m_{j\to i}^l}{\sum_{j=1}^N w(r_{ij}, r_0)} \tag{16}$$

In the l-th layer of message passing, the node embedding for node i is updated as follows:

$$h_i^l = \sigma\left(w_l' M_i^l + b_l' h_i^{l-1}\right), l = 1,2,\dots,L \tag{17}$$

Here, w_l' and b_l' denote the learnable weight vector and bias term in the l-th layer, respectively.

The decoder maps the node embeddings from the final message-passing layer to the desired output. Specifically, after L layers of recursive message passing, the output q_i is obtained by decoding the final node embedding h_i^L using a learnable decoder function $embedding()$. The decoder is implemented as a multilayer perceptron, analogous to the encoder described in Eq. (12), but with its own set of learnable weights and biases.

$$q_i = embedding(h_i^t) \tag{18}$$

Here, q_i represents the mapping from the input features to the pressure of each particle and will be denoted as f_{GNN} hereafter. In this study, all encoder and decoder MLPs are implemented with three layers, while the number of recursive message-passing layers is set to two.

In our GAT model for pressure prediction, the Exponential Linear Unit (ELU) activation function, Huber loss, and Adam optimizer are jointly employed to address the inherent challenges of particle-based fluid simulations and enhance the model's robustness. The ELU activation function is chosen for its ability to introduce a smooth exponential response for negative inputs, which effectively mitigates the vanishing gradient problem and captures the complex nonlinear interactions among fluid particles. Moreover, its continuous differentiability facilitates the representation of multiscale pressure variations. The ELU activation function is defined as follows in Eq. (19):

$$f(x) = \begin{cases} x, (x > 0) \\ \alpha(e^x - 1), (x < 0) \end{cases} \tag{19}$$

In the ELU activation function, the parameter α is a positive constant that determines the saturation value for negative inputs. Larger values of α allow the negative outputs to extend further, while smaller values compress them toward zero. The exponential term e^x is applied only when the input x is negative, ensuring that the function remains smooth and provides non-zero gradients even in the negative region. Collectively, these characteristics result in an activation function that behaves linearly for positive inputs and smoothly decays toward $-\alpha$ for negative inputs, thereby improving gradient flow compared with ReLU.

The Huber loss function is employed to handle noise and heterogeneity in the simulation data. Its quadratic behavior for small errors and linear penalty for large deviations effectively mitigates the influence of outliers arising from particle collisions, free-surface fluctuations, or numerical instabilities, while preserving fine pressure details in regions with smooth gradients.

$$L_\delta(r) = \begin{cases} \frac{1}{2}r^2, |r| \le \delta \\ \delta(r - \frac{1}{2}\delta), |r| > \delta \end{cases} \tag{20}$$

The Huber loss introduces a parameter δ that governs the transition between the quadratic and linear regimes of the loss function. Specifically, when the absolute prediction error $|r|$ is smaller than δ, the loss behaves quadratically, providing smooth gradients similar to the mean squared error (MSE). When $|r|$ exceeds δ, the loss becomes linear, resembling the mean absolute error (MAE) and thereby reducing the influence of outliers. This design allows the Huber loss to combine the sensitivity of MSE for small errors with the robustness of MAE for large deviations [22].

The Adam optimizer was selected for its adaptive learning rate mechanism, which facilitates navigation of the high-dimensional, nonconvex loss landscape, addresses the sparse and irregular gradients commonly encountered in particle-based simulations, and mitigates early training instabilities. Collectively, these design choices enable stable training trajectories and effective generalization to unseen flow regimes, thereby demonstrating the model's suitability for particle-based fluid–structure interaction analyses.

2.3 Incorporating GAT with MPS

2.3.1 Input parameters

To ensure satisfactory prediction accuracy, it was determined that the inputs to the GAT model must include variables that effectively represent hydrodynamic characteristics. To prevent the GAT model from functioning solely as a time-series predictor, certain physical quantities already computed at the current time step were incorporated into the input parameters. In this study, a blended source term method (Eq. (11)) was employed in the PPE of the MPS model. Since the particle number density n^* and velocity divergence $\nabla \cdot v^*$ at the current time step appear in the source term, these quantities were included as input parameters to represent the current particle distribution and instantaneous compressibility effects. Moreover, due to the continuous nature of pressure variation, the pressure from the previous time step p_t was also included as an input parameter. Prior studies, such as the investigation combining the MPS model with a CNN for free-surface flow modeling [12], have indicated that including these variables significantly improves prediction performance.

Based on the above considerations, the field data of n^*, $\nabla \cdot v^*$, and p_t are adopted as input parameters to train the GAT model for predicting the pressure $p_{t+\Delta t}$. The mapping function f_{GAT} from input to output by the GAT model can be expressed as:

$$p_{t+\Delta t} = f_{GAT}\,(\,p_t, n^*, \nabla \cdot v^*\,) \tag{21}$$

It is noteworthy that in previous studies applying GNNs to ISPH [23], pressure was decomposed into static and dynamic components. The GNN was used solely to predict the dynamic pressure, while the static pressure was calculated directly from particle depth. This approach allowed the model to focus on the more variable dynamic pressure, thereby reducing predictive error. However, within the MPS framework, determining the static pressure requires identifying free-surface particles at the vertical positions corresponding to the current particles, which entails substantial additional computational cost. Consequently, the direct pressure prediction approach proposed by Yao [10] was adopted in this study, and the results demonstrate that

satisfactory predictive accuracy can be achieved.

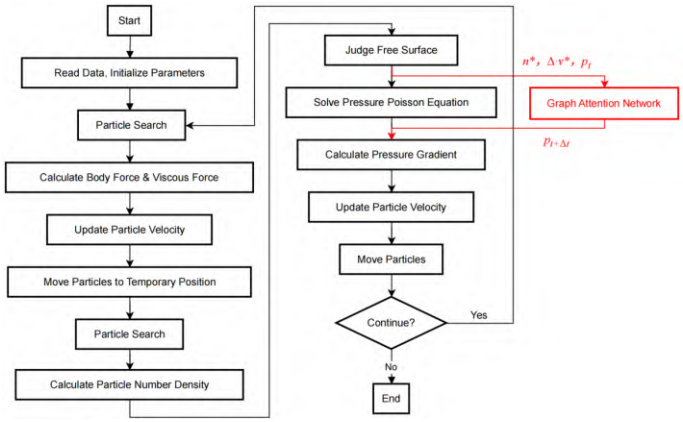


FIGURE 2: PARTICLE VELOCITY AND POSITION UPDATE IN MPS-GAT MODEL

2.3.2 Training data

The overall accuracy of the model is also influenced by the quality of the training dataset. In this study, a high-fidelity training dataset was generated using a Python implementation of the standard MPS solver, employing the BiConjugate Gradient Stabilized (BiCGSTAB) method. This dataset was subsequently used to train the GAT model, ensuring that the learned representations were informed by accurate pressure distributions. The training conditions for the MPS model are summarized in Table 1, which details the parameter settings, flow regimes, and boundary conditions employed during the training phase, providing a comprehensive overview of the simulation scenarios and numerical configurations used to generate the dataset. Two benchmark cases are employed to train the GAT model in this study: a liquid tank sloshing model and a dam-breaking model.

Table 1: TRAINING WORK CONDITIONS OF MPS MODEL

Parameters	Value
Initial particle distance r_0	$0.025m$
The particle support domain radius r_e	$2.1r_0$
Gravitational acceleration g	$9.8m/s^2$
Kinematic viscosity coefficient ν	1.0e-6
Hybrid source term coefficient γ	0.01
Time step Δt	0.0005s
Total time t	2s
Fluid column size (l×h) of dam break	$0.6\times0.8m$
Tank size (L×H) of dam break	$2.0\times1.0m$
Fluid column size (l×h) of tank sloshing	$2.0\times0.4m$
Tank size (L×H) of tank sloshing	$2.0\times1.0m$

In the final dataset, 75% of the total simulations are selected as the training set, comprising 60 simulations in total. These simulations include 48 tank sloshing cases and 12 dam breaking cases. For each of these 60 simulations, the first 8000 time steps are used for training, while the remaining 2000 time steps constitute an interpolation test set. The remaining 20 simulations consist of 16 tank sloshing cases and 4 dam breaking cases, which are reserved as an extrapolation test set to evaluate the model performance under previously unseen initial and boundary conditions.

In the present work, the Huber loss function is employed, combining the advantages of both L2 and L1 losses by applying a quadratic penalty to small errors and a linear penalty to large errors. Two primary types of boundary conditions are considered: free-surface and solid boundaries. During training, discrepancies between predicted and reference pressures at the free surface are ignored. Instead, the pressure at free-surface particles is explicitly constrained to zero, thereby enforcing the free-surface boundary condition $p = 0$.

Regarding solid boundary conditions, neighboring nodes of fluid particles may include solid particles, whose pressures influence the pressure distribution of the surrounding fluid. To account for this interaction, both fluid and solid particles are incorporated into the training process. Consequently, the trained GAT model is applied to predict pressure values for both fluid and solid particles. For solid particles, additional refinement is performed by deriving their pressure from the pressures of neighboring fluid particles.

3. RESULTS AND DISCUSSION

For comparative evaluation, the conventional MPS method, which computes pressure by solving the PPE, is also implemented as a reference. The performance of the proposed GAT-based method is compared against this traditional MPS approach, thereby demonstrating improvements in pressure prediction accuracy and overall robustness under varying flow conditions.

Pressure probes are placed at multiple locations within the flow field. At each probe location, the local pressure is calculated as the average pressure of particles within a prescribed neighborhood, as shown in Figure 3. These values are recorded over time to generate pressure time-history curves. The pressure histories predicted by the GAT model are then compared with those obtained from the PPE-based MPS solver to quantitatively assess the accuracy of the GAT-based pressure field prediction.

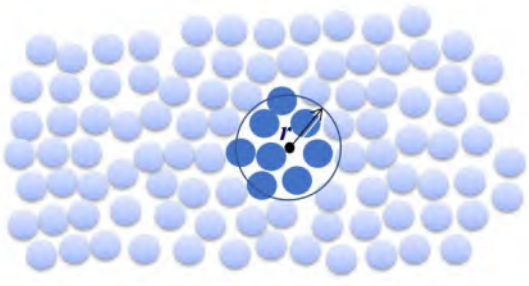


FIGURE 3: SCHEMATIC DIAGRAM OF PRESSURE PROBES

3.1 DAM BREAKING FLOW

As a classical validation case for Lagrangian fluid simulations, the dam breaking is first considered. A rectangular

column of water with the width l and the height h is confined in a water tank with a horizontal length of L and the height H, as shown in Figure 4. In the dam break model, the transient evolution of an initially confined fluid, following the abrupt removal of a barrier, is simulated, thereby assessing the method's capability to capture complex free-surface dynamics and rapid pressure changes.

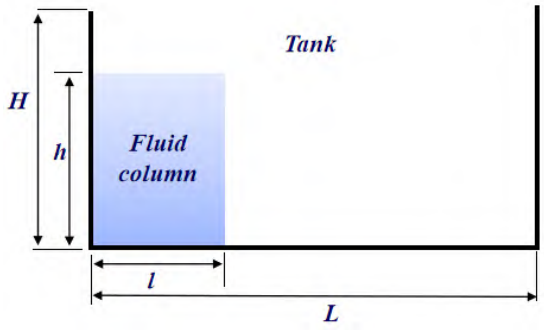
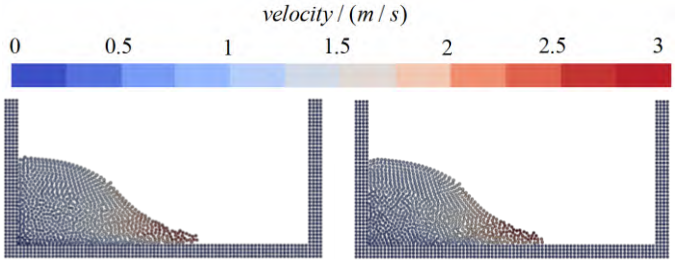


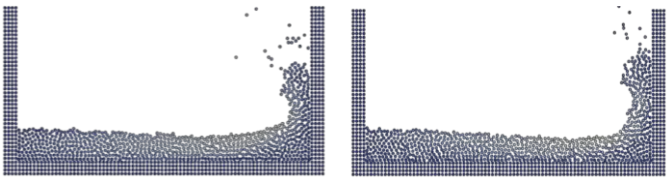
FIGURE 4: SKETCH OF THE DAM BREAKING MODEL

Training and testing datasets were generated via a series of MPS-PPE dam breaking simulations (Figure 4). An initial particle spacing of $DX = 0.025m$ and a time step of $\Delta t = 0.0005s$ were prescribed, and 200 frames at 0.01s intervals were recorded per simulation to capture the flow evolution from barrier removal to impact against the downstream wall. The GAT model $f_{GAT}(p_t, n^*, \nabla \cdot v^*)$ was trained on this database and subsequently validated on an independent dam breaking scenario $(h = 0.8L, l = 0.3L, L = 2.0m)$.

Pressure contours and particle distributions at representative instants reveal that the GAT model faithfully reproduces MPS-PPE results, accurately capturing free-surface deformation and jet formation.Pressure probes were deployed at multiple locations within the flow field to record the temporal evolution of the pressure field. Particle velocity field snapshots at different time steps are compared (Figure 5), and corresponding pressure field snapshots are presented (Figure 6). In both figures, the left panels are computed using the MPS-PPE solver, while the right panels are predicted by the MPS-GAT model.

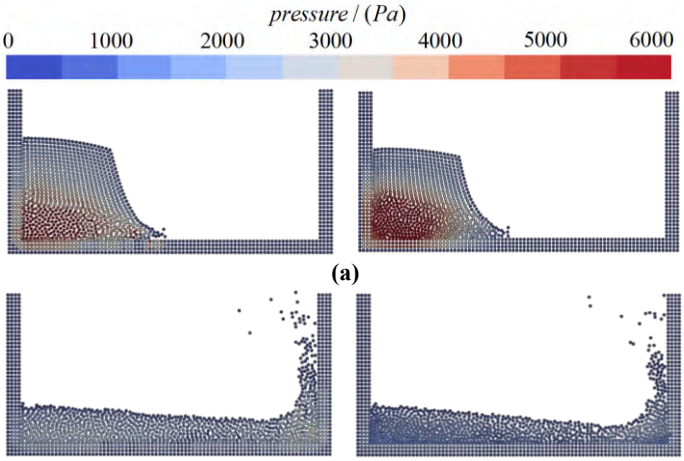


(a)



(b)

FIGURE 5: PARTICLE DISTRIBUTION AND VELOCITY CONTOUR COMPARISON: MPS-GAT VS. MPS-PPE IN DAM BREAKING SIMULATION

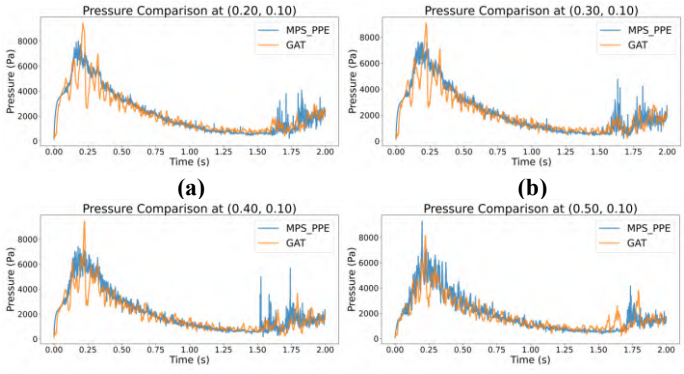


(a)

(b)

FIGURE 6: PARTICLE DISTRIBUTION AND PRESSURE CONTOUR COMPARISON: MPS-GAT VS. MPS-PPE IN DAM BREAKING SIMULATION

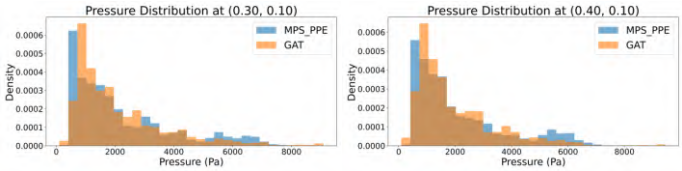
Additionally, pressure time histories at various spatial locations are compared to evaluate the temporal accuracy of the predicted pressure fields (Figure 7). Pressure frequency distributions are analyzed to further highlight the accuracy of the predicted pressures, by verifying that the dominant frequency components of the signals are well preserved (Figure 8). Furthermore, the pressure time history of a representative single particle is illustrated to provide a detailed local comparison between the MPS-PPE solver and MPS-GAT, emphasizing their consistency in capturing transient pressure dynamics (Figure 9).



(a) (b)

(c) (d)

FIGURE 7: PRESSURE HISTORY COMPARISON: MPS-GAT VS. MPS-PPE AT SELECTED MONITORING POINTS IN DAM BREAKING SIMULATION



(a) **(b)**

FIGURE 8: PRESSURE FREQUENCY DISTRIBUTION COMPARISON: MPS-GAT VS. MPS-PPE IN DAM BREAKING SIMULATION

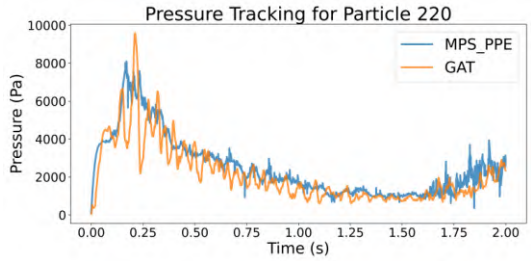


FIGURE 9: SINGLE-PARTICLE PRESSURE HISTORY: COMPARISON BETWEEN MPS-GAT AND MPS-PPE IN DAM BREAKING SIMULATION (PARTICLE ID: 220)

3.2 TANK SLOSHING FLOW

The liquid tank sloshing model examines fluid-structure interaction under external excitations within a confined container, which introduces significant dynamic pressure variations that influence both the fluid flow and the structural response.

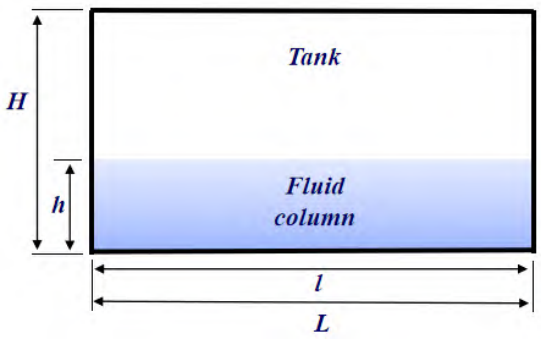
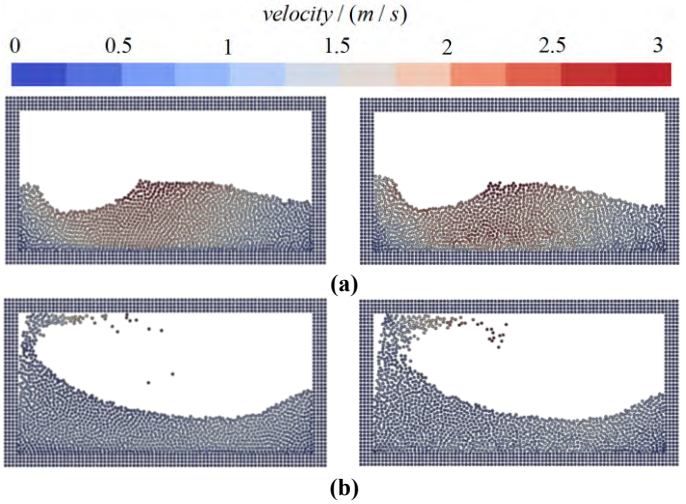


FIGURE 10: SKETCH OF THE TANK SLOSHING MODEL

Both the conventional MPS-PPE solver and the MPS-GAT model were evaluated on a series of tank sloshing cases (Figure 10). For each tank sloshing simulation, 200 snapshots were recorded at intervals of 0.01s to capture the dynamic evolution of the free surface and the pressure field throughout the oscillatory motion. The GAT model $f_{GAT}(p_t, n^*, \nabla \cdot v^*)$ was trained on this dataset and subsequently validated using an independent dam breaking scenario $(h = L, l = 0.2L, L = 2.0m)$.

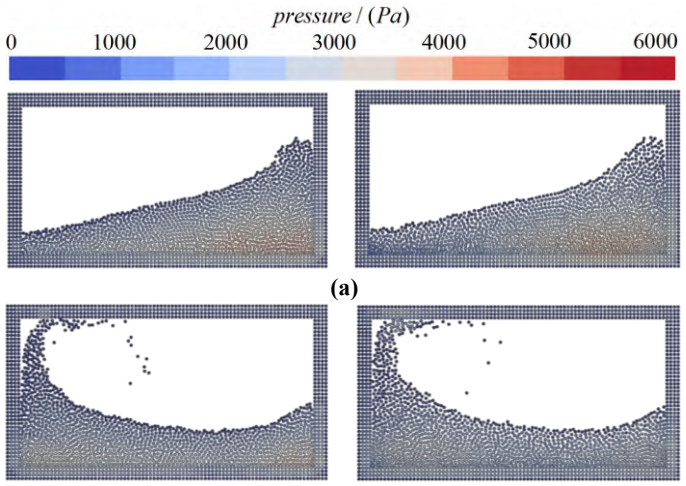
As illustrated in Figure 11 and Figure 12, the spatial distributions of velocity and pressure are presented at several representative instants over a tank sloshing cycle. The velocity and pressure fields predicted by the MPS-GAT model exhibit a high level of consistency with those obtained from the MPS-PPE solver. In particular, key flow phenomena such as free surface deformation and the generation of sloshing induced jets are clearly reproduced. This agreement indicates that the proposed MPS-GAT model is capable of accurately capturing the complex and highly nonlinear flow characteristics associated with sloshing motion.



(a)

(b)

FIGURE 11: PARTICLE DISTRIBUTION AND VELOCITY CONTOUR COMPARISON: MPS-GAT VS. MPS-PPE IN TANK SLOSHING SIMULATION



(a)

(b)

FIGURE 12: PARTICLE DISTRIBUTION AND PRESSURE CONTOUR COMPARISON: MPS-GAT VS. MPS-PPE IN TANK SLOSHING SIMULATION

To provide a quantitative assessment of the model performance, pressure probes were arranged at multiple predefined locations inside the tank. The corresponding pressure time histories are compared in Figure 13. The GAT based

predictions closely match the MPS-PPE results in both amplitude and phase over the entire simulation duration. These comparisons demonstrate that the proposed MPS-GAT model can reliably reproduce the unsteady pressure response generated by tank sloshing, confirming its effectiveness for pressure field prediction in strongly transient free surface flows.

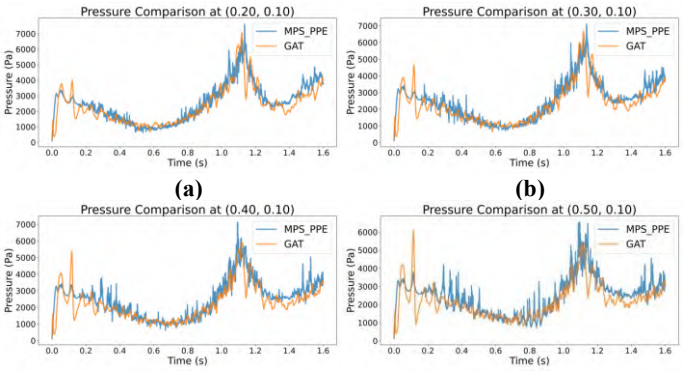


FIGURE 13: PRESSURE HISTORY COMPARISON: MPS-GAT VS. MPS-PPE AT SELECTED MONITORING POINTS IN TANK SLOSHING SIMULATION

To further assess the accuracy of the predicted pressure fields, frequency domain analyses of the pressure signals at the same probe locations are conducted. As shown in Figure 14, the dominant spectral components obtained from the MPS-GAT model are in close agreement with those derived from the MPS-PPE solver, indicating that the main oscillatory characteristics of the pressure response are accurately reproduced.

In addition, the pressure time history of a representative individual particle is examined to enable a detailed local comparison between the MPS-GAT model and the MPS-PPE solver. The results, presented in Figure 15, demonstrate that the MPS-GAT model is able to follow the transient pressure evolution at the particle level, thereby further confirming its capability to accurately capture short term pressure fluctuations and dynamic response features.

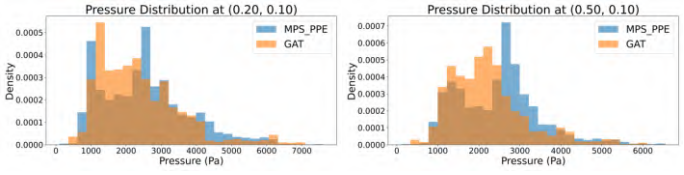


FIGURE 14: PRESSURE FREQUENCY DISTRIBUTION COMPARISON: MPS-GAT VS. MPS-PPE IN TANK SLOSHING SIMULATION

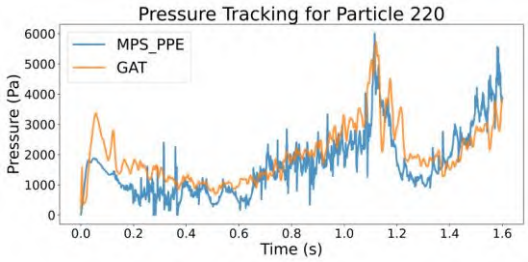


FIGURE 15: SINGLE-PARTICLE PRESSURE HISTORY: COMPARISON BETWEEN MPS-GAT AND MPS-PPE IN TANK SLOSHING SIMULATION (PARTICLE ID: 220)

3.3 ACCURACY VALIDATION

The accuracy of the predicted pressure fields is assessed using three standard metrics: Root Mean Square Error (*RMSE*), Mean Absolute Percentage Error (*MAPE*), and the coefficient of determination (R^2). These metrics provide complementary insights into the prediction performance.

RMSE measures the average magnitude of the error between the predicted and reference values. It is sensitive to large deviations. A lower *RMSE* indicates better predictive accuracy, with values closer to zero representing smaller deviations from the true values. It is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(p_i^{pred} - p_i^{true})^2} \quad (1)$$

MAPE expresses the average absolute error as a percentage of the true values and is useful for understanding the relative error magnitude. Lower *MAPE* values indicate better performance, with values below 20% generally considered acceptable for most engineering applications. It is given by:

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\frac{|P_i^{pred} - P_i^{true}|}{P_i^{true}} \quad (2)$$

R^2 evaluates how well the predicted values match the true values in terms of overall variance. An R^2 value close to 1 indicates a strong correlation between predicted and true values, reflecting high model performance. It is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(p_i^{pred} - p_i^{true})^2}{\sum_{i=1}^{N}(p_i^{true} - \bar{p}^{true})^2} \quad (3)$$

Together, these three metrics provide a comprehensive assessment of the model's accuracy and generalization capability in predicting pressure fields.In the following, we present the *RMSE*, *MAPE*, and R^2 values of the pressure fields obtained from both the PPE method and the GAT model under the dam breaking and tank sloshing scenarios. These results allow for a direct comparison of the predictive performance between the traditional numerical solver and the data-driven approach across different flow conditions.

The results are summarized in Table 2 below, providing a clear comparison of *RMSE*, *MAPE*, and R^2 for pressure field

predictions between the PPE method and the GAT model for both the dam breaking and tank sloshing cases.

TABLE 2: PERFORMANCE METRICS COMPARISON FOR PRESSURE FIELD PREDICTIONS: MPS-PPE VS. MPS-GAT IN DAM BREAKING AND TANK SLOSHING CASES

Case	*RMSE*	*MAPE*	R^2
Dam breaking	550.3333	12.82%	0.8778
Tank sloshing	567.0912	15.50%	0.7803

3.4 RESOLUTION SCALING

The GAT model was initially trained using low resolution datasets, with each simulation consisting of approximately 2000 particles. Its performance was then evaluated on higher resolution tank sloshing cases, in which the particle spacing dx was reduced from 0.025m to 0.0075m, resulting in roughly 20000 particles per simulation. To further examine the generalization capability of the model, additional tests were conducted with an even finer discretization, where dx was decreased to 0.002 m and the total particle count exceeded 100000.

Although the model was trained solely on the coarse resolution data, it was applied directly to these much denser particle configurations without any retraining or parameter adjustment. Comparisons with the pressure fields computed using the MPS-PPE solver reveal that the GAT based predictions exhibit significantly improved smoothness and substantially reduced high frequency oscillations. In particular, the pressure contours produced by the GAT model appear more spatially uniform, while spurious transient fluctuations are effectively suppressed, as illustrated in Figure 16. These results demonstrate the robustness of the proposed approach and its enhanced predictive stability under increasingly refined spatial discretizations.

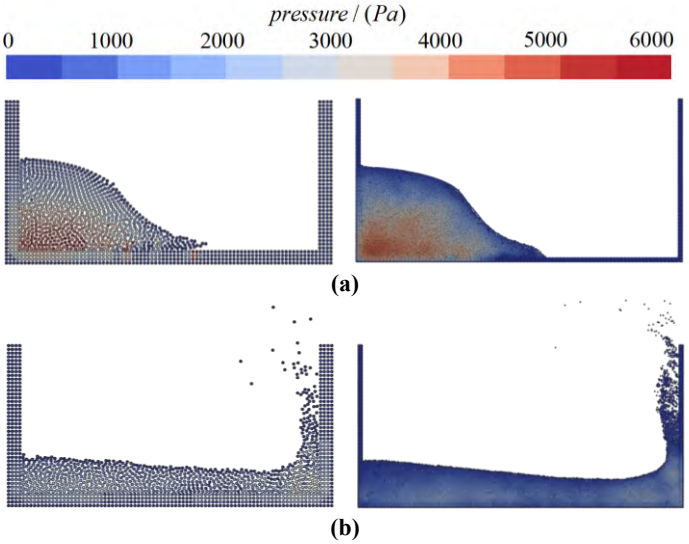


FIGURE 16: PARTICLE DENSIFICATION EFFECTS ON PRESSURE FIELD: BASE MPS-PPE (LEFT) VS. MPS-GAT (RIGHT) IN DAM BREAKING

Additionally, to verify the model's accuracy under high-resolution conditions, a dam breaking case was supplemented in which particle resolution was significantly increased and simulated using the MPS-PPE method. The resulting high-resolution pressure field was then compared with the prediction generated by the MPS-GAT model (Figure 17).

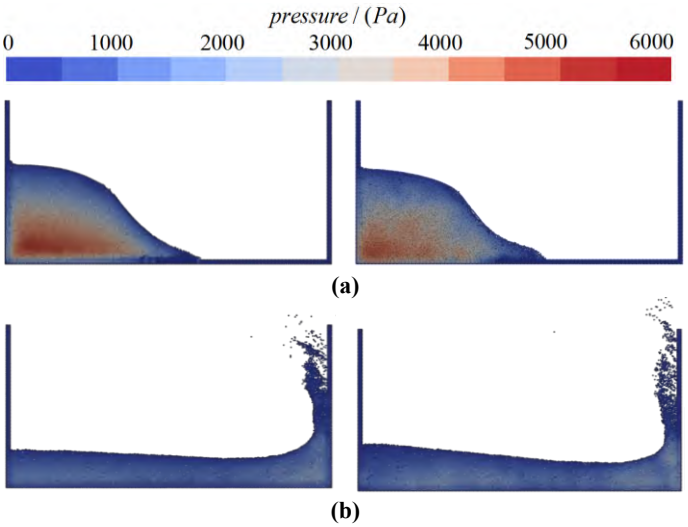


FIGURE 17: PRESSURE FIELD COMPARISON: MPS-PPE (LEFT) VS. MPS-GAT (RIGHT) AFTER PARTICLE DENSIFICATION IN DAM BREAKING

Comparisons of pressure time histories at different spatial locations are presented to further evaluate the predictive capability of the GAT model under high-resolution conditions (Figure 18), and the corresponding pressure frequency distributions are analyzed to examine the consistency of dynamic pressure characteristics between the two methods (Figure 19). Due to the significant alterations in particle indexing and the dynamic behavior of particles resulting from the refined spatial resolution, the pressure evolution of a single particle is no longer tracked in this scenario.

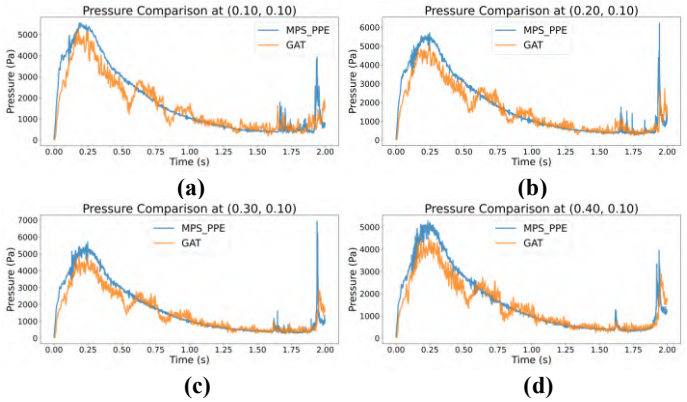
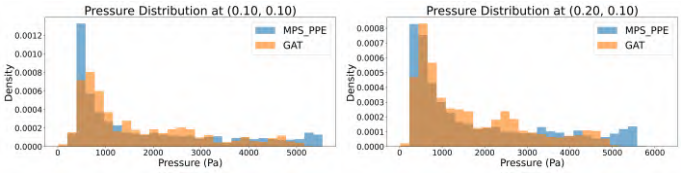


FIGURE 18: COMPARISON OF PRESSURE TIME HISTORIES AT DIFFERENT FLOW POSITIONS: MPS-PPE VS. MPS-GAT IN POST-DENSIFICATION DAM BREAKING SIMULATION



(a) (b)

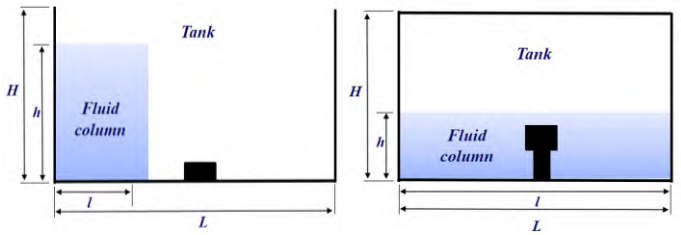
FIGURE 19: COMPARATIVE PRESSURE HISTOGRAM ANALYSIS: MPS-PPE AND MPS-GAT PERFORMANCE ACROSS FLOW FIELD POSITIONS AFTER PARTICLE DENSIFICATION

3.5 OBSTACLE INCLUSION

To further evaluate the generalization capability of the proposed approach, the GAT model trained exclusively on unobstructed dam breaking and tank sloshing cases was directly applied to flow configurations involving internal obstacles, without any additional retraining. In the dam breaking case, a rectangular obstacle was introduced, with a width of $0.2L$ and a height of $0.3L$. The upstream face of the obstacle was positioned at a distance of $0.4L$ downstream from the initial barrier and centered along the lateral direction of the domain, as shown in Figure 20a. In the tank sloshing case, a T-shaped support structure was mounted on the tank bottom to represent floor stiffeners. The horizontal flange had a length of $0.6L$, while the vertical web extended to a height of $0.2L$, with the centroid of the structure located at the mid length of the tank, as illustrated in Figure 20b.

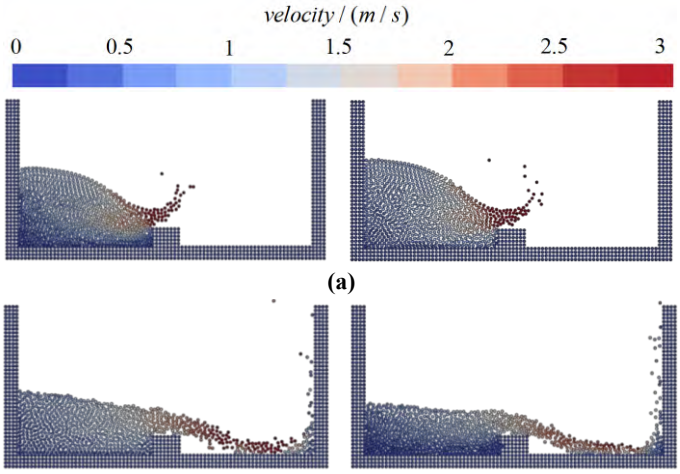
Despite the complete absence of obstructed flow scenarios in the training dataset, the GAT model was applied without modification and exhibited robust predictive performance. The pressure field in the vicinity of the obstacles was reproduced with high fidelity, and key flow features such as free surface deformation, localized pressure amplification, and pressure fluctuations induced by flow acceleration and vortex shedding were accurately captured. These predictions show close agreement with the results obtained from the MPS-PPE solver.

A detailed comparison of velocity field distributions is presented in Figure 21, demonstrating that the flow patterns predicted by the MPS-GAT model are consistent with those computed using the MPS-PPE method. Corresponding pressure field distributions are compared in Figure 22, further confirming the spatial accuracy of the GAT based predictions. In addition, pressure time histories recorded at several critical probe locations and the corresponding pressure frequency distributions are shown in Figure 23 and Figure 24, where strong agreement is observed in both amplitude and phase. To provide a more localized temporal assessment, the pressure time history of a representative individual particle is examined in Figure 25. The close correspondence between the MPS-GAT predicted and MPS-PPE computed results confirms that the proposed model is capable of accurately capturing transient pressure dynamics even in complex obstructed flow environments.



(a) (b)

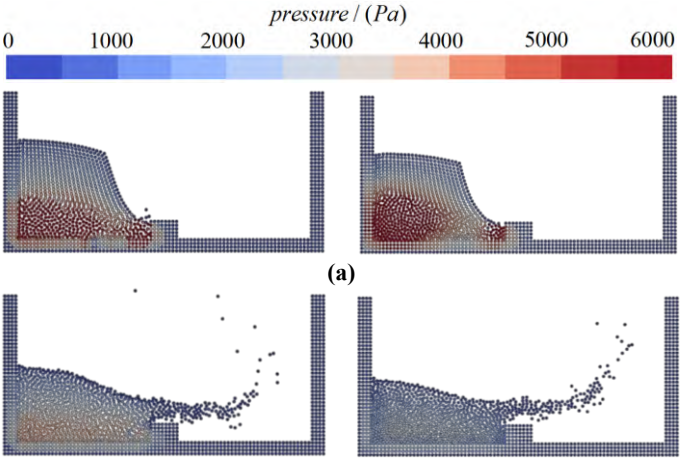
FIGURE 20: EXPERIMENTAL SETUP: DAM BREAKING AND TANK SLOSHING MODEL WITH OBSTACLES



(a)

(b)

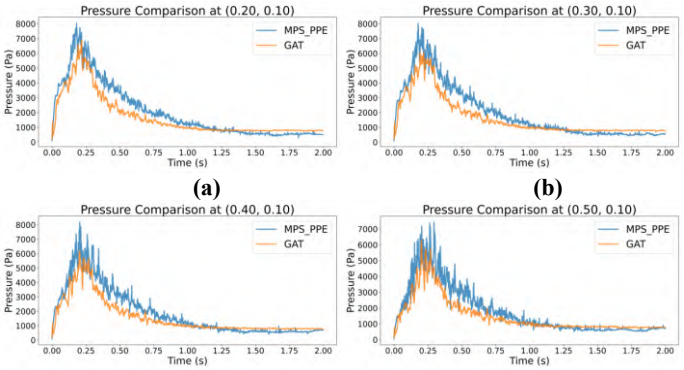
FIGURE 21: PARTICLE DISTRIBUTION AND VELOCITY FIELD COMPARISON: MPS-GAT VS. MPS-PPE IN OBSTRUCTED DAM BREAKING SIMULATION



(a)

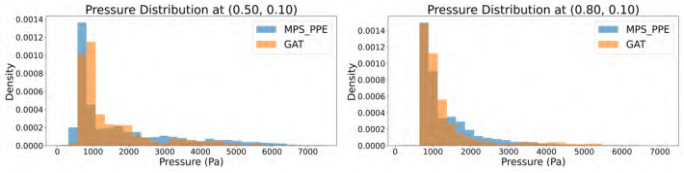
(b)

FIGURE 22: PARTICLE DISTRIBUTION AND PRESSURE FIELD COMPARISON: MPS-GAT VS. MPS-PPE IN OBSTRUCTED DAM BREAKING SIMULATION



(a) (b) (c) (d)

FIGURE 23: PRESSURE TIME HISTORY COMPARISON AT MULTIPLE POSITIONS: MPS-PPE VS. MPS-GAT IN OBSTRUCTED DAM BREAKING SIMULATION



(a) (b)

FIGURE 24: PRESSURE FREQUENCY DISTRIBUTION COMPARISON AT MULTIPLE MONITORING POSITIONS: MPS-PPE VS. MPS-GAT IN OBSTRUCTED DAM BREAKING SIMULATION

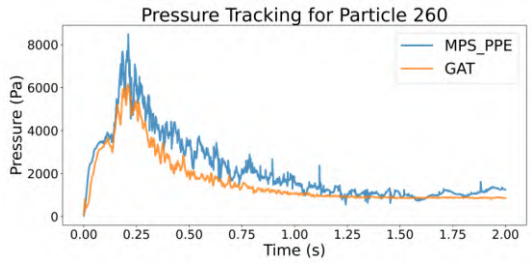
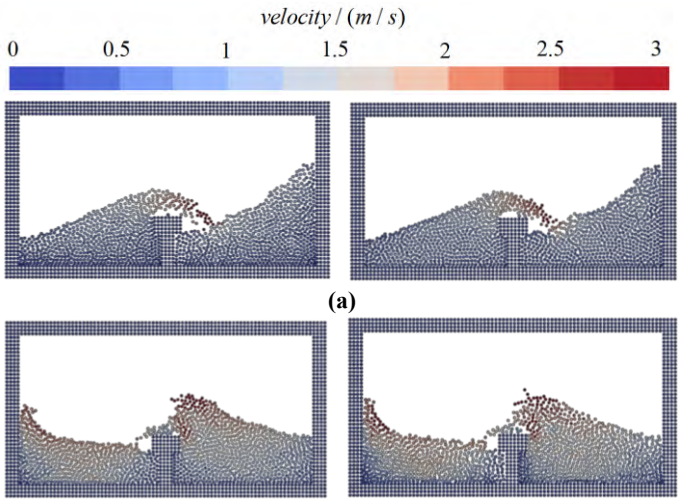


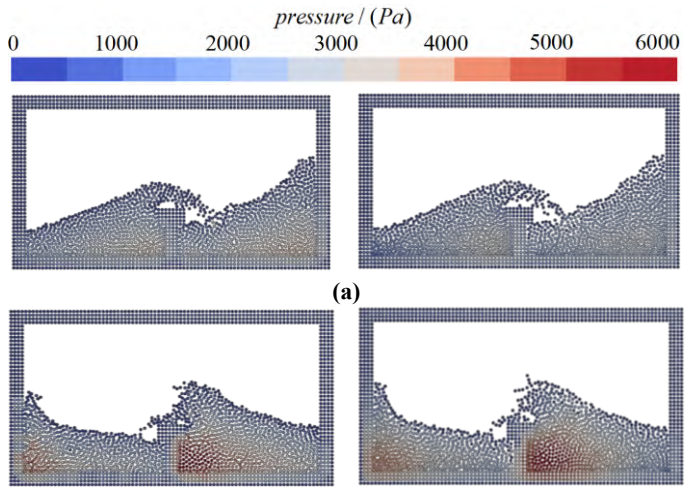
FIGURE 25: SINGLE-PARTICLE PRESSURE DYNAMICS COMPARISON: MPS-PPE VS. MPS-GAT IN OBSTRUCTED DAM BREAKING FLOW (PARTICLE ID: 260)



(a)

(b)

FIGURE 26: PARTICLE DISTRIBUTION AND VELOCITY FIELD COMPARISON: MPS-GAT VS. MPS-PPE IN OBSTRUCTED TANK SLOSHING SIMULATION

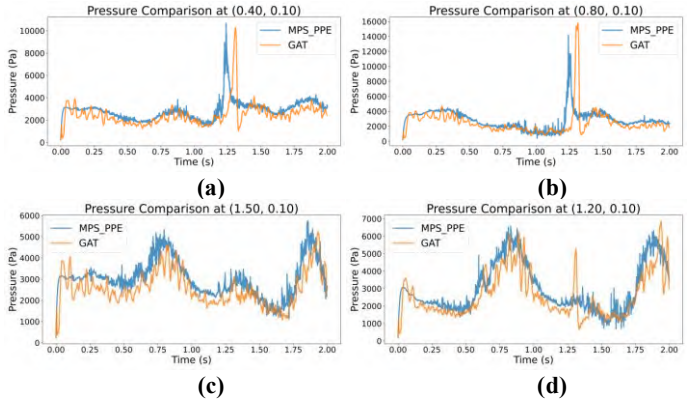


(a)

(b)

FIGURE 27: PARTICLE DISTRIBUTION AND PRESSURE FIELD COMPARISON: MPS-GAT VS. MPS-PPE IN OBSTRUCTED TANK SLOSHING SIMULATION

Similarly, sloshing in a tank with internal obstacles was also investigated. The velocity and pressure fields in the presence of obstacles were analyzed and compared to assess the predictive performance of the GAT model under more complex flow conditions. Comparisons of velocity fields are presented (Fig. 26), followed by pressure-field cloud maps that highlight the influence of internal structures on pressure distribution (Fig. 27). Pressure time histories at multiple probe locations are depicted to evaluate the temporal accuracy of the predicted pressure response (Fig. 28), while the pressure time history of a representative single particle is illustrated to provide a localized assessment of prediction fidelity (Fig. 29).



(a) (b) (c) (d)

FIGURE 28: PRESSURE TIME HISTORY COMPARISON AT MULTIPLE POSITIONS: MPS-PPE VS. MPS-GAT IN OBSTRUCTED TANK SLOSHING SIMULATION

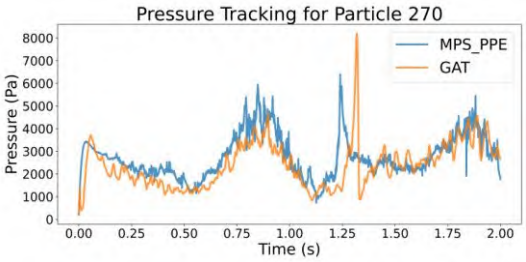


FIGURE 29: SINGLE-PARTICLE PRESSURE DYNAMICS COMPARISON: MPS-PPE VS. MPS-GAT IN OBSTRUCTED TANK SLOSHING FLOW (PARTICLE ID: 270)

3.6 COMPUTATIONAL EFFICIENCY

The profile of devices used for this test is as follows: NVIDIA GeForce RTX 4060 Ti, 16.0GB memory. The algorithms are predominantly coded in Python, with the regression model being built using the PyTorch deep learning library and trained on a GPU. The computations are efficiently distributed across both the CPU and GPU for enhanced processing capabilities.The GPU implementation of GAT model uses CUDA. The particle numbers and comparisons of time cost for these tests are shown in Table 3.

TABLE 3: PARTICLE NUMBERS AND COMPUTATION TIME COMPARISON: MPS-PPE VS. MPS-GAT

Case	Description	Particle number	MPS-PPE time cost(s/it)	MPS-GAT time cost(s/it)	Time Reduction Factor
1	Dam breaking	1440	2.15	0.71	3.028
2	Tank sloshing	2304	3.07	1.01	3.039
3	Dam breaking with obstacles	1480	2.17	0.72	3.014
4	Tank sloshing with obstacles	2304	3.12	1.02	3.059
5	Spherical tank sloshing	6075	21.52	3.40	6.329
6	Dam breaking with particle densification_1	10640	54.40	5.13	10.604
7	Dam breaking with particle densification_2	22432	184.92	11.74	15.751
8	Dam breaking with particle densification_3	128032	1310.94	32.29	40.599

The accuracy of the predicted pressure fields is evaluated using *RMSE*, *MAPE*, and R^2, which together provide a comprehensive assessment of prediction performance from both absolute and relative error perspectives, as well as model fit. The results are summarized in Table 4, which presents a detailed comparison of *RMSE*, *MAPE*, and R^2 for the pressure field predictions obtained by the MPS-PPE method and the MPS-GAT model. The evaluation covers the dam breaking and tank sloshing models with added obstacles, as well as three dam breaking cases with different levels of particle refinement.

TABLE 4: GENERALIZATION PERFORMANCE METRICS COMPARISON: MPS-PPE VS. MPS-GAT FOR PRESSURE FIELD PREDICTION (*RMSE, MAPE, R^2*)

Case	*RMSE*	*MAPE*	*R^2*
Dam breaking with obstacles	701.0961	29.89%	0.8514
Tank sloshing with obstacles	701.8735	18.24%	0.8090
Dam breaking with particle densification_1	550.3333	28.82%	0.8278
Dam breaking with particle densification_2	549.9440	23.35%	0.8591
Dam breaking with particle densification_3	506.6913	21.84%	0.8704

4. CONCLUSION

A novel data-driven hybrid framework is proposed in which the conventional MPS pressure Poisson equation solver is replaced by a graph attention network, hereafter referred to as MPS-GAT. Within this framework, the trained GAT model directly predicts pressure fields, thereby eliminating the need for the iterative and computationally expensive solution of the pressure Poisson equation. The input feature set includes the divergence of the intermediate velocity field, particle number density, and pressure values from the previous time step, enabling the network to infer pressure corrections in a physically informed manner. Free-surface and solid-boundary conditions are explicitly imposed to preserve incompressibility and to ensure an accurate representation of fluid–structure interactions at domain interfaces.

To achieve a balance between accuracy and robustness, a Huber-loss-based objective function is adopted, providing sensitivity to small prediction errors while mitigating the influence of outliers associated with turbulent fluctuations. Viscous dissipation effects are explicitly accounted for through a penalty term on velocity divergence in the loss formulation. In addition, Lagrangian particle transport characteristics are incorporated via input features that encode particle displacements and local neighborhood interactions, allowing the network to capture the intrinsic dynamics of particle-based flow evolution.

The performance of MPS-GAT was systematically evaluated using two canonical free-surface flow benchmarks, namely dam breaking and tank sloshing. For both cases, direct comparisons were conducted against the standard MPS-PPE solver with respect to flow-field accuracy, pressure time history at representative probe locations, and overall computational efficiency.

Based on the numerical investigations, the following conclusions can be drawn.

(1) Flow-field fidelity. The pressure distributions and free-surface profiles predicted by MPS-GAT exhibit excellent agreement with those obtained using the conventional MPS-PPE method. Discrepancies in both spatial structure and pressure magnitude are minimal, indicating that the GAT-based solver

reliably captures the dominant hydrodynamic features of highly transient and nonlinear free-surface flows.

(2) Pressure time history curves. The temporal evolution of pressure at selected probe locations is accurately reproduced by MPS-GAT throughout the simulations. Both phase alignment and amplitude levels closely match those of the MPS-PPE reference solutions, demonstrating that the proposed model effectively captures the timing and intensity of dynamic pressure fluctuations. This capability is particularly important for the reliable prediction of transient loading in sloshing and impact-dominated flow scenarios.

(3) Computational time cost. The computational advantage of MPS-GAT becomes increasingly pronounced as the particle count grows. For simulations involving approximately 2,000 particles, MPS-GAT achieves a speedup of about three times relative to MPS-PPE. This acceleration increases to roughly sixteen times for cases with 20,000 particles, and reaches up to forty times when the number of particles exceeds 100,000. These results indicate that MPS-GAT offers substantial efficiency gains for large-scale simulations, particularly in three-dimensional applications where particle counts increase dramatically.

Collectively, these findings demonstrate that MPS-GAT constitutes a highly accurate and computationally efficient alternative to traditional PPE-based pressure solvers in particle-based free-surface flow simulations.

In addition to accuracy and efficiency, the generalization capability of MPS-GAT was examined in detail. Generalization performance was assessed by applying the trained model to simulation scenarios that extend beyond the conditions represented in the original training and testing datasets. Two categories of generalization tests were considered, namely resolution scaling and obstacle inclusion. For resolution scaling, the model was trained using coarse particle discretizations characterized by relatively large particle spacing and support radii. The trained model was then applied, without retraining, to simulations with progressively refined discretizations achieved by reducing particle spacing and support radius. For obstacle inclusion, the training dataset consisted exclusively of unobstructed flow configurations. In subsequent evaluations, rectangular and T-shaped obstacles were introduced into dam breaking and tank sloshing scenarios, respectively, and the unchanged model was applied to assess its response to complex boundary-induced flow perturbations.

The results indicate that the MPS-GAT model developed in this study exhibits strong generalization capability under both increased spatial resolution and the presence of previously unseen geometric constraints. Nevertheless, several limitations remain. Future work will expand the training dataset to encompass a broader range of flow regimes and discretization levels. Boundary condition enforcement strategies will be further refined, and the loss function will be augmented with additional physics-based constraints to enhance model robustness. Moreover, the development of a single generalized GAT model trained on mixed datasets covering multiple flow types will be pursued to improve overall versatility.

The present study is limited to single-phase, two-dimensional free-surface flows. As a first step in future research, the transferability of the single-phase-trained MPS-GAT model will be evaluated on simplified multiphase configurations. Subsequently, dedicated retraining using gas–liquid multiphase datasets will be conducted to improve the model's ability to capture interfacial dynamics and density-driven effects. In parallel, the extension of the MPS-GAT framework to fully three-dimensional free-surface flows will be investigated, with particular emphasis on scalability, memory efficiency, and the accurate representation of complex three-dimensional flow structures.

REFERENCES

[1] KOSHIZUKA S, OKA Y. Moving-Particle Semi-Implicit Method for Fragmentation of Incompressible Fluid[J/OL]. Nuclear Science and Engineering, 1996[2025-03-11]. https://www.tandfonline.com/doi/abs/10.13182/NSE96-A24205. DOI:10.13182/NSE96-A24205.

[2] KOSHIZUKA S, NOBE A, OKA Y. Numerical analysis of breaking waves using the moving particle semi-implicit method[J/OL]. International Journal for Numerical Methods in Fluids, 1998, 26(7): 751-769. DOI:10.1002/(SICI)1097-0363(19980415)26:7%3C751::AID-FLD671%3E3.0.CO;2-C.

[3] ZHANG Y, WAN D cheng, ZHAO W wen. Apply MPS method to simulate liquid sloshing in LNG tank[C/OL]. 2012[2025-09-12]. https://dx.doi.org/.

[4] YANG Y, TANG Z, WAN D. Numerical simulations of 3D liquid sloshing flows by MPS method[C/OL]. 2014[2025-09-12]. https://dx.doi.org/.

[5] JENA D, BISWAL K C. A numerical study of violent sloshing problems with modified MPS method[J/OL]. Journal of Hydrodynamics, 2017, 29(4): 659-667. DOI:10.1016/S1001-6058(16)60779-5.

[6] XIE F ze, ZHAO W wen, WAN D cheng. CFD simulations of three-dimensional violent sloshing flows in tanks based on MPS and GPU[J/OL]. Journal of Hydrodynamics, 2020, 32(4): 672-683. DOI:10.1007/s42241-020-0039-8.

[7] GINGOLD R A, MONAGHAN J J. Smoothed particle hydrodynamics: theory and application to non-spherical stars[J/OL]. Monthly Notices of the Royal Astronomical Society, 1977, 181(3): 375-389. DOI:10.1093/mnras/181.3.375.

[8] DUAN G, CHEN B. Comparison of parallel solvers for Moving Particle Semi-Implicit method[J/OL]. Engineering Computations, 2015, 32(3): 834-862. DOI:10.1108/EC-02-2014-0029.

[9] KOSHIZUKA S, SHIBATA K, KONDO M, et al. Moving Particle Semi-implicit Method: A Meshfree Particle Method for Fluid Dynamics[M]. Academic Press, 2018.

[10] YAO Q, WANG Z, ZHANG Y, et al. Towards real-time fluid dynamics simulation: a data-driven NN-MPS method and its implementation[J/OL]. Mathematical and Computer Modelling of Dynamical Systems, 2023, 29(1): 95-115. DOI:10.1080/13873954.2023.2184835.
[11] SCARSELLI F, GORI M, AH CHUNG TSOI, et al. The Graph Neural Network Model[J/OL]. IEEE Transactions on Neural Networks, 2009, 20(1): 61-80. DOI:10.1109/TNN.2008.2005605.
[12] ZHANG N, YAN S, MA Q, et al. A CNN-supported Lagrangian ISPH model for free surface flow[J/OL]. Applied Ocean Research, 2023, 136: 103587. DOI:10.1016/j.apor.2023.103587.
[13] ZHANG N, YAN S, MA Q, et al. Numerical simulation of wave-floater interactions using ISPH_GNN trained on data for wave-only cases[J/OL]. Ocean Engineering, 2024, 306: 118041. DOI:10.1016/j.oceaneng.2024.118041.
[14] TANAKA M, MASUNAGA T. Stabilization and smoothing of pressure in MPS method by Quasi-Compressibility[J/OL]. Journal of Computational Physics, 2010, 229(11): 4279-4290. DOI:10.1016/j.jcp.2010.02.011.
[15] LEE B H, PARK J C, KIM M H, et al. Step-by-step improvement of MPS method in simulating violent free-surface motions and impact-loads[J/OL]. Computer Methods in Applied Mechanics and Engineering, 2011, 200(9): 1113-1125. DOI:10.1016/j.cma.2010.12.001.
[16] KIPF T N, WELLING M. Semi-Supervised Classification with Graph Convolutional Networks[A/OL]. arXiv, 2017[2025-03-13]. http://arxiv.org/abs/1609.02907. DOI:10.48550/arXiv.1609.02907.
[17] ZHOU J, CUI G, HU S, et al. Graph neural networks: A review of methods and applications[J/OL]. AI Open, 2020, 1: 57-81. DOI:10.1016/j.aiopen.2021.01.001.
[18] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph Attention Networks[A/OL]. arXiv, 2018[2025-03-13]. http://arxiv.org/abs/1710.10903. DOI:10.48550/arXiv.1710.10903.
[19] GASTEIGER J, BOJCHEVSKI A, GÜNNEMANN S. Predict then Propagate: Graph Neural Networks meet Personalized PageRank[A/OL]. arXiv, 2022[2025-03-13]. http://arxiv.org/abs/1810.05997. DOI:10.48550/arXiv.1810.05997.
[20] SANCHEZ-GONZALEZ A, GODWIN J, PFAFF T, et al. Learning to Simulate Complex Physics with Graph Networks[C/OL]//Proceedings of the 37th International Conference on Machine Learning. PMLR, 2020: 8459-8468[2025-03-11]. https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html.
[21] LI Y, TARLOW D, BROCKSCHMIDT M, et al. Gated Graph Sequence Neural Networks[A/OL]. arXiv, 2017[2025-03-11]. http://arxiv.org/abs/1511.05493. DOI:10.48550/arXiv.1511.05493.
[22] HUBER P J. Robust estimation of a location parameter[M/OL]//KOTZ S, JOHNSON N L. Breakthroughs in Statistics: Methodology and Distribution. New York, NY: Springer, 1992: 492-518[2025-09-26]. https://doi.org/10.1007/978-1-4612-4380-9_35. DOI:10.1007/978-1-4612-4380-9_35.
[23] ZHANG N, YAN S, MA Q, et al. A hybrid method combining ISPH with graph neural network for simulating free-surface flows[J/OL]. Computer Physics Communications, 2024, 301: 109220. DOI:10.1016/j.cpc.2024.109220.