



City Research Online

City, University of London Institutional Repository

Citation: Ma, Q., Zhang, N., Yan, S. & Li, Q. (2024). Numerical simulation of wave-floater interactions using ISPH_GNN trained on data for wave-only cases. *Ocean Engineering*, 306, 118041. doi: 10.1016/j.oceaneng.2024.118041

This is the published version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/32864/>

Link to published version: <https://doi.org/10.1016/j.oceaneng.2024.118041>

Copyright: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

Reuse: Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

City Research Online:

<http://openaccess.city.ac.uk/>

publications@city.ac.uk



Research paper

Numerical simulation of wave-floater interactions using ISPH_GNN trained on data for wave-only cases

Ningbo Zhang, Shiqiang Yan, Qingwei Ma^{*}, Qian Li

School of Science & Technology, City, University of London, London, UK

ARTICLE INFO

Keywords:

ISPH
PPE
Wave-floater interactions
Graph neural network (GNN)
Data
Wave-only cases

ABSTRACT

As a mesh-free approach, the incompressible Smoothed Particle Hydrodynamics (ISPH) method has been often used for simulating wave-structure interaction problems. In the conventional ISPH method, the pressure-projection phase of solving the pressure Poisson's equation (PPE) is the most time-consuming. In recent years, the machine learning (ML) techniques has gradually shown their potential in accelerating the computational fluid dynamics. In this paper, the graph neural network (GNN) supported ISPH method (ISPH_GNN), in which the GNN replaces solving the PPE for the fluid pressure in the conventional ISPH, is adopted for numerical simulations of wave-floater interactions. To the best of the authors' knowledge, this is the first work to study the wave-floater interactions by using GNN supported ISPH method. More importantly, this paper demonstrates that the GNN trained only on data for simpler wave-only cases (i.e. no structure in the wave fields) can be satisfactorily applied to the cases for wave-floater interactions. More specifically, the paper will show this by using the ISPH_GNN with such trained GNN model to simulate various different cases, including the decay tests of a box, a floating box subjected to a wave, the interaction between wave and a moored floating breakwater and the violent green water impact on a floating structure. In most of the cases, the numerical results are validated by comparing with experimental data. Agreement between them is surprisingly satisfactory, being as good as those obtained by the conventional ISPH. The paper will also show that the ISPH_GNN requires much less computational time (97 times less for the cases concerned) than the conventional ISPH for estimating pressure involved in wave-floater interactions. This reveals a great potential that one can train the GNN using the datasets for simpler cases and then use the ISPH_GNN to simulate wave-floater interaction problems.

1. Introduction

The study of waves interactions with floating bodies has been receiving extensive attentions in the design and operation of the offshore and marine structures for the safety and survivability of the structures. In recent years, as a Lagrangian meshless method that discretizes the computational domain by particles, the Smoothed Particle Hydrodynamics (SPH) method has been proved to be a promising modelling tool for simulating the fluid-structure interactions (Sun et al., 2015; Ren et al., 2015; Khayyer et al., 2018; He et al., 2020, 2023; Zhang et al., 2021; Luo et al., 2021; Zhang et al., 2023a). As an important stream of the SPH, the incompressible SPH (ISPH) has also been applied to modelling the wave-structure interactions with producing stable and accurate pressure fields (Shao and Gotoh, 2004; Shao, 2010; Zheng et al., 2018; Zhang et al., 2021; Khayyer et al., 2021) and securing good volume conservation properties (Gotoh et al., 2014). In ISPH method,

the incompressible Navier-Stokes (NS) equation and continuity equation are solved using the projection method and the pressure is evaluated by solving a PPE. The accuracy and convergence of the ISPH are largely affected by the numerical schemes to solve the PPE (Ma et al., 2016). Despite some works have been made to improve the ISPH (Fourtakas et al., 2021; Tsuruta et al., 2023), solving the PPE remains to be of the most time-consuming part in the ISPH.

Recently, the machine learning (ML) approaches have been widely applied to fluid dynamics. Some researchers have used the ML techniques to accelerate the numerical process through replacing the time-consuming or challenging execution part in the conventional procedure by appropriate ML models. For example, Zhang and Duraisamy (2015) adopted the trained ML model to replace the conventional turbulence models for predicting the turbulent production term. Ling et al. (2016) developed a novel neural network architecture for the prediction of Reynolds stress anisotropy tensor. Some works have also been made

^{*} Corresponding author.

E-mail address: q.ma@city.ac.uk (Q. Ma).

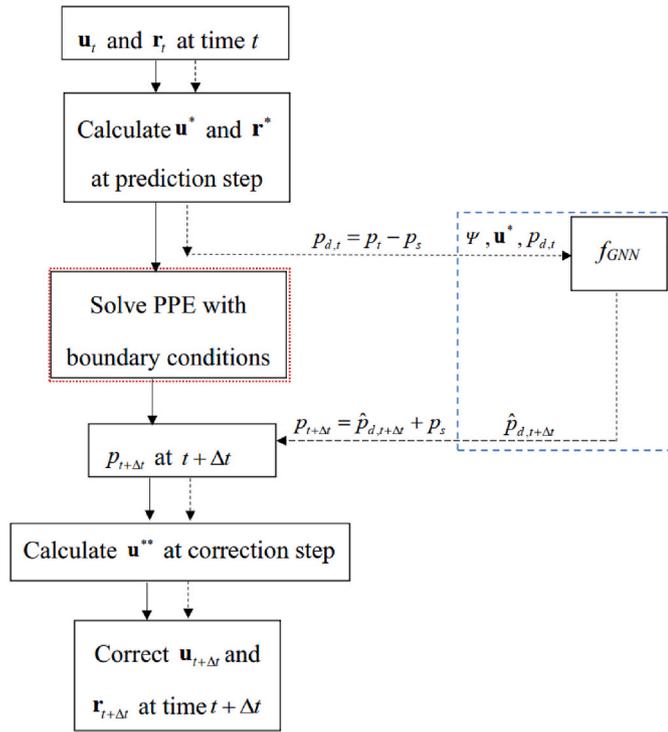


Fig. 1. Flowchart of ISPH incorporating the GNN (solid arrows denotes the flowchart of the conventional ISPH and dashed arrows indicates the flowchart of the ISPH_GNN).

to use the convolutional neural networks (CNNs) to replace solving the PPE for predicting the fluid pressure in the Navier-Stokes or Euler equations using the projection method. Tompson et al. (2017) proposed a data-driven method based on the CNN architecture to solve the pressure projection for accelerating Eulerian fluid simulation; A

smartfluidnet framework based on CNN was developed by Dong et al. (2019) to automatically use multiple neural network models to accelerate Eulerian fluid simulation. Xiao et al. (2019) proposed a CNN-based model, in which the discretization structure and the intermediate velocities were used as the inputs to predict the solutions of the large linear equations resulted from the discretization of the PPE; Zhang et al.

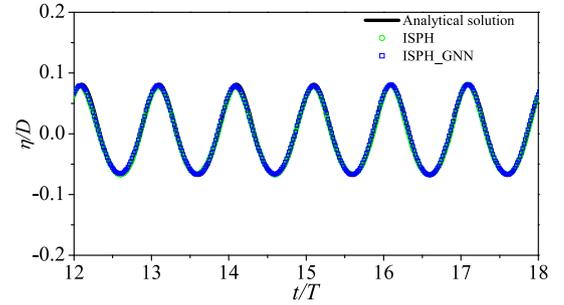


Fig. 4. The comparisons of wave profiles between different ISPH results and analytical solution at the measuring point G_1 located at $x/D = 15$.

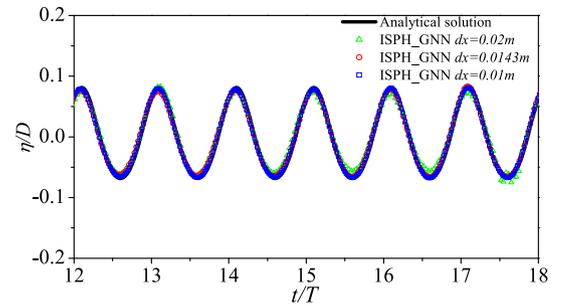


Fig. 5. Comparisons of wave profiles at $x/D = 15$ between analytical solution and ISPH_GNN results with different particle spacings.

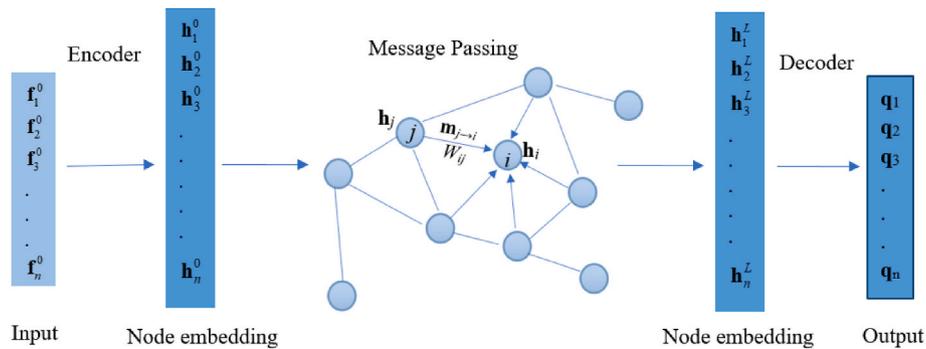


Fig. 2. Diagram of the GNN network architecture.

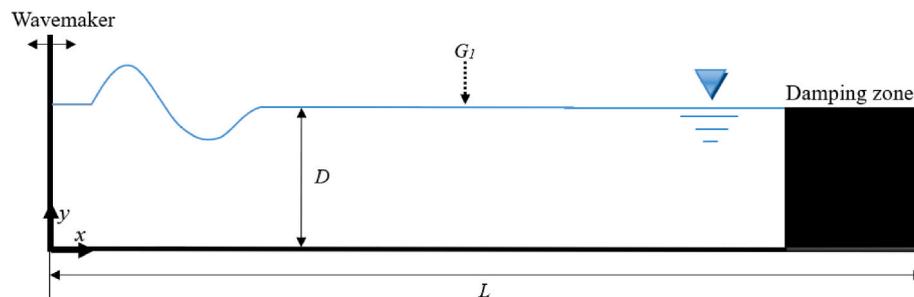


Fig. 3. Schematic setup of the numerical wave tank.

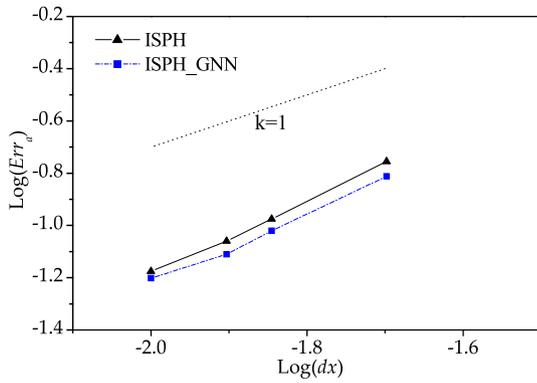


Fig. 6. Errors of numerical results in the regular wave case with different particle resolutions.

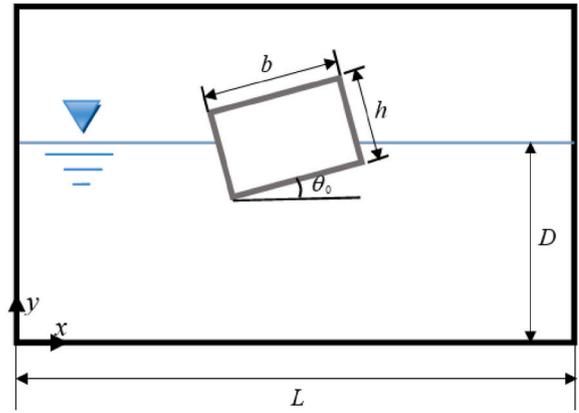


Fig. 9. Floating box with an angle of inclination in water.

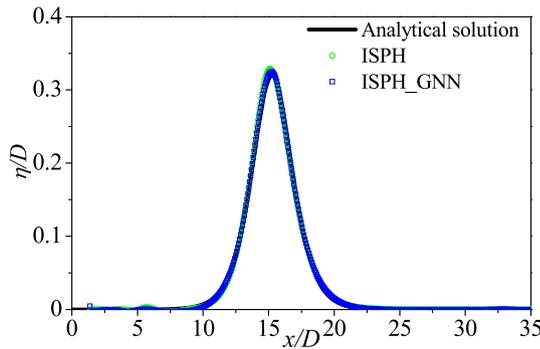


Fig. 7. The comparisons of solitary wave profiles between numerical results and analytical solution at $\tilde{t} = 20$ in the case with $H = 0.32D$.

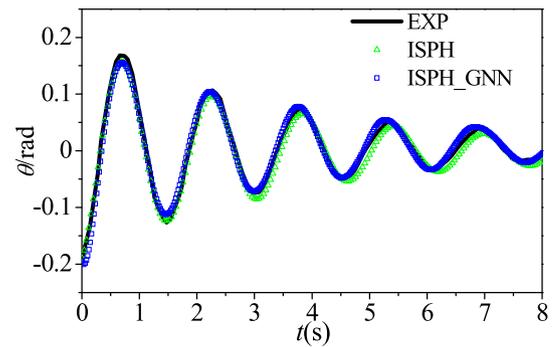


Fig. 10. The comparison of the roll motion in the case of the damped rolling decay.

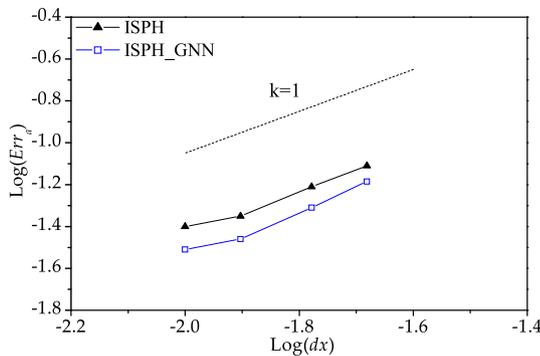


Fig. 8. Averaged errors Err_a of numerical results corresponding to different particle spacing in the solitary wave propagation.

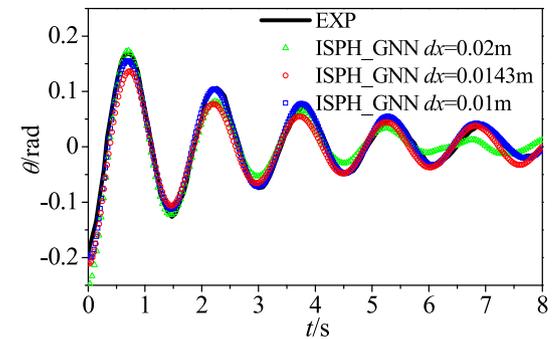


Fig. 11. Time history of roll angles corresponding to different particle spacings from ISPH_GNN.

(2023b) combined the CNN with ISPH for the free-surface flow simulation and used CNN model to replace solving the PPE for the fluid pressure.

Because CNNs perform the convolutions on a uniform Cartesian grid, therefore, the inputs (such as the field data, domain geometry) for CNNs need to be expressed based on the regular grids. This restriction makes CNNs have inherent limitations and pose challenges for their application to particle-based physical problems, in which particles are irregularly distributed. As the geometric deep learning framework, graph neural networks (GNNs) (Gori et al., 2005) are based on graphs which are formed by information at nodes which can be distributed in any way, regularly or irregularly. Therefore, GNNs have the ability to address the limitation of CNN discussed above and offer key advantages for particle-based simulation, e.g., Lagrangian ISPH simulation, in which

the data are usually represented at irregularly-distributed particles.

In recent years, the application of GNNs has grown very rapidly to the fluid simulation. Belbute-Peres et al. (2020) incorporated an embedded differentiable fluid simulator with the graph convolutional network to develop a hybrid graph neural network for fluid flow prediction. Pfaff et al. (2020) proposed a GNN model based on the unstructured mesh to simulate the incompressible flow around cylinders as well as the compressible flow around airfoils. A similar GNN model was employed by Chen et al. (2021) as a surrogate model to study the pressure and velocity fields in simulation of the incompressible flow around a bluff body. Gao and Jaiman (2022) presented a quasi-monolithic GNN-based framework for data-driven reduced-order modeling flow around an elastically mounted cylinder, where a multi-layer perceptron was used to evolve mesh displacements. Lino et al. (2022) proposed a data-driven framework including the multi scale

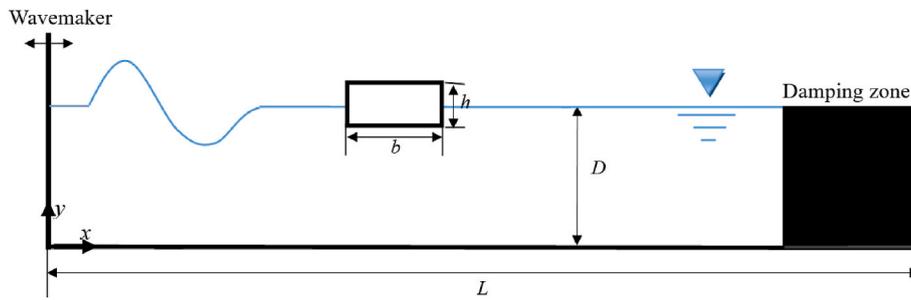


Fig. 12. Sketch of wave interaction with a freely floating box.

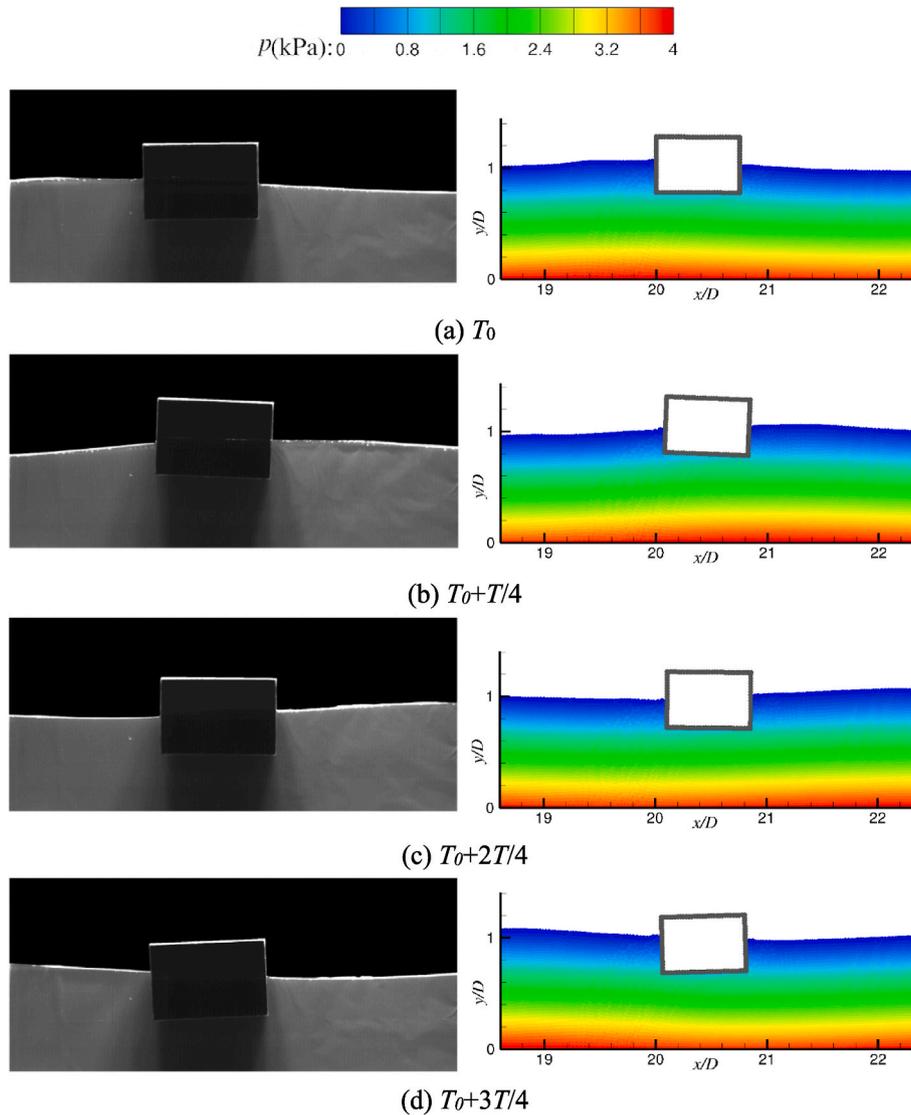


Fig. 13. The comparisons of the motions of the floating box between experimental photos (Ren et al., 2015) (left) and ISPH_GNN simulations (right) at different instants.

rotation-equivariant graph neural networks for learning the unsteady Eulerian fluid dynamics, where the domain was discretized into an unstructured set of fixed nodes. Some efforts have also been made to apply GNNs for modeling fluid dynamics simulations in Lagrangian particle (meshfree) system. In this regard, Sanchez-Gonzalez et al. (2020) presented a machine learning framework with graph neural network to simulate fluids and fluid interacting with static obstacles. Kumar and Vantassel (2022) developed a Graph Network Simulator (GNS) by

discretizing the domain with nodes and the links connecting the nodes for fluid modeling. Li and Farimani (2022) developed a data-driven approach based on GNN to simulate fluid within the Lagrangian particle representation. Zhang et al. (2024b) combined the GNN with ISPH for simulating the free-surface flows. They mainly focused on the development of the GNN to speed up fluid simulation without considering structures. In their approach, the cases generating data for training GNN is largely in the similar scope of cases they studied using the trained

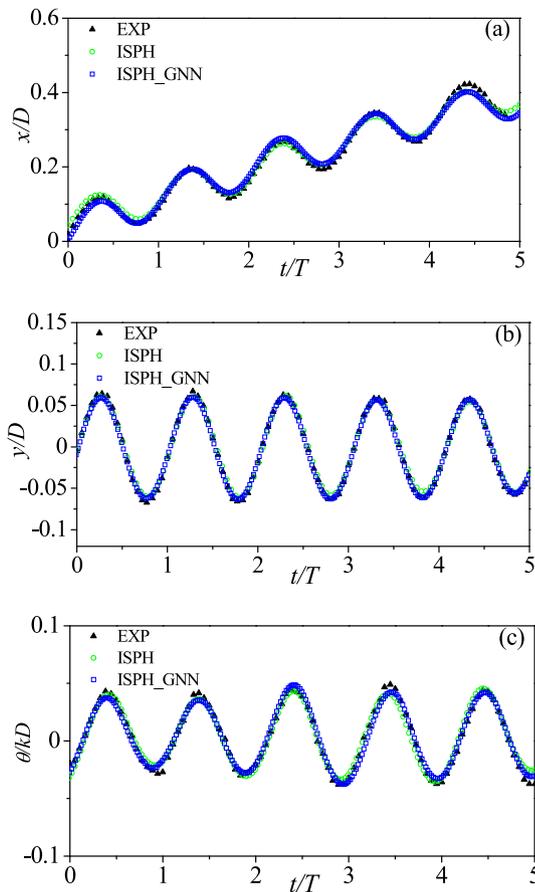


Fig. 14. The comparisons of time histories of motions of the floating box: (a) sway; (b) heave and (c) roll.

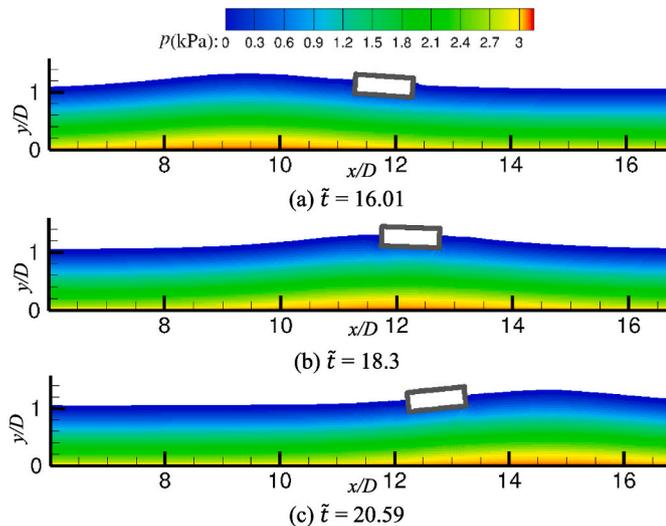


Fig. 15. The particles distribution with pressure field from the ISPH_GNN in the case of solitary wave interactions with a floater at different time instants.

GNN.

While these existing GNN-based works have demonstrated the potential of GNNs for various fluid flow applications, these works mainly focus on fluid simulation and flow around fixed body and/or domain boundaries. To the best of our knowledge, there are no relevant works that combines the GNN with the ISPH for the application to wave-floater interactions simulations. More interestingly, no publication has

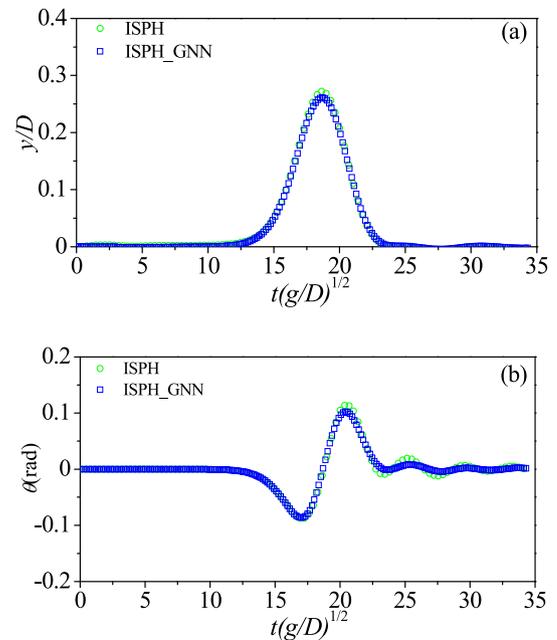


Fig. 16. Time histories of the motions of the floating box under solitary wave with $H = 0.24D$: (a) heave and (b) roll.

demonstrated that the GNN trained using data for simple cases without structures can be applied to more complex cases involving wave-floater interaction.

In this paper, we adopt the GNN supported ISPH method (ISPH_GNN) to simulate wave-floater interactions. In this method, the GNN model developed by Li and Farimani (2022) is combined with our in-house ISPH solver (Zhang et al., 2018) to replace solving the PPE for pressure in the ISPH. More significant contribution of this paper is to show that ISPH_GNN with GNN trained by using datasets generated only from simple wave propagation cases without any structure can be applied to simulate relatively complex cases for wave-floater interactions.

2. Mathematical and numerical formulation

2.1. Lagrangian ISPH

In the ISPH, the fluid is governed by the incompressible NS equation and the continuity equation that are respectively given as,

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \mathbf{g} + \nu\nabla^2\mathbf{u} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

where \mathbf{u} , p and ρ are, respectively, the velocity, pressure and density of the fluid, \mathbf{g} is the gravitational acceleration and ν is the kinematic viscosity of the fluid. Eq. (1) is written in a Lagrangian frame and D/Dt is the material derivative following the motion of fluid particles. On the solid boundaries, the following Neumann boundary conditions for velocity and pressure, respectively, are imposed,

$$\mathbf{u} \cdot \mathbf{n} = \mathbf{U} \cdot \mathbf{n} \quad (3)$$

$$\mathbf{n} \cdot \nabla p = \rho(\mathbf{n} \cdot \mathbf{g} - \mathbf{n} \cdot \ddot{\mathbf{U}}) \quad (4)$$

where \mathbf{U} and $\ddot{\mathbf{U}}$ are the velocity and acceleration of the solid boundary, respectively; \mathbf{n} is the corresponding unit normal vector. In addition, the following free-surface condition is imposed,

$$p = 0 \quad (5)$$

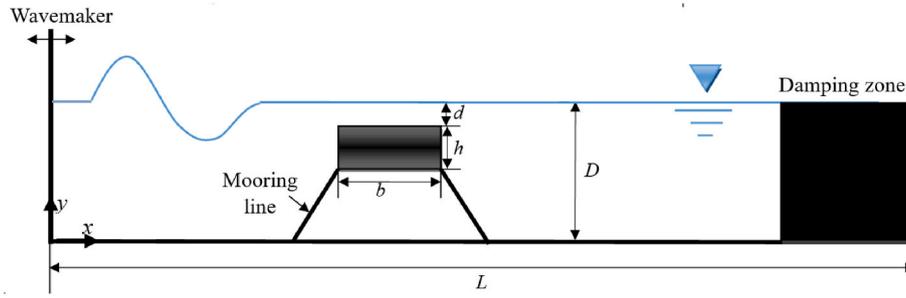


Fig. 17. Sketch of regular wave interaction with a moored floating body.

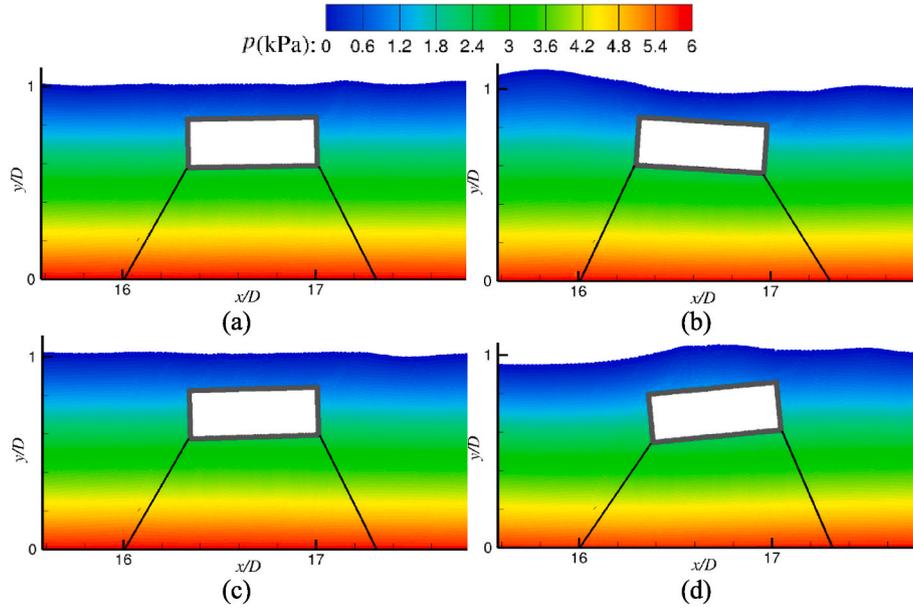


Fig. 18. The particles distribution with pressure field from the ISPH_GNN at (a) T_0 ; (b) $T_0+T/4$; (c) $T_0+2T/4$ and (d) $T_0+3T/4$.

A projection method is used to solve the governing equations and boundary conditions, which is composed of three main stages.

- (1) Prediction stage: calculating the intermediate fluid velocity \mathbf{u}^* and the intermediate position vector \mathbf{r}^* at the particle using

$$\mathbf{u}^* = \mathbf{u}_t + \Delta \mathbf{u}^* \quad (6)$$

$$\Delta \mathbf{u}^* = (\mathbf{g} + \nu \nabla^2 \mathbf{u}) \Delta t \quad (7)$$

$$\mathbf{r}^* = \mathbf{r}_t + \mathbf{u}^* \Delta t \quad (8)$$

where \mathbf{u}_t and \mathbf{r}_t are the velocities and positions of particles at the time t , respectively; Δt is the time step size; \mathbf{u}^* and \mathbf{r}^* are the intermediate velocities and positions of particles.

- (2) Pressure projection stage: By substituting Eqs. (6) and (7) to Eqs. (1) and (2), it is not difficult to derive the PPE,

$$\nabla^2 p_{t+\Delta t} = \frac{\rho \nabla \cdot \mathbf{u}^*}{\Delta t} \quad (9a)$$

This equation is derived to satisfy a divergence free condition. Xu et al. (2009) revealed that the divergence-free approach based on Eq. (9a) may suffer from a numerical instability caused by ill-distributed particles following the Lagrangian movement of particles. According to numerical tests presented in Ma and Zhou (2009) and also suggested by Zheng et al. (2014), the better results for violent water waves can be obtained by using a mixed term on the right-hand side of Eq. (9a).

Following this, the following alternative form of PPE is applied to all internal fluid particles,

$$\nabla^2 p_{t+\Delta t} = \frac{\Psi}{\Delta t} \quad (9b)$$

where $\Psi = \alpha \frac{\rho - \rho^*}{\Delta t} + (1 - \alpha) \rho \nabla \cdot \mathbf{u}^*$ with ρ^* being the intermediate density of particle estimated by considering the movement of particles, and α is the blending coefficient. According to the numerical investigation in Zhang et al. (2018, 2021, 2024a), the value of α is taken as 0.01 in this work.

- (3) Correction stage: After the PPE with the boundary conditions are solved, the velocities change during the correction step are estimated by

$$\mathbf{u}^{**} = -\frac{\Delta t}{\rho} \nabla p_{t+\Delta t} \quad (10)$$

The velocities of particles at time $t + \Delta t$ are corrected using

$$\mathbf{u}_{t+\Delta t} = \mathbf{u}^* + \mathbf{u}^{**} \quad (11)$$

The positions of particles are updated by

$$\mathbf{r}_{t+\Delta t} = \mathbf{r}_t + \frac{\mathbf{u}_t + \mathbf{u}_{t+\Delta t}}{2} \Delta t \quad (12)$$

Further details of the numerical implementations including the numerical discretization of the PPE, the numerical approaches for imposing the boundary conditions, the gradient and the divergence

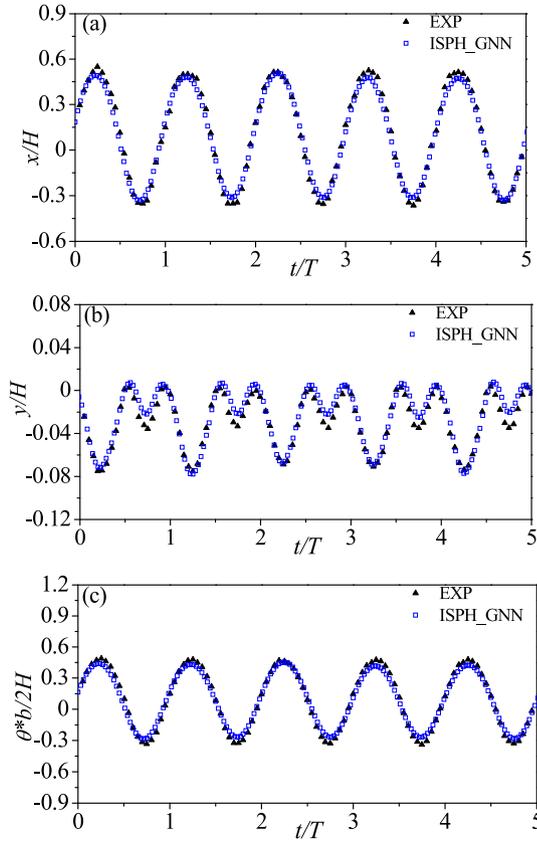


Fig. 19. Comparison of experimental and simulated time histories of the breakwater motions: (a) sway; (b) heave; (c) roll.

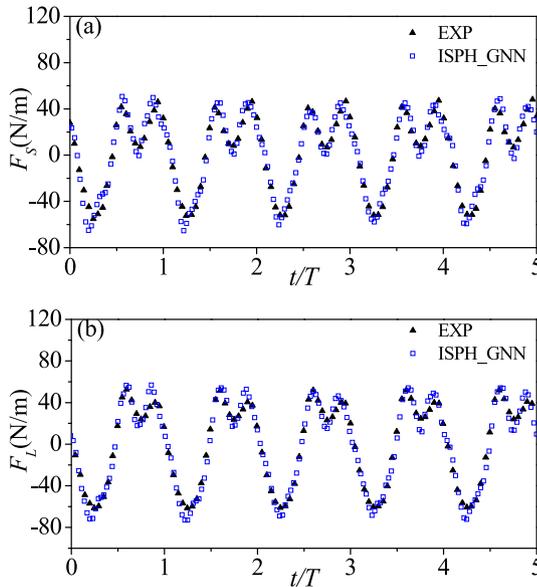


Fig. 20. The comparisons of time histories of the mooring forces: (a) on the seaward side and (b) on the leeward side.

operator in the conventional ISPH can be found in Zhang et al. (2021).

2.2. Motion of floating body

The floating body is considered as rigid body in this work. In 2D cases, the motion of the floating body can be described by the Newton's

second law as

$$M \frac{d\mathbf{V}}{dt} = \mathbf{F} + M\mathbf{g} \quad (13)$$

$$\mathbf{I} \frac{d\boldsymbol{\Omega}}{dt} = \mathbf{T} \quad (14)$$

and the movements of the floating body are estimated using

$$\frac{d\mathbf{r}_G}{dt} = \mathbf{V} \quad (15)$$

$$\frac{d\boldsymbol{\theta}}{dt} = \boldsymbol{\Omega} \quad (16)$$

where \mathbf{F} and \mathbf{T} are the force and torque action on the floating body; M and \mathbf{I} are the mass and inertia matrices of the floating body, respectively; \mathbf{V} is the translational velocity at the gravity centre; $\boldsymbol{\Omega}$ is the angular velocity of floating body around the gravity centre; $\boldsymbol{\theta}$ denotes the Euler angle (Yan and Ma, 2007).

The moving solid boundaries of the floating body are represented by boundary particles. These boundary particles can be treated to move with the floating body, the corresponding velocity of the boundary particle can be obtained by

$$\mathbf{u}_b = \mathbf{V} + \boldsymbol{\Omega} \times (\mathbf{r}_b - \mathbf{r}_G) \quad (17)$$

where \mathbf{r}_b is the position vector for the boundary particle and \mathbf{r}_G is the position vector of the centroid of the floating body. The velocity \mathbf{u}_b calculated by Eq. (17) is used in the continuity equation. The acceleration can be obtained by

$$\dot{\mathbf{u}}_b = \frac{d\mathbf{V}}{dt} + \frac{d\boldsymbol{\Omega}}{dt} \times (\mathbf{r}_b - \mathbf{r}_G) + \boldsymbol{\Omega} \times (\mathbf{u}_b - \mathbf{V}) \quad (18)$$

In the ISPH implementation, the force and torque on the floating body can be calculated by the following discretized equations

$$\mathbf{F} = \sum_{b=1}^N \mathbf{n}_b p_b dx \quad (19)$$

$$\mathbf{T} = \sum_{b=1}^N (\mathbf{r}_b - \mathbf{r}_G) \times \mathbf{n}_b p_b dx \quad (20)$$

where p_b is the pressure of floater boundary particle b ; dx is the initial particle spacing; \mathbf{n}_b is the inner normal vector of boundary particle b ; N is the number of floater boundary particle.

2.3. Incorporating GNN with ISPH

In the ISPH_GNN, the GNN is adopted to replace solving the PPE for pressure in the conventional ISPH. The flowchart of ISPH_GNN is illustrated in Fig. 1. The brief discussions on some issues related to the GNN model, the combining ISPH with the GNN as well as training and implementing the GNN are given here. More details can be found in Zhang et al. (2024b).

2.3.1. GNN model

The GNN (Li and Farimani, 2022) used in this paper can be regarded to perform convolutions on graphs defined by nodes and their spatial relations (Wu et al., 2020). This GNN model (Fig. 2) includes three steps: encoding, message-passing, and decoding (Gao and Jaiman, 2022; Lino et al., 2022).

The encoding step aims to convert the input data into network embeddings which can be used for graph convolutions. Considering a particle i in Fig. 2, the input \mathbf{f}_i^0 is encoded into node embedding \mathbf{h}_i^0 via a learnable encoder $en()$ based on the multi-layer perceptron (MLP) with N layers:

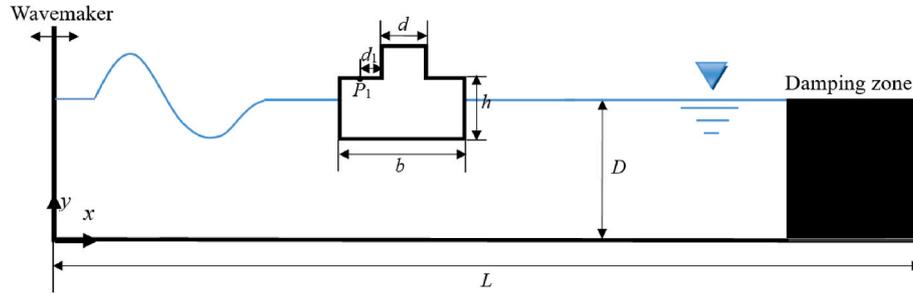


Fig. 21. Sketch of green water impact on the floating structure.

$$en(\mathbf{f}_i^l) = \sigma(\mathbf{w}_l \mathbf{f}_i^{l-1} + b_l) \quad \text{with } l = 1, 2, 3, \dots, N \quad (21)$$

$$\mathbf{h}_i^0 = en(\mathbf{f}_i^N) \quad (22)$$

where $\sigma(\cdot)$ is an activation function, \mathbf{w}_l and b_l are the learnable weight vector and bias term of the l -th layer in MLP; \mathbf{f}_i^l is the node feature at the l -th layer in MLP. The \mathbf{h}_i^0 will be used as the input of graph convolution described below.

In the message passing block, the influence of a neighbouring node j on the node i in the l -th layer convolution is represented by $\mathbf{m}_{j \rightarrow i}^l$ which is given as

$$\mathbf{m}_{j \rightarrow i}^l = \mathbf{h}_j^{l-1} W(r_{ij}, r_0) \quad (23)$$

where \mathbf{h}_j^{l-1} denotes the node embedding from the neighbouring node j in the $(l-1)$ -th layer of message passing, and the weight is given by $W(r_{ij}, r_0) = \begin{cases} r_0/r_{ij} - 1 & 0 < r_{ij} \leq r_0 \\ 0 & r_{ij} > r_0 \end{cases}$. The influence from all the neighbouring nodes is aggregated and given by

$$\mathbf{M}_i^l = \frac{\sum_{j=1}^N \mathbf{m}_{j \rightarrow i}^l}{\sum_{j=1}^N W(r_{ij}, r_0)} \quad (24)$$

In the l -th layer of message passing, the node embedding of the node i is calculated by

$$\mathbf{h}_i^l = \sigma(\mathbf{w}_l \mathbf{M}_i^l + b_l' \mathbf{h}_i^{l-1}) \quad \text{with } l = 1, 2, \dots, L \quad (25)$$

where $\sigma(\cdot)$ is an activation function, \mathbf{w}_l and b_l' are the learnable weight vector and bias term of the l -th layer.

After the message passing with L layers, the output \mathbf{q}_i (the pressure in this work) can be obtained by decoding the final embedding \mathbf{h}_i^L using a learnable decoder $em(\cdot)$ based on the multi-layer perceptron (MLP) with N layers, which is similar to Eq. (21) but with different learnable weights \mathbf{w}_l' and bias b_l' :

$$em(\mathbf{h}_i^L) = \sigma(\mathbf{w}_l' \mathbf{h}_i^{L-1} + b_l') \quad \text{with } l = L+1, L+2, \dots, L+N \quad (26a)$$

$$\mathbf{q}_i = em(\mathbf{h}_i^{L+N}) \quad (26b)$$

2.3.2. Input parameters

As indicated above, GNN is used to replace PPE of pressure, i.e., Eq. (9b). To train the GNN, the inputs of the GNN model need to include the variables, which reflect the characteristics of Eq. (9b). In this paper, a blended form (Eq. (9b)) including the right hand side $\frac{\Psi}{\Delta t}$ is used in the PPE of the ISPH model. Therefore, the Ψ is adopted as one of the input parameters. In addition, the intermediate velocity, \mathbf{u}^* can reflect the gravity and the viscous effects on the water wave. According to the study of combining ISPH with CNN model (ISPH_CNN) for free-surface flows

modelling in Zhang et al. (2023b), the better prediction performance can be obtained by including \mathbf{u}^* as one of input parameters than without considering it. Therefore, we also use \mathbf{u}^* as an input parameter for the GNN model here. In addition, the study of Zhang et al. (2023b) has demonstrated that adoption of the pressure at the previous time step in the inputs can improve the prediction accuracy for the pressure at current step. Therefore, we also consider the pressure at the previous time step in the input parameters.

However, it should be noted that in both the training and predictions of GNN, the total pressure p is split into two parts: the reference pressure p_s , and the rest part of pressure p_d , i.e., $p = p_d + p_s$. The reference pressure p_s can be directly calculated by $\rho g \tilde{z}$, where \tilde{z} is the vertical distance from the fluid particle to its nearest free-surface particle. Based on the splitting, the GNN is trained using the data for p_d and also predicts the corresponding part of pressure \hat{p}_d after training. Based on our numerical tests, the overall results with the pressure predicted by the GNN trained in this way are significantly better than those obtained by using GNN to directly handle total pressure.

According to the above discussions, the term Ψ , the intermediate velocity \mathbf{u}^* and the pressure $p_{d,t} = p_t - p_s$ are adopted as the input parameters to train the GNN model for predicting the pressure $\hat{p}_{d,t+\Delta t}$. The function f_{GNN} from input to output by the GNN model can be expressed as:

$$\hat{p}_{d,t+\Delta t} = f_{GNN}(\Psi, \mathbf{u}^*, p_{d,t}, c_p) \quad (27a)$$

where c_p are the network coefficients, i.e., the weights and bias involved in Eqs. (21), (25) and (26), which are optimized during training process as discussed in Section 2.3.1.

In general, after the GNN model is trained, the pressure $\hat{p}_{d,t+\Delta t}$ can be predicted based on the input data of Ψ , \mathbf{u}^* and $p_{d,t}$ by the GNN model f_{GNN} with the network coefficients c_p . After the prediction by the GNN model f_{GNN} , the total pressure of fluid particles used in Eq. (1) for updating the velocity field can be obtained by

$$p_{t+\Delta t} = \hat{p}_{d,t+\Delta t} + p_s \quad (27b)$$

2.3.3. Training data

In this paper, the GNN model is trained on data only generated from wave propagation cases without any structure. These cases are simulated by using the consistent second order ISPH (ISPH_CQ) (Zhang et al., 2024a). In the ISPH_CQ, the governing equations and boundary conditions are the same as the conventional ISPH, but the derivatives for Laplacian operator, velocity divergence and corresponding Neumann boundary condition and the pressure gradient are numerically approximated by the quadric SFDI (Yan et al., 2020). More details about the ISPH_CQ can be found in (Yan et al., 2020; Zhang et al., 2024a).

The datasets are generated for two kinds of waves: the regular waves and solitary waves without any object and structure appearing in the wave fields. The waves are simulated in a numerical wave tank as sketched in Fig. 3, where D and L are the mean water depth and the length of the tank, respectively. A wavemaker is placed at the left end of the tank to generate the incident waves. The parameters for the cases

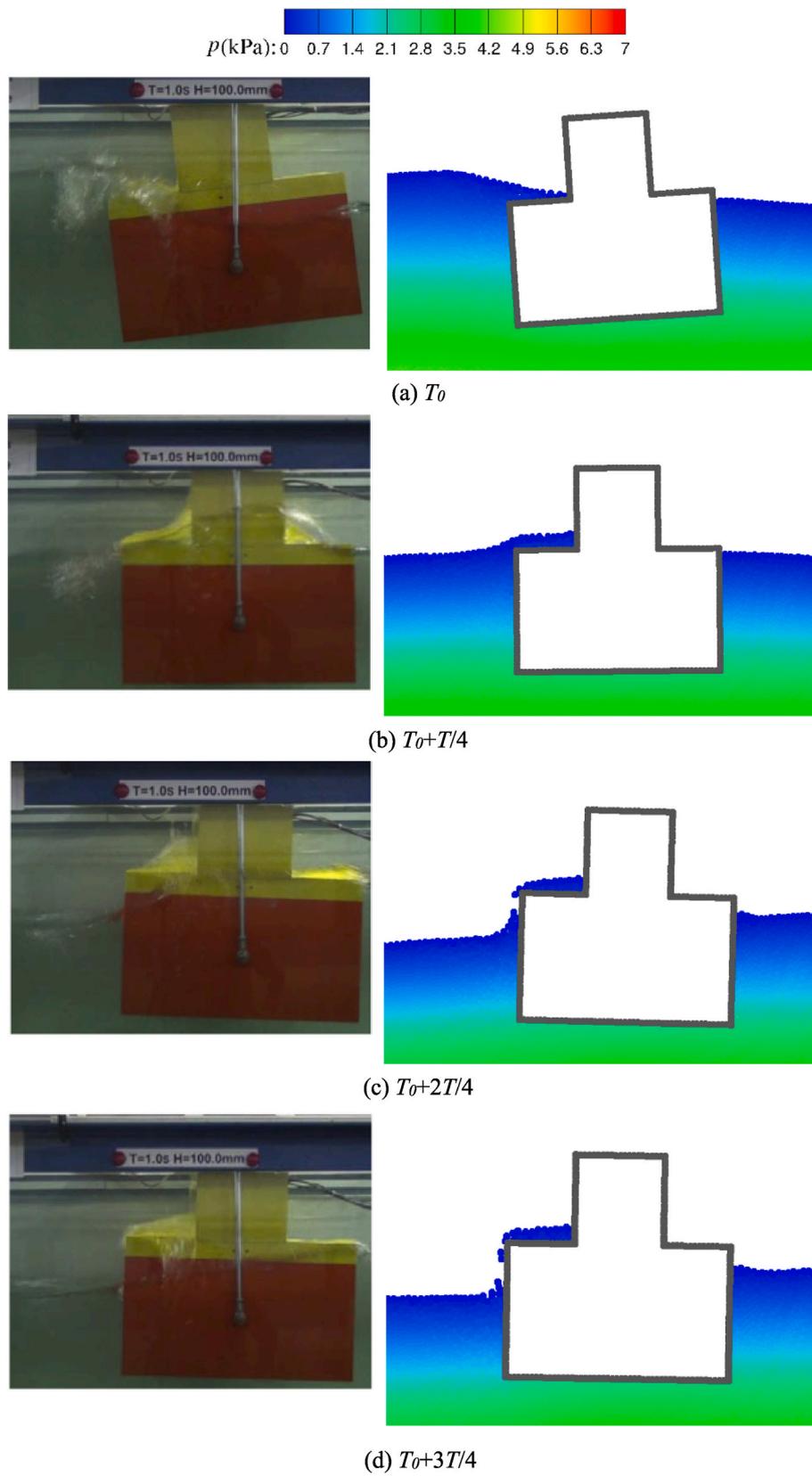


Fig. 22. Comparisons of the floater movement progress during green water impact between laboratory photos (Zheng et al., 2018) (left) and ISPH_GNN simulations (right) at different instants.

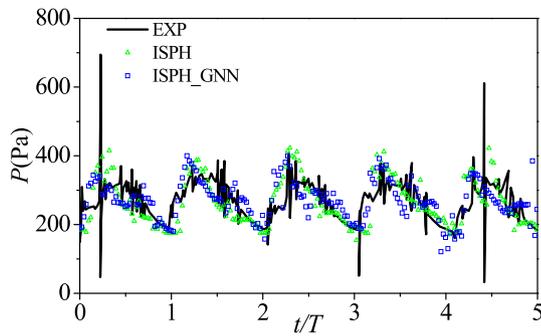


Fig. 23. Time histories of the impact pressure on deck at P_1 .

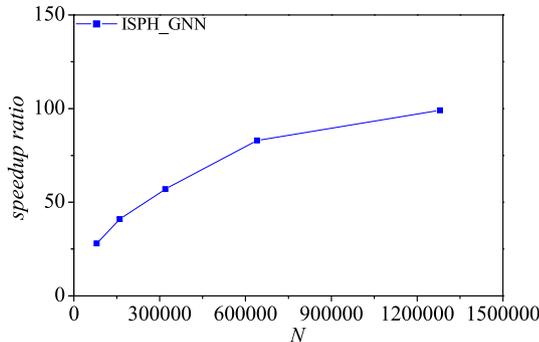


Fig. 24. Speedup ratio averaged over computational periods.

Table 1

Case configurations for the regular wave interactions with a freely floating box.

Case	D (m)	L (m)	dx (m)	N
1	0.4	20	0.01	80,000
2	0.4	40	0.01	160,000
3	0.4	80	0.01	320,000
4	0.8	80	0.01	640,000
5	0.8	160	0.01	1280,000

used for generating the training database are randomly selected. For example, for each of the regular wave cases, the wave height H is randomly selected in a range from $0.05D$ to $0.25D$ whereas the wave period is from 1.0 s to 1.2 s, respectively. The water depth $D = 0.4$ m and the length of the tank $L = 25D$ are used in these cases. The initial particle size of 0.01m and the time step size of 0.001s are used in ISPH_CQ to run these cases. Totally, 40 cases are simulated with each case including 400 frames recorded with the time interval of 0.05 s. For each of the solitary wave cases, the wave height H is selected in a range from $0.2D$ to $0.4D$, whereas water depth D is from 0.25 m, 0.275 m–0.3 m. The length of the tank $L = 40D$ is used. The initial particle size of 0.0125 m and the time step size of 0.00125 s are used to produce the corresponding database. Totally, 60 cases are simulated with each case including 400 frames recorded with the time interval of 0.0125 s.

For training purpose, the datasets need to be split into training sets and test sets. The training sets include 30 regular wave cases and 40 solitary wave cases. They are selected randomly from the total cases simulated. The simulated cases except for the training sets are used as test sets.

2.3.4. Training and implementing the GNN

With the training and testing data sets available, the GNN is trained on the datasets described in above section to minimize the loss function, which is the sum of the squared error between the prediction and actual values of pressure in this work:

$$f_{obj} = \sum_i^n s_i (\hat{p}_{d,i} - p_{d,i})^2 \quad (28)$$

where $\hat{p}_{d,i}$ and $p_{d,i}$ are the predicted and actual pressure obtained from the training data at particle i , respectively. s_i is a weight for considering the effects of boundaries. The weight term considering this and the free surface is proposed as

$$s_i = \begin{cases} 3 & d_b \leq 3.0 \cdot dx \\ 2 & 3.0 \cdot dx < d_b \leq 6.0 \cdot dx \\ 1 & d_b > 6.0 \cdot dx \\ 0 & \text{free surface} \end{cases} \quad (29)$$

which d_b is the distance from a particle to a boundary and dx is initial particle spacing.

During the training process, the difference between the predicted and actual pressure on the free surface is not taken into account. The free-surface condition Eq. (5) can be satisfied by enforcing the pressure of all free-surface particle to be zero when predicting the pressure by GNN model. In addition, to avoid the error introduced by imposing solid boundary condition, only inner fluid particles are considered, ignoring the particles representing solid boundaries during the training and testing stages of the GNN. Although the boundary conditions treatment for the wall and floating body may not be perfect, it is effective as the results demonstrate below in Sections 4 and 5. Other methods for dealing with the boundary conditions will be explored in our future work.

After the model is trained, it will be implemented in the numerical practice of the ISPH_GNN as illustrated in Fig. 1. After predicting the fluid pressure at inner particles by using the trained GNN model, the pressure p_b at the boundary particles on the wall boundary and floating body boundary are both obtained directly from the neighbouring fluid particles,

$$p_b = \frac{\sum_{f \in \text{fluid}} p_f W_{bf} + (\mathbf{g} - \dot{\mathbf{U}}) \cdot \sum_{f \in \text{fluid}} \rho_f (\mathbf{r}_f - \mathbf{r}_b) W_{bf}}{\sum_{f \in \text{fluid}} W_{bf}} \quad (30)$$

where p_f is the pressure of the fluid particle, ρ_f is the density of fluid, W_{bf} is a kernel weight function, M is total number of the neighbouring fluid particles of the boundary particles, \mathbf{r}_f and \mathbf{r}_b are the position vector of fluid and boundary particles.

3. Simulation of wave propagation using the trained model

In this part, before the ISPH_GNN is used for the wave-floater interaction simulation, the ISPH_GNN is applied to the regular wave and solitary wave propagation simulations to evaluate the corresponding performance of the ISPH_GNN in terms of the prediction accuracy and convergence.

Firstly, the ISPH_GNN is applied to simulate the regular wave propagation, where the wave tank with $D = 0.4$ m and $L = 50D$ is used. The regular waves are generated by a piston wavemaker using the linear wavemaker theory (Gotoh et al., 2004). To prevent the wave reflections from the right end of the tank, an artificial damping region with a length of 3 wavelength is attached to the right end. In the damping region, the fluid velocity is corrected by a velocity damping during the correction stage of the projection procedure (Wei and Kirby, 1995). We choose the case with wave height $H = 0.15D$ and wave period $T = 1.2$ s. The case considered here does not coincide with any case for training. For such wave condition, the second order Stokes wave theory (Madsen, 1971) is valid and provides the reference value for the error evaluation in the present numerical investigation. Fig. 4 shows the comparison of the wave profiles from numerical simulations using the initial particle spacing $dx = 0.01$ m with the analytical solution from the second-order

Stokes waves at $x/D = 15$. The result predicted by ISPH_GNN shows good agreement with that from analytical solution.

Fig. 5 gives the corresponding surface elevations obtained by ISPH_GNN model with different particle spacings dx , ranging from 0.01 m to 0.02 m, i.e. $D/dx = 80$ to 40, for the convergence investigation. It can be seen in Fig. 5 that the numerical results become closer to the analytical solution with reducing the particle spacing. The errors of numerical results are shown in Fig. 6, in which the error is computed by

$$Err_a = \frac{\sqrt{\sum_{i=1}^N (\eta_{i,n} - \eta_{i,f})^2}}{\sqrt{\sum_{i=1}^N \eta_{i,f}^2}} \quad (31)$$

where $\eta_{i,n}$ is the surface elevation at i -th time step from the ISPH_GNN and the $\eta_{i,f}$ is the corresponding analytical solution, N denotes the total number of time steps from $t/T = 12$ to 18. For convenience, the dash line representing first-order ($k = 1$) is included in the figure. It can be seen in Fig. 6 that the error of the ISPH_GNN reduces linearly as the particle size decreases, which demonstrates the linear convergence similar to that in the conventional ISPH. As can also be observed from Fig. 6 that the error of the ISPH_GNN is lower than the conventional ISPH, which indicates that the ISPH_GNN is slightly more accurate than the conventional ISPH if the particle spacing is the same.

Further attention is paid to the solitary wave propagation with $H = 0.32D$ and $D = 0.25$ m. Again this case does not coincide with any case used for training. The solitary wave is generated by the wavemaker using the technique presented in Ma and Zhou (2009). For comparison, the corresponding free-surface profiles at $\tilde{t} = 20$ ($\tilde{t} = t\sqrt{g/D}$) between different ISPH results with initial particle spacing 0.01 m, i.e. $D/dx = 25$, are shown in Fig. 7. This figure confirms a good agreement between the ISPH_GNN result and the Boussinesq solution (Lee et al., 1982). The corresponding errors of results from the ISPH_GNN and ISPH methods for this case with different particle spacing are plotted in Fig. 8. The error is computed in the same way as that used in Eq. (31) but $\eta_{i,n}$ and $\eta_{i,f}$ are replaced by the wave elevations at i -th particle obtained from the numerical results and the analytical solution for this case; N is the total number of free-surface particles in the subdomain from $x/D = 10.0$ to $x/D = 21.0$ in Fig. 7. Overall, the ISPH_GNN has a similar convergence rate to the conventional ISPH, but it has a slightly higher accuracy than the conventional ISPH.

4. Numerical investigations of wave-floater interactions

In this section, we will demonstrate that the ISPH_GNN with the GNN model trained on wave-only propagation cases can be used to simulate the cases with wave-floater interactions. The cases considered include the roll decay test, the motions of a freely floating body under regular and solitary waves, the interaction between regular wave and a moored floating breakwater and the green water on the floating structure. These wave-floater interactions cases are beyond and more complex than the wave-only cases used to generate the training data for the training of GNN model. The corresponding results from the conventional ISPH (Zhang et al., 2018) are also presented for the purpose of comparison.

4.1. Roll decay test

Firstly, the case of roll decay of a floating box is studied, as shown in Fig. 9, in which a floating rectangular box with the width $b = 0.75D$ and height $h = 0.5D$ is placed with the initial angle $\theta^0 = 15^\circ$ in the still water. The mass and the corresponding moment of inertia of the box are 30 kg and 0.325 kg m², respectively. In this case, the tank length $L = 7.5D$ and the mean water depth $D = 0.4$ m is adopted. The floating box is homogeneous and its centre of mass is placed at $(3.75D, 1.0D)$.

The comparison of the roll angle time history between the numerical results and the experimental data (Ren et al., 2015) is presented in Fig. 10. The initial particle size dx is taken as $D/dx = 40$, and the time step is determined by $dt/dx = 0.1$. It can be observed that the floating box starts to roll and eventually stops at a balanced position after the release from its original position. The result predicted by ISPH_GNN shows good agreement with that from the experimental data, although some differences are visible. Compared with the experimental data in Fig. 10, the errors of the roll angle are about 13.2% for ISPH_GNN and 13.5% for ISPH, respectively. To investigate the convergence of the model, Fig. 11 gives the computed results by ISPH_GNN with different spatial resolutions. As can be found in Fig. 11 that the numerical results tend to approach the experimental data more closely as the particle number increases, which implies the convergence of numerical results in spatial domains.

4.2. Motions of a freely floating body in waves

Secondly, ISPH_GNN with the trained f_{GNN} is used to study the motions of a floating body in waves. The response of the freely floating box subjected to regular waves is considered, which has been experimentally investigated by Ren et al. (2015). A sketch of the present domain is given in Fig. 12. The water depth is $D = 0.4$ m. The width and the height of the rectangular box are $b = 0.75D$ and $h = 0.5D$, respectively. The density of the body is uniform, i.e. 500 kg/m³, yielding to a half-immersed condition at a still water (the initial draft is 0.25D). The centroid of the box is initially placed at $(20.0D, 1.0D)$.

Fig. 13 shows a series of snapshots comparing the ISPH_GNN results with the initial particle spacing $dx = 0.01$ m, i.e. $D/dx = 40$, and experimental photos for the motion at different time instants in the case with the wave height $H = 0.1D$ and the wave period $T = 1.2$ s. As can be seen in Fig. 13(a) that the body drifts upwards under the wave action. Then, it rotates clockwise in Fig. 13(b) and returns to horizontality at $T_0 + 2T/4$ in Fig. 13(c). Then the body starts to rotate anti-clockwise as shown in Fig. 13(d). From Fig. 13, the ISPH_GNN method can produce consistent and smooth pressure fields near the floating box.

Fig. 14 shows the comparison of the corresponding time histories of the motion of the floating box with experimental data (Ren et al., 2015). Generally, the agreement between numerical results and experimental data is satisfactory, although some differences are visible from the experimental data. Based on the experimental results in Fig. 14, the errors of the sway motion are about 4.4% for ISPH_GNN and 4.1% for ISPH in Fig. 14(a), the errors of the heave motion are about 7.7% for ISPH_GNN, and 8.9 % for ISPH in Fig. 14(b) and the errors of the roll motion are about 13.8% for ISPH_GNN, and 15.2 % for ISPH in Fig. 14 (c).

The ISPH_GNN with f_{GNN} is further used to simulate the solitary wave interactions with a floating body. For this purpose, a solitary wave with the wave height $H = 0.27D$ and the water depth $D = 0.3$ m is studied. This rectangular body has dimensions with the width $b = 1.0D$ and the height $h = 0.33D$. It is also half-immersed in the water initially. The centroid of the floating box is initially placed at $(11.67D, 1.0D)$. The initial particle spacing $dx = 0.01$ m is adopted, i.e. $D/dx = 30$, in this case. Fig. 15 depicts the interaction process between the solitary wave and floating body computed by ISPH_GNN. As can be observed from Fig. 15(a) that as the wave front approaches, the floating box starts to rotate in clockwise direction. And the floating box reaches its maximum climbing point under the wave action at $\tilde{t} = 18.3$ as shown in Fig. 15(b). Then after the solitary wave propagates over the body, it begins to run down and rotate in anti-clockwise direction in Fig. 15(c). Fig. 16 gives the comparisons of the heave and roll motions of the floating box between ISPH and ISPH_GNN. Generally, the agreement between them is satisfactory.

4.3. Interaction between regular wave and a moored submerged floating breakwater

Further, the ISPH_GNN with f_{GNN} is applied to study the interaction between regular waves and a moored floating breakwater. The corresponding experiments have been carried out by Peng et al. (2013) and the configuration of the numerical simulation is illustrated in Fig. 17. In this case, the water depth $D = 0.6$ m is adopted. The mass and inertia moment of the structure are taken as 42 kg and 0.64 kg m², respectively. The breadth and the height of the breakwater are $b = 0.667D$ and $h = 0.25D$, respectively. The submerged depth of the breakwater (from the upper surface of the body to the mean water depth) is $d = 0.17D$. The mooring lines, which can be treated as the light spring model (Hong et al., 2005; Ren et al., 2017), are attached at lower corners of the body and anchored on the tank bed. To balance the moored body in static water, the pre-tension is added on the mooring line during the numerical simulation. And the mooring force can be estimated by

$$|\mathbf{F}_m| = \begin{cases} |\mathbf{F}_{m0}| + k(l_m - l_{m0}) & |\mathbf{F}_{m0}| > k(l_{m0} - l_m) \\ 0 & |\mathbf{F}_{m0}| > k(l_{m0} - l_m) \end{cases} \quad (32)$$

where $|\mathbf{F}_{m0}|$ is the pretension force added on the mooring line, which equals to the difference between the buoyancy force and the floater weight. $k = 1.2 \times 10^5$ N/m is the elasticity coefficient of mooring line used in this case. l_{m0} and l_m are the initial and transient length of the mooring line, respectively.

Correspondingly, the torque acting on the moored body from the mooring line can be obtained by

$$\mathbf{T}_m = (\mathbf{r}_m - \mathbf{r}_G) \times \mathbf{F}_m \quad (33)$$

where \mathbf{r}_m is the position vector of the joints between the mooring line and the moored body.

The case with wave height $H = 0.0767D$ and wave period $T = 1.0$ s is studied here. The initial particle spacing is taken as 0.01 m, i.e. $D/dx = 60$. The snapshots for the pressure field during the regular wave propagation over the moored floater computed by the ISPH_GNN are given in Fig. 18. As can be seen from Fig. 18(a)–(b), the floater begins to move anti-clockwise as the wave front approaches, and then it returns to the initial position in Fig. 18(c). Then the floating breakwater rotates to the maximum angle in the clockwise direction as shown in Fig. 18(d). It can also be seen that the ISPH_GNN provides quite consistent and smooth pressure fields near the moored floating breakwater at different time instants.

The comparisons between the experimental and numerical results for the sway, heave and roll of the breakwater are shown in Fig. 19. It can be observed from Fig. 19 that the results of the ISPH_GNN agree well with the experimental data, although there is small difference especially at the trough of the heave movement. In addition, Fig. 20 gives the time histories of the mooring force, in which F_s is the mooring force from the mooring line on the seaward side and F_L is the leeside mooring force. The trend of the mooring forces is similar to that of the heave movement. The overall good agreement between the experimental results and the ISPH_GNN results can also be seen in Fig. 20, although ISPH_GNN slightly overestimates the mooring forces as compared with the experimental results. The good agreement between the experimental results and the ISPH_GNN results indicates a promising potential of the ISPH_GNN on modelling the wave-floater interaction problems.

4.4. Green water impact on a floating structure

To further check the performance and robustness of the ISPH_GNN, the case of green water impact on the floating structure as shown in Fig. 21 is studied here. The corresponding physical experiment was carried out by Zheng et al. (2018). The water depth is $D = 0.7$ m and the initial draft of the structure is $0.357D$. The main structure at the bottom is $b = 0.714D$ in breadth and $h = 0.429D$ in height. The superstructure is a square structure with the size $d = 0.286D$. The center of mass of the structure is located at $0.188D$ from the bottom and the moment of inertia

of the structure is 6.531 kg m². A pressure sensor P_1 with a horizontal distance $d_1 = 0.0714D$ from the upper vertical plate is located on the deck to measure the pressure of inside deck. The initial particle spacing $dx = 0.01$ m is adopted, i.e. $D/dx = 70$.

Fig. 22 gives the comparisons of the snapshots of the motion of the floating structure under the green water impact within one wave period obtained by the ISPH_GNN and experiments (Zheng et al., 2018) for the wave height $H = 0.143D$ and wave period $T = 1.0$ s. As can be seen in Fig. 22(a) that when the severe green water occurs, the floater rolls to the wave-coming direction. Then the floater returns to the wave propagation direction as shown in Fig. 22(c). The phenomenon of flow returning from the deck to the water is also observed from the ISPH_GNN simulations in Fig. 22(d). Generally, both the wave surface deformations and floater movement simulated by the ISPH_GNN are quite similar to the experimental photos.

To investigate the accuracy of the ISPH_GNN in pressure prediction, the comparisons of the time histories of the green water impact pressures on the deck from ISPH and ISPH_GNN methods with the experimental data of Zheng et al. (2018) for the wave condition $H = 0.143D$ and $T = 1.2$ s are illustrated in Fig. 23. Taking the experimental result from $t/T = 0$ to $t/T = 5$ as the reference, the errors of the green water impact pressures in Fig. 23 are about 18.7% for ISPH_GNN and 19.5% for ISPH, respectively, which shows that ISPH_GNN and the conventional ISPH have the similar level of accuracy. This also implies that the ISPH_GNN has the potential capability for the simulation of violent wave-floater interactions with the wave breaking and green water impact on deck.

4.5. Computational efficiency

The computational efficiency is one important issue to assess the performance of the ISPH_GNN model for the wave-floater interactions. To investigate the computational efficiency of the ISPH_GNN, the speedup ratio, the computational time per step required for solving PPE in the conventional ISPH to the corresponding computational time of the GNN model for the pressure prediction, is given in Fig. 24. For this purpose, the cases of regular wave interactions with a freely floating box (section 5.2) with different computational configurations (Table 1) are considered. These cases are carried out on the workstation with the main specification as follows: Intel Xeon Platinum 8268 CPU at 2.9 GHz, 256 GB RAM and NVIDIA GeForce RTX 3090 with 24 GB RAM.

As can be seen, the pressure estimation in the conventional ISPH can be significantly speeded up by using the trained GNN model. With the growth of particle numbers, the speed up ratio increase significantly. For example, at 1280k particles, the ISPH_GNN can achieve about 97 times acceleration for the pressure prediction compared with the conventional ISPH. This indicates that the ISPH_GNN method is particularly suitable for large-scale (large number of particles) numerical simulation.

5. Conclusion

This paper presents a new approach for simulating the wave-floater interactions. In this approach, the graphic neural network (GNN) replaces solving PPE for pressure in conventional ISPH method. More significantly, the GNN is trained on datasets obtained for the wave-only cases without any structure or objects involved. The ISPH combined with the GNN trained in such a way is applied to simulating the cases for wave-floater interaction. These cases include the roll decay test of a box, a floating box subjected to a wave, the regular wave interactions with a moored floating body and the violent green water impact on a floating object. The numerical results have been compared with experimental data in most of the cases. Surprising agreement between them has been shown.

In addition, the method can produce similar results to those obtained by using the conventional ISPH with slightly smaller error than that of the conventional ISPH in most of cases if their particle spacing is the same. However, the present method takes much less computational

time, which can be 97 times less, compared to the conventional ISPH at similar level of accuracy for the cases considered. The significance of this paper lies in revealing a fact that the GNN trained using data for simple cases can be applied to more complex cases for wave-floater interactions.

Above conclusions and observations are based only on the cases considered in this paper. They shed very positive light but they need to be confirmed using more different cases. On the basis of the datasets just for regular and solitary waves in this paper, more cases for other waves with different parameters, including broken waves may be added to the training data in future work. In addition, only two-dimensional wave-floater interactions have been studied in this paper. Three-dimensional wave-floater interactions needs to be investigated in future work.

CRedit authorship contribution statement

Ningbo Zhang: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Shiqiang Yan:** Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Qingwei Ma:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Qian Li:** Software, Resources, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors gratefully acknowledge the financial support of EPSRC projects (EP/V040235, EP/T026782 and EP/T00424X).

References

- Belbute-Peres, F.D.A., Economon, T., Kolter, Z., 2020. Combining differentiable PDE solvers and graph neural networks for fluid flow prediction. In: In International Conference on Machine Learning. PMLR, pp. 2402–2411.
- Chen, J., Hachem, E., Viquerat, J., 2021. Graph neural networks for laminar flow prediction around random two-dimensional shapes. *Phys. Fluids* 33 (12).
- Dong, W., Liu, J., Xie, Z., Li, D., 2019. Adaptive neural network-based approximation to accelerate eulerian fluid simulation. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–22.
- Fourtakas, G., Rogers, B.D., Nasar, A.M., 2021. Towards pseudo-spectral incompressible smoothed particle hydrodynamics (ISPH). *Comput. Phys. Commun.* 266, 108028.
- Gao, R., Jaiman, R., 2022. Quasi-monolithic graph neural network for fluid-structure interaction. arXiv preprint arXiv:2210.04193.
- Gori, M., Monfardini, G., Scarselli, F., 2005. A new model for learning in graph domains. In: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, vol. 2. IEEE, pp. 729–734.
- Gotoh, H., Shao, S., Memita, T., 2004. SPH-LES model for numerical investigation of wave interaction with partially immersed breakwater. *Coast Eng. J.* 46 (1), 39–63.
- He, F., Zhang, H., Huang, C., Liu, M., 2020. Numerical investigation of the solitary wave breaking over a slope by using the finite particle method. *Coast Eng.* 156, 103617.
- He, F., Zhang, Y., Jiang, H., Huang, C., 2023. Numerical investigation of solitary wave breaking over a slope based on multi-phase smoothed particle hydrodynamics. *Phys. Fluids* 35 (2).
- Hong, S.Y., Kim, J.H., Cho, S.K., Choi, Y.R., Kim, Y.S., 2005. Numerical and experimental study on hydrodynamic interaction of side-by-side moored multiple vessels. *Ocean Eng.* 32 (7), 783–801.
- Khayyer, A., Gotoh, H., Falahaty, H., Shimizu, Y., 2018. An enhanced ISPH-SPH coupled method for simulation of incompressible fluid-elastic structure interactions. *Comput. Phys. Commun.* 232, 139–164.
- Khayyer, A., Shimizu, Y., Gotoh, H., Hattori, S., 2021. Multi-resolution ISPH-SPH for accurate and efficient simulation of hydroelastic fluid-structure interactions in ocean engineering. *Ocean Eng.* 226, 108652.
- Kumar, K., Vantassel, J., 2022. GNS: a generalizable Graph Neural Network-based simulator for particulate and fluid modeling. arXiv preprint arXiv:2211.10228.
- Lee, J.J., Skjelbreia, J.E., Raichlen, F., 1982. Measurement of velocities in solitary waves. *J. Waterw. Port. Coast. Ocean Div.* 108 (2), 200–218.
- Li, Z., Farimani, A.B., 2022. Graph neural network-accelerated Lagrangian fluid simulation. *Comput. Graph.* 103, 201–211.
- Ling, J., Kurzawski, A., Templeton, J., 2016. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* 807, 155–166.
- Lino, M., Fotiadis, S., Bharath, A.A., Cantwell, C.D., 2022. Multi-scale rotation-equivariant graph neural networks for unsteady Eulerian fluid dynamics. *Phys. Fluids* 34 (8).
- Luo, M., Khayyer, A., Lin, P., 2021. Particle methods in ocean and coastal engineering. *Appl. Ocean Res.* 114, 102734.
- Ma, Q.W., Zhou, J., 2009. MLPG_R method for numerical simulation of 2D breaking waves. *Comput. Model. Eng. Sci.: Comput. Model. Eng. Sci.* 43 (3), 277–304.
- Ma, Q.W., Zhou, Y., Yan, S., 2016. A review on approaches to solving Poisson's equation in projection-based meshless methods for modelling strongly nonlinear water waves. *J. Ocean Eng. Marine Energy* 2, 279–299.
- Madsen, O.S., 1971. On the generation of long waves. *J. Geophys. Res.* 76 (36), 8672–8683.
- Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., Battaglia, P.W., 2020. Learning mesh-based simulation with graph networks. arXiv preprint arXiv:2010.03409.
- Peng, W., Lee, K.H., Shin, S.H., Mizutani, N., 2013. Numerical simulation of interactions between water waves and inclined-moored submerged floating breakwaters. *Coast Eng.* 82, 76–87.
- Ren, B., He, M., Dong, P., Wen, H., 2015. Nonlinear simulations of wave-induced motions of a freely floating body using WCSPH method. *Appl. Ocean Res.* 50, 1–12.
- Ren, B., He, M., Li, Y., Dong, P., 2017. Application of smoothed particle hydrodynamics for modeling the wave-moored floating breakwater interaction. *Appl. Ocean Res.* 67, 277–290.
- Shao, S.D., 2010. Incompressible SPH flow model for wave interactions with porous media. *Coast Eng.* 57 (3), 304–316.
- Shao, S.D., Gotoh, H., 2004. Simulating coupled motion of progressive wave and floating curtain wall by SPH-LES model. *Coast Eng. J.* 46 (2), 171–202.
- Sun, P.N., Ming, F.R., Zhang, A.M., 2015. Numerical simulation of interactions between free surface and rigid body using a robust SPH method. *Ocean Eng.* 98, 32–49.
- Tompson, J., Schlachter, K., Sprechmann, P., Perlin, K., 2017. Accelerating eulerian fluid simulation with convolutional networks. In: International Conference on Machine Learning. PMLR, pp. 3424–3433.
- Tsuruta, N., Khayyer, A., Gotoh, H., 2023. Development of Advective Dynamic Stabilization scheme for ISPH simulations of free-surface fluid flows. *Comput. Fluid* 266, 106048.
- Wei, G., Kirby, J.T., 1995. Time-dependent numerical code for extended Boussinesq equations. *J. Waterw. Port. Coast. Ocean Eng.* 121 (5), 251–261.
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y., 2020. A comprehensive survey on graph neural networks. *IEEE Transact. Neural Networks Learn. Syst.* 32 (1), 4–24.
- Xu, R., Stansby, P., Laurence, D., 2009. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *J. Comput. Phys.* 228 (18), 6703–6725.
- Yan, S., Ma, Q.W., 2007. Numerical simulation of fully nonlinear interaction between steep waves and 2D floating bodies using the QALE-FEM method. *J. Comput. Phys.* 221 (2), 666–692.
- Yan, S.Q., Ma, Q.W., Wang, J.H., 2020. Quadric SFDI for laplacian discretisation in Lagrangian meshless methods. *J. Mar. Sci. Appl.* 19, 362–380.
- Zhang, C., Rezavand, M., Hu, X., 2021. A multi-resolution SPH method for fluid-structure interactions. *J. Comput. Phys.* 429, 110028.
- Zhang, Z.J., Duraisamy, K., 2015. Machine learning methods for data-driven turbulence modeling. In: 22nd AIAA Computational Fluid Dynamics Conference, p. 2460.
- Zhang, N.B., Zheng, X., Ma, Q.W., Duan, W.Y., Khayyer, A., Lv, X.P., Shao, S.D., 2018. A hybrid stabilization technique for simulating water wave-structure interaction by incompressible smoothed particle hydrodynamics (ISPH) method. *J. Hydro-Environ. Res.* 18, 77–94.
- Zhang, N.B., Yan, S.Q., Ma, Q.W., Zheng, X., 2021. A QSFDI based Laplacian discretisation for modelling wave-structure interaction using ISPH. *Appl. Ocean Res.* 117, 102954.
- Zhang, N.B., Ma, Q.W., Zheng, X., Yan, S.Q., 2023a. A two-way coupling method for simulating wave-induced breakup of ice floes based on SPH. *J. Comput. Phys.* 488, 112185.
- Zhang, N.B., Yan, S.Q., Ma, Q.W., Guo, X.H., Xie, Z.H., Zheng, X., 2023b. A CNN-supported Lagrangian ISPH model for free surface flow. *Appl. Ocean Res.* 136, 103587.
- Zhang, N.B., Yan, S.Q., Ma, Q.W., Khayyer, A., Guo, X.H., Zheng, X., 2024a. A consistent second order ISPH for free surface flow. *Comput. Fluid* 274, 106224.
- Zhang, N.B., Yan, S.Q., Ma, Q.W., Li, Q., 2024b. A hybrid method combining ISPH with graph neural network for simulating free-surface flows. *Comput. Phys. Commun.* 301, 109220.
- Zheng, X., Lv, X.P., Ma, Q.W., Duan, W.Y., Khayyer, A., Shao, S., 2018. An improved solid boundary treatment for wave-float interactions using ISPH method. *Int. J. Nav. Archit. Ocean Eng.* 10 (3), 329–347.
- Zheng, X., Ma, Q.W., Duan, W.Y., 2014. Incompressible SPH method based on Rankine source solution for violent water wave simulation. *J. Comput. Phys.* 276, 291–314.