Heliyon 10 (2024) e25187

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Physically-based simulation for oil leakage and diffusion on river using heterogeneous graph attention network

Yuanfeng Lian [a,b,*], Hanzhao Gao [a], Lianen Ji [a,b], Shaohua Dong [c]

[a] Department of Computer Science and Technology, China University of Petroleum, Beijing, 102249, China
[b] Beijing Key Laboratory of Petroleum Data Mining, Beijing, 102249, China
[c] National Engineering Laboratory for Pipeline Safety, China University of Petroleum, Beijing, 102249, China

ARTICLE INFO

Keywords:
Heterogeneous graph attention network
Fluid simulation
Multiphase flow
Smoothed particle hydrodynamics
Mixture model

ABSTRACT

Once the oil pipeline leakage accident occurs on the river, the simulation of the leakage diffusion range is of great significance for the designation of emergency rescue plans. The existing methods cannot show the precise leakage diffusion process consistent with the physical law for crude oil on the river and the simulation suffers high run-time complexity. This paper proposed a two-phase leakage simulation for oil and water combined with the physical process of smoothed particle hydrodynamics (SPH) and graph attention network. A new and efficient method—Mixture Tension Divergence-Free SPH (MTDF-SPH)—that the mixture model and the surface tension model are introduced to the divergence-free smoothed particle hydrodynamics (DFSPH) for simulating the mixing and decomposition effects of immiscible phases. To further accelerate the leakage diffusion process, we design a physics-aware heterogeneous graph attention network (PAGATNet), based on Attention Graph Network Block (AGNB) and Feature-Response Knowledge Distillation (FRKD) to enhance the network's ability for extracting the particle features of physical properties. The experimental results on different test cases show the accuracy, robustness and effectiveness of our method than those of the state-of-the-art in two-phase leakage simulation of crude oil on the river.

1. Introduction

The physically-based multiple-fluid simulation has essential research value in various areas, such as film special effects, industrial production, disaster simulation, etc. With the development of oil exploration and transportation, long-distance crude oil pipelines usually need to cross the rivers. These pipelines can leak out easily due to the scour and erosion of the river water. The leakage will flow down the river quickly and cause heavy damage to the biological and natural environment in a short time [1][2]. Even though some efforts have been made [3][4][5][6][7][8][9], the task of solving the simulating for oil leakage and diffusion on river accurately and efficiently is still very challenging.

Over the past decades, the Lagrangian particle methods have received considerable attention in fluid simulation. Maslo et al. [10] proposed the lattice Boltzmann method in large-scale oil spill modeling to predict the location, the concentration and the stranded area of crude oil at sea, which was validated against satellite image data of crude oil spills. Dhavalikar and Choudhari [11] predicted oil spill trajectories on the sea surface using a random walk technique. Molteni et al. [12] applied SPH numerical methods to observe

* Corresponding author at: Department of Computer Science and Technology, China University of Petroleum, Beijing, 102249, China.
E-mail address: lianyuanfeng@cup.edu.cn (Y. Lian).

https://doi.org/10.1016/j.heliyon.2024.e25187
Received 16 February 2023; Received in revised form 13 January 2024; Accepted 22 January 2024

Available online 26 January 2024

wave-drop coupling and to study the 'walking droplet' phenomenon in crude oil. Moreover, the Lagrangian method provided a new perspective for studying crude oil diffusion on the water surface. Soussi et al. [13] introduced an offshore oil spill trajectory dynamics simulation model based on the Lagrangian method to simulate the spread of oil spill trajectories on the sea. By studying the mechanism of oil leakage due to turbulent production and buoyancy, Violeau et al. [14] predicted the boom and oil spill motion in wave flume.

Some recent efforts mainly focus on the smoothed particle hydrodynamic method due to the Lagrangian nature, which gives it a unique advantage in multiphase flow simulations. Becker and Teschner [15] developed the weakly compressible SPH method (WCSPH), which calculated the position of particles by introducing a surface tension model. Solenthaler and Pajarola [16] introduced the Pairwise Force SPH model (PF-SPH) to simulate two-phase and three-phase flows by adding pairwise forces to the Navier-Stokes equations. Macklin and Müller [17] presented the Position Based Fluids model (PBF) to improve simulation accuracy by formulating and solving for a set of position constraints for the constant density. Ihmsen et al. [18] generalized the Implicit Incompressible SPH method (IISPH), which could accelerate the simulation of large scenarios of fluids by relating density to pressure. Bender and Koschier [19] proposed the Divergence-free SPH (DFSPH) approach, which solved the fluid compressibility problem by adding the divergence-free and constant density solver. Despite the recent success of several SPH methods for fluid simulation, those models cannot achieve robust and efficient results due to not considering the physical law for crude oil on the river.

For the multiphase fluid simulation problems, the SPH methods could be summarized as miscibility and immiscibility. Following the SPH immiscible fluids simulation approach, Premžoe et al. [20] presented the Moving-Particle Semi-Implicit method (MPS) to solve Navier-Stokes equations. Akinci et al. [21] used the momentum-conserving two-way coupling method of SPH to deal with immiscible fluid. Furthermore, Müller et al. [22] introduced the concept of volume fraction, which represented the spatial distribution of different phases. Liu et al. [23] improved the volume fraction optimization SPH method to simulate multi-fluid flow. Moreover, Ren et al. [24] presented a versatile and robust SPH simulation approach for multiple-fluid flows using an improved mixture model. Some recent works [25] [26][27][28] introduced a generalized simulation scheme for multiple fluid permeable porous materials. In this paper, the crude oil leakage diffusion on the river is regarded as a two-phase immiscible flow (oil and water) process with complex interfaces. Different from the above works, our method for two-phase fluid embedded the mixture model based on DFSPH and the surface tension constraint.

Recent researches on deep learning methods have enabled the development to solve fluid simulation problems. An end-to-end neural network FluidsNet for Lagrangian fluid simulation was presented by Zhang et al. [29], directly extracting fluid particle features to predict the result and avoiding the limitations of artificially designed features. Methods [30][31][32] that can improve prediction precision, used the physically-based module to capture complex topological position of particles and general physical relationships. Um et al. [33] proposed the fluid generation method to improve the visual fidelity of hybrid particle fluid simulations, which learned the physical laws for simulating small-scale fluids precisely. Battaglia et al. [34] attempted to construct a general-purpose framework that could dynamically predict object interaction in complex systems and infer the state of particle motion in rigid-body and non-rigid-body dynamics. Haghighat et al. [35] proposed the PINN method to simulate the coupling and deformation effects of single-phase and multi-phase flows in porous media. Ummenhofer et al. [36] presented a Lagrangian fluid simulation method with convolutional networks, which extended N-D convolution to the continuous domain for simulating arbitrary effects of different materials with improved accuracy and speed. Some approaches also used graph neural networks for fluid simulation. Tompson et al. [37] represented the physical states of particles based on the graph network and calculated the particle states based on the message-passing mechanism. It can build the interrelationships between the fluid and other materials. The Fluid Graph Networks (FGN) [38] used the message-passing mechanism to transform the information, which had two accurate, stable and computationally efficient graph neural network structures. Even though the deep learning methods for fluid simulation have been improved, the models are still not focused on the attention mechanism for refining the model. In this work, we proposed a heterogeneous graph attention network to represent the relationship between different phase particles. Moreover, the physics-aware module transfers the teacher network features into the student network for fulfilling the knowledge distillation.

Our main contributions can be summarized as follows. We proposed a method called MTDF-SPH to simulate the diffusion of crude oil on the river surface. Hence, the specific simulation process conforms to the physical laws of immiscible multiphase flow diffusion. Secondly, in order to obtain more powerful simulation properties, we designed a physics-aware graph attention network (PAGATNet) based on a heterogeneous graph attention mechanism. Besides, we trained the PAGATNet using the data set generated by the MTDF-SPH method and optimized it by using knowledge distillation to accelerate the simulation speed. Finally, the experiments on the accuracy, robustness and efficiency with different test cases have shown good improvements.

The paper is organized as follows. Section 2 presents an improved Lagrangian two-phase flow simulation method, MTDF-SPH. Then, Section 3 presented a physics-aware heterogeneous graph attention network based on PAGATNet. Experiments and analysis are carried out in section 4. Section 5 discusses and analyzes the stability and accuracy of the computational findings and our future work.

2. MTDF-SPH

In this section, we constructed a divergence-free simulation method called "Mixture Tension Divergence-Free SPH" (MTDF-SPH) that incorporates the mixture model and surface tension based on the framework of smoothed particle hydrodynamics. We believe that MTDF-SPH allows a powerful simulation of the fluids' phases for the directions and positions.

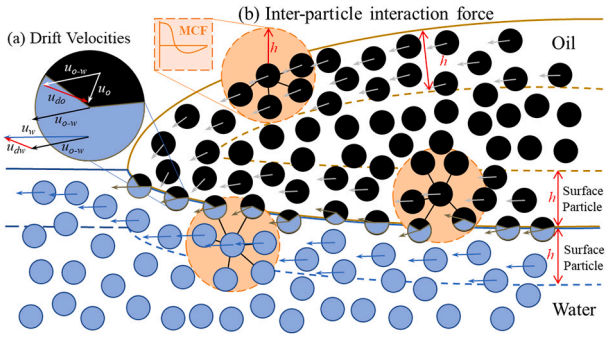

Fig. 1. Schematic diagram of the MTDF-SPH for oil-water surface (black filled circle is oil particle, black-blue filled circle is the two-phase mixture particle, blue filled circle is water particle). (a) The two-phase mixture particle has two phases with unit density. The drift velocities u_{do}, u_{dw} are obtained from the mixture velocity u_{o-w} and the velocities of each phase u_o and u_w. (b) The sum of the inter-particle forces is non-zero for surface particles which turns out to be the surface tension, such as the particle between the solid line and the broken line in the diagram. The sum of the inter-particle forces among the inner particles is zero. The inter-particle force is modified by Modified Cosine Force (MCF) to make the force function smoother, which is repulsive at the short distance and attractive at the long distance.

2.1. Lagrangian fluid simulation

In the mathematical theory of multiple fluid, the continuity equation (Eq. (1)) for the oil and the water is:

$$\left|\frac{\partial}{\partial t}\left(\alpha_o\rho_o\right)+\nabla\cdot\left(u_o\alpha_o\rho_o\right)\right|+\left|\frac{\partial}{\partial t}\left(\alpha_w\rho_w\right)+\nabla\cdot\left(u_w\alpha_w\rho_w\right)\right|=0 \tag{1}$$

And, the momentum equation (Eq. (2)) is:

$$\left|\begin{array}{l}\frac{\partial}{\partial t}\left(u_o\alpha_o\rho_o\right)+\nabla\cdot\left(u_ou_o\alpha_o\rho_o\right)-\\ g\alpha_o\rho_o+\alpha_o\nabla p_o-\nabla\cdot\left(\alpha_oT_o\right)-F_o\end{array}\right|+\left|\begin{array}{l}\frac{\partial}{\partial t}\left(u_w\alpha_w\rho_w\right)+\nabla\cdot\left(u_wu_w\alpha_w\rho_w\right)-\\ g\alpha_w\rho_w+\alpha_w\nabla p_w-\nabla\cdot\left(\alpha_wT_w\right)-F_w\end{array}\right|=0 \tag{2}$$

where subscript o and w indicates the phases of oil and water, p is the pressure, g is the external force, T is the viscous stress tensor, F is the source of momentum at the interface. The fluid phase is represented by its volume fraction α, velocity u, and density ρ. The volume fractions α_w and α_o are bounded between 0 and 1, and they must add up to 1. The continuum and momentum equations (Eqs. (3)-(9)) for the mixture model can be expressed as:

$$\frac{\partial}{\partial t}\rho_{o-w}+\nabla\cdot\left(\rho_{o-w}u_{o-w}\right)=0 \tag{3}$$

$$u_{o-w}=\frac{1}{\rho_{o-w}}\left(\alpha_o\rho_ou_o+\alpha_w\rho_wu_w\right) \tag{4}$$

$$\rho_{o-w}=\alpha_o\rho_o+\alpha_w\rho_w \tag{5}$$

$$p_{o-w}=\alpha_op_o+\alpha_wp_w \tag{6}$$

$$\frac{\partial}{\partial t}u_{o-w}+\left(u_{o-w}\cdot\nabla\right)u_{o-w}=-\frac{\nabla p_{o-w}}{\rho_{o-w}}+g+\frac{\nabla\cdot\left(\alpha_oT_o+\alpha_wT_w\right)}{\rho_{o-w}}-\frac{\nabla\cdot\left(\alpha_o\rho_ou_{do}u_{do}+\alpha_w\rho_wu_{dw}u_{dw}\right)}{\rho_{o-w}} \tag{7}$$

$$u_{do}=u_o-u_{o-w} \tag{8}$$

$$u_{dw}=u_w-u_{o-w} \tag{9}$$

where ρ_{o-w} and p_{o-w} are pressure and density for the mixture of oil and water, T_o and T_w are the viscous stress tensor of oil and water, u_{o-w} is the mixture velocity, u_{do} and u_{dw} are the drift velocities of the water phase and the oil phase respectively, as shown in Fig. 1(a).

2.2. Surface tension model based on MCF

Surface tension is one of the most important physical properties in liquid-liquid two-phase simulations. The SPH methods for simulating immiscible two-phase fluids (e.g., water and oil) can result in heterogeneous particle stagnation due to strong numerical oscillations at the intersection. To address this problem, we use a surface tension model based on Modified Cosine Force (MCF) to deal with the oil-water surface tension [39]. In the MCF, the inter-particle force (Eq. (10)) between the particles i and j is defined as:

$$\frac{dv_i^c}{dt}=-\frac{1}{m_i}F_i^c \tag{10}$$

where v_i^c is the velocity component generated by the tension on the particle, and F_i^c is the net force of the surface particles from neighboring particles, the inter-particle interaction force equation (Eq. (11)) is defined as:

$$F_i^c = \begin{cases} \sum_j s_{ij} \cos\left(\frac{3\pi}{2h}\left|r_{ij}\right|\right) \frac{r_{ij}}{\left|r_{ij}\right|} W_{ij}, \left|r_{ij}\right| \le h \\ 0, \left|r_{ij}\right| \ge h \end{cases} \tag{11}$$

where s_{ij} is the strength factor of the interaction force between the particles, r_{ij} is the distance of particle i and j, x is the coordinate, h is the nucleus radius of the particle and $W_{ij} = W\left(x_i - x_j, h\right)$ is a smoothing kernel function with support radius h, as illustrated in Fig. 1(b).

2.3. MTDF-SPH solver

For enforcing the condition $d\rho/dt = 0$, the solver will determine a set of pressures for the particle and its neighbors to correct the divergence error. The pressure of particle i is determined by the following equation (Eq. (12)):

$$F_i^p = -\frac{m_i}{\rho_i} \nabla p_i \tag{12}$$

where the mixture pressure gradient is calculated by means of the differential equations (Eqs. (13)-(15)), as follows:

$$\nabla p_i = \varepsilon \nabla \rho_i = \varepsilon \left(\sum_j m_j \frac{p_i + p_j}{2\bar{\rho}_j} \nabla W_{ij} \right) \tag{13}$$

$$F_i^p = -\frac{m_i}{\rho_i} \varepsilon_i \left(\sum_j m_j \frac{p_i + p_j}{2\bar{\rho}_j} \nabla W_{ij} \right) \tag{14}$$

$$\bar{\rho}_j = \sum_j m_i \nabla W_{ij} \tag{15}$$

where p_i is the pressure at particle i, $\bar{\rho}_j$ is the interpolated density of particle j [40], ε is the stiffness parameter. In addition, considering the pressure of particle i acting on the neighboring particle j, there is:

$$F_{j \leftarrow i}^p = \frac{m_i}{\rho_i} \varepsilon m_j \frac{p_{mi} + p_{mj}}{2\bar{\rho}_j} \nabla W_{ij} \tag{16}$$

The change rate in the density of particle i is calculated as follows:

$$\frac{d\rho_i}{dt} = \sum_j m_j \left(v_i - v_j\right) \nabla W_{ij} \tag{17}$$

$$\Delta v_i = \Delta t \frac{F_i^p}{m_i} \tag{18}$$

$$\Delta v_j = \Delta t \frac{F_{j \leftarrow i}^p}{m_j} \tag{19}$$

Substitute Eq. (18) and Eq. (19) into Eq. (17), we can obtain:

$$\frac{d\rho_i}{dt} = -\Delta t \sum_j m_j \left(\frac{F_i^p}{m_i} - \frac{F_{j \leftarrow i}^p}{m_i} \right) \nabla W_{ij} \tag{20}$$

Substitute Eq. (13) and Eq. (16) into Eq. (20), we can obtain Eqs. (21)-(23) for the stiffness parameter ε as follows:

$$\varepsilon = \frac{1}{\Delta t} \cdot \frac{d\rho_i}{dt} \cdot \frac{\rho_i}{\left|\sum_j m_j \nabla W_{ij}\right|^2 + \sum_j \left|m_j \nabla W_{ij}\right|^2} \cdot \frac{2\bar{\rho}_j}{p_i + p_j} \tag{21}$$

$$\vartheta_i = \frac{\rho_i}{\left|\sum_j m_j \nabla W_{ij}\right|^2 + \sum_j \left|m_j \nabla W_{ij}\right|^2} \tag{22}$$

$$\tau_i = \frac{2\bar{\rho}_j}{p_i + p_j} \tag{23}$$

where ϑ_i depends on the current position, τ_i is the mixture parameter. If we compute pressure forces with the resulting parameter, the condition $D\rho/Dt = 0$ will be fulfilled. Nevertheless, the stiffness parameters of neighboring particles depend on each other, and iterative calculations are required. In the special case where particle i has fewer neighboring particles, the denominator ϑ_i may be unstable. We clamp the denominator if ϑ_i gets smaller than 10^{-6} in order to solve this problem. Finally, the net force of the pressure on particle i is:

$$F_{i,\,\text{total}}^p = F_i^p + \sum_j F_{i \leftarrow j}^p = -\frac{m_i}{2} \sum_j m_j \left(p_i + p_j\right) \left(\frac{\varepsilon_i}{\rho_i \bar{\rho}_j} + \frac{\varepsilon_j}{\rho_j \bar{\rho}_i} \right) \nabla W_{ij} \tag{24}$$

where $F^{p}_{i\leftarrow i}$ denotes the pressure of neighboring particle j on particle i. The dispersion-free solution algorithm uses parallel Jacobi iterative solutions and is described as Algorithm 1.

Algorithm 1: Mixture divergence-free Algorithm.

Input: The velocity v_i^*, parameter ϑ_i, parameter τ_i
Output: The velocity v_i^*
1: Initialize threshold η
2: **while** $\left(\left(\frac{d\rho}{dt}\right)_{\text{avg}} > \eta\right) \vee (\text{iterative} < 1)$ **do**
3: **for** i in particles **do**
4: $\varepsilon_i = \frac{1}{\Delta t}\frac{d\rho_i}{dt}\vartheta_i\tau_i$
5: $\varepsilon_j = \frac{1}{\Delta t}\frac{d\rho_j}{dt}\vartheta_j\tau_j$
6: $v_i^* = v_i^* - \frac{\Delta t}{2}\sum_j m_j\left(p_i + p_j\right)\left(\frac{\varepsilon_i}{\rho_i\tilde{\rho}_j} + \frac{\varepsilon_j}{\rho_j\tilde{\rho}_i}\right)\nabla W_{ij}$
7: **end for**
8: **end while**
9: **return** v_i^*

The constant density solver can minimize the deviation $\rho - \rho_0$, which is similar to the dispersion-free solver. The density is calculated by the Euler integral in Eq. (25):

$$\rho_i' = \rho_i + \Delta t\sum_j m_j\Delta v_{i-j}^*\nabla W_{ij} \tag{25}$$

where $\Delta v_{i-j}^* = v_i^* - v_j^*$. To correct the density error of the pressure $\Delta\rho_i' = \rho_i' - \rho_0$, there are Eqs. (26) and (27):

$$\Delta\rho_i' = \Delta t^2\sum_j m_j\left(\frac{F_i^p}{m_i} - \frac{F^p_{j\leftarrow i}}{m_j}\right)\nabla W_{ij} \tag{26}$$

$$\varepsilon_i = \frac{1}{\Delta t^2}\Delta\rho_i'\cdot\vartheta_i\cdot\tau_i \tag{27}$$

The constant density solution algorithm is described as Algorithm 2.

Algorithm 2: Mixture constant density Algorithm.

Input: The velocity v_i^*, parameter ϑ_i, parameter τ_i
Output: The velocity v_i^*
1: Initialize threshold η
2: **while** $\left(\rho_{\text{ag}} - \rho_0 > \eta\right) \vee (\text{iterative} < 2)$ **do**
3: **for** i in particles **do**
4: **compute** $\Delta\rho'$
5: $\varepsilon_i = \frac{1}{\Delta t^2}\Delta\rho_i'\cdot\vartheta_i\cdot\tau_i$
6: $\varepsilon_j = \frac{1}{\Delta t^2}\Delta\rho_j'\cdot\vartheta_j\cdot\tau_j$
7: $v_i^* := v_i^* - \frac{\Delta t}{2}\sum_j m_j\left(p_i + p_j\right)\left(\frac{\varepsilon_i}{\rho_i\tilde{\rho}_j} + \frac{\varepsilon_j}{\rho_j\tilde{\rho}_i}\right)\nabla W_{ij}$
8: **end for**
9: **end while**
10: **return** v_i^*

2.4. MTDF-SPH simulation

Based on the SPH method, the value of x_i is calculated from the positions of its neighboring particles, and the density ρ_i (Eq. (28)) at x_i can be expressed as:

$$\rho_i = \sum_j m_j W_{ij} \tag{28}$$

where m_j is the mass of particle j, $W_{ij} = W\left(x_i - x_j, h\right)$ is the Gaussian-like kernel function and the radius is h. After calculating the density, the pressure field (Eq. (29)) of the fluid is defined as follows:

$$p_i = \frac{\varepsilon_i\rho_0}{\xi}\left(\left(\frac{\rho_i}{\rho_0}\right)^{\xi} - 1\right) \tag{29}$$

Among them, based on the study by Ren et al. [24], we introduce the mixture constraint to DFSPH for considering the effect of phase volume fraction when calculating the pressure. As Eq. (24), the pressure (Eq. (30)) is as follows:

$$F^{p}_{i,\text{ total}} = -\frac{m_i}{2}\sum_j m_j\left(p_i + p_j\right)\left(\frac{\varepsilon_i}{\rho_i\bar{\rho}_j} + \frac{\varepsilon_j}{\rho_j\bar{\rho}_i}\right)\nabla W_{ij} \tag{30}$$

Inspired by Aly et al. [39], we introduce a modified surface tension term into our calculations. In this approach, the effect of surface tension is taken into account by calculating the combined forces on the particles. The formula (Eq. (31)) for surface tension is defined as follows:

$$F^{c}_{i} = \sum_j s_{ij}\cos\left(\frac{3\pi}{2h}\left|r_{ij}\right|\right)\frac{r_{ij}}{\left|r_{ij}\right|}W_{ij} \tag{31}$$

The paired particle-particle interaction forces [41] (Eq. (32)) and the equation (Eq. (33)) of particle motion are as follows:

$$F_{i,\text{ total}} = F^{p}_{i,\text{ total}} + g\left(\alpha_o\rho_o + \alpha_w\rho_w\right) + \nabla\cdot\left(\alpha_o T_o + \alpha_w T_w\right) - \nabla\cdot\left(\alpha_o\rho_o u_{do}u_{do} + \alpha_w\rho_w u_{dw}u_{dw}\right) \tag{32}$$

$$\frac{d\left(m_i v_i\right)}{dt} = F_{i,\text{ total}} + F^{c}_{i} \tag{33}$$

where $F_{i,\text{ total}}$ is the total force on particle i, F^{c}_{i} is the surface tension. The inter-particle forces are calculated for all surface particles. This can ensure the computational efficiency and the numerical stability.

The simulation process of the method is described in Algorithm 3. Each iteration of the simulation algorithm has four main steps: The first step searches particle neighborhoods and initializes the densities and the factors. The second step calculates the values of all non-pressure forces F^{adv}, such as gravity, surface tension and viscosity. It adjusts the time step according to the Courant-Friedrich-Levy (CFL) condition $\Delta t \le \Phi/\left\|v_{\max}\right\|$, which Φ is the particle diameter and $v_{\max}$ is the maximum particle velocity. In line 11, we calculate the predicted velocity v^{*}_{i} by using the non-pressure forces. The third step uses a constant density solver to calculate the particle's current position and velocity value. The constant density solver uses the predicted velocity and the pre-calculated factors ϑ_i and τ_i to determine the pressure in each neighborhood for correcting the density errors. The fourth step uses a dispersion-free solver to calculate the velocity of the particle so that it satisfies the no-dispersion state condition. Both solvers use the same ϑ_i and τ_i factors which can significantly reduce the computational effort. Then, the neighborhood N_i, the density ρ_i and the factors ε_i should be updated. In line 18, the dispersion-free solver calculation has to satisfy the pressure $D\rho/Dt = 0$, so the velocity field is divergence-free. Finally, the particle updates the position by using the velocity v^{*}_{i}.

Algorithm 3: Fluid Simulation Algorithm.

Input: The physical properties of all particles
Output: The velocity v^{*}_{i}
1: **for** i in particles **do**
2: particles $N_i(0)$ within radius h
3: init densities $\rho_i(0)$, parameter $\vartheta_i(0)$ and parameter $\tau_i(0)$
4: **end for**
5: **while** $(t < T)$ **do**
6: **for** i in particles **do**
7: $F^{adv}_{i}(t) = F^{\text{gravity}}_{i} + F^{\text{viscosity}}_{i} + F^{\text{surface tension}}_{i}$
8: **end for**
9: **for** i in particles **do**
10: $\Delta t \le \Phi/\left\|v_{\max}\right\|$
11: $v^{*}_{i} = v^{*}_{i} + \Delta t F^{adv}_{i}/m_i$
12: input $v^{*}_{i}, \vartheta_i, \tau_i$ to **Algorithm 2**
13: update positions $x_i(t+\Delta t) = x_i(t) + \Delta t v^{*}_{\text{i}}$
14: **end for**
15: **for** i in particles **do**
16: particles $N_i(t+\Delta t)$ within radius h
17: update $\rho_i(t+\Delta t)$, $\vartheta_i(t+\Delta t)$ and $\tau_i(t+\Delta t)$
18: input $v^{*}_{i}, \vartheta_i(t+\Delta t), \tau_i(t+\Delta t)$ to **Algorithm 1**
19: update velocities $v^{*}_{i} = v_i t + v_i \Delta t$
20: **end for**
21: **end while**
22: **return** v^{*}_{i}

3. PAGATNet design

Inspired by the graph network-based simulators [42], a new network architecture for crude oil diffusion simulation, "Physics-Aware Graph Attention Network", called PAGATNet, is proposed to extract features from oil and water phases for obtaining more detailed feature information. Moreover, the knowledge distillation is designed to efficiently improve the robustness and convergence of the network.

We design three types of different attention modules to construct the architecture of AGNB, which can extract different types of force features for the multiple-fluid particles. However, relying only on river water or crude oil pipelines without the topology and

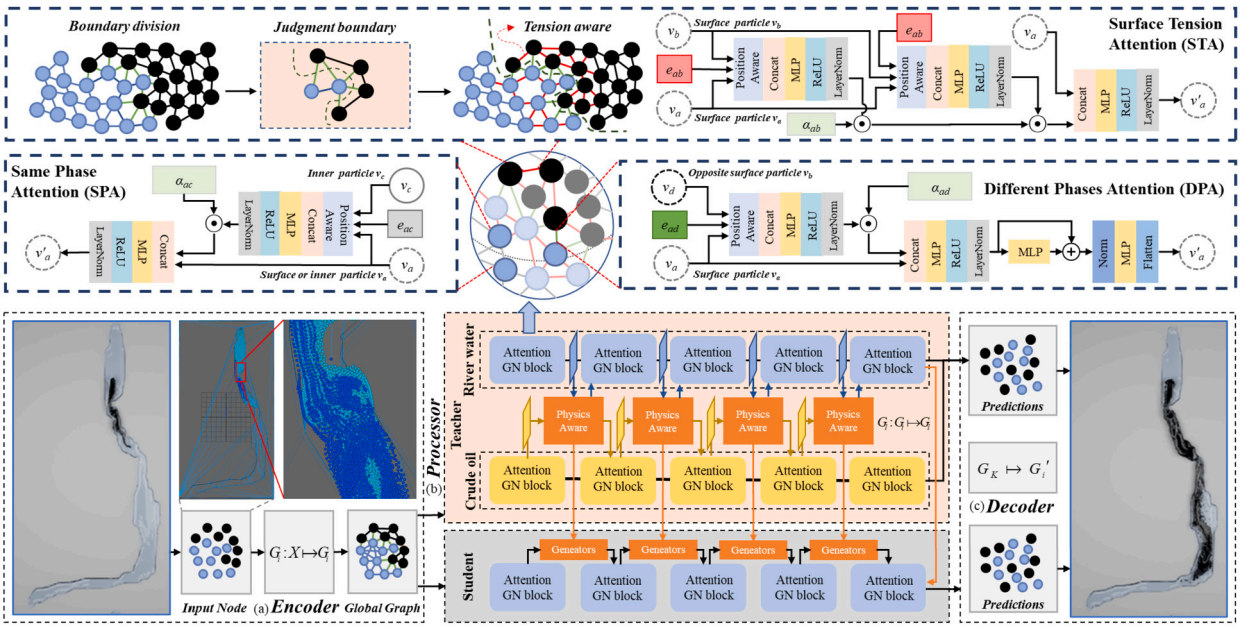

Fig. 2. The framework of our proposed network PAGATNet consists of three components: Encoder, Processor and Decoder. (a) The Encoder gets particles' state vector X and constructs graph G_i. (b) The Processor contains the teacher network and the student network which have triple attention mechanisms of different attention mechanisms between different particle-particle relations. These particle-particle relations can be divided into three groups. The first group contains oil surface-surface and water surface-surface relations. The second group contains oil surface-inner, oil inner-inner, water surface-inner, water inner-inner relations. The third group contains oil surface-water surface relation. Position-aware and physics-aware are used in different attention mechanisms. The teacher network is connected to the student network through knowledge distillation to guide the training of the student network. (c) The Decoder acquires the network output's graph to compute the particles' subsequent positions.

positioning information of water/oil nodes makes the processor module less accurate in simulation. Therefore, we combine the AGNB with physics-aware modules aiming at solving the problem. Another two challenges we address in this paper are model compression and simulation acceleration. Although the teacher model can accurately describe the process of multiple-fluid interaction, it cannot guarantee lightweight and efficiency. Thus, we propose the feature-response knowledge distillation (FRKD) to compress the structure and fine-tune the model.

As shown in Fig. 2, the MTDF-SPH simulator is used to generate the dataset which trained the network to predict the future position and velocity. The neighboring particles can be the same or different kinds of particles, and the number of neighboring particles is not fixed.

3.1. Simulation on PAGATNet

Our PAGATNet consists of three main steps-Encoder, Processor and Decoder.

Encoder definition. The Encoder applies physical dynamics to get trajectory of states $X_{i,t} = \left(x_i, \dot{x}_{i,t-5}, \cdots, \dot{x}_{i,t-1}, f_i\right)$, where x is the position of the particle, $\dot{x}$ is the velocity of the particle, and f_i is the static material properties of the particle. The graph $G_i = \left(V_i, E_i, u\right)$ is constructed from input X_i. $V_i \in \mathbb{R}^{A \times D_i}$ and $E_i \in \mathbb{R}^{B \times Q_i}$ are the feature sets of node and edge, and u_i is the velocity of particle i.

Processor definition. The Processor consists of ten AGNBs which are embedded in position-aware attention mechanisms. The heterogeneous graph is input to the oil/water blocks to update the oil/water node parameters. Getting the parameters from the AGNB, the physics-aware modules share the parameters and update the weights of the graph, which can be expressed as Eqs. (34) and (35):

$$e'_{ij} = \varphi_e\left(e_{ij}, v_i, v_j, u_i\right) \tag{34}$$

$$v'_i = \varphi_v\left(v_i, e'_{ij}, u_i\right) \tag{35}$$

where φ_v and φ_e are the attribute processing functions for node and edge respectively. v_i and e_{ij} are the feature of node i and edge ij. In the Processor, the propagation of l layer can be expressed as Eq. (36):

$$G_l = \left(V_l, E_l, u_l\right) \tag{36}$$

where $V_l \in \mathbb{R}^{A \times D_l}$ and $E_l \in \mathbb{R}^{B \times Q_l}$ are the outputs of the l-th layer for the feature sets of node and edge. They are defined as Eqs. (37) and (38):

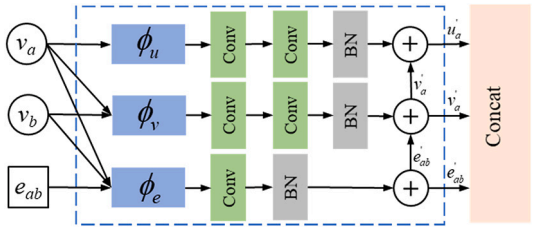

Fig. 3. Structure of the position-aware module. The Conv denotes the convolution, the BN denotes the batch normalization and the Concat denotes the concatenation. v is the node, e is the edge between the nodes, u is the velocity. After updating and aggregation, the features of the next layer are obtained.

$$V_l = \delta_o\left(V_i\right) \tag{37}$$

$$E_l = \delta_e\left(E_i\right) \tag{38}$$

where δ_o and δ_e are the attribute encoding functions for node and edge respectively. The final output graph is $G_K = \left(V_K, E_K, u_K\right)$. As shown in Fig. 3, the position-aware module is used to capture the position relationships between the nodes to enhance the feature expressiveness. For the input v and e, the feature matrix is obtained by the extracting function ϕ. The convolution features and the aggregation results from different channels are used to update the features of edges e'_{ab}, nodes v'_a and velocities u'_a. The process (Eqs. (39)-(41)) can be expressed as:

$$e'_{ab} = BN\left(C_e\left(\phi_e\left(e_{ab}, v_a, v_b\right)\right)\right) \tag{39}$$

$$v'_a = BN\left(C_v\left(\phi_v\left(v_a, v_b\right)\right)\right) + \sum_{b\in N} \phi_v\left(e'_{ab}\right) \tag{40}$$

$$u'_a = BN\left(C_u\left(\phi_u\left(v_a\right)\right)\right) + \phi_u\left(v_a\right) \tag{41}$$

where ϕ_e, ϕ_v and ϕ_u are the functions of edges, nodes and velocities respectively. C_e, C_v and C_u represent the convolution operations of the edges, nodes and velocities, respectively. BN is batch normalization, N represents the neighboring particle set.

In order to improve the ability of the network to describe the force feature of the particles, the surface tension attention (STA) module, the same phase attention (SPA) module and different phases attention (DPA) module are proposed.

We use the position-aware module to calculate the node vectors and the edge vectors. The multi-layer perceptron M, concatenation operation *cat*, rectified linear unit σ and LayerNorm layer *LN* are applied to improve the training stability. The function φ_A (Eq. (42)) can be written as:

$$\varphi_A = LN\left\{\sigma\left\{M\left\{\text{cat}\left[\text{pos}\left(e_{ab}, v_a, v_b\right)\right]\right\}\right\}\right\} \tag{42}$$

The function (Eq. (43)) of output pipeline can be defined as φ_B:

$$\varphi_B = LN\left\{\sigma\left\{M\left\{\text{cat}\left[W^A, W^B\right]\right\}\right\}\right\} \tag{43}$$

$$\varphi^v_{sta} = \varphi_B\left\{v_a, \left\{\varphi_A\left(e_{ab}, v_a, v_b\right) \cdot \left[\alpha_{ab} \cdot \varphi_A\left(e_{ab}, v_a, v_b\right)\right]\right\}\right\} \tag{44}$$

$$\varphi^v_{spa} = \varphi_B\left\{\left(\left[\varphi_A\left(e_{ac}, v_a, v_c\right) \cdot \alpha_{ac}\right], v_a\right)\right\} \tag{45}$$

$$\varphi^v_{dpa} = Fla\left\{M\left\{\text{Norm}\left\{M\left\{\varphi_B\left\{\left[v_a, \alpha_{ad} \cdot \varphi_A\left(e_{ad}, v_a, v_d\right)\right]\right\}\right\} + \varphi_B\left\{\left[v_a, \alpha_{ad} \cdot \varphi_A\left(e_{ad}, v_a, v_d\right)\right]\right\}\right\}\right\}\right\} \tag{46}$$

where subscript *sta*, *spa* and *dpa* indicates the surface tension attention function (Eq. (44)), the same phase attention function (Eq. (45)) and the different phase attention function (Eq. (46)), respectively. *Fla* is the flattened layer.

Decoder definition. The Decoder uses the features of graph G_K obtained from the Processor to calculate the position of each particle, and the decoding equation (Eq. (47)) is as follows:

$$\ddot{x} = \delta'_v\left(v_i\right), v_i \in V_K \tag{47}$$

where δ'_v is the node property decoding function, and we use Euler integration for accelerating to update the particle positions, which is shown as Eqs. (48) and (49):

$$\dot{x}_{i,t} = \ddot{x}_{i,t-1}\Delta t + \dot{x}_{i,t-1} \tag{48}$$

$$x_{i,t+1} = \dot{x}_{i,t}\Delta t + x_{i,t} \tag{49}$$

where $\ddot{x}$ is the acceleration of the particle. So, the approximation function (Eq. (50)) for this learnable simulator is expressed as:

$$\ddot{x}_{i,t} = d\left(X_{i,t}\right) \tag{50}$$

The loss function L_{acc} is used to predict the per particle's acceleration, which is defined as Eq. (51):

$$L_{acc} = \left\| d\left(X_{i,t}\right) - \ddot{p}_{i,t} \right\|^2 \tag{51}$$

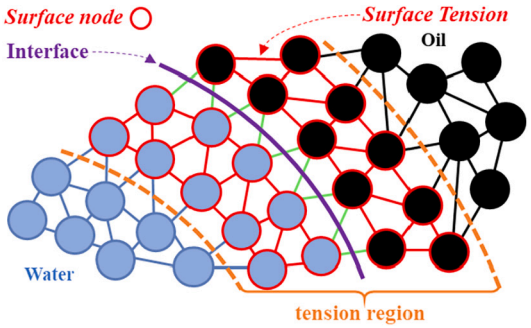

Fig. 4. Schematic of the surface nodes in interfacial flow. (blue filled circle is water node, black filled circle is oil node, red outline circle is surface node for water or oil, red line is surface tension, purple line is the interface between oil and water.)

Table 1
Three types of links between the nodes.

Type	Meaning
Surface Tension (ST)	oil surface-surface, water surface-surface
Same Phase (SP)	oil surface-inner, oil inner-inner, water surface-inner, water inner-inner
Different Phases (DP)	oil surface-water surface

where $\ddot{p}_{i,t}$ is the target acceleration of the particle i at time t.

3.2. Heterogeneous graph attention mechanisms

In the MTDF-SPH model, oil and water particles have different physical properties. The fluid surface tension terms are non-zero for surface particles and zero for internal fluid particles. Here, we propose heterogeneous graph attention mechanisms as the weight function for prediction. The node types of the heterogeneous graph include oil and water. The types (e.g., water and oil) of nodes can be divided into two groups: surface nodes and inner nodes.

As shown in Fig. 4, the surface node is detected by scanning the adjacent nodes in the two steps' topological distance (the set of nodes that can be positioned within two steps) around the center node. If the neighbors are not the same type, the center node is a surface node. Otherwise, it is an inner node. Here, the algebraic method with normalization property of interpolation kernel functions on the support domain is used to detect the interface/surface particles, the discrete form [43] is defined as Eq. (52):

$$\sum_j W_{ij} v_j + W_i v_j \begin{cases} = \gamma, & \text{inner node} \\ < \gamma, & \text{surface node} \end{cases} \tag{52}$$

where W is the kernel function that correlates particles with the support radius of two steps' topological distance, j represents the neighboring particles of i in the same phase, the threshold value γ is set as 1.

The links between the nodes are classified into three types in Table 1. Fig. 4 shows the schematic of the surface nodes in interfacial flow. These different types of links use different attention mechanisms, which are shown in Fig. 1. For the STA module, we combine the position-aware module with MLP to capture the position relationships between the nodes for extracting surface tension features and updating the attention matrix. The SPA part is designed to retain the same phase information, such as oil surface-inner, oil inner-inner, water surface-inner and water inner-inner relations. The same phase particles mainly correlate to self-property. Therefore, we remove the layer φ_A from STA to generate effective node features. In the multiple-fluid simulation with different phases particles, we integrate the MLP-based structure into SPA to construct the DPA module for improving the features' representation capability of different phases fluids interaction.

The attention layer φ_{att} contains the node features $v = \{v_1, v_2, \cdots, v_A\}, v_l \in \mathbb{R}^{D_l}$ and edge features $e = (e_1, e_2, \cdots, e_B), e_l \in \mathbb{R}^{Q_l}$, where the subscripts A and B denote the number of nodes and edges respectively, D and Q denote the feature dimensions of nodes and edges respectively, and l is the number of frame layers.

Assuming that the central node is i and the weight matrix of every node is $W \in \mathbb{R}^{D_i \times D_i}$, the attention $\phi_a : \mathbb{R}' \times \mathbb{R}' \to \mathbb{R}$ (Eq. (53)) between node i and node j can be formulated as follows:

$$\eta_{ij}^{att} = \phi_a \left(W\vec{h}_i, W\vec{h}_j \right) \tag{53}$$

where h is the attention radius, and η^{att} is the attention factor. Since node i is only influenced by nodes within radius h, so the neighboring nodes $j \in N_i$ need to be calculated. In addition, another three attention networks ϕ_{ST}, ϕ_{SP} and ϕ_{DP} for the three types of links can be written as Eq. (54).

$$\eta_T^{\text{att}} = \phi_T \left(W\vec{h}_i, W\vec{h}_j \right) \tag{54}$$

where T is the type of links (ST, SP or DP) between the nodes. The normalization operation equation (Eq. (55)) for all neighboring nodes are as follows:

$$v_i' = \sigma \left(\sum_{j \in N_i} \varphi_v \left(W v_j, \beta_{ij}^{att} \vec{e_i'} \right) \right) \tag{55}$$

The relation vector for the physics-aware module is generated by the linear transformation matrix, we have Eq. (56):

$$P_m' = W^{\varsigma} P_m \tag{56}$$

where P_m' is the relationship vector. By clustering the features of neighboring particles, the oil-water particle node aggregation formula (Eq. (57)) can be obtained:

$$v_{ij}' = \sigma \left(\frac{1}{\chi} \sum_{\varsigma=1}^{\chi} \sum_{j \in N_i} \varphi_v \left(W^{\varsigma} v_j^{(n-1)}, \beta_{ij}^{\varsigma} \vec{e}_i^{(n-1)} \right) \right) \tag{57}$$

Finally, the cross-entropy is used to calculate the loss (Eq. (58)) by the label values and predicted values:

$$L_{att} = - \sum_{v_i \in V} \ln \left(\theta \cdot v_i^* \right) \tag{58}$$

where v_i^* is the output of the node vector for the last fully connected layer, and θ is the parameters of the perceptron.

3.3. FRKD design

In order to reduce the increased computation caused by the attention mechanism and physics-aware module, we compress the structure of PAGATNet and transfer the information from the teacher model to the student network with a knowledge distillation. The FRKD is used to compress and fine-tune the model. As shown in Fig. 2, the teacher network is connected to the student network through knowledge distillation to guide the learning of the student network by using the AGNB and physics-aware module.

The physics-aware module can guide the student network to learn privileged information. It can improve the representation learning ability of the teacher network and facilitate information exchange between the oil and water branches. In order to minimize the feature difference between the teacher network and the student network, the physics-aware module is embedded into the teacher network to get the perception ability of the local geometric structure that can reduce the feature difference effectively. As shown in Fig. 5, the generator is used to synthesize the data, which uses the local features containing the nodes and edges parameters from the shared features of the student network and the teacher network. Thus the KD minimizes the maximum distance between the teacher and the student. The details of the physics-aware module and the generator structure can be seen in Fig. 5. The Eqs. (59) and (60) for the generator is expressed as follows:

$$v_{i,stu}' = f_{\text{adjust}} \left(v_{i,stu}, v_{i,tch} \right) = v_{i,stu} + \lambda_l \times \mathrm{W}_{i,tch}^{v} \tag{59}$$

$$e_{i,stu}' = f_{\text{adjust}} \left(e_{i,stu}, e_{i,tch} \right) = e_{i,stu} + \lambda_l \times W_{i,tch}^{e} \tag{60}$$

where $v_{i,stu}$ and $v_{i,tch}$ are the vectors of node i from the student and the teacher network respectively, and $W_{i,tch}^{e}$ and $W_{i,tch}^{v}$ are the parameter matrices of the edge and node vectors from the teacher network respectively. λ_l is the feature parameter of the l layer in knowledge distillation. The teacher network is connected to the student network through knowledge distillation and guides the training of the student network. The network output knowledge distillation process (Eq. (61)) is as follows:

$$\eta_i^{T-S} = \left\| V_{li}^{T} - V_{li}^{S} \right\| \tag{61}$$

where η_i^{T-S} denotes the distillation loss of node i between the teacher output feature map and the student output feature map. V_{li}^{T} and V_{li}^{S} denotes the output graph of node i in the teacher and student network respectively. Then, the weighted distillation loss (Eq. (62)) is defined as:

$$L_{T-S} = \sum_i \omega_i \eta_i^{T-S} \tag{62}$$

where L_{T-S} represents the response-based knowledge distillation loss, ω_i is the weight of the validation loss. Besides network output distillation, we further enhance the student network by the feature-based knowledge distillation. The generating module of each student layer is associated with the aware module of the teacher layer. Differences of structural features between node i and neighboring node j are defined η_{ij}^{D} (Eq. (63)) shown below:

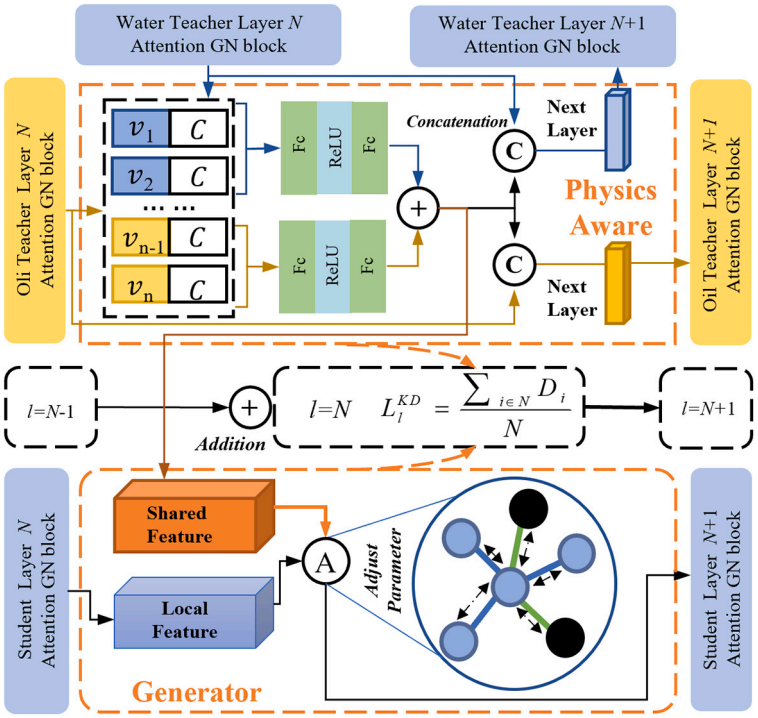

Fig. 5. Structure of the physics-aware module and generator framework. The *Fc* denotes the fully connected layer, the *ReLu* denotes the linear rectification function, the C next to v is the set of nodes which connected the v, l represents the layer number of the network.

$$\eta_{ij}^{D} = \left\| G_i - G_j \right\| \tag{63}$$

Replacing the η_{ij}^{att} in Eq. (26) with η_{ij}^{D} then obtain the new normalized formula. β_{ij}^{TD} and β_{ij}^{SD} denote the local structure vectors of the teacher network and the student network respectively. For each central node i, the similarity (Eq. (64)) between the local structures of students and teachers is as follows:

$$D_i = \sum_{k \in N_i} \beta_{ik}^{SD} \log\left(\frac{\beta_{ik}^{SD}}{\beta_{ik}^{TD}}\right) \tag{64}$$

where D indicates the smaller difference between the two local structures and the more similar distribution of local structures. Therefore, we calculate the similarity of the distribution on all nodes of a given graph and obtain the loss function (Eq. (65)) for knowledge distillation as follows:

$$L_l^{KD} = \frac{\sum_{i \in N} D_i}{N} \tag{65}$$

where N denotes the number of all nodes, L_l^{KD} denotes the l-th inter-layer knowledge distillation loss function. The multi-layer knowledge distillation loss function L_{MKD} (Eq. (66)) is defined as:

$$L_{MKD} = \sum_{(S_l, T_l) \in KD} L_l^{KD} \tag{66}$$

where *KD* denotes the set of knowledge distillation modules, and S_l, T_l denote the l-th attention layer of the student model and the teacher model respectively, and thus the overall loss (Eq. (67)) is obtained as:

$$L_{KD} = L_{T-S} + \beta L_{MKD} \tag{67}$$

where L_{KD} is the overall loss function for knowledge distillation, and the hyperparameter β is used to balance the two separate loss terms.

3.4. Loss function

As discussed above, we get the loss function L_{acc} for per-particle accelerations, L_{att} for graph attention mechanisms and L_{KD} for FRKD. Thus, the overall loss function L (Eq. (68)) of PAGATNet is defined as follows:

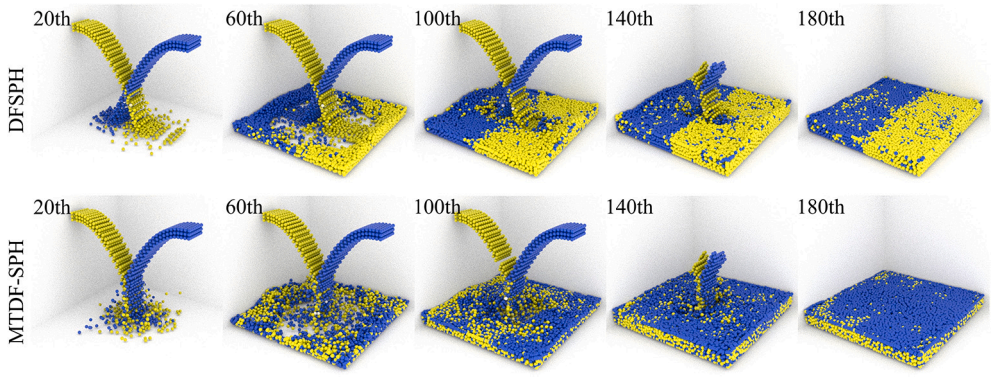

Fig. 6. Two-phase fluid particle interaction results (yellow and blue). (a) Top: 20th-180th of DFSPH, the two phases are not mixed and are separated into left and right sides with a clear interface. (b) Bottom: 20th-180th of MTDF-SPH, the two phases are mixed and when the mixing is finished, they are mixed due to the difference in density between the immiscible phases.

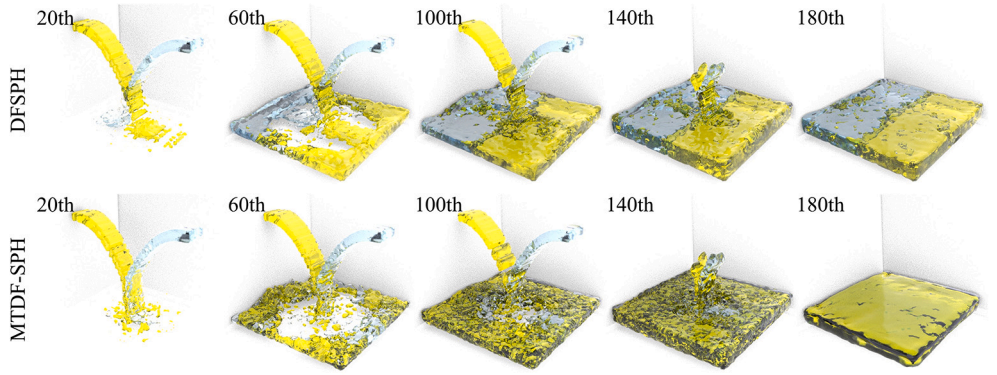

Fig. 7. Two-phase fluid interaction rendered results (yellow liquid, blue liquid). (a) Top: 20th-180th of DFSPH. (b) Bottom: 20th-180th of MTDF-SPH.

$$L = \lambda_1 L_{acc} + \lambda_2 L_{att} + \lambda_3 L_{KD} \tag{68}$$

where λ_1, λ_2 and λ_3 are the hyper-parameters.

4. Experiments and discussion

The experimental environment of our model is based on the Ubuntu 18.04 operating system with an Intel Xeon Bronze 3206R CPU, 8GB RAM and an Nvidia GeForce GTX 3060 GPU (GPU has 16 GB RAM) as hardware support. The MTDF-SPH algorithm uses C++ language under the Visual Studio 2017. The OpenGL is used as a real-time simulation tool of particles, and we finish the offline realistic rendering by Maya. The PAGATNet algorithm is implemented by Python language, and Blender is used to render the generated .ply file in simulation. Our network uses the Adam algorithm [44] as the optimizer, and the initial learning rate is set as 1e-4, the end learning rate as 1e-6, and the batch size as 1.

For the MTDF-SPH two-phase fluid simulation, experiments are conducted comparatively for each of the SPH methods such as WCSPH [15], PFSPH [45], PBF [17], IISPH [18], DFSPH [19] algorithms and our algorithm, as is shown in Figs. 10, 11 and 12 from left to right. The methods of GNS [42] and DPI-Nets [46] are used to compare with the PAGATNet. Our datasets typically contained 4000 train trajectories, 4000 validation trajectories and 2000 test trajectories, each are simulated for 2000 timesteps.

4.1. MTDF-SPH two-phase fluid simulation

In the MTDF-SPH experiment, the particle radius is set to 0.025, and all particles are in the gravity field. The DFSPH and MTDF-SPH two-phase fluid-particle interaction and rendering results are given in Fig. 6 and Fig. 7. There are 5000 particles to simulate the flow of crude oil from the suspended pipeline to the water surface and the approximate maximum number of particles is 5000. The DFSPH method simulates 180 steps with 30.50 seconds, and the MTDF-SPH method simulates 180 steps with 34.51 seconds.

In the beginning, two sets of fluids are injected into the vessel in opposite directions, and fluids fall to the bottom after the collision of particles. In the results of the DFSPH method, the two sets of fluids do not collide after touching. They pass directly into each other and then fall to the bottom of the vessel. The interaction focuses on collisions between the particles and the container boundary with few collisions between the particles. In the results of the MTDF-SPH method, the two groups of fluids collide and fall to the bottom of the vessel after touching.

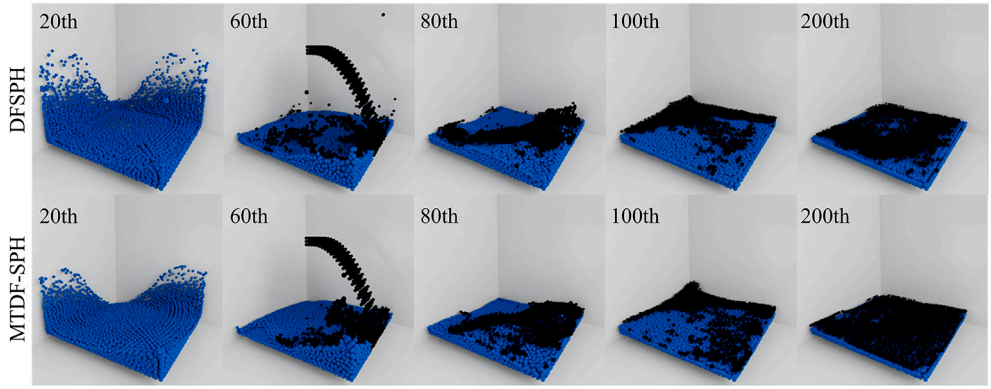

Fig. 8. Results of oil and water particle interaction (black and blue). (a) Top: 20th-200th of DFSPH, the two phases are not mixed and are divided into two layers with the liquid splashing at frame 60th and distribute not evenly at the end of the mixing. (b) Bottom: 20th-200th of MTDF-SPH, the two phases are mixed, and the oil particles distribute in the upper layer when the mixing is finished.

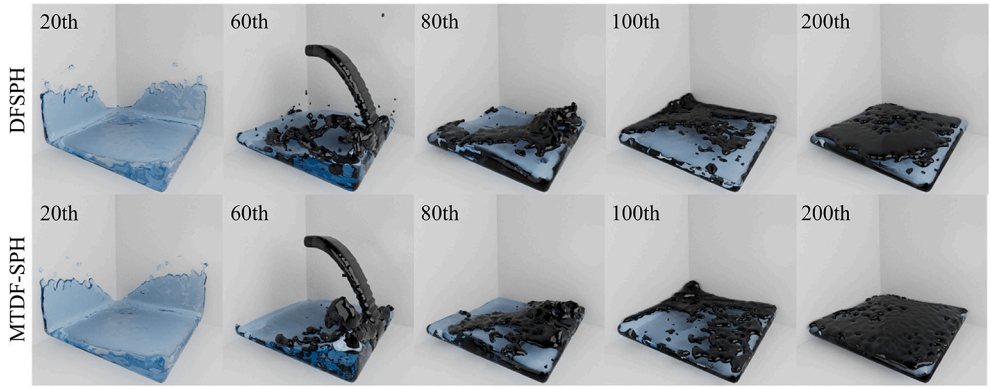

Fig. 9. Oil and water liquid interaction rendered results (black liquid, blue liquid). (a) Top: 20th-200th of DFSPH. (b) Bottom: 20th-200th of MTDF-SPH.

At the 140th frame, the two groups of fluids in the DFSPH method gradually show a gathering tendency. In our method, the two groups of fluids gradually merge due to surface tension between the fluids of particles.

When the overall fluid stabilizes at the 200th frame in the DFSPH, the blue fluid is concentrated in the vessel's right side and the yellow fluid is concentrated in the left side of the vessel. In contrast, the particles in containers of our method are uniformly mixed and separated into two layers with a clear interface.

As shown in Fig. 8, the black and blue particles represent the crude oil particles and the water particles. We set the parameter of the oil phase (0.94×density and 4×viscosity) and the water phase (1×density and 1×viscosity). There are 1500 particles to simulate the flow of crude oil from the suspended pipeline to the water surface. The approximate maximum number of particles is 6000, and There. The DFSPH method simulates 200 steps with 32.88 seconds, and the MTDF-SPH method simulates 200 steps with 38.53 seconds. It can be seen that the particles for the DFSPH method are splashed at frame 60th when the two fluids collide and interact with each other, which does not respect the physical properties of crude oil.

In MTDF-SPH, the two fluids are more concentrated in the interaction process. At the 200th frame, when the two-phase flow is stabilized, the lower-dense fluid floats on the higher-density fluid. The MTDF-SPH results have a more uniform distribution than the DFSPH method, including oscillatory movement over time due to the surface tension. Fig. 9 shows the rendering results of the two methods of simulation that MTDF-SPH can improve the computational accuracy well.

4.2. Comparison of MTDF-SPH leakage simulation

To investigate the influence of the MTDF-SPH, we import an actual topography of a specific crude oil transportation pipeline crossing a river in China based on the remote-sensing image. The details of the three test conditions are shown in Table 2. The river channel is filled with water particles, and we set up particle emitters on the upstream river's left bank, middle area and right bank to represent the oil spill points respectively.

As shown in Fig. 10, the crude oil spill in case 1 is located close to the left river bank. The particles simulated by the WCSPH method are clustered on the right bank at frame 3000th (in the red box). The leakage edges are zigzag and irregular during diffusion. The PFSPH method shows a backflow of crude oil in the river from top to bottom. In the DFSPH method, the leakage of crude oil spreads at the beginning of the spill and touches the right river bank before reaching the narrow channel due to the particles piling

Table 2
Description of the test conditions. The Leakage port location is divided into the left bank, the middle area of the river and the right bank.

Test case	Leakage port location	Water flow speed	Leakage rate
Test Case 1	Left	1.5 m/s	1
Test Case 2	Middle	1.5 m/s	1
Test Case 3	Right	1.5 m/s	1

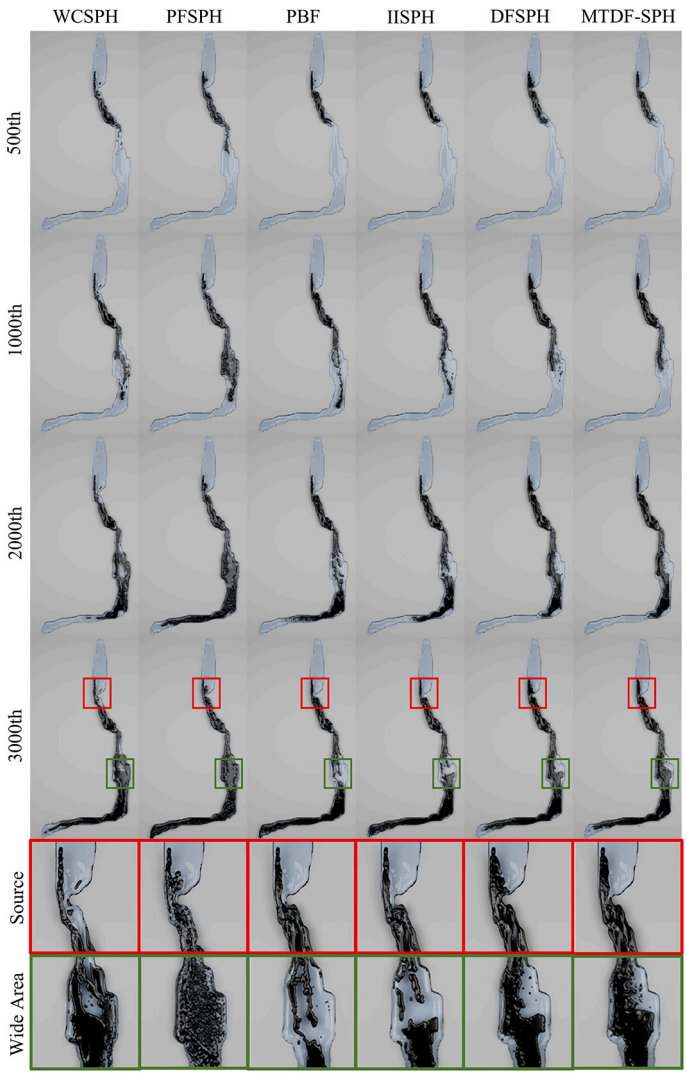

Fig. 10. The simulation results of the WCSPH, PFSPH, PBF, IISPH, DFSPH and MTDF-SPH models at frames 500-3000th for river crude oil spills of Case 1. The six columns from left to right are the top views of the simulation results for each model, and the results of different time are represented from top to bottom. In the river, the water flows from top to bottom and the black area shows the extended distribution of crude oil. The red box represents a zoom-in view of the source of the crude oil spill in frame 3000th, where the spill originates near the left bank of the river. The green box represents a zoom-in view of a wide area of the river downstream of frame 3000th.

up together at the boundary. The simulation results of the PBF, IISPH and MTDF-SPH methods show that the crude oil flows from the spill mouth along the left side of the river bank into the narrow mouth of the turn with no particles clustered at the boundary. In the middle reaches of the spill area at frame 3000th (in the green box), the river channel's width ranges from small to large with significant interruptions in crude oil flow for the WCSPH, PBF and IISPH methods.

In Fig. 11, we can see that the crude oil in Case 2 spills from the middle of the river and a small amount of crude oil accumulates on the right bank of the upstream narrow mouth at frame 3000th (in the red box) in both methods of WCSPH and IISPH. The PFSPH

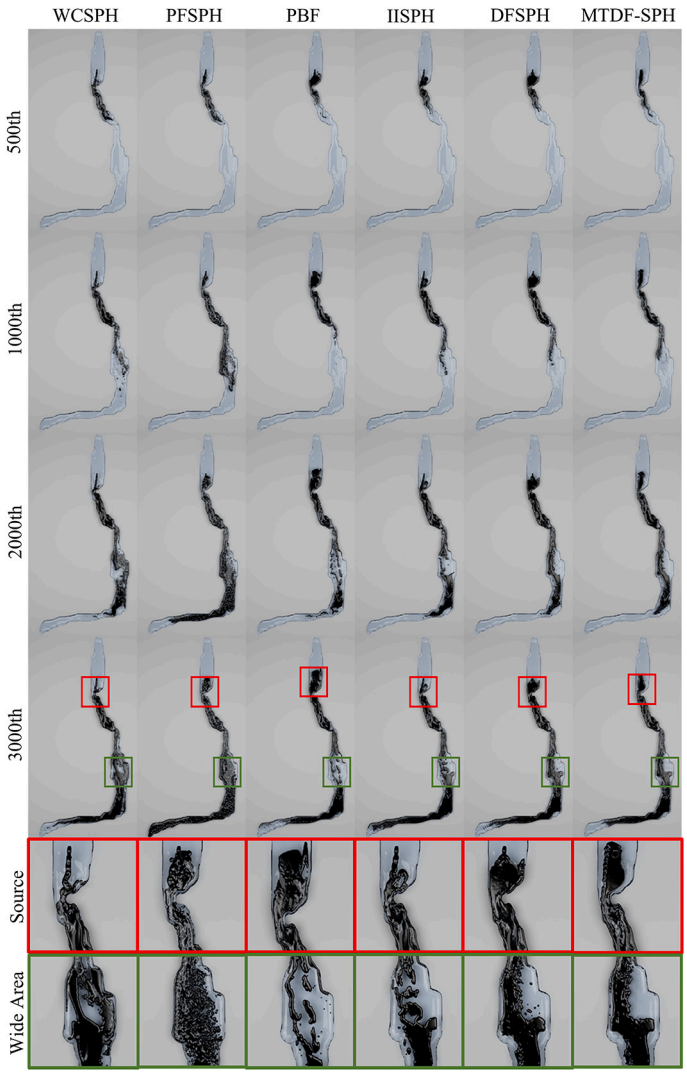

Fig. 11. The simulation results of the WCSPH, PFSPH, PBF, IISPH, DFSPH and MTDF-SPH models at frames 500-3000th for river crude oil spills of Case 2. The six columns from left to right are the top views of the simulation results for each model, and the results of different time are represented from top to bottom. In the river, the water flows from top to bottom and the black area shows the extended distribution of crude oil. The red box represents a zoom-in view of the source of the crude oil spill in frame 3000th, where the spill originates in the middle of the river. The green box represents a zoom-in view of a wide area of the river downstream of frame 3000th.

and PBF methods simulation results show crude oil's backflow from downstream to upstream with abnormal spreading. The crude oil touches the left and right sides in a clumped shape during the simulation of the DFSPH method. In contrast, in the MTDF-SPH method, crude oil flows in the middle of the river without touching the banks on both sides in a bundle shape. In the middle reaches of the spill area (within the green box) at frame 3000th, the river changes from narrow to wide. The simulation results of WCSPH, PBF and IISPH methods for leakage of crude oil show different degrees of fracture, accumulation and bifurcation. The PFSPH method simulates crude oil evenly spreading on the river surface with little influence from the flow. The DFSPH and MTDF-SPH methods simulate crude oil spreading downstream and regathering at river narrowing.

As shown in Fig. 12, the location of the crude oil spill in Case 3 is close to the right side of the bank from upstream to downstream of the river. At the frame 3000th upstream of the spill area (in the red box), a small amount of the crude oil leakage has backflow on the right side in the WCSPH method. The serious backflow occurred in the PFSPH method and the PBF method, which are the same as the second group of experimental results. The DFSPH method simulation showed unusual spreading in the upstream of the river. In the MTDF-SPH method, on the other hand, the spilled crude oil keeps following the right side of the river from the spill to the downstream. In the middle region of the spill at frame 3000th (within the green box), the simulated results of the WCSPH, PBF and IISPH methods are incorrect for the phenomena effects such as splashing and breaking waves during the simulation. The PFSPH method for simulating crude oil uniformly flat on the river surface without touching the right bank in broad areas. However, the results of MTDF-SPH crude oil are more concentrated for particle dispersion, following the river flow direction without droplet splashing.

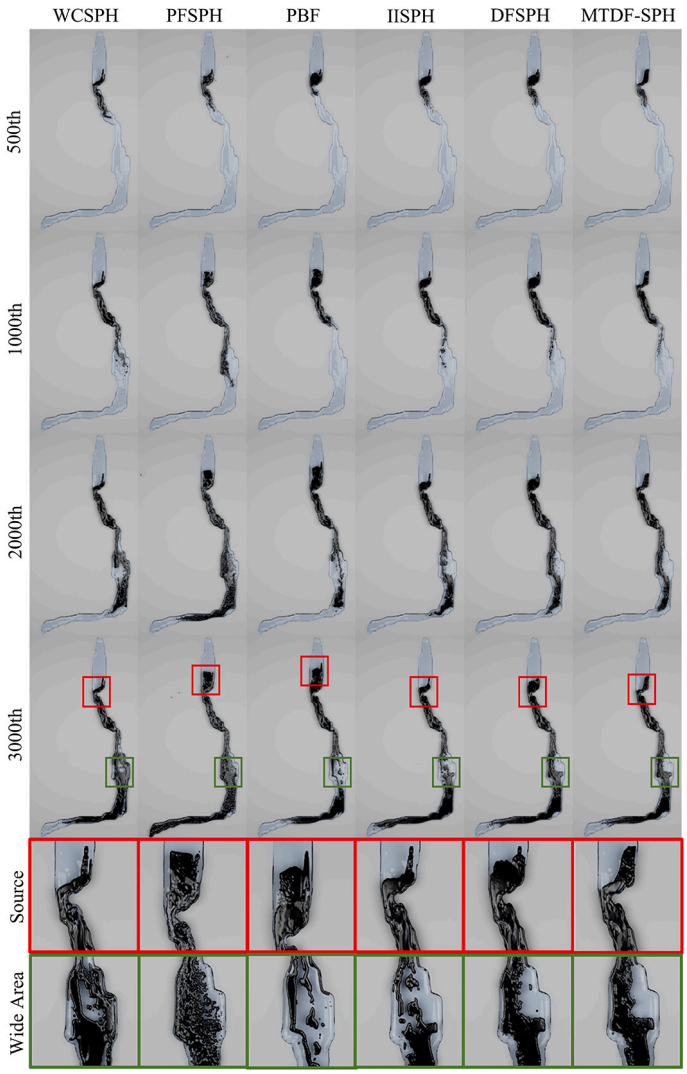

Fig. 12. The simulation results of the WCSPH, PFSPH, PBF, IISPH, DFSPH, and MTDF-SPH models at frames 500-3000th for river crude oil spills of Case 3. The six columns from left to right are the top views of the simulation results for each model, and the results of different time are represented from top to bottom. In the river, the water flows from top to bottom and the black area shows the extended distribution of crude oil. The red box represents a zoom-in view of the source of the crude oil spill in frame 3000, where the spill originates near the right bank of the river. The green box represents a zoom-in view of a wide area of the river downstream of frame 3000.

Experimental results show that the location of leakage upstream greatly influences the simulation results and the effects are smaller in the downstream. MTDF-SPH has the best simulation results in the downstream of the river compared to the above simulation algorithms. The simulation of crude oil downstream is relatively better than upstream because the error transfer of the simulation. The MTDF-SPH method goes one step further than DFSPH in simulation results and achieves better results in the DFSPH-based river crude oil spill simulation task. Considering the interaction between particles of different phases and surface tension in multiphase flow, our method combines the solutions of dispersion-free and density-constant to reduce the cumulative errors in the simulation process and to match the physical laws of crude oil spill diffusion in a flowing two-phase flow. In our simulation studies, the ground truth information refers to the Norilsk oil spill occurred on 29 May 2020 with a large oil spill into the Ambarnaya river [47]. Compared with the Sentinel-2 data, it can be seen that the simulation results of our methods have high authenticity.

In order to demonstrate the efficiency of our method, the experiments with six different algorithms to get the simulation time for the simulation Case 1 in Table 3.

Due to the mixture model and surface tension model, which are added to the SPH algorithm, the simulation time of oil leakage is increased. In Table 3, we can see that the MTDF-SPH simulation time is increased. At the frame of 3100th, the time required by MTDF-SPH is increased by 13.84% compared with DFSPH. So, we should improve the time efficiency of the MTDF-SPH algorithm.

Table 3
Comparison of T frame timings for river oil leakage simulation algorithms.

Experiment	T = 600	T = 1100	T = 1600	T = 2100	T = 2600	T = 3100
WCSPH	2260.67 s	2670.59 s	3229.49 s	3852.19 s	4539.52 s	5274.10
PFSPH	6533.43 s	11082.40 s	15869.90 s	16800.30 s	17838.90 s	19176.10 s
PBF	2233.81 s	3972.96 s	4971.35 s	6070.57 s	9409.95 s	13300.60 s
IISPH	3805.47 s	6341.60 s	7157.57 s	8090.76 s	9551.53 s	10991.20 s
DFSPH	2848.12 s	4913.76 s	5504.07 s	5981.93 s	7126.96 s	8568.16 s
MTDF-SPH	**3238.26 s**	**5579.96 s**	**6285.72 s**	**6753.50 s**	**8024.29 s**	**9753.63 s**

Table 4
Parameter setting for each experiment.

Domain	Number of particles (Approximate)	Epoch	Batch size	Trajectory length	Trajectories		
					Train	Validation	Test
Two-phase interaction	7k	200	4	200	400	400	200
Oil-water interaction	7k	200	4	200	400	400	200
Test Case 1	100k	200	2	3500	4000	4000	2000
Test Case 2	100k	200	2	3500	4000	4000	2000
Test Case 3	100k	200	2	3500	4000	4000	2000

Table 5
Comparison of run times and the model error based on the river crude oil spill pollution dataset.

Model	Mean Squared Error	Time for T frame			
		T = 40	T = 70	T = 100	T = 130
MTDF-SPH	-	13.72 s	23.87 s	38.43 s	47.23 s
DPI-Nets	0.0397	11.79 s	20.02 s	29.07 s	36.27 s
GNS	0.0242	11.05 s	18.01 s	25.01 s	31.45 s
PAGATNet	**0.0038**	**8.41 s**	**14.34 s**	**19.97 s**	**25.71 s**

4.3. PAGATNet simulation experiments

For the computational complexity problems of the SPH algorithms, such as the MTDF-SPH, we propose the PAGATNet network for simulation, which can effectively reduce the bottleneck of time complexity.

In order to demonstrate the simulation accuracy of PAGATNet for crude oil on the river surface, we trained the PAGATNet with the dataset generated by the MTDF-SPH algorithm. We compared it with the methods of GNS and DPI-Nets. In Figs. 13 to 17, the frame i in Case 1 from Section 4.2 are set as the initial state and predicts the leakage in frame T. The rows in these figures from top to bottom represent MTDF-SPH, DPI-Nets, GNS and PAGATNet. Table 4 gives the Parameter setting for each experiment.

As shown in Table 5, MTDF-SPH as the ground truth is compared with DPI-Nets, GNS and PAGATNet. The total approximate maximum number of particles is 10k. We use the mean square error as our main metric between ground truth and predicting results. It can be seen that the PAGATNet has lower error than other deep learning models across all cases we tested.

The flow effects of the crude oil diffusion simulated using MTDF-SPH are given in Fig. 13. At the beginning of the leak diffusion as the 230th frame, the crude oil flows from the spill location and starts to touch the narrow bend of the river. At the 260th frame, the leakage flows downstream along the right bank. At the 290th frame, the leakage gradually aggregates on the concave bank of the river. At the 320th frame, the crude oil spread along the concave bank of the river under the effect of inertia and impulse, and it changes the shape of the distribution. During the metaphase of the leak diffusion, as in Fig. 14 at 530th frame, the leakage has spread to the center of the river and gathered in clumps. At the 560th frame, the crude oil is gradually dispersed and spread around the river. At the 620th frame, the river becomes narrow and the leakage aggregates and adheres to the river bank. The oil film transitioned from a small-scale centralized state to a large-scale decentralized state. At the end of the diffusion, as in Fig. 15, the river channel changes from narrow to wide, and the velocity of water is decreased. At the 1030th frame, the crude oil enters the broad river channel and some of the crude oil gathers while some of the crude oil spreads in a cluster shape. The frame 1090th reflects the flow direction of the river more obviously and the oil film scatter in the multi-block. In frame 1120th, the shape of oil is shown as thin strips, and the mixture of oil and water diffuses to the downstream of the river.

As shown in Figs. 13, 14 and 15, the leak point is on the left bank of the river. The GNS method predicts enormous differences in shape with the MTDF-SPH results at the source of the river. At the edge of the crude oil spill diffusion, the DPI-Nets method has some gaps in the downstream crude oil diffusion body while the shape of the downstream crude oil spill body of the GNS method differs significantly from the MTDF-SPH calculation. The simulation results of PAGATNet have no spatial gaps in the downstream

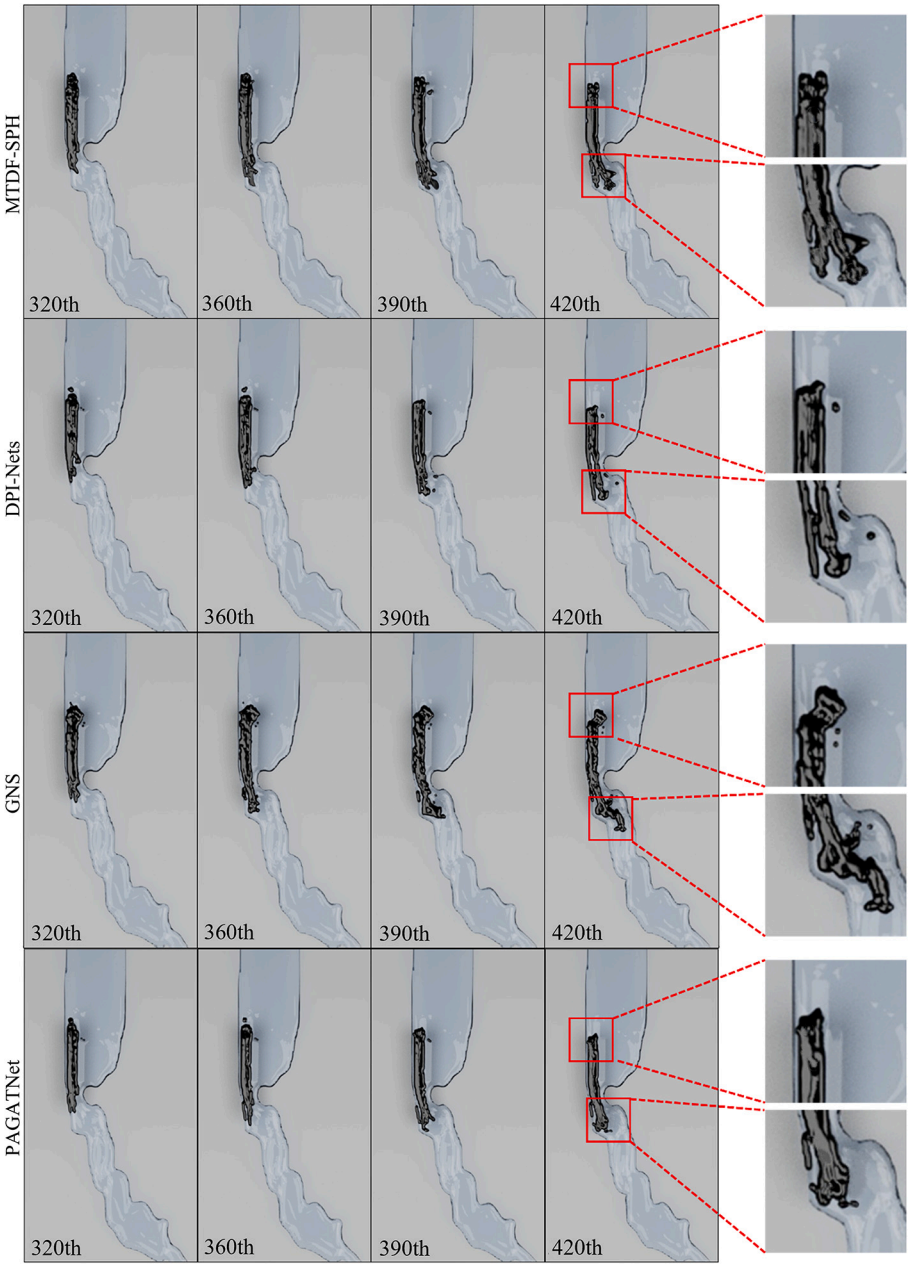

Fig. 13. Simulation results for the MTDF-SPH, DPI-Nets, GNS and PAGATNet models for the river crude oil spill of Case 1 from 320th to 420th frame. From top to bottom is a top view of the simulation results for each model, and the six columns present the simulation results at different times from left to right. In the river, the water flows from top to bottom, and the black areas show the extended distribution of crude oil. The upper red box in each image represents a zoom-in view of the crude oil spill source in the river at frame 420th, and the spill source is near the left bank of the river. The lower red box represents a zoom-in view of the edge of the river crude oil spill spread at frame 420th.

crude oil spill body, and the details are well-preserved and closer to the MTDF-SPH results shown in Fig. 15. After the extensional stage, the oil will fill the river's surface and shift downstream, and the particle number of DPI-Nets and GNS becomes smaller than MTDF-SPH. However, GNS predicts a similar fluid distribution as MTDF-SPH, showing three divergent flows. PAGATNet also predicts three divergent flows and the predicted number of particles is similar to MTDF-SPH results.

As shown in Fig. 16, in Case 2, the source of the leak is in the middle of the river and spread along the centerline of the river. The DPI-Nets method shows significant reflux at 420th and the predicted results of the GNS method also reflect backflow at 420th. At the same time, the distribution of the predicted effects of PAGATNet is similar to the MTDF-SPH results. The leakage will spread over the river at the spreading segments and the downstream diffusion edge. The PAGATNet predicts the number of particles and the diffusion range of crude oil which is similar to the MTDF-SPH results. The results of DPI-Nets using the same number of particles are

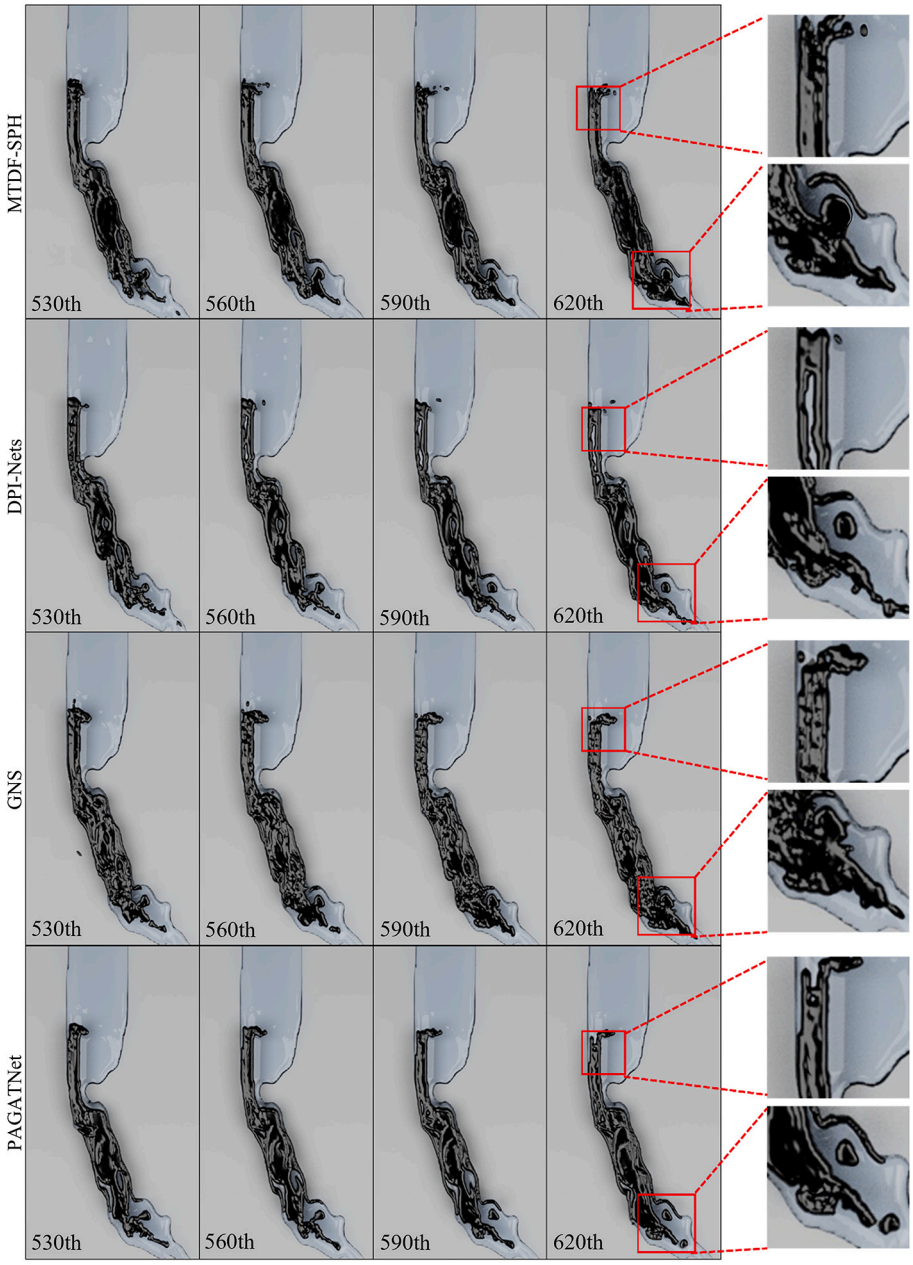

Fig. 14. The simulation results of the MTDF-SPH, DPI-Nets, GNS and PAGATNet models for the river crude oil spill of Case 1 from 520th to 620th frame. From top to bottom is a top view of the simulation results for each model, and the six columns present the simulation results at different times from left to right. In the river, the water flows from top to bottom, and the black areas show the extended distribution of crude oil. The upper red box in each image represents a zoom-in view of the crude oil spill source in the river at frame 620th, and the spill source is near the left bank of the river. The lower red box represents a zoom-in view of the edge of the river crude oil spill spread at frame 620th.

fragmented and fractured with the dispersion distribution. The predicted shape of GNS is similar to the MTDF-SPH at the upstream, but the diffusion distance is inaccurate.

As shown in Fig. 17 of Case 3, the simulation results of the DPI-Nets method show that the crude oil collisions at the left bank with the wrong location. The prediction leakage of GNS touches the left bank of the river and the downstream diffusion range is inconsistent with the MTDF-SPH results. In the PAGATNet results, the collision location of crude oil with the left bank of the river and the diffusion range downstream are closer to the actual situation. In the beginning, the leakage will gradually move to the river bank and cling to the bank. However, the leakage spreads all over the river after the spreading segments. The DPI-Nets predictions show the same number of particles but different diffusion ranges. Moreover, the artifacts of GNS in the boundary region are due to the instability of prediction.

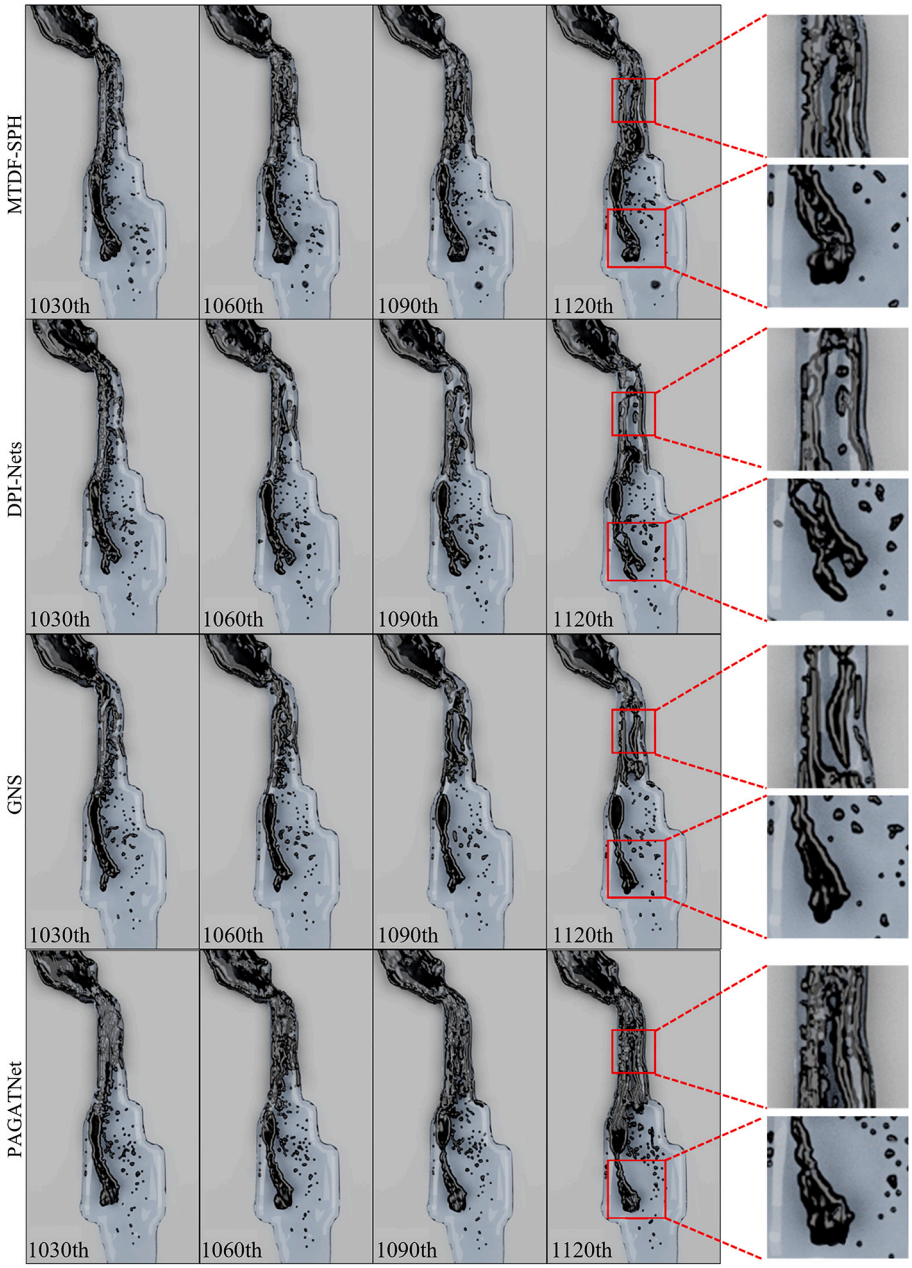

Fig. 15. Simulation results for the MTDF-SPH, DPI-Nets, GNS and PAGATNet models for the river crude oil spill of Case 1 from 1030th to 1120th frame. From top to bottom is a top view of the simulation results for each model, and the six columns present the simulation results at different times from left to right. In the river, the water flows from top to bottom, and the black areas show the extended distribution of crude oil. The upper red box in each image represents a zoom-in view of the narrower river widths at frame 1120th, and the spill source is near the left bank of the river. The lower red box represents a zoom-in view of the edge of the river crude oil spill spread at frame 1120th.

According to the above experimental results, we can see that the PAGATNet is closer to the MTDF-SPH simulation results than the other two methods for different test conditions, times and basins. Considering the model's prediction error decreases with the increase of training time, the PAGATNet shows better performance for immiscible two-phase fluid in the oil leakage case with the graph attention mechanism designed in Section 3.2.

In order to verify the efficiency, MTDF-SPH, DPI-Nets, GNS and PAGATNet are compared to simulate the crude oil spills on real terrain, as shown in Table 5. We can see that the PAGATNet significantly reduces the overall computation time. When the number of simulation frames is 130th, PAGATNet required 45.57% less time than MTDF-SPH. It can also be seen that the PAGATNet significantly improves the efficiency of simulation calculation.

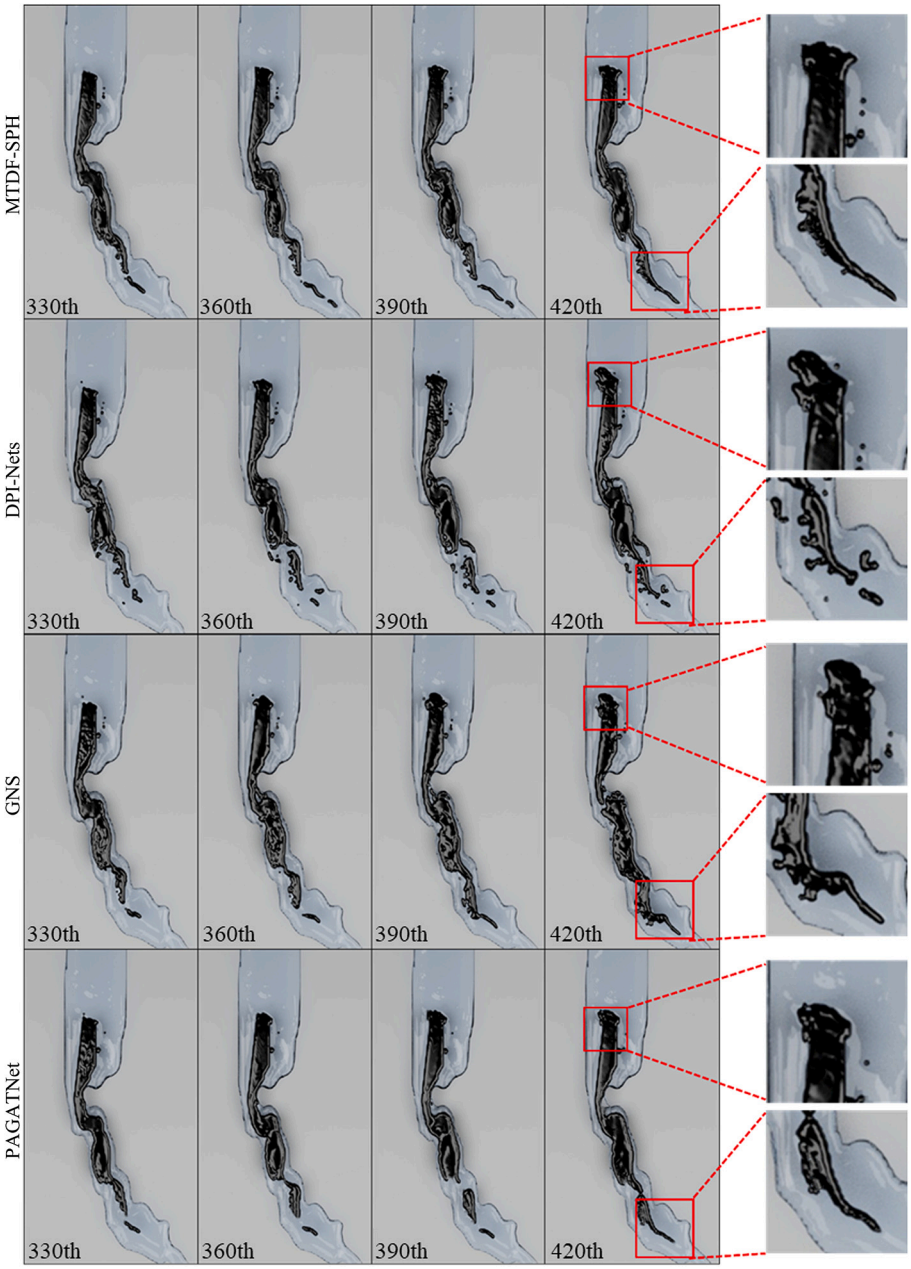

Fig. 16. Simulation results for the MTDF-SPH, DPI-Nets, GNS, and PAGATNet models for the river crude oil spill of Case 2 from 330th to 420th frame. From top to bottom is a top view of the simulation results for each model, and the six columns present the simulation results at different times from left to right. In the river, the water flows from top to bottom, and the black areas show the extended distribution of crude oil. The upper red box in each image represents a zoom-in view of the crude oil spill source in the river at frame 420th, and the spill source is in the middle of the river. The lower red box represents a zoom-in view of the edge of the river crude oil spill spread at frame 420th.

4.4. Ablation experiments

In this section, we inspect the ablation effects of each module for PAGATNet by comparing the results before and after adding the AGNB, the FRKD and the physics-aware module, respectively. To examine the impact of the AGNB, we remove attention mechanism from our processor and replace the AGNB with Graph Networks (GN) [48], which we call PAGATNet_AGNB.

As shown in Fig. 18, the edges of leakage by GNS are very different from those of MTDF-SPH. The particles simulated by PAGATNet_AGNB are sparsely distributed for the reason that PAGATNet_AGNB cannot process surface tension and multiple-fluid interaction. Although the PAGATNet has a slightly different shape with MTDF-SPH at the leading edge of oil, it can simulate the spill along the river bank of upper side realistically.

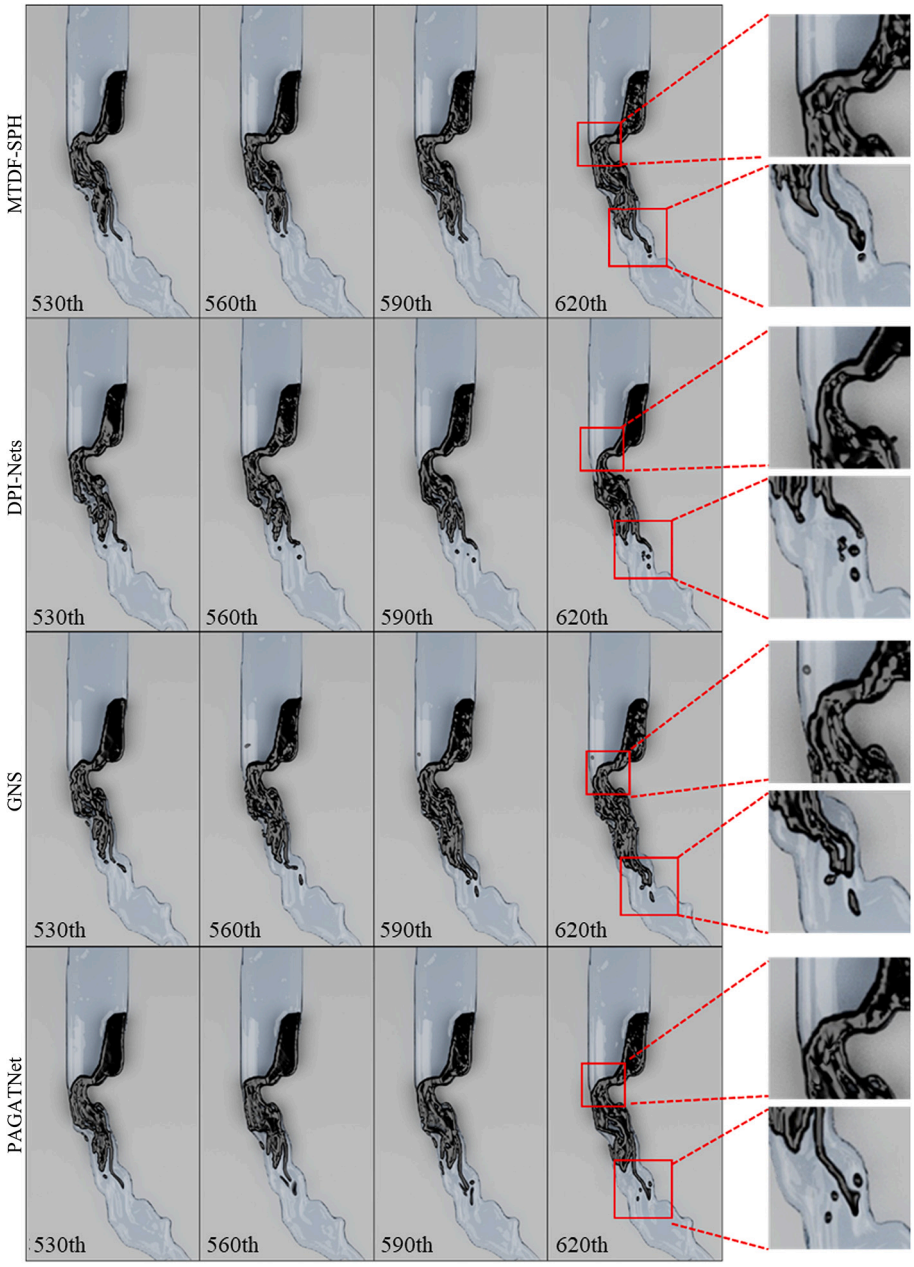

Fig. 17. Simulation results for the MTDF-SPH, DPI-Nets, GNS and PAGATNet models for the river crude oil spill of Case 3 from 530th to 620th frame. From top to bottom is a top view of the simulation results for each model, and the six columns present the simulation results at different times from left to right. In the river, the water flows from top to bottom, and the black areas show the extended distribution of crude oil. The upper red box in each image represents a zoom-in view of the river bend at frame 620th, and the spill source is near the right bank of the river. The lower red box represents a zoom-in view of the edge of the river crude oil spill spread at frame 620th.

In order to verify the rationality of the physics-aware module design in PAGATNet, we remove the physics-aware block and retrain the network. The retraining results are shown in Fig. 19. After removing the physics-aware module, the leakage of the crude oil is scattered at step 75th which becomes more severe with the simulation time increases. Also, inertia makes the empty area in the leakage body more dispersed downstream. In contrast, the leakage diffuses more continuously with the physics-aware module adopted, which improves the stability and reality of the simulation.

Meanwhile, we verify the rationality of the knowledge distillation and test the teacher and student networks using the dataset generated by the MTDF-SPH algorithm. Table 6 shows that the student network took less time than the teacher network to simulate the same data. It also can be seen from Fig. 20 that differences in the results are limited between the teacher and student network. Therefore, the student network reserves the accuracy of the simulation in the teacher network. The local graph of features makes full

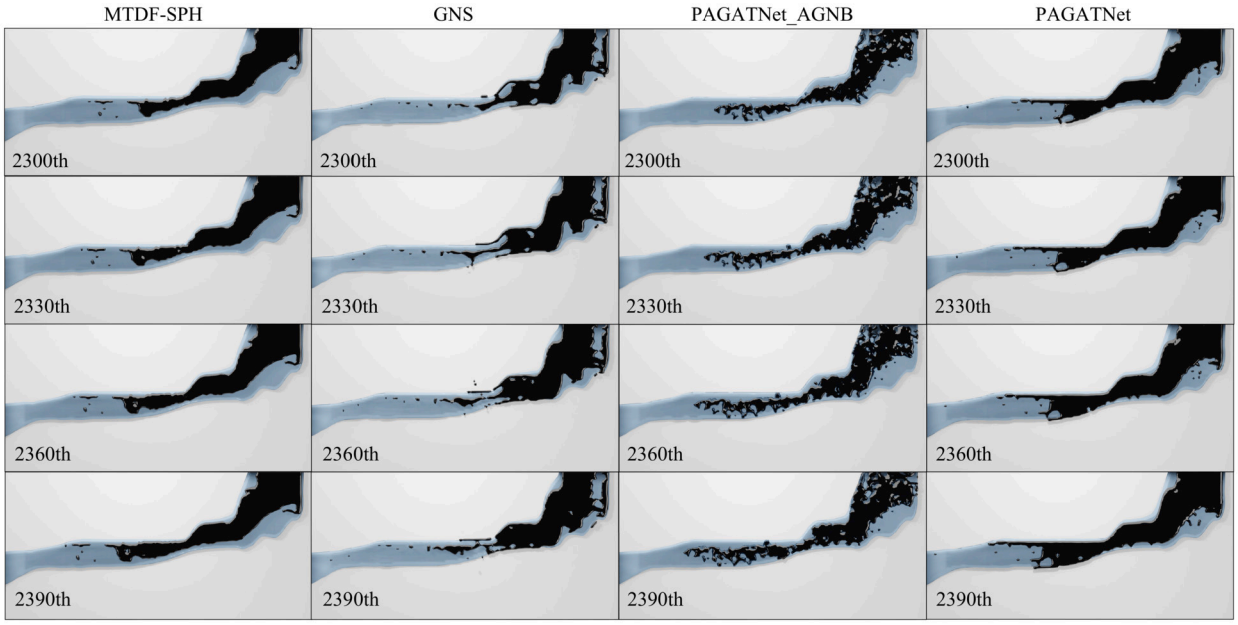

Fig. 18. Simulation results for the MTDF-SPH, GNS, PAGATNet_AGNB and PAGATNet modules for the river crude oil spill of Case 1 from 2300th to 2390th frame. From left to right is a top view of the simulation results for each model, and the four rows show the simulation results at different times from top to bottom.

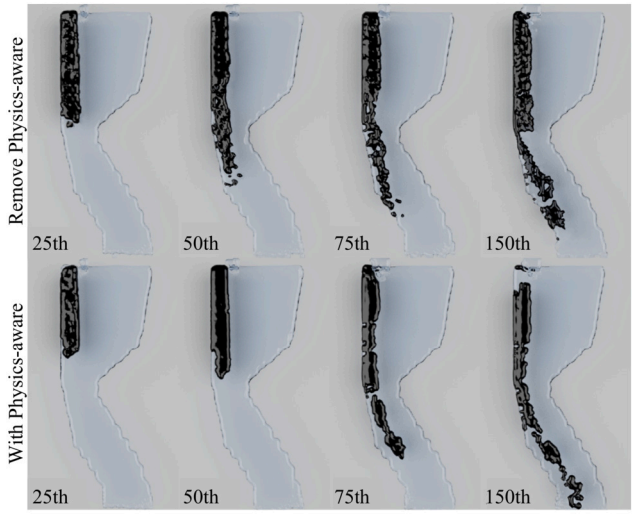

Fig. 19. Comparison results of the attentional ablation experiment simulation. Top: removing the physics-aware module. Bottom: with the physics-aware module.

Table 6
Comparison of the time to T frame for the teacher and the student network.

Model	Time taken to generate T frame			
	T = 30	T = 60	T = 90	T = 120
Teacher network	9.04 s	17.97 s	26.47 s	35.79 s
Student Network	4.99 s	9.87 s	14.45 s	19.65 s

and complete use of the knowledge from the teacher model. Due to the teacher network keeps learning the law of interaction of oil and water particles which promotes the learning ability of the student network, the student network can pay more attention to the learning of the processor ability for the graph network.

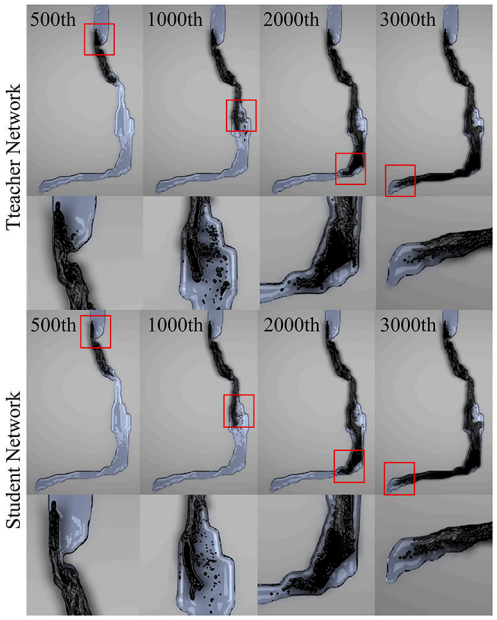

Fig. 20. Comparative simulation results of the attentional ablation experiment. Top: the teacher network simulation results. Bottom: the student network simulation results.

5. Conclusion

In this paper, we have investigated a new and robust two-phase leakage simulation method for diffusing crude oil on the river surface. A simulation model MTDF-SPH is proposed based on the models of the mixture, surface tension and DFSPH method. Then, to accelerate the two-phase flow simulation, the heterogeneous graph attention network PAGATNet is designed by using a knowledge distillation method. Our approach is demonstrated to be superior through evaluations on several cases and can improve the stability and reality of the simulation for the leakage of crude oil on the river. We believe that the novel fusion of SPH methods and graph attention neural network is a promising direction since it can be used to meet the requirement of relevant leakage applications of two-phase flow.

As the oil spills into the river, it has many processes: evaporation, dissolution and emulsification, etc. The phenomenon of evaporation can affect the simulation results due to the change of the particle density. The process of emulsification will cause changes in particle viscosity which can affect the migration trajectory of oil particles. To maintain the simulation for physical properties, one possible extension for this work would be to consider the transformation process. Another possible future work is to incorporate weather factors, such as wind and air temperature, to address simulation issues with complex environments.

CRediT authorship contribution statement

Yuanfeng Lian: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Writing – original draft, Writing – review & editing. **Hanzhao Gao:** Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Lianen Ji:** Data curation, Formal analysis, Supervision, Validation, Writing – review & editing. **Shaohua Dong:** Data curation, Formal analysis, Supervision, Validation, Writing – review & editing.

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.

Acknowledgements

This research was partially funded by Strategic Cooperation Technology Projects of CNPC and CUPB: ZLZX2020-05.

References

[1] M. de Oliveira Soares, C.E.P. Teixeira, L.E.A. Bezerra, S.V. Paiva, T.C.L. Tavares, T.M. Garcia, J.T. de Araújo, C.C. Campos, S.M.C. Ferreira, H. Matthews-Cascon, et al., Oil spill in South Atlantic (Brazil): environmental and governmental disaster, Mar. Policy 115 (2020) 103879.

[2] D.M. Leme, M.A. Marin-Morales, Chromosome aberration and micronucleus frequencies in Allium cepa cells exposed to petroleum polluted water—a case study, Mutat. Res., Genet. Toxicol. Environ. Mutagen. 650 (2008) 80–86.
[3] Y. Ma, P.C.P. Medini, J.R. Nelson, R. Wei, T.H. Grubesic, J.A. Sefair, R. Maciejewski, A visual analytics system for oil spill response and recovery, IEEE Comput. Graph. Appl. 41 (2020) 91–100.
[4] F. Yu, Y. Yin, Simulation and 3d visualization of oil spill on the sea, in: 2011 IEEE International Symposium on VR Innovation, IEEE, 2011, pp. 213–216.
[5] L. Yang, Y. Chu, P. Liu, L. Wei, D. Guan, Numerical simulation of spill oil diffusion in offshore oil pipeline, in: 2018 3rd International Conference on Modelling, Simulation and Applied Mathematics (MSAM 2018), Atlantis Press, 2018, pp. 99–104.
[6] X. Feng, Y. Shi, Numerical study of containment of spilled medium-viscosity oil in wave–current flow, J. Eng. Mech. 145 (2019) 04019056.
[7] G. He, X. Lyu, K. Liao, Y. Li, L. Sun, A method for fast simulating the liquid seepage-diffusion process coupled with internal flow after leaking from buried pipelines, J. Clean. Prod. 240 (2019) 118167.
[8] H. Fu, L. Yang, H. Liang, S. Wang, K. Ling, Diagnosis of the single leakage in the fluid pipeline through experimental study and cfd simulation, J. Pet. Sci. Eng. 193 (2020) 107437.
[9] H. Ji, M. Xu, W. Huang, K. Yang, The influence of oil leaking rate and ocean current velocity on the migration and diffusion of underwater oil spill, Sci. Rep. 10 (2020) 1–15.
[10] A. Maslo, J. Panjan, D. Žagar, Large-scale oil spill simulation using the lattice Boltzmann method, validation on the Lebanon oil spill case, Mar. Pollut. Bull. 84 (2014) 225–235.
[11] A.S. Dhavalikar, P.C. Choudhari, Prediction of oil spill trajectory on the ocean surface using mathematical modeling, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 15 (2022) 5894–5905.
[12] D. Molteni, E. Vitanza, O.R. Battaglia, Smoothed particles hydrodynamics numerical simulations of droplets walking on viscous vibrating liquid, Comput. Fluids 156 (2017) 449–455.
[13] A. Soussi, C. Bersani, R. Sacile, D. Bouchta, A. El Amarti, H. Seghiouer, D. Nachite, J. Al Miys, An oil spill trajectory model: validation in the Mediterranean Sea, in: 2019 International Symposium on Systems Engineering (ISSE), IEEE, 2019, pp. 1–6.
[14] D. Violeau, C. Buvat, K. Abed-Meraïm, E. de Nanteuil, Numerical modelling of boom and oil spill with sph, Coast. Eng. 54 (2007) 895–913.
[15] M. Becker, M. Teschner, Weakly compressible sph for free surface flows, in: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2007, pp. 209–217.
[16] B. Solenthaler, R. Pajarola, Predictive-corrective incompressible sph, in: ACM SIGGRAPH 2009 Papers, 2009, pp. 1–6.
[17] M. Macklin, M. Müller, Position based fluids, ACM Trans. Graph. 32 (2013) 1–12.
[18] M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, M. Teschner, Implicit incompressible sph, IEEE Trans. Vis. Comput. Graph. 20 (2013) 426–435.
[19] J. Bender, D. Koschier, Divergence-free sph for incompressible and viscous fluids, IEEE Trans. Vis. Comput. Graph. 23 (2016) 1193–1206.
[20] S. Premžoe, T. Tasdizen, J. Bigler, A. Lefohn, R.T. Whitaker, Particle-based simulation of fluids, in: Computer Graphics Forum, Wiley Online Library, 2003, pp. 401–410.
[21] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, M. Teschner, Versatile rigid-fluid coupling for incompressible sph, ACM Trans. Graph. 31 (2012) 1–8.
[22] M. Müller, B. Solenthaler, R. Keiser, M. Gross, Particle-based fluid-fluid interaction, in: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2005, pp. 237–244.
[23] S. Liu, Q. Liu, Q. Peng, Realistic simulation of mixing fluids, Vis. Comput. 27 (2011) 241–248.
[24] B. Ren, C. Li, X. Yan, M.C. Lin, J. Bonet, S.M. Hu, Multiple-fluid sph simulation using a mixture model, ACM Trans. Graph. 33 (2014) 1–11.
[25] B. Ren, B. Xu, C. Li, Unified particle system for multiple-fluid flow and porous material, ACM Trans. Graph. 40 (2021) 1–14.
[26] S.M. Tayyebi, M. Pastor, M.M. Stickle, Two-phase sph numerical study of pore-water pressure effect on debris flows mobility: Yu Tung debris flow, Comput. Geotech. 132 (2021) 103973.
[27] E. Kazemi, S. Tait, S. Shao, Sph-based numerical treatment of the interfacial interaction of flow with porous media, Int. J. Numer. Methods Fluids 92 (2020) 219–245.
[28] H. Liang, S. He, Z. Chen, W. Liu, Modified two-phase dilatancy sph model for saturated sand column collapse simulations, Eng. Geol. 260 (2019) 105219.
[29] Y. Zhang, X. Ban, F. Du, W. Di, Fluidsnet: end-to-end learning for Lagrangian fluid simulation, Expert Syst. Appl. 152 (2020) 113410.
[30] L. Ladickỳ, S. Jeong, B. Solenthaler, M. Pollefeys, M. Gross, Data-driven fluid simulations using regression forests, ACM Trans. Graph. 34 (2015) 1–9.
[31] X. Jia, J. Willard, A. Karpatne, J.S. Read, J.A. Zwart, M. Steinbach, V. Kumar, Physics-guided machine learning for scientific discovery: an application in simulating lake temperature profiles, ACM/IMS Trans. Data Sci. 2 (2021) 1–26.
[32] N. Muralidhar, J. Bu, Z. Cao, L. He, N. Ramakrishnan, D. Tafti, A. Karpatne, Physics-guided deep learning for drag force prediction in dense fluid-particulate systems, Big Data 8 (2020) 431–449.
[33] K. Um, X. Hu, N. Thuerey, Liquid splash modeling with neural networks, in: Computer Graphics Forum, Wiley Online Library, 2018, pp. 171–182.
[34] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, et al., Interaction networks for learning about objects, relations and physics, Adv. Neural Inf. Process. Syst. 29 (2016).
[35] E. Haghighat, D. Amini, R. Juanes, Physics-informed neural network simulation of multiphase poroelasticity using stress-split sequential training, Comput. Methods Appl. Mech. Eng. 397 (2022) 115141.
[36] B. Ummenhofer, L. Prantl, N. Thuerey, V. Koltun, Lagrangian fluid simulation with continuous convolutions, in: International Conference on Learning Representations, 2019.
[37] J. Tompson, K. Schlachter, P. Sprechmann, K. Perlin, Accelerating Eulerian fluid simulation with convolutional networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 3424–3433.
[38] Z. Li, A.B. Farimani, Graph neural network-accelerated Lagrangian fluid simulation, Comput. Graph. 103 (2022) 201–211.
[39] A.M. Aly, M. Asai, Y. Sonda, Modelling of surface tension force for free surface flows in isph method, Int. J. Numer. Methods Heat Fluid Flow 23 (2013) 479–498.
[40] B. Solenthaler, R. Pajarola, Density contrast sph interfaces, 2008.
[41] A.M. Tartakovsky, P. Meakin, Pore scale modeling of immiscible and miscible fluid flows using smoothed particle hydrodynamics, Adv. Water Resour. 29 (2006) 1464–1478.
[42] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, P. Battaglia, Learning to simulate complex physics with graph networks, in: International Conference on Machine Learning, PMLR, 2020, pp. 8459–8468.
[43] E.S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, P. Stansby, Comparisons of weakly compressible and truly incompressible algorithms for the sph mesh free particle method, J. Comput. Phys. 227 (2008) 8417–8436.
[44] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint, arXiv:1412.6980, 2014.
[45] A.M. Tartakovsky, A. Panchenko, Pairwise force smoothed particle hydrodynamics model for multiphase flow: surface tension and contact line dynamics, J. Comput. Phys. 305 (2016) 1119–1146.
[46] Y. Li, J. Wu, R. Tedrake, J.B. Tenenbaum, A. Torralba, Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids, arXiv preprint, arXiv:1810.01566, 2018.
[47] C. Wu, Z. Chen, C. An, K. Lee, B. Wang, M.C. Boufadel, Z. Asif, Examining an oil spill plume mapping method based on satellite nir data, J. Environ. Inf. Lett. 5 (2021) 17–26.
[48] P.W. Battaglia, J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al., Relational inductive biases, deep learning, and graph networks, arXiv preprint, arXiv:1806.01261, 2018.