A Pioneering Neural Network Method for Efficient and Robust Fuel Sloshing Simulation in Aircraft

Yu Chen, Shuai Zheng*, Nianyi Wang, Menglong Jin, Yan Chang

School of Software Engineering, Xi'an Jiaotong University, Xi'an, 710049, China
xjtu_chenyu@163.com, shuaizheng@xjtu.edu.cn

Abstract

Simulating fuel sloshing within aircraft tanks during flight is crucial for aircraft safety research. Traditional methods based on Navier-Stokes equations are computationally expensive. In this paper, we treat fluid motion as point cloud transformation and propose the first neural network method specifically designed for simulating fuel sloshing in aircraft. This model is also the deep learning model that is the first to be capable of stably modeling fluid particle dynamics in such complex scenarios. Our triangle feature fusion design achieves an optimal balance among fluid dynamics modeling, momentum conservation constraints, and global stability control. Additionally, we constructed the Fueltank dataset, the first dataset for aircraft fuel surface sloshing. It comprises 320,000 frames across four typical tank types and covers a wide range of flight maneuvers, including multi-directional rotations. We conducted comprehensive experiments on both our dataset and the take-off scenario of the aircraft. Compared to existing neural network-based fluid simulation algorithms, we significantly enhanced accuracy while maintaining high computational speed. Compared to traditional SPH methods, our speed improved approximately 10 times. Furthermore, compared to traditional fluid simulation software such as Flow3D, our computation speed increased by more than 300 times.

Introduction

Aircraft fuel sloshing simulation intersects multiple disciplines, including computer graphics, 3D computer vision, fluid dynamics, etc. The fuel tank is a key component of an aircraft fuel system. During large-angle maneuvers, fuel sloshing can cause violent fluid deformation and shifts in the center of gravity, both of which can endanger flight safety. Consequently, the simulation of fuel sloshing during flight is a critical research topic in aviation applications.

There are two typical representations of fluid in the area of Computational Fluid Dynamics: grid-based methods and particle-based (also known as SPH). Traditional fuel sloshing simulations typically use grid-based fluid simulation software such as Ansys Fluent and Flow3D. The grid-based approach segments the fluid region into a grid of cells or voxels. This approach can maintain high numerical stability and has been widely applied in engineering. However, it requires significant computation time and storage, often taking weeks or even months to complete, and it struggles with handling complex boundaries. In recent years, fluid simulation methods based on SPH (Solenthaler and Pajarola 2009; Bender and Koschier 2015; Ye et al. 2019; Li, Liu, and Zheng 2018; Koschier et al. 2020) have gained traction in fuel sloshing simulation (Zheng et al. 2021; Calderon Sanchez et al. 2019), such as DualSPHysics (Alshaer, Rogers, and Li 2017; Pourabdian, Omidvar, and Morad 2017), due to their relatively higher efficiency and better handling of free surface flows. However, despite these advantages over grid-based methods, the inherently high computational cost of solving Navier-Stokes equations remains, which is highly inconvenient for practical applications.

Neural network-based approaches for learning physics have brought new vitality to this field (Ling, Kurzawski, and Templeton 2016; Tompson et al. 2017; Morton et al. 2018; Saha, Dash, and Mukhopadhyay 2021). By adopting the particle-based perspective of SPH and integrating methods from 3D computer vision, we treat fluids as point clouds with velocity vectors. This allows us to model complex fluid dynamics using neural networks, thereby avoiding the computationally intensive Navier-Stokes equations and significantly improving computational efficiency.

However, existing neural network methods (Ummenhofer et al. 2019; Shao, Loy, and Dai 2022; Prantl et al. 2022; Li et al. 2018) are not sufficiently robust, which is suitable only for simple scenarios, such as the freefall of fluid blocks or dam-break flows. In contrast, the internal structure of a fuel tank is much more complex, with ribs and holes that allow fuel flow between compartments. Current neural network methods lack sufficient learning capacity and the necessary constraints to guide the modeling of fluid dynamics in such intricate scenarios, making them prone to breakdown under such challenging conditions.

In general, the main contributions of this paper include:

- We propose a neural network that is the first to be robust enough to simulate fuel sloshing in complex fuel tanks, achieving a speed improvement of over 300 times compared to traditional fluid simulation software such as Flow3D, and nearly 10 times compared to traditional SPH methods.
- We introduce three key capabilities for fluid simulation

*Corresponding author

neural networks: fluid dynamics modeling, physical law constraints, and global stability control. To achieve an optimal balance among these factors, we design the Triangle Feature Fusion, significantly enhancing the accuracy of neural network-based fluid simulation methods.

- We have constructed the first fuel sloshing surface dataset, encompassing 320,000 frames across four typical tank types and covering various possible flight maneuvers. Compared to existing datasets, this dataset features more complex fluid motions and scenarios, including multi-directional rotations.
- We conducted comprehensive experiments on the Fueltank dataset and aircraft takeoff generalization scenarios, demonstrating that our network surpasses all previous methods in multiple evaluation metrics.

Related Work

Fluid Dynamics Modeling

When applying point cloud perspectives to SPH fluid particles, an intuitive initial approach might be to model the fluid particles and their interactions using the nodes and edges of a graph neural network (Shao, Loy, and Dai 2022; Li et al. 2018; Battaglia et al. 2016; Sanchez-Gonzalez et al. 2020; Mrowca et al. 2018). However, the complexity of processing large graph structures with numerous nodes and edges makes the convolution operations for each node in GNN highly time-consuming. Some graph-based fluid simulation methods are even slower than the traditional SPH algorithms. Given that the accuracy of current neural network-based methods has not yet surpassed traditional algorithms, it is essential for the computational speed of neural network algorithms to be significantly faster than that of traditional methods to maintain a competitive edge. Graph-based methods rely on particle discretization, but fluid mechanics in the real world are described by continuous partial differential equations rather than discrete graph structures. From this perspective, continuous convolution (Wang et al. 2018; Ummenhofer et al. 2019; Thomas et al. 2019; Winchenbach and Thuerey 2024) are more appropriate than graph convolutions for fluid dynamics modeling. Among these, the CConv (Ummenhofer et al. 2019) has been validated by (Prantl et al. 2022; Chen et al. 2024) as a stable foundation for fluid dynamics modeling, enhancing the neural network's ability to learn fluid dynamics.

Physical Law Constraints

Although the continuous convolution kernel can effectively model fluid dynamics, it lacks the constraints of physical laws and additional knowledge. This limitation restricts its accuracy ceiling (Cai et al. 2021). (Prantl et al. 2022) innovatively applied an anti-symmetric design to CConv, introducing the Anti-Symmetric Convolutional Kernel (ASCC) and demonstrating that this design effectively maintains the law of momentum conservation. However, the hard constraints introduced by the anti-symmetric design reduce ASCC's learning ability for fluid dynamics. To balance this trade-off, the authors introduced a multi-scale feature module, which is relatively time-consuming. In this way, the full potential of CConv and ASCC has not yet been fully explored. In this paper, we propose an innovative feature fusion method that achieves an optimal balance in this trade-off.

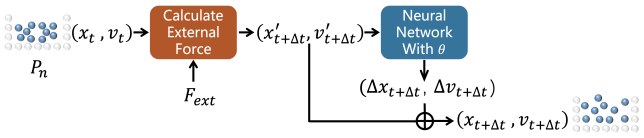


Figure 1: Position-based fluids scheme. It first computes external forces to obtain intermediate fuel particle states. Then, our neural network with trainable parameters θ predicts position and velocity changes induced by internal forces.

Global Stability Control

Besides the local interactions between fluid particles, we also note that learning global fluid features can enhance the neural network's performance in global stability control. Previous works can effectively extract global features from point clouds (Qi et al. 2017a,b; Chen and Shi 2023; Wang et al. 2019; Wu et al. 2024; Liu et al. 2022). In this task, we require a simple and efficient global feature extraction layer, which enables the neural network to broadly control the overall fluid motion within a reasonable range (e.g., fluid particles in a closed box should not move outside the box). The extracted global feature does not need to be highly detailed, as fluid dynamics primarily focuses on the interactions between local particles. The global feature serves as additional information to make the framework more robust.

Method

Our approach employs the position-based fluids (PBF) scheme(Macklin and Müller 2013; Zhang et al. 2015), as illustrated in Figure 1. Given a set of fuel particles P_n, it takes their positions x_t and velocities v_t at time step t as input, and outputs the final results x_{t+1} and v_{t+1} at time step $t+1$. The main goal of this task is to closely model the physical dynamics, enabling accurate prediction of the particles' positions and velocities at the subsequent time step.

Continuous Convolution Kernel

Inspired by (Ummenhofer et al. 2019; Prantl et al. 2022; Chen et al. 2024), our network incorporates two types of continuous convolution kernels, *CConv* and *ASCC*. For a point cloud P_n consisting of n points indexed by i, each corresponding to values f_i at positions x_i, the *CConv* at position $x \in P_n$ is described as follows:

$$\begin{aligned} CConv_g &= (f * g)(x) \\ &= \sum_{i \in \mathcal{N}(x,R)} a(x_i, x)\, f_i g(\Lambda(x_i - x)). \end{aligned} \tag{1}$$

Here, $\mathcal{N}(x, R)$ denotes the set of points located within a radius R centered at position x. The filter function g employs a mapping function $\Lambda(r)$ to transform a unit ball into a unit cube. The window function a scales with the distance between x_i and x, and becomes zero beyond distance R.

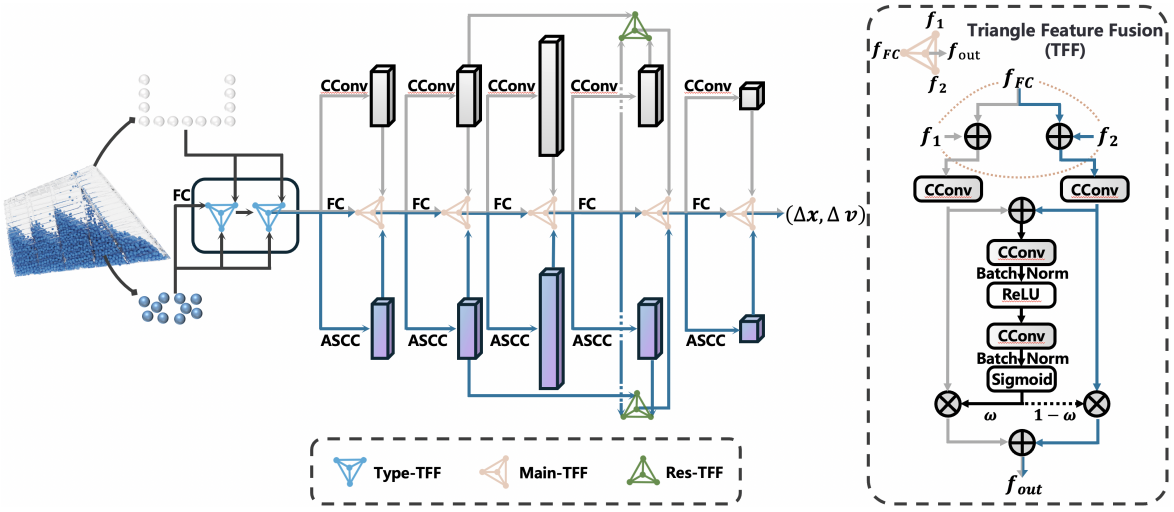


Figure 2: The architecture of our network and Triangle Feature Fusion (TFF). The three types of TFF modules share the same architecture but serve three distinct roles in different positions within the network. The Type-TFF handles type-aware input for fuel and tank particles. The Main-TFF integrates three pathways to balance fluid dynamics modeling, physical constraints, and global stability. The Res-TFF adds a residual connection between the second and fourth layers.

ASCC, a specialized variant of *CConv*, uses an antisymmetric kernel to enforce strict constraints, enabling the capture of fluid dynamics while conserving momentum. This is accomplished by halving the learnable kernel parameters along an axis, mirroring the second half about the kernel's center, and then negating the mirrored values. The ASCC formulation is as follows:

$$\begin{aligned} ASCC_{g_s} &= (f * g_s)(x) \\ &= \sum_{i \in \mathcal{N}(x,R)} a(x_i, x)(f + f_i) g_s(\Lambda(x_i - x)). \end{aligned} \tag{2}$$

Where g_s is the anti-symmetric continuous convolution kernel and f is the corresponding value at position x. For two points x and y in P_n, the interparticle force is given by:

$$F_{xy} = (f_x + f_y)(-g_s(x - y)) = -F_{yx}. \tag{3}$$

This ensures that:

$$\int_{x \in P_n} \int_{y \in P_n} F_{xy} = 0. \tag{4}$$

The antisymmetric kernel design enforces momentum conservation in *ASCC*, but this constraint limits its learning capacity, complicating the problem. Conversely, *CConv* offers better learning and modeling capabilities but lacks adherence to physical laws. To achieve a balance among these two aspects and global stability control, we propose the Triangle Feature Fusion in the next section.

Pipeline with Triangle Feature Fusion (TFF)

The architecture of our network is shown in Figure 2. We propose three types of TFF modules with a shared architecture, as illustrated in the right part of Figure 2. However, they serve different roles in different positions within the network. Initially, the fuel particles and tank particles, along with the global fluid features obtained through a fully connected (FC) layer, are input into the Type-TFF. This module is designed to handle the fluid-solid coupling for different types of inputs.

After passing through two Type-TFF modules, the output features are fed into three separate paths: CConv-based, ASCC-based, and FC. The network architecture consists of five layers. In each layer, features from the three paths undergo fusion through the Main-TFF before being passed to the next layer. This step is crucial for achieving the optimal balance of our three key capabilities. Both CConv and ASCC are based on continuous convolution, providing strong learning and modeling abilities for fluid dynamics. ASCC also introduces a hard constraint for momentum conservation, guiding the network to adhere to physical laws. Additionally, the FC layer extracts global features of the fluid. Incorporating this simple fully connected layer enables the neural network to broadly control the overall fluid motion within a reasonable range (e.g., fluid particles in a closed box should not move outside the box).

Notably, we incorporate a residual connection between the second and fourth layers through the Res-TFF, which benefits the training optimization process.

The architecture of the Triangle Feature Fusion Module is depicted on the right side of Figure 2. The features f_{FC} output from the FC layer are first concatenated with f_1 and f_2 separately. After passing through a *CConv* layer, the two resulting features are concatenated again and pass through several layers of continuous convolutions and ReLU activation. Then a sigmoid function is applied to obtain a fusion weight ω between 0 and 1. This weight is used to fuse the features of two pathways, resulting in f_{out}. The mathematical formula is described as follows:

$$f_1{}' = (f_1) \oplus (f_{FC}). \tag{5}$$

$$f_2{}' = (f_2) \oplus (f_{FC}). \tag{6}$$

$$\omega = \lambda(\phi(f_1{}') \oplus \phi(f_2{}')). \tag{7}$$

$$f_{out} = \omega \times f_1{}' + (1 - \omega) \times f_2{}'. \tag{8}$$

In Equation 7, λ comprises two *CConv* layers, *RELU* activation, and the *Sigmoid* function. ϕ represents the *CConv* function(Equation 1). In Type-TFF, f_1 and f_2 represent the fuel particles and the tank particles. In Main-TFF, they represent the features output by the *CConv* layer and the *ASCC* layer. In Res-TFF, they represent the features output by the second and fourth layers.

Training Strategy

Our loss function is defined as the mean absolute error (MAE) of the position values between the predicted and ground truth, weighted by the neighboring points count:

$$L(t) = \sum_i e^{-\frac{c_i}{c_{avg}}} \left\| x_i^{t+1} - \hat{x}_i^{t+1} \right\|_2^{\gamma}. \tag{9}$$

Here, x_i^{t+1} and $\hat{x}_i^{t+1}$ are the predicted and ground truth positions at time step $t+1$. c_i is the neighbor count for particle i. c_{avg} denotes the average neighbor count, set to 40.

Unlike previous methods (Chen et al. 2024; Ummenhofer et al. 2019) that compute loss and gradients over the next two frames, our network runs for $N = W + T$ steps, with each step's input based on the result of the previous prediction. Here, W represents the preprocessing frames and T represents the frames over which the loss and gradients are actually computed. During training, the value of W is adjusted according to training progress and the difficulty of the examples, but loss and gradients are always computed over the T steps, as formulated in Equation 10. This strategy helps the network self-correct over longer sequences, improving long-term prediction stability without increasing memory usage.

$$L_{sum} = \frac{1}{T} \sum_{t=0}^{T} L(t) \tag{10}$$

Fuel Surface Sloshing Dataset

Limitation of Existing Datasets

Existing SPH simulation datasets for fluid simulation (Ummenhofer et al. 2019; Chen et al. 2024; Shao, Loy, and Dai 2022) have the following limitations:

1) These datasets include only simple fluid motions, such as free fall or dam break in simple cubic containers, which are inadequate for capturing the fluid dynamics in complex scenarios.
2) These datasets only focus on scenarios where the system is horizontally placed and subjected to constant vertically downward gravity. However, in real-life scenarios, rotations are common, and forces may come from various directions, especially during the flight of aircraft fuel tanks.
3) They only focus on water-based simulations. In addition, the fluid motion in fuel tanks often involves surface sloshing rather than fluid blocks falling. This makes existing datasets inadequate for training neural networks to perform well in fuel tank environments.

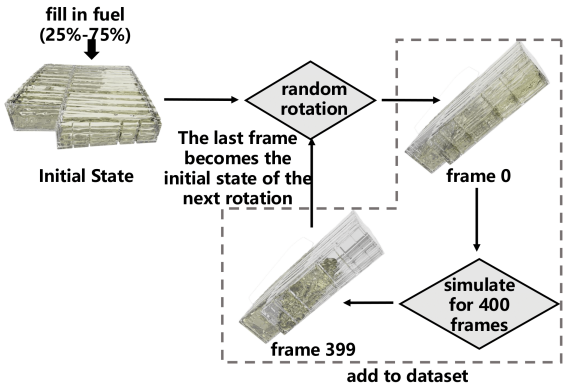


Figure 3: Construction strategy of Fueltank dataset. We use an iterative generation strategy, producing 400 frames per iteration. The final frame of the current iteration serve as the initial state for the next iteration.

Such simple fluid datasets are inadequate for capturing the complex fuel dynamics in tanks with intricate rib structures. To facilitate a more comprehensive study of aircraft fuel sloshing, it is necessary to construct a dedicated fuel surface dataset. Instead of the commonly used freefall or dam-break scenarios of fluid blocks, we adopt a surface scheme that better represents the actual fuel sloshing inside fuel tanks. Additionally, we employ an iterative strategy to generate data, enriching the initial surface shapes and sloshing scenarios.

Construction Strategy

We construct the Fueltank dataset based on four common types of aircraft fuel tanks: Tank Types I, II, III and IV. We generate the SPH fluid dataset using the traditional DFSPH solver (Bender and Koschier 2015), known for its high-fidelity simulations. The particle radius is set to 0.025 m and the fuel density to 782.885 kg/m^3.

Our construction strategy is illustrated in Figure 3. In each scenario, we first randomly fill a specific fuel quantity, ranging from 25% to 75% of the tank's capacity, in a stationary horizontal tank. Then we apply random rotations to the pitch and roll angles, from -90 to 90 degrees. Notably, these rotations are performed in a very short time, assuming the fuel remains stationary during the rotation. Once the rotation is complete, the fuel begins to flow for 400 frames. We add these frames to the dataset, and the final frame of the current rotation serve as the initial state for the next rotation. We generated 200 iterations for each fuel tank type. This iterative strategy makes our dataset encompass a diverse range of flight maneuvers, which significantly enhances the robustness of neural networks in fuel simulation. Examples from the Fueltank dataset are shown in Figure 4.

Experiments

Experimental Setup

We employ the PyTorch framework and the Adam optimizer for training, using a batch size of 2. The initial learning rate is set to 0.002 and is halved at steps 15000, 25000, ..., 55000.

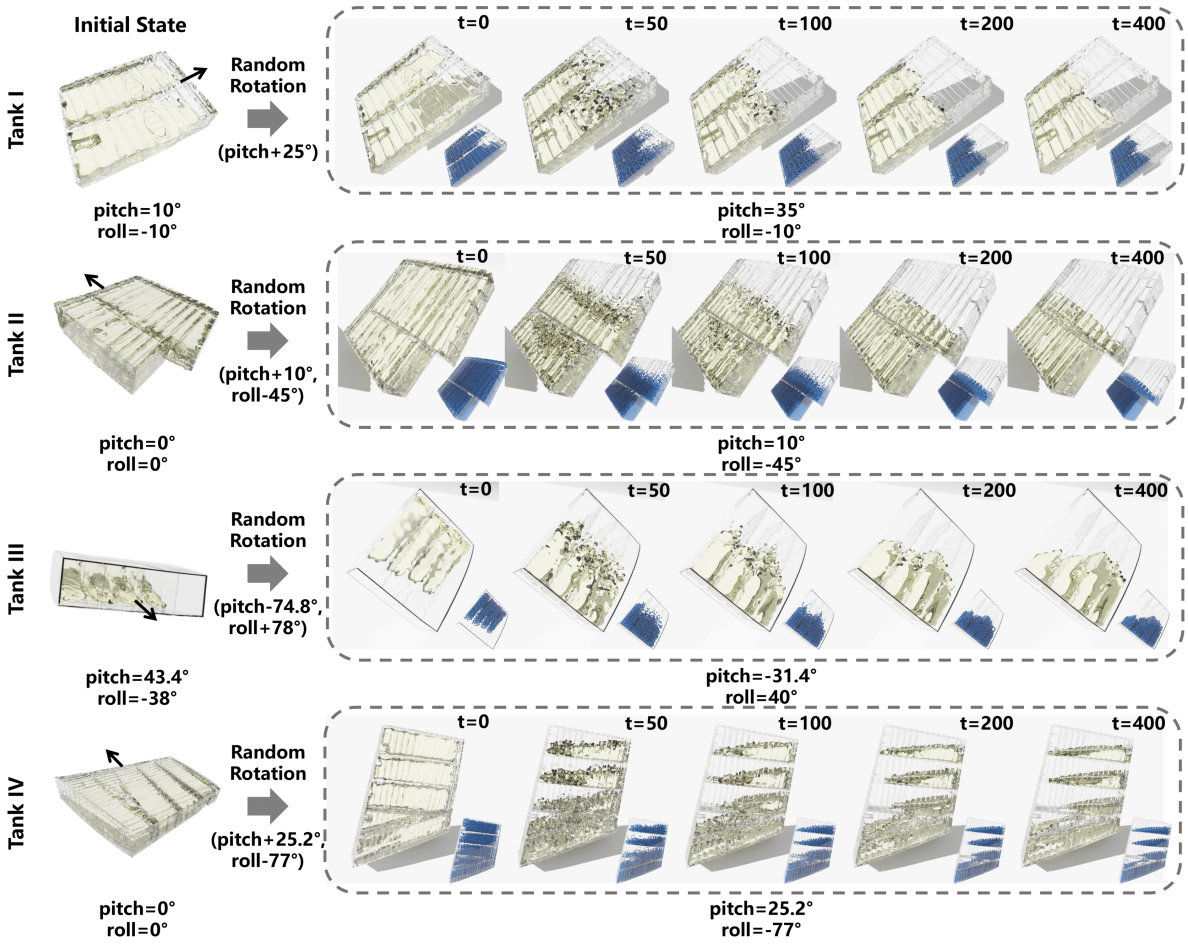


Figure 4: Examples of four tank types from the Fueltank dataset. The fuel tank undergoes random pitch and roll rotations at frame 0 and flows over the next 400 frames. The thumbnails depict the fuel surface from the SPH particle perspective.

The networks are trained for a total of 60,000 iterations on an NVIDIA A800. Specifically, the particle radius is set to $h = 0.025m$, and spherical filters with a spatial resolution of [4,4,4] are utilized, featuring a radius of $R = 4.5h$.

Evaluation Metrics

Initially, we use the Chamfer Distance (CD) and Earth Mover Distance (EMD) as our evaluation metrics. We calculated the CD and EMD for the next two frames to evaluate the accuracy of the network's short-term prediction.

Additionally, to assess the long-term stability of the fluid simulation, we computed the average distance from the ground-truth particles to the closest predicted particle over the entire sequence of n frames:

$$d^n = \frac{1}{N}\sum_{i=1}^{N} \min_{x^n \in X^n} \|\hat{x}_i^n - x^n\|_2 . \tag{11}$$

Where X^n is the set of predicted particle positions for the frame n, $\hat{x}_i^n$ is the ground-truth position of particle i, and N is the total number of particles.

Furthermore, to evaluate the network's adherence to physical laws, we also introduced the maximum density error, which reflects the fluid's incompressibility and stability:

$$e = \left| 1 - \frac{\max_i \rho(x_i)}{\max_i \rho(\hat{x}_i)} \right| \tag{12}$$

We conducted experiments on our proposed Fueltank dataset, using the traditional SPH method, DFSPH (Bender and Koschier 2015), as ground truth, as it closely approximates real-world fluid dynamics. We compared our method against the traditional methods and existing neural networks.

Experiments on Fueltank Dataset

The results of the comparative experiments are shown in Table 1 and Figure 5. Existing methods struggle in complex scenario. Specifically, the qualitative results of the continuous convolution methods, CConv (Ummenhofer et al. 2019) and DMCF (Prantl et al. 2022), show particles falling vertically continuously. This occurs because they overfit to a specific gravity direction and cannot handle varying rotational angles, causing the networks' predictions for internal forces ultimately converge to zero in all directions. According to the PBF scheme, as discussed in Figure 1, when the internal forces are zero, the fluid moves downward due to gravity.

Method	CD (mm)		EMD (mm)		n-frame Sequence Error d^n (mm)	Max Density Error (g/cm^3)	Time (s)
	t+1	t+2	t+1	t+2			
Grid-based	-	-	-	-	-	-	>50
SPH-based	-	-	-	-	-	-	>1
CConv	1.518	3.786	0.598	0.978	160.841	0.173	**0.021**
DMCF	1.472	3.488	0.185	0.323	128.902	0.025	0.602
TIE	1.528	3.734	0.196	0.372	140.813	0.089	1.213
DualFluidNet	1.232	3.169	0.157	0.328	35.318	0.013	0.199
Ours	**1.149**	**2.759**	**0.147**	**0.259**	**25.310**	**0.008**	<u>0.158</u>

Table 1: Quantitative experiments. Grid-based and SPH-based methods are traditional fluid simulation methods. Their simulation results can be regarded as ground truth but require long computation times. Existing neural networks cannot stably simulate fuel sloshing in complex fuel tanks, leading to substantial numerical errors, particularly in long-term predictions. Only our network achieves low errors while significantly reducing computation time.

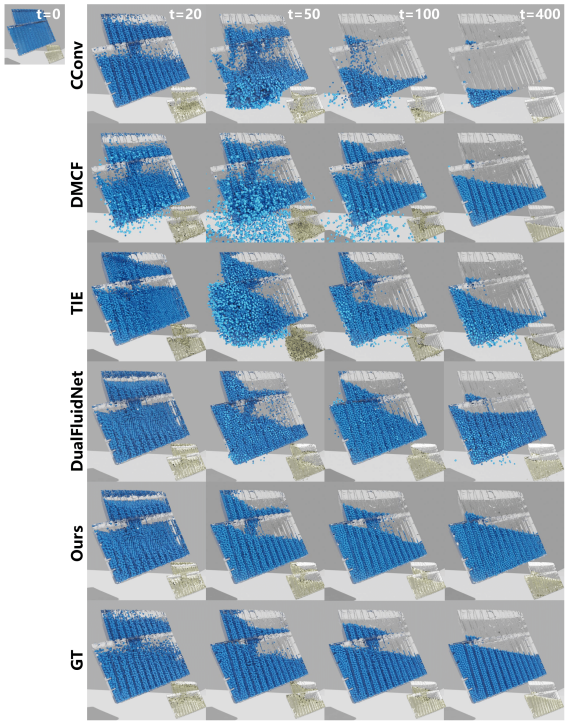
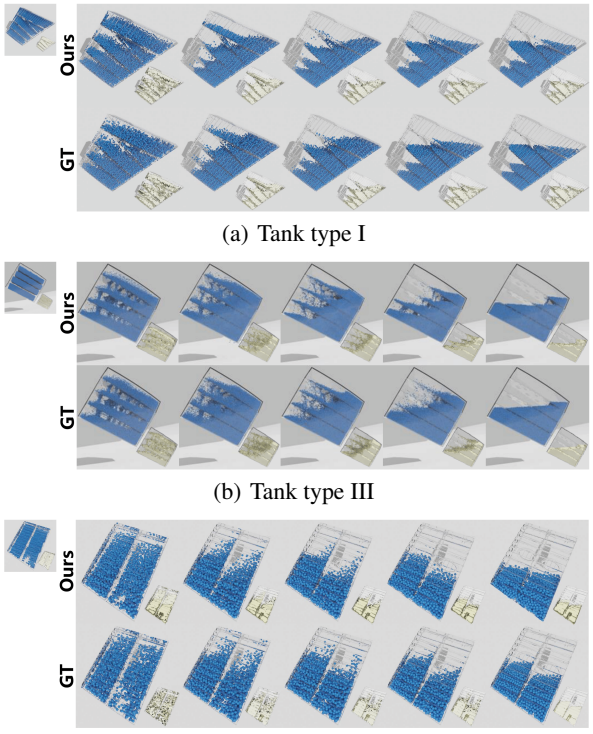


Figure 5: Qualitative experiments on Tank II show that existing neural network methods fail in complex scenarios, whereas our method provides stable fluid simulations comparable to traditional methods.



(a) Tank type I

(b) Tank type III

(c) Tank type IV

Figure 6: Qualitative experiments on other three tanks.

We also conducted experiments on the current state-of-the-art methods: TIE (Shao, Loy, and Dai 2022), based on graph convolution, and DualFluidNet (Chen et al. 2024), based on continuous convolution. TIE's inadequate fluid dynamics modeling causes particles to disperse erratically during simulation. Additionally, the large number of nodes and edges processed by the graph convolution and the incorporation of transformers result in significant computation time. For DualFluidNet, although it is the most stable among the existing methods, it still experiences particles escaping the fuel tank. Furthermore, since DualFluidNet computes loss and gradients only over the next two frames, its long-term stability is inferior to our network's W+T training strategy, as discussed in .

Our method is the first neural network robust enough to simulate fuel sloshing in complex fuel tanks, achieving accuracy comparable to traditional methods while significantly increasing speed. Like the stability of a triangle, our proposed triangle feature fusion achieves an optimal balance among the three key capabilities, providing exceptionally stable fluid simulation capabilities. Figure 6 shows the performance of our network in the other three fuel tanks.

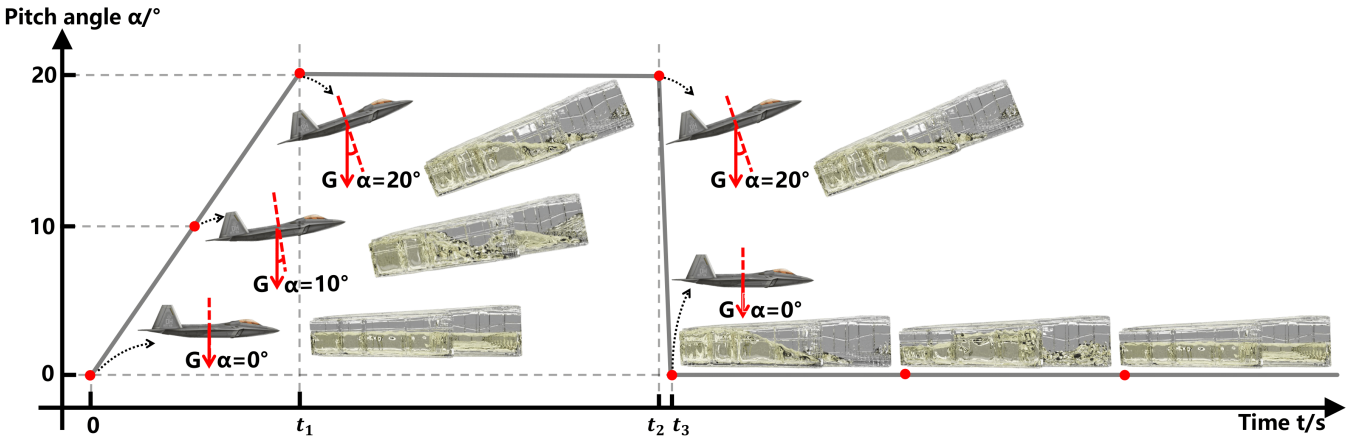


Figure 7: We simulated the aircraft's takeoff process, where the pitch angle gradually increases from 0 to 20 degrees at a pitch rate of 2 degrees per second from t_0 to t_1. It maintains 20 degrees from t_1 to t_2 , and then rapidly returns to 0 degrees from t_2 to t_3. From t_2 to t_3, which is a very short period, the fuel surface can be assumed as unchanged.

Method	Sequence Error d^n (mm)	Max Density Error (g/cm^3)
w/o Main-TFF	149.521	0.098
w/o Type-TFF	99.237	0.032
w/o Res-TFF	39.717	0.024
w/o CConv	181.518	0.144
w/o ASCC	87.922	0.054
w/o FC	33.317	0.013
Ours	**25.310**	**0.008**

Table 2: Quantitative results of the ablation study, showing the contributions of each TFF module type and the integration of the three features in the Main-TFF module.

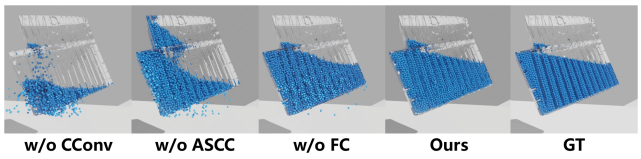


Figure 8: Qualitative results of the ablation study, further validating the necessity and contributions of our triangle feature fusion to network performance.

Ablation Study

We conducted an ablation study to verify the effectiveness of each module, as shown in Table 2 and Figure 8. Specifically, in the w/o CConv scenario, replacing CConv reduced the network's ability to model fluid dynamics, resulting in fluid collapse. In the w/o ASCC scenario, replacing ASCC retained some learning and modeling capability but lacked the guidance of physical laws, leading to deviations from normal physical behavior. Comparing w/o FC and Ours, adding the FC layer can improve global fluid control stability. The Type-TFF enhances the capability to handle type-aware input for fuel and tank particles. The Main-TFF integrates three pathways to balance fluid dynamics modeling, physical constraints, and global stability. The Res-TFF adds a residual connection to benefit the training optimization. Each module has a distinct and indispensable role, collectively forming an efficient and robust framework.

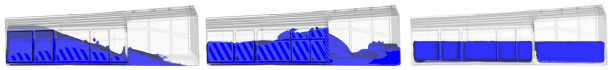


Figure 9: Simulation results after t_3 using traditional fluid simulation software Flow3D, which employs the traditional grid-based method.

Fuel Sloshing Simulation in Flight

To validate the generalization of our method in real aircraft takeoff scenarios, we simulated the aircraft's takeoff process, where the pitch angle gradually increases from 0 to 20 degrees, stabilizes, and then rapidly returns to 0 degrees, as shown in Figure 7. It shows that our network can generalize to real-world fuel simulation scenarios robustly and stably. Additionally, we compared the simulation results after t_3 using the famous fuel simulation software Flow3D, as shown in Figure 9. For the entire simulation, Flow3D takes nearly 10 hours, while our method requires only 2 minutes and achieves comparable accuracy.

Conclusion

In this paper, we proposed the first neural network robust enough for fuel sloshing simulation in complex fuel tanks. Our approach optimally balances fluid dynamics modeling, physical law constraints, and global stability control through a triangle feature fusion. We also constructed the first comprehensive fuel surface sloshing dataset. Our network significantly enhances simulation speed while ensuring accuracy and stability in aircraft applications. This network can also generalize to other fluids and scenarios, providing a robust and efficient framework for fluid simulation.

References

Alshaer, A. W.; Rogers, B. D.; and Li, L. 2017. Smoothed Particle Hydrodynamics (SPH) modelling of transient heat transfer in pulsed laser ablation of Al and associated free-surface problems. *Computational Materials Science*, 127: 161–179.

Battaglia, P.; Pascanu, R.; Lai, M.; Jimenez Rezende, D.; et al. 2016. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems*, 29.

Bender, J.; and Koschier, D. 2015. Divergence-free smoothed particle hydrodynamics. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics symposium on computer animation*, 147–155.

Cai, S.; Mao, Z.; Wang, Z.; Yin, M.; and Karniadakis, G. E. 2021. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12): 1727–1738.

Calderon Sanchez, J.; González, L.; Marrone, S.; Colagrossi, A.; and Gambioli, F. 2019. A SPH simulation of the sloshing phenomenon inside fuel tanks of the aircraft wings.

Chen, Y.; and Shi, P. 2023. Rotation-Invariant Completion Network. *arXiv preprint arXiv:2308.11979*.

Chen, Y.; Zheng, S.; Jin, M.; Chang, Y.; and Wang, N. 2024. DualFluidNet: An attention-based dual-pipeline network for fluid simulation. *Neural Networks*, 177: 106401.

Koschier, D.; Bender, J.; Solenthaler, B.; and Teschner, M. 2020. Smoothed particle hydrodynamics techniques for the physics based simulation of fluids and solids. *arXiv preprint arXiv:2009.06944*.

Li, B.; Liu, H.; and Zheng, S. 2018. Multidisciplinary topology optimization for reduction of sloshing in aircraft fuel tanks based on SPH simulation. *Structural and multidisciplinary optimization*, 58: 1719–1736.

Li, Y.; Wu, J.; Tedrake, R.; Tenenbaum, J. B.; and Torralba, A. 2018. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. *arXiv preprint arXiv:1810.01566*.

Ling, J.; Kurzawski, A.; and Templeton, J. 2016. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807: 155–166.

Liu, Z.; Wu, S.; Xu, C.; Wang, X.; Zhu, L.; Wu, S.; and Feng, F. 2022. Copy motion from one to another: Fake motion video generation. *arXiv preprint arXiv:2205.01373*.

Macklin, M.; and Müller, M. 2013. Position based fluids. *ACM Transactions on Graphics (TOG)*, 32(4): 1–12.

Morton, J.; Jameson, A.; Kochenderfer, M. J.; and Witherden, F. 2018. Deep dynamical modeling and control of unsteady fluid flows. *Advances in Neural Information Processing Systems*, 31.

Mrowca, D.; Zhuang, C.; Wang, E.; Haber, N.; Fei-Fei, L. F.; Tenenbaum, J.; and Yamins, D. L. 2018. Flexible neural representation for physics prediction. *Advances in neural information processing systems*, 31.

Pourabdian, M.; Omidvar, P.; and Morad, M. R. 2017. Multiphase simulation of liquid jet breakup using smoothed particle hydrodynamics. *International Journal of Modern Physics C*, 28(04): 1750054.

Prantl, L.; Ummenhofer, B.; Koltun, V.; and Thuerey, N. 2022. Guaranteed conservation of momentum for learning particle-based fluid dynamics. *Advances in Neural Information Processing Systems*, 35: 6901–6913.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Saha, P.; Dash, S.; and Mukhopadhyay, S. 2021. Physics-incorporated convolutional recurrent neural networks for source identification and forecasting of dynamical systems. *Neural Networks*, 144: 359–371.

Sanchez-Gonzalez, A.; Godwin, J.; Pfaff, T.; Ying, R.; Leskovec, J.; and Battaglia, P. 2020. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, 8459–8468. PMLR.

Shao, Y.; Loy, C. C.; and Dai, B. 2022. Transformer with implicit edges for particle-based physics simulation. In *European Conference on Computer Vision*, 549–564. Springer.

Solenthaler, B.; and Pajarola, R. 2009. Predictive-corrective incompressible SPH. In *ACM SIGGRAPH 2009 papers*, 1–6.

Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6411–6420.

Tompson, J.; Schlachter, K.; Sprechmann, P.; and Perlin, K. 2017. Accelerating eulerian fluid simulation with convolutional networks. In *International Conference on Machine Learning*, 3424–3433. PMLR.

Ummenhofer, B.; Prantl, L.; Thuerey, N.; and Koltun, V. 2019. Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations*.

Wang, S.; Suo, S.; Ma, W.-C.; Pokrovsky, A.; and Urtasun, R. 2018. Deep Parametric Continuous Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5): 1–12.

Winchenbach, R.; and Thuerey, N. 2024. Symmetric Basis Convolutions for Learning Lagrangian Fluid Mechanics. *arXiv preprint arXiv:2403.16680*.

Wu, S.; Liu, Z.; Zhang, B.; Zimmermann, R.; Ba, Z.; Zhang, X.; and Ren, K. 2024. Do as I Do: Pose Guided Human Motion Copy. *IEEE Transactions on Dependable and Secure Computing*.

Ye, T.; Pan, D.; Huang, C.; and Liu, M. 2019. Smoothed particle hydrodynamics (SPH) for complex fluid flows: Recent developments in methodology and applications. *Physics of Fluids*, 31(1).

Zhang, S.; Yang, X.; Wu, Z.; and Liu, H. 2015. Position-based fluid control. In *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*, 61–68.

Zheng, S.; Gao, F.; Zhang, Z.; Liu, H.; and Li, B. 2021. Topology optimization on fuel tank rib structures for fuel sloshing suppression based on hybrid fluid–solid SPH simulation. *Thin-Walled Structures*, 165: 107938.