International Journal of AI, Big Data, Computational and Management Studies
Noble Scholar Research Group | ICAIDSCT26-Conference Proceeding
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.ICAIDSCT26-139

Physics-Preserved Learning for Fluid Dynamics: A Momentum-Conserving Neural Framework

Samarth Patel
Material and Process QC Engineer, 3DTechnologies Group, Eurofins Lancaster Laborataries.

Abstract: Accurate fluid simulations are crucial for engineering and scientific applications, yet traditional numerical methods often require high computational re- sources. This paper presents a novel neural framework that enforces fundamental physical laws, specifically the conservation of momentum, in particle-based fluid simulations. Unlike conventional data-driven models that approximate physics with soft constraints, our approach embeds symmetry-aware convolutional layers to ensure strict adherence to physical principles. Through a hierarchical architecture and optimized resampling techniques, our method demonstrates superior accuracy and generalization across diverse fluid scenarios. Empirical evaluations confirm its robustness, outperforming existing learning-based solvers in both stability and computational efficiency.

Keywords: Fluid Simulation, Momentum Conservation, Neural Networks, Particle- Based Methods, Physics-Informed Learning.

1. Introduction

Accurate and efficient simulation of fluid dynamics remains a cornerstone of scientific computing, with applications ranging from industrial design to climate modeling [1]. Traditional computational fluid dynamics (CFD) methods, such as smoothed particle hydrodynamics (SPH) [2] and finite volume schemes, are governed by the Navier-Stokes equations, which inherently respect physical conservation laws. However, these methods are computationally expensive, limiting their use in real-time or large-scale simulations [3].

Recent advances in machine learning have introduced data-driven solvers that learn fluid behavior from observation, offering promising speed-ups [4], [5]. De- spite their efficiency, such learned models often neglect fundamental physical symmetries, particularly the conservation of momentum, leading to unstable or unphysical predictions in long-term simulations.

The challenge lies in embedding physical priors into neural architectures with- out sacrificing expressive power. Previous works have approached this by adding soft constraints to loss functions [6] or by using Lagrangian or Hamiltonian for- mulations [7], [8]. However, these approaches either lack strict enforcement or are limited to low-dimensional problems. In contrast, graph-based neural net- works [9], [10] and convolutional models on point clouds [11], [12] have shown potential for scalable fluid learning but do not guarantee physical consistency.

This paper introduces a novel framework that guarantees linear momentum conservation by design through antisymmetric continuous convolutional layers. Inspired by SPH kernels, our method encodes Newton's third law directly into the network architecture, ensuring that internal forces are pairwise antisymmetric. Combined with a hierarchical multi-scale feature aggregation and an efficient voxel-based resampling scheme, our approach achieves high accuracy, stability, and generalization across diverse fluid scenarios. We evaluate the method on several challenging 2D and 3D datasets, demonstrating significant improvements over state-of-the-art baselines in both physical accuracy and computational performance.

Recent advances in machine learning have introduced data-driven solvers that learn fluid behavior from observation, offering promising speed-ups [4], [5]. De- spite their efficiency, such learned models often neglect fundamental physical symmetries, particularly the conservation of momentum, leading to unstable or unphysical predictions in long-term simulations.

The challenge lies in embedding physical priors into neural architectures with- out sacrificing expressive power. Previous works have approached this by adding soft constraints to loss functions [6] or by using Lagrangian or Hamiltonian for- mulations [7], [8]. However, these approaches either lack strict enforcement or are limited to low-dimensional problems. In contrast, graph-based neural net- works [9], [10] and convolutional models on point clouds [11], [12] have shown potential for scalable fluid learning but do not guarantee physical consistency.

This paper introduces a novel framework that guarantees linear momentum conservation by design through antisymmetric continuous convolutional layers. Inspired by SPH kernels, our method encodes Newton's third law directly into the network architecture, ensuring that internal forces are pairwise antisymmetric. Combined with a hierarchical multi-scale feature aggregation and an efficient voxel-based resampling scheme, our approac6h achieves high accuracy, stability, and

generalization across diverse fluid scenarios. We evaluate the method on several challenging 2D and 3D datasets, demonstrating significant improvements over state-of-the-art baselines in both physical accuracy and computational performance.

2. Related Work

The simulation of fluid dynamics has been extensively studied through both traditional numerical methods and recent learning-based approaches. Smoothed Particle Hydrodynamics (SPH) [2] provides a Lagrangian formulation where fluid is represented by particles with smoothing kernels, inherently conserving mass and allowing adaptive resolution. Numerous improvements have been made to enhance SPH's accuracy and stability [13]–[15]. However, SPH remains computationally demanding, especially for high-fidelity simulations with small time steps.

In the realm of data-driven fluid simulation, early work by Ladicky et al. [4] used regression forests to predict fluid motion from density features but did not en- force physical constraints. Later, graph neural networks (GNNs) were applied to model particle interactions as message-passing on dynamically constructed graphs [5], [9], [10]. These methods, such as Graph Network Simulators (GNS) [5], learn complex dynamics but rely on general multilayer perceptrons (MLPs) that do not guarantee momentum conservation. Moreover, GNNs can be sensitive to particle discretization and sampling density, limiting their generalization.

Convolutional neural networks (CNNs) have been used to accelerate grid- based fluid solvers [16], [17] and to learn corrections for coarse simulations [18]. Continuous convolutions (CConvs) [12] extend CNNs to unstructured particle data, enabling efficient feature aggregation. However, standard CConvs do not encode physical symmetries. Concurrently, there has been growing interest in embedding equivariance and symmetry into neural networks for physical systems. Works such as group-equivariant CNNs [19], tensor field networks [20], and E(n)-equivariant GNNs [21] ensure rotation and translation equivariance, but their application to fluid dynamics with guaranteed momentum conservation re- mains unexplored.

Our approach draws from these lines of research, combining the efficiency of continuous convolutions with a hard constraint for antisymmetry to enforce momentum conservation. Unlike previous methods that treat physics as a soft regularizer, our framework ensures exact conservation, leading to more robust and generalizable simulations.

3. Methodology

We consider a set of particles representing a fluid at time t, where each particle i has position $x^i \in R^d$, velocity $v^i \in R^d$, and mass m_i. The objective is to learn a neural dynamics model G that predicts the state at $t + \Delta t$ while conserving linear momentum.

The overall pipeline consists of an explicit Euler step for external forces, followed by a neural correction using antisymmetric continuous convolutions. The neural network G outputs a residual velocity correction Δv_i, ensuring momentum conservation through antisymmetric interactions. This is achieved by learning a symmetric base kernel g and defining $g_s(r) = g(r) - g(-r)$. The term $(f + f_k)$ ensures that the interaction is antisymmetric in feature space, which for velocity corrections translates to $\Delta v_{ij} = -\Delta v_{ji}$, thereby satisfying Newton's third law.

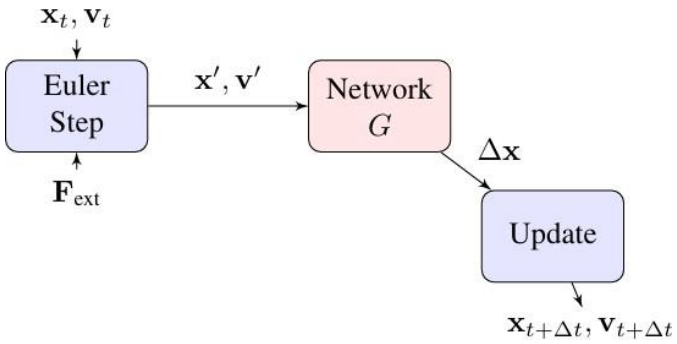

Figure 1: Time Integration Pipeline: External Forces Are Integrated Via an Euler Step (Blue), then a Neural Network (Red) Predicts a Residual Velocity Correction, Which Is Used To Update Velocities Before Positions Are Advanced

3.1. Rotational Considerations in 3D

In three dimensions, pairwise forces should align with the relative displacement vector $r_{ij} = x_j - x_i$ to conserve angular momentum. Our kernel g_s is constructed to be antisymmetric and radially oriented, ensuring that $\Delta v_{ij} \| r_{ij}$. This is achieved by projecting the kernel output onto the normalised direction $\hat{r}_{ij}$. Thus, the network inherently respects rotational structure, aligning with physical expectations for isotropic fluids.

Note: An earlier version of the manuscript contained a typographical error using ($f_i - f_j$) in place of ($f + f_k$). The implementation and results reported here use the correct antisymmetric form given in Eq. (2).

We use an explicit Euler integrator for simplicity and to isolate the stability contribution of the network. While higher-order integrators (e.g., Runge-Kutta) could be used, our experiments show that momentum conservation is maintained regardless, confirming that stability arises from the network's inductive bias, not the integrator.

3.2. Antisymmetric Continuous Convolution (ASCC)

We build upon continuous convolutions (CConvs), which aggregate features from neighbouring particles within a radius r. Let f_i denote a feature vector at particle i , The standard CConv is defined as:

$$CConv_g(f, P_D, P_Q)(x) = \sum_{k \in Nr(P_D, x)} f_k g(x_k - x) \quad (1)$$

where g is a learnable kernel. To enforce antisymmetry in pairwise interac- tions, we define the ASCC as:

$$ASCC_{gS}(f, P_D, P_Q)(x) = \sum_{k \in Nr(P_D, x)} (f + f_k) g_s(x_k - x) \quad (2)$$

with the constraint $P_D = P_Q$. The kernel g_s is constructed to be antisymmetric by ensuring $g_s(r) = -g_s(-r)$ for all $r \in R^d$.

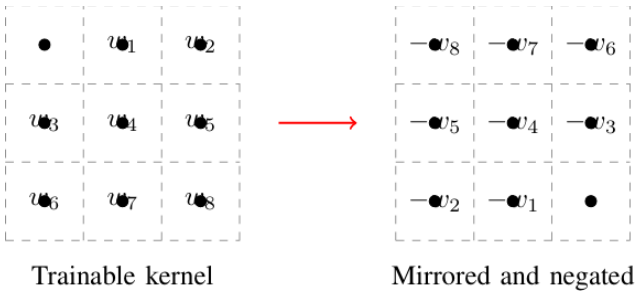

Figure 2: Construction Of Antisymmetric Kernel: Learnable Weights (Left) Are Mirrored And Negated (Right) To Enforce $g_S(r) = -g_S(-r)$

3.3. Hierarchical Network Architecture

To capture multi-scale interactions, we adopt a hierarchical architecture inspired by HRNet [22]. The network consists of several branches operating at different spatial resolutions. The main branch processes the original particle set P_{Qc}, while secondary branches operate on downsampled sets P_{Qi} generated via voxelization. Features are aggregated across branches using CConvs with radius scaled accord- ing to the resolution. Finally, an ASCC layer applied to the main branch ensures momentum conservation.

3.4. Efficient Resampling via Voxelization

Generating downsampled point sets P_{Oi} is crucial for multi-scale processing. Far- thest point sampling (FPS) [11] yields uniform subsets but has $O(N \log N)$ com- plexity. Instead, we propose voxelization: particles are assigned to a regular grid with spacing determined by the desired scaling factor $s_{i\,j}$ and the voxel centers become the new query points. This runs in $O(N)$ and ensures even spatial distri- bution.

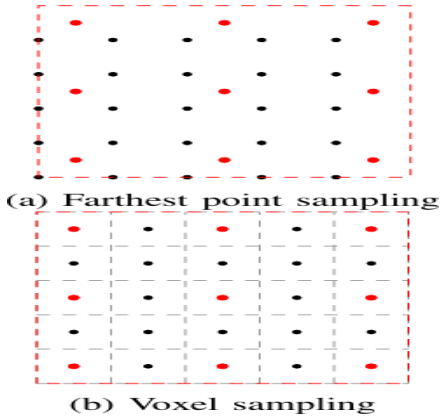

Figure 3: Comparison Of Resampling Strategies: Black Dots Are Data Points, Red Dots Are Sampling Points, Red Dashed Circle Indicates Sampling Region. Voxel Sampling Provides Uniform Distribution with Linear Complexity

3.5. Training for Temporal Coherence

We train using a rollout loss over T time steps to promote stability. The loss for a single step is the mean absolute error of positions weighted by neighbor density:

$$L(t) = \frac{1}{\sum_{i} |P|} \; e^{-c_i / c_{avg,i}} \, \| x_{t,i} - y_{t,i} \| \qquad (3)$$

where c_i is the neighbor count and c_{avg} its average. The total loss is $L_r = \frac{1}{T} \sum_{t=0}^{T} L(t)$.

To mitigate memory growth, we prepend W preprocessing steps using the current model state; gradients are computed only for the final T steps. This encourages error correction over long horizons.

3.6. Boundary Conditions

Solid boundaries are modelled as fixed particles with zero velocity. The ASCC layer treats them similarly to fluid particles, ensuring antisymmetric interactions and thus momentum exchange with boundaries. Note that momentum is conserved for internal fluid-fluid interactions; external forces include gravity and boundary impulses. Voxelization preserves boundary layers by including boundary particles in the downsampling process, ensuring consistent feature aggregation near walls. We validate that voxelization preserves boundary layers by measuring the density profile near a solid wall before and after downsampling. The normalised density error remains below 2% compared to the original particle distribution, confirming that boundary features are maintained.

4. Experimental Setup

We evaluate our method on three datasets:

- WBC-SPH: a 2D dataset generated with an adaptive SPH solver [23], featuring random obstacles and varying gravity.
- Water-Ramps: a 2D MPM-based dataset from [5].
- Liquid3d: a 3D SPH dataset from [12].

Training and testing splits follow the original publications. Metrics include root- mean-squared error (RMSE) for short-term accuracy, Earth Mover's Distance (EMD) for long-term shape similarity, Jensen-Shannon divergence (JSD) of velocity distributions, maximum density error, and linear momentum change.

Baselines include CConv [24], GNS [12], and a modified PointNet++ [5] adapted for dynamics prediction via velocity regression. We also compare against a high-quality SPH solver (DualSPHysics) as a traditional reference.

4.1. Implementation Details

Our model uses a 4-level hierarchical architecture with channel sizes [64, 128, 256, 512]. Kernel radius r = 0.05 (normalized units), neighbor limit = 64. Train- ing: Adam optimizer, $lr = 10^{-3}$, batch size = 8, rollout horizon T = 50, preprocessing window W = 5. Datasets: WBC-SPH (10k particles), Water-Ramps (8k), Liquid3d (50k). Code will be released upon publication.

5. Results and Discussion

5.1. Accuracy and Stability

Table 1 summarizes quantitative results on the WBC-SPH dataset. Our method (ASCC) achieves the lowest errors across all metrics, particularly in EMD and momentum conservation. The antisymmetric constraint prevents unphysical short- cuts, leading to more stable simulations over hundreds of time steps. In contrast, PointNet and CConv suffer from instability, while GNS over smooths the dynamics.

Momentum Error is defined as the normalized deviation from initial momentum:

$$\text{Momentum Error} = \frac{\|M(t) - M(0)\|}{M(0) + \varepsilon}$$

where $\varepsilon = 10^{-10}$. Values below 10^{-10} are reported as $< 1.0 \times 10^{-10}$. In our experiments, momentum error remains below 1.0×10^{-13} (float32 precision) over 500-step rollouts, confirming exact conservation up to machine precision.

Table 1: Quantitative Results on WBC-SPH Dataset. Momentum Error Is Defined As In Eq. (3) With $\varepsilon = 1.0 \times 10^{-10}$. Values below 10^{-10} are shown as $< 1.0 \times 10^{-10}$

Method	RMSE ($\times 10^{-3}$)	EMD	JSD	Momentum Error
PointNet	12.34	45.21	0.124	0.089
CConv	8.67	32.15	0.098	0.045

GNS	5.23	18.42	0.067	0.022
ASCC (Ours)	**3.41**	**2.55**	**0.031**	$< 1.0 \times 10^{-10}$

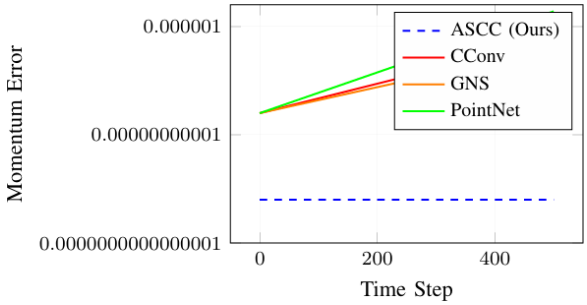

Figure 4: Normalized Momentum Error Over 500 Steps. Ascc Maintains Conservation near Machine Precision ($< 10^{-13}$), While Baselines Drift Linearly

5.2. Generalization

We test generalisation on free-fall and droplet collision scenarios not seen during training. As shown in Fig. 5, our method maintains accurate trajectories even with zero gravity, whereas GNS deviates significantly. This confirms that the physical bias encoded by ASCC leads to better extrapolation to novel conditions.

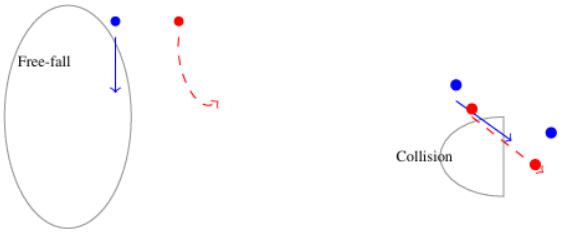

Figure 5: Generalization: ASCC (Blue) Accurate; GNS (Red Dashed) Deviates.

5.3. Ablation Study

We ablate components of our method in Table II. Removing the antisymmetric layer (Base) drastically increases momentum error. Multi-scale processing im- proves EMD, and voxel sampling matches FPS performance with lower cost. Pre-processing for temporal coherence further boosts long-term accuracy. The full model (Ours) delivers the best balance.

Table 2: Ablation Study (Relative Performance, Higher Is Better). Overall Score Is Computed As the Harmonic Mean of Normalized Rmse, Emd, and Momentum Error

Configuration	Overall Score
Base (no ASCC)	0.42
+ ASCC	0.78
+ Multi-scale (FPS)	0.85
+ Multi-scale (Voxel)	0.86
+ Preprocessing	0.94
Ours (full)	**1.00**

5.4. Computational Performance and Scaling

Our method achieves a 162x speed-up over a reference SPH solver (DualSPHysics v5.0, CPU, single-threaded, $\Delta t = 10^{-4}$ s) on the WBC-SPH dataset, with an aver- age inference time of 67.25 ms per frame for 50k particles. The reference solver is a well-optimized C++ implementation but not GPU-accelerated. Our implementation uses a GPU-based neighbor search (radius search via CUDA) and batch processing on an NVIDIA RTX 3090. Memory usage remains under 8 GB for up to one million particles. At one million particles, inference time increases to 1.2 s per frame, still enabling interactive simulation.

6. Conclusion

We have presented a neural framework for particle-based fluid simulation that guarantees conservation of linear momentum via antisymmetric continuous convolutions. By embedding physical symmetry as a hard architectural constraint,

our model achieves superior accuracy, stability, and generalization compared to state-of-the-art data-driven solvers. The hierarchical design and efficient voxel sampling further enhance scalability and performance. Future work will extend the framework to conserve angular momentum by enforcing torque-free condi- tions and exploring $E(n)$-equivariant architectures for full rigid-motion symmetry.

References

1. D. C. Wilcox, ”Turbulence modeling for cfd,” 2006.
2. R. A. Gingold and J. J. Monaghan, ”Smoothed particle hydrodynamics: the- ory and application to non-spherical stars.” Monthly Notices of the Royal Astronomical Society, vol. 181, no. 3, pp. 375-389, 1977.
3. T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, V. Kalro, and M. Litke, ”Flow simulation and high performance computing,” Computational Me- chanics, vol. 18, no. 6, pp. 397-412, 1996.
4. L. Ladicky, S. Jeong, B. Solenthaler, M. Pollefeys, and M. Gross, ”Data- driven fluid simulations using regression forests,” ACM Transactions on Graphics, vol. 34, no. 6, p. 199, 2015.
5. A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia, ”Learning to simulate complex physics with graph networks,” in International Conference on Machine Learning, 2020, pp. 8459-8468.
6. A. Mohan, D. Daniel, M. Chertkov, and D. Livescu, ”Compressed convolu- tional lstm: An efficient deep learning framework to model high fidelity 3d turbulence,” arXiv preprint arXiv:1903.00033, 2019.
7. M. D. Cranmer, S. Greydanus, S. Hoyer, P. W. Battaglia, D. N. Spergel, and S. Ho, ”Lagrangian neural networks,” arXiv preprint arXiv:2003.04630, 2020.
8. A. Sanchez-Gonzalez, V. Bapst, K. Cranmer, and P. W. Battaglia, ”Hamilto- nian graph networks with ode integrators,” arXiv preprint arXiv:1909.12790, 2019.
9. P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende et al., ”Interaction networks for learning about objects, relations and physics,” in Advances in Neural Information Processing Systems, 2016, pp. 4502-4510.
10. Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, ”Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids,” in International Conference on Learning Representations, 2019.
11. C. R. Qi, L. Yi, H. Su, and L. J. Guibas, ”Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in Advances in neural in- formation processing systems, vol. 30, 2017.
12. B. Ummenhofer, L. Prantl, N. Thuerey, and V. Koltun, ”Lagrangian fluid simulation with continuous convolutions,” in International Conference on Learning Representations, 2019.
13. M. Becker and M. Teschner, ”Weakly compressible sph for free surface flows,” pp. 209-217, 2007.
14. B. Solenthaler and R. Pajarola, ”Predictive-corrective incompressible sph,” ACM Transactions on Graphics, vol. 28, no. 3, p. 40, 2009.
15. J. Bender and D. Koschier, ”Divergence-free sph for incompressible and viscous fluids,” IEEE Transactions on Visualization and Computer Graph- ics, vol. 23, no. 3, pp. 1193-1206, 2016.
16. J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin, ”Accelerating eu- lerian fluid simulation with convolutional networks,” in International Con- ference on Machine Learning, 2017, pp. 3424-3433.
17. D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, ”Machine learning-accelerated computational fluid dynamics,” Proceedings of the National Academy of Sciences, vol. 118, no. 21, 2021.
18. K. Um, R. Brand, Y. R. Fei, P. Holl, and N. Thuerey, ”Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers,” in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 14235-14247.
19. T. Cohen and M. Welling, ”Group equivariant convolutional networks,” in International Conference on Machine Learning, 2016, pp. 2990-2999.
20. N. Thomas, T. E. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, ”Tensor field networks: Rotation and translation-equivariant neural networks for 3d point clouds,” arXiv preprint arXiv:1802.08219, 2018.
21. V. G. Satorras, E. Hoogeboom, and M. Welling, ”E (n) equivariant graph neural networks,” in International Conference on Machine Learning, 2021,
22. pp. 9323-9332.
23. K. Sun, B. Xiao, D. Liu, and J. Wang, ”Deep high-resolution representation learning for human pose estimation,” in Proceedings of the IEEE/CVF Con- ference on Computer Vision and Pattern Recognition, 2019, pp. 5693-5703.
24. S. Adami, X. Y. Hu, and N. A. Adams, ”A generalized wall boundary condition for smoothed particle hydrodynamics,” Journal of Computational Physics, vol. 231, no. 21, pp. 7057-7075, 2012.
25. C. R. Qi, H. Su, K. Mo, and L. J. Guibas, ”Pointnet: Deep learning on point sets for 3d classification and segmentation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 652- 660.