DPGNN: Differentiable Physics- and Geometry-Assisted Neural Network in Clifford Algebra for 2D Flow Estimation

A. Pepe[a], **M. Montanari**[a], **Z. Liu**[a]

[a]Intelligent Cloud Technologies Lab
Huawei Munich Research Center
Riesstraße 25, 0992 München

Abstract

We propose DPGNN, a differentiable physics- and geometry-aided pipeline for the solution of 2D partial differential equations describing the turbulent flow of a fluid around obstacles. We build upon the work of [1], that proposes a hybrid framework through a differentiable solver-in-the-loop strategy called differentiable physics-assisted neural network (DPNN). While the DPNN outperforms other supervised learning approaches by keeping the network solutions sound from a physical perspective, it lacks explicit geometrical information about the problem. We include such geometrical information by working in two-dimensional (2D) Geometric Algebra, $G(2,0,0)$, replacing the ResNet of the DPNN with a Clifford ResNet and by recasting the inputs in multivector form, with scalar part encoding information about the obstacles and vector part encoding information about the fluid velocity. DPGNN, which modifies DPNN minimally, achieves lower minima during training, it shows comparable performances to its geometry agnostic counterpart at a fraction of the number of its trainable parameters and it is more robust when estimating the fluid flow for large Δt.

1 Introduction

Computational Fluid Dynamics (CFD) is a branch of engineering focusing on the numerical simulation of phenomena such as fluid flow and heat transfer. Recently, there has been a growing interest in combining CFD with machine learning (ML) techniques to enhance simulation capabilities [6]. ML architectures can help accelerate CFD simulations drastically by reducing the computational resources, they can adapt to changes in the geometry of the problem or in the boundary conditions of the fluid flow.

ML approaches, however, are only as good as the quality of the datasets they are trained on, and results stemming from such approaches might not just be approximate, but also unphysical, as ML architectures do not solve fluid dynamics equations explicitly, but rather learn from patterns in the data. For this reason, several physics-aided pipelines have been proposed, in order to keep prediction via ML grounded in physical principles [12, 4, 11, 5]. An example of this is given in [1], in which a differentiable CFD solver is included in the training loop to keep the networks prediction sound from a physical perspective as much as possible.

We believe that information about the geometry of the problem to be solved, however, is just as crucial as the physical laws governing it. We embed such geometrical information through Clifford Algebra, which has been gaining increasing recognition in the design of geometry-aware pipelines in several fields, including physics [2, 8], computer vision [9, 10] and computational biology [7].

2 Problem Formulation

The 2D Navier-Stokes partial differential equations (PDEs) describe the flow of an incompressible fluid. They read as follows:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

in which $\mathbf{u} = (u, v)$ is the velocity vector, p is pressure, ρ is density, and ν is the kinematic viscosity. Eq. 1 represents the momentum equation, expressing the conservation of momentum in the (x, y) directions, accounting for advection, pressure, and viscosity, while Eq. 2 represents the continuity equation.
We look at the fluid flow past multiple obstacles in a rectangular domain at moderate Reynolds number (i.e. laminar flow) to estimate the 2D velocity field profile and wakes formations. We follow the same approach of [1] and mantain the dataset identical, with the same obstacle configuration design.
We recast the problem into a multivector-to-multivector regression problem. The input of our DPGNN pipeline takes as input a multivector $\mathbf{x}$ in 2D Geeometric Algebra $G(2, 0, 0)$ with one scalar component and two vector components along e_1, e_2 of the form

$$\mathbf{x} = m_f \cdot 1 + ue_1 + ve_2, \tag{3}$$

as opposed to DPNN that takes three real-valued channels (m_f, u, v). In Eq. 3, u and v are the components of $\mathbf{u}$ along the e_1 and e_2 direction, respectively, and m_f is a binary mask indicating the shape and the position of the obstacles.
The problem is formulated as a supervised learning problem with rollout $m = 4$ time instants, with time step $\Delta t = 0.1s$ and loss measured between ground truth and predicted multivectors $\mathbf{x}, \hat{\mathbf{x}}$ as

$$\mathcal{L} = \frac{1}{m}\sum_{i=0}^{m} \|\langle \mathbf{x} \rangle_1^i - \langle \hat{\mathbf{x}} \rangle_1^i\|_2 = \frac{1}{m}\sum_{i=0}^{m} \|\mathbf{u}^i - \hat{\mathbf{u}}^i\|_2, \tag{4}$$

in which $\langle \mathbf{x} \rangle_1^i = \mathbf{u}^i$ indicates the velocity field at time i.
‘

3 DPGNN

The overal DPGNN pipeline is shown in Fig. 1. The reference solver, FoamExtends, provides the 2D vector field of the fluid velocity. Each fluid velocity snapshot, or *frame*, is downsampled by a factor of 3 for agile processing and fed into the base solver, PhiFlow [3], to provide the next frame after $\Delta t = 0.1s$. No learning is involved in this step. We

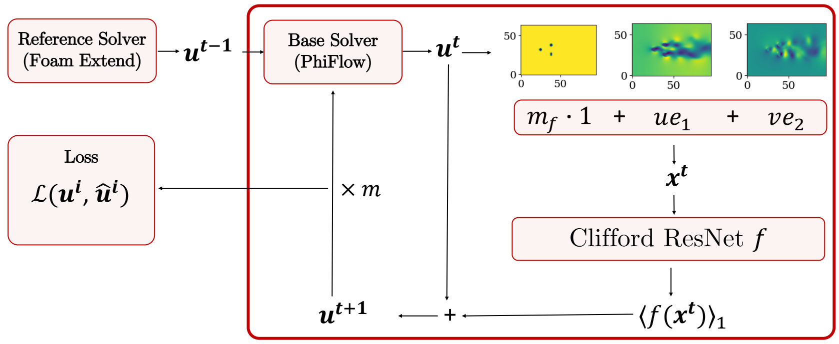


Figure 1: The DPGNN pipeline. We borrow from the DPNN the solver-in-the-loop strategy through PhiFlow. We add the geometric information by embedding data in multivector form and employing an architecture in Clifford Algebra.

recast the output of the base solver (i.e. the velocity field at the next time instant) plus the obstacle mask in multivector form according to Eq. 3. A ResNet architecture working in Clifford Algebra f, modelled after [2] is trained to give an estimate of the velocity field at the next time step according to:

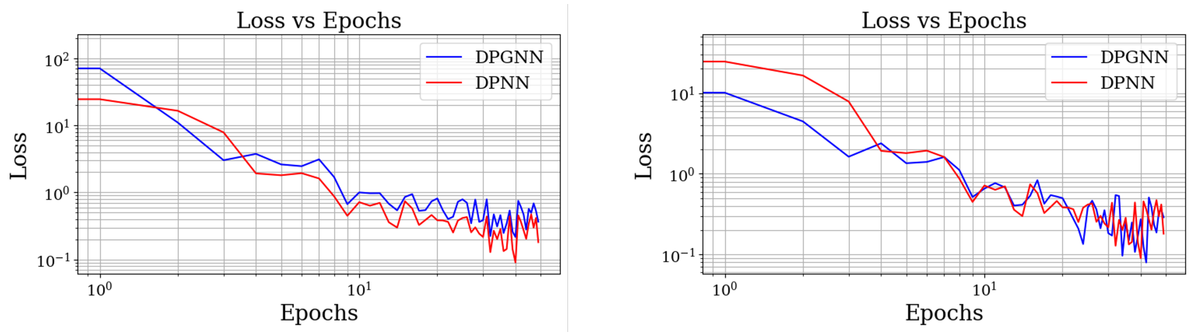
$$\mathbf{u}^{i+1} = f(\mathbf{u}^i) + \mathbf{u}^i = \Delta\mathbf{u}^i + \mathbf{u}^i, \tag{5}$$

meaning that the Clifford ResNet predicts the incremental difference between two consecutive time steps. The vector field $\mathbf{u}^{i+1}$ is fed again into the base solver. This loop is repeated for $m = 4$ rollout steps, with loss function evaluated for each frame.
The ResNet architecture is composed by 20 blocks of 2D convolutional layers paired with ReLU activation function and a skip connection every two blocks, for a total of 600,000 trainable parameters. Weights and biases are full multivectors in $G(2, 0, 0)$, inputs are multivectors with only scalar and vector component and outputs are multivectors only with vector component.
We train the network for 50 epochs. Data are divided into 50 batches and each batch goes through 96 iterations for a total of three nested loops. We employed the Adam optimizer and constant learning rate of 10^{-4}. The training set includes 3000 frames obtained through computational fluid dynamics simulations via FoamExtend run for 300s. The first 1500 frames are discarded to ensure steady state flow. The test set includes 100 frames. The obstacles are arranged in a total of 100 different configurations.

4 Results

We report results obtained with our physics- *and* geometry-aided pipeline trained for 50 epochs and compare them with those presented in [1]. In Fig. 2 we report the training loss for the DPNN and the DPGNN for two different number of convolutional channels. Note how the training profile of the DPGNN at 16 channels is almost overlapping to the one of the DPNN with 520,000 trainable parameters. This shows that the geometric embedding of data allows to reach similar performances of real-valued networks with roughly 25% fewer trainable parameters.
Similarly, when the number of parameters is comparable, with 20 channels for the DPGNN and 32 channels for the DPNN pipelines respectively, the training curve starts

at almost half a decade lower loss value for our DPGNN compared to the geometry-agnostic approach, and it reaches lower minima. Note that the number of channels of our DPGNN has been picked so to yield the smallest difference in number of parameters with respect to the original DPNN.



(a) DPGNN with 16 channels, 387714 parameters.

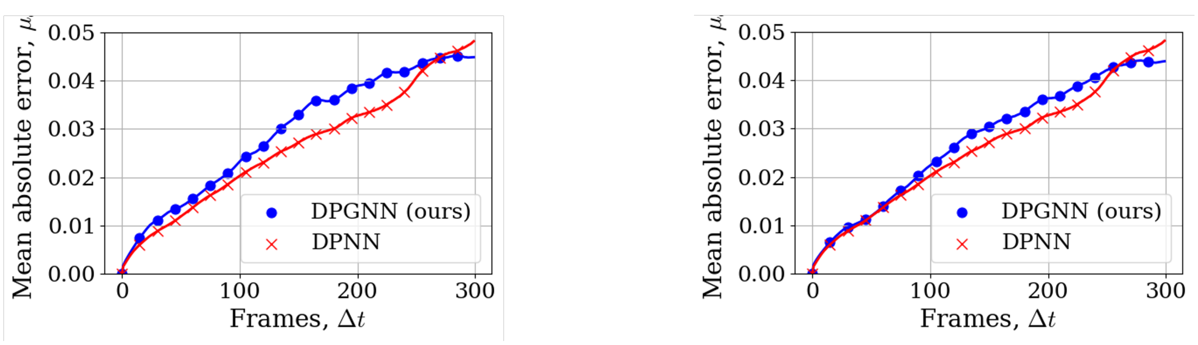
(b) DPGNN with 20 channels, 604642 parameters.

Figure 2: Training losses for DPNN and DPGNN.

In Fig. 3 we report the mean absolute error (MAE) between ground truth (GT) and estimated velocity field evaluated at different time instants Δt for a sample test case. The MAE is defined as

$$\mu = \frac{1}{MN}\sum_{m}^{M}\sum_{n}^{N}(|u_{mn} - \hat{u}_{mn}|) + (|v_{mn} - \hat{v}_{mn}|), \tag{6}$$

in which M, N are width and length of the 2D grid, respectively and x_{mn} is the field $x = u, v$ evaluated at grid indices (m, n).
Our pipeline is comparable to DPNN for the first samples of the velocity fields and tends to plateau for large values of Δt, meaning it is more effective at capturing the long-term evolution of the fluid flow profile.



(a) DPGNN with 16 channels, 387714 parameters.

(b) DPGNN with 20 channels, 604642 parameters.

Figure 3: Mean absolute error losses for test case 39 with DPNN and DPGNN.

In Fig. 4 we compare the GT vorticity flowfield at 5 different time instants $t = \{60\Delta t, 100\Delta t, 180\Delta t, 240\Delta t, 300\Delta t\}$ with respect to GT obtained with the DPNN and

DPGNN approaches for a sample test case. Note how the wakes formation is estimated correctly for both approaches with minimally noticeable differences, despite the DPGNN having a smaller number of convolutional channels. For large Δt, on the other hand, a more significant deviation with respect to GT can be noticed in both pipelines.

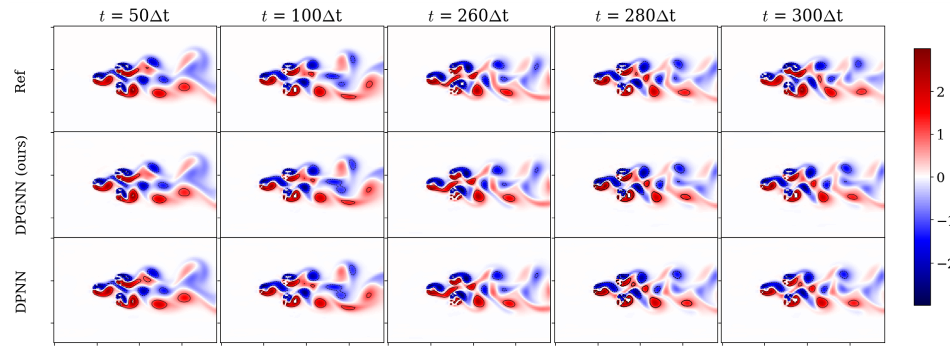


Figure 4: Vorticity flowfield for 5 different time instants.

In Fig. 5 we show the vector components u, v of $\mathbf{u}$ as a function of Δt in three different locations on the 2D grid, in correspondence of the the wakes of the downstream cylinders along the centerline axis. In agreement with Fig. 3, our DPGNN does a better job at estimating the velocity vector field for high Δt. When $\Delta t \geq 250$, in fact, the DPGNN closely follows the GT pattern, especially when $x \leq 64$.

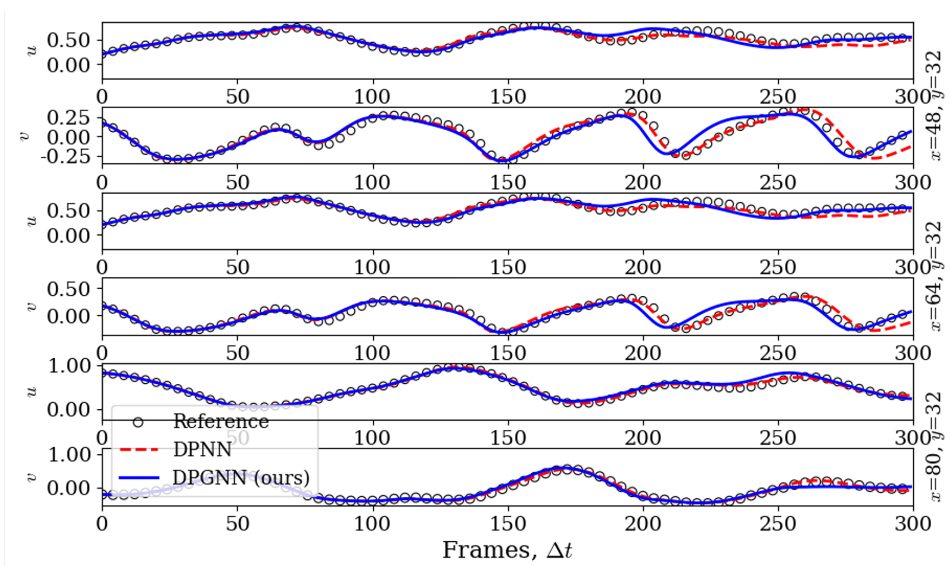
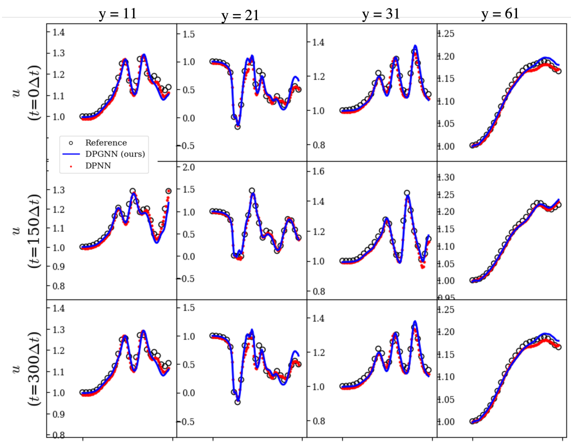
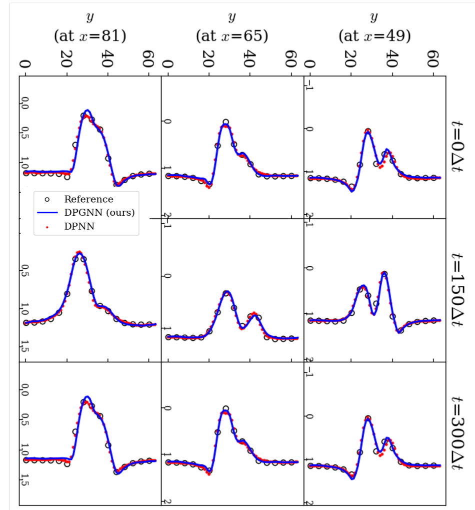


Figure 5: u, v components of the velocity field as a function of time in 3 probes locations (x_n, y_n).

In Fig. 6, cross sections of the u, v components of the velocity field $\mathbf{u}$ are shown for a selected test case. Both DPNN and DPGNN yields satisfactory estimation of the flow profiles, but our DPGNN slightly outperforms the geometry-agnostic approach in certain locations and at $\Delta t = 300$ for a smaller number of channels.



(a) Cross-section of the u component.



(b) Cross-section of the v component.

Figure 6: Sections of the velocity fields in different position for different Δt

5 Conclusions & Limitations

We have introduced DPGNN, a pipeline that combines a differentiable physics-aided approach through a solver-in-the-loop strategy (introduced with DPNN) with a ResNet in Geometric Algebra for the solution of 2D Navier-Stokes PDEs in laminar flow condition. To the best of our knowledge, it represents the first example of a physics-aided GA network.

DPGNN modifies DPNN minimally. The two main differences are: (1) input data are recast in $G(2,0,0)$ multivector form, with the obstacles' mask as scalar component and u, v as vector components; (ii) the ResNet architecture, the best performing within the DPNN pipeline, is replaced by a Clifford ResNet as described in [2] also in $G(2,0,0)$.

The DPGNN allows to achieve two things: (i) similar performances to DPNN at a fraction of trainable parameters and convolutional channels; (2) better resilience as time increases and the flow estimation becomes more challenging.

The DPGNN, on the other hand, also presents some limitations: firstly, as it minimally changes DPNN, offers a smaller improvement compared to the one offered by NNs over CFD methods, as shown in [1]. Second, the GA embedding increases the tensors' dimensionality, meaning that a GA approach might not scale well, especially if data require an embedding in 3D space rather than 2D.

Future work might include the comparison of other architectures in GA, a study on the effects of the weights and biases embedding (whether multivector, vector or scalar only) and alternative input data embeddings.

References

[1] Shuvayan Brahmachary and Nils Thuerey. “Unsteady Cylinder Wakes from Arbitrary Bodies with Differentiable Physics-Assisted Neural Network”. In: (2023). arXiv preprint arXiv:arXivID. DOI: 10.48550/arXiv.2308.04296. URL: https://arxiv.org/abs/2308.04296.

[2] Johannes Brandstetter et al. "Clifford neural layers for pde modeling". In: *arXiv preprint arXiv:2209.04934* (2022).

[3] Philipp Holl et al. "phiflow: A differentiable pde solving framework for deep learning via physical simulations". In: *NeurIPS workshop*. Vol. 2. 2020.

[4] George Em Karniadakis et al. "Physics-informed machine learning". In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440.

[5] Temoor Muther et al. "Physical laws meet machine intelligence: current developments and future directions". In: *Artificial Intelligence Review* 56.7 (2023), pp. 6947–7013.

[6] Dhruvil Panchigar et al. "Machine learning-based CFD simulations: a review, models, open threats, and future tactics". In: *Neural Computing and Applications* 34.24 (2022), pp. 21677–21700.

[7] Alberto Pepe, Sven Buchholz, and Joan Lasenby. "Clifford Group Equivariant Neural Network Layers for Protein Structure Prediction". In: *Northern Lights Deep Learning Conference*. PMLR. 2024, pp. 205–211.

[8] Alberto Pepe, Sven Buchholz, and Joan Lasenby. "GA-ReLU: an activation function for Geometric Algebra Networks applied to 2D Navier-Stokes PDEs". In: *ICLR 2024 Workshop on AI4DifferentialEquations In Science*. 2024. URL: `https://openreview.net/forum?id=T54Hn1e98P`.

[9] Alberto Pepe, Joan Lasenby, and Sven Buchholz. "CGAPoseNet+ GCAN: A Geometric Clifford Algebra Network for Geometry-aware Camera Pose Regression". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 6593–6603.

[10] David Ruhe et al. "Geometric clifford algebra networks". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 29306–29337.

[11] Yubiao Sun, Ushnish Sengupta, and Matthew Juniper. "Physics-informed deep learning for simultaneous surrogate modeling and PDE-constrained optimization of an airfoil geometry". In: *Computer Methods in Applied Mechanics and Engineering* 411 (2023), p. 116042.

[12] Xingang Zhao et al. "On the prediction of critical heat flux using a physics-informed machine learning-aided framework". In: *Applied Thermal Engineering* 164 (2020), p. 114540.