# Session 4

## Physics-informed Models

| | |
|---|---|
| **S4-A3** | Christian Bauer — Robust estimation of pressure and velocity gradients from one time step in turbulent pipe flows via two-stage physics-informed neural networks |
| **S4-A5** | Jincheng Zhang — Physics-informed neural networks for RANS simulations with data-driven turbulence modelling |
| **S4-A6** | Wanshu Li — PGP-GNN: A Physics-Guided Prior Graph Neural Network for Efficient Prediction of Hypersonic Chemical Non-Equilibrium Aerodynamic Heating |
| **S4-A7** | Simon Wassing — CFD-informed Neural Networks (CINNs) – Integrating an industrial-grade CFD solver into Physics-Informed Neural Networks |
| **S4-A9** | Gulliver Van Essche — Physics-informed learning of unsteady fluid systems via dynamical invariants |
| **S4-A11** | Matthias Möller — IGANets: Blending isogeometric analysis with operator learning |
| **S4-A14** | Qiang Liu (Giacomo Baldan) — Pareto-Optimal Generative Modeling with Physical Constraints via Physics-Based Flow Matching |
| **S4-A15** | Tamara Gammaidoni — Physics-Informed ML for CFD Applications: Forward Modeling of Microchannel Heat Transfer and Inverse Identification in Hydrogen Injection |
| **S4-A18** | Hyungmin Shin — A Two-Stage Physics-Informed Neural Network for Temperature Reconstruction from |
| **S4-A20** | Yunde Su — SIM-PINN: Segregated, Iterative, and Multi-Stage Physics-Informed Neural Network for Incompressible Flow Simulations |
| **S4-P1** | Jiahao Song — FENN: Feature-enhanced neural network for solving partial differential equations involving fluid mechanics |
| **S4-P2** | Michael Mommert — Benchmarking PINN architectures for temperature-reconstruction by determining viscosity deviations |
| **S4-P4** | Zishan M. Mahmood — Data-Driven and Physics-Informed Graph Neural Networks for Flow Field Prediction in Two-Dimensional Channel Flows Featuring Bluff Body Obstacles |
| **S4-P8** | Ian Hubbard — Fidelity-Scaling via Physics-Informed Graph Learning on Marine Propeller Blades |
| **S4-P10** | Amelie Lam — Solver-In-The-Loop Training with Multistep Inputs for Capturing Fine-Scale Turbulence |
| **S4-P12** | Roberto Dal Monte — Probabilistic Near-Wall Turbulence Modeling via Physics-Guided Diffusion Model |
| **S4-P13** | Yuling Han — Physics-informed latent neural operator for three-dimensional compressor cascade flow prediction |
| **S4-P17** | Paolo Guida — Real-time control of multiphase processes with learned operators |
| **S4-P19** | Faras Brumand-Poor — Physics-Informed Neural Networks for Thermohydrodynamic Modeling of Fuel Injection Pumps with Fuel-Dependent Properties |
| **S4-P21** | Stefan Posch — Physics-informed Neural Networks and Differentiable Chemistry for Solving the Unsteady Flamelet Equation |

# Robust estimation of pressure and velocity gradients from one time step in turbulent pipe flows via two-stage physics-informed neural networks

[1]Robin Barta; [2]Christian Bauer*; [1]Gholamhossein Bagheri

[1]*Laboratory for Fluid Physics Pattern Formation and Biocomplexity, Max Planck Institute for Dynamics and Self-Organization 37077 Göttingen, Germany*
[2]*Institute of Aerodynamics and Flow Technology, German Aerospace Center (DLR), 37073 Göttingen, Germany*

## Abstract

Physics-informed neural networks (PINNs) have emerged as a promising approach in fluid mechanics for reconstructing unknown flow quantities—such as pressure and temperature—from sparse velocity measurements by embedding the governing equations of fluid flow into the training process [1,2]. Although PINNs are well suited to handling spatially sparse data [3], they are typically trained using large datasets across multiple time steps, which makes them computationally expensive. Here, we propose a novel PINN framework that reconstructs pressure fields and energy dissipation rates in non-thermal turbulent flows using only a single time step of sparsely sampled three-dimensional velocity data involving training with the pressure Poisson equation,

As input, we employ synthetic particle data generated by a direct numerical simulation (DNS) of turbulent pipe flow at a friction Reynolds number  with varying inter-particle spacings. While DNS data are used at this stage, the ultimate objective is to apply the framework to experimental particle data obtained from fast two-pulse measurement systems [4]. To evaluate the method under realistic conditions, we added simulated measurement noise to the data. The proposed PINN effectively corrects the velocity field, yielding over 90% accuracy in both pressure reconstruction and energy dissipation estimation. These results highlight the framework's strong potential for processing realistic, noisy experimental data.

[1] Mommert M, Barta R, Bauer C, Volk MC, Wagner C (2024). Comput. Fluids, 283, 106419
[2] Barta R, Volk MC, Bauer C, Wagner C, Mommert M (2025). Meas. Sci. Technol., 36(7), 075304
[3] Di Leoni PC, Agarwal K, Zaki TA, Meneveau, C, Katz J (2023). Exp. Fluids, 64(5):95
[4] Thiede B, Nordsiek F, Kim Y, Bodenschatz E, Bagheri G (2025). Atmos. Meas. Tech., 18(21):5999–6019

**Keywords: Physics-informed neural networks (PINN), pressure assimilation, dissipation rates**

# Physics-informed neural networks for RANS simulations with data-driven turbulence modelling

[1]Jincheng Zhang

[1]*School of Engineering, University of Warwick, Coventry CV4 7AL, UK*

## Abstract

Turbulent flows are of great interest for many engineering applications. With the fast development of scientific machine learning, a lot of works are emerging on data-driven turbulence modelling and machine-learning-augmented CFD simulations, aiming at better performance over the traditional turbulence models. On the other hand, the recent development of physics-informed neural networks (PINNs) offers an alternative framework to the traditional numerical approaches for flow simulations, particularly for parametric flow simulations. To push the research boundary, this work proposes an integrated data-driven turbulence modelling and simulation framework based on physics-informed neural networks. From the perspective of data-driven turbulence modelling, most works in the current literature rely on a baseline RANS model within their simulation framework. In this work, the data-driven modelling of the Reynolds stress transport equations is investigated without relying empirical models, thus avoiding limitations imposed by the baseline models. From the numerical simulation perspective, there are several attempts in the current literature on using PINNs for RANS simulations. Their performances are encouraging but are still facing great challenges particularly on the training convergence, considering that they all employed traditional turbulence models which were developed within the numerical paradigm but not tailored for the PINN framework. In this work, by developing a data-driven turbulence model and subsequently employing it within the same PINN framework, the consistency between the inverse problem (i.e. model discovery) and the forward problem (i.e. solution inference) is enabled. The proposed framework is applied to a set of flows over periodic hills of parameterized geometries. The preliminary results show that the mean flow quantities and the turbulence quantities are both obtained with good accuracy, and this is true for both the training case and the test cases.

**Keywords: Physics-informed neural networks; RANS simulations; Data-driven turbulence model**

# PGP-GNN: A Physics-Guided Prior Graph Neural Network for Efficient Prediction of Hypersonic Chemical Non-Equilibrium Aerodynamic Heating

*Wanshu Li [1], Zhiyu Duan[1], Wenwen Zhao[1], Weifang Chen [1]*

[1] *School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China*

**Abstract:** Accurate and efficient prediction of the chemical non-equilibrium aerodynamic heating remains a fundamental challenge in hypersonic vehicle design. High-fidelity Computational Fluid Dynamics (CFD) simulations are computationally prohibitive due to stiff chemical source terms, whereas purely data-driven approaches often suffer from limited physical interpretability and poor generalization across varying catalytic wall conditions. To address these limitations, a Physics-Guided Prior Graph Neural Network (PGP-GNN) is proposed. The central idea of the proposed framework is a physics-based multi-fidelity fusion strategy, in which a computationally inexpensive perfect-gas solution is employed as a low-fidelity physical model before capturing the topological structure of the flow field. A dual-branch graph-based architecture is constructed to explicitly decouple the total wall heat flux into a conductive component, governed primarily by macroscopic thermal gradients, and a diffusive component driven by surface catalytic reactions. Numerical experiments conducted on double-ellipsoid and HTV-2 configurations indicate that the proposed framework provides a sevenfold speedup relative to real-gas CFD simulations, while maintaining high predictive fidelity. The model exhibits robust performance even under complex regimes involving Mach number and wall temperature extrapolation alongside catalytic efficiency interpolation. This work effectively bridges the gap between physical mechanism decoupling and data-driven efficiency, offering a robust tool for rapid aerothermal design.

# CFD-informed Neural Networks (CINNs) –

# Integrating an industrial-grade CFD solver into Physics-Informed Neural Networks

[1]Simon Wassing*; [1]Tobias Leicht; [1]Philipp Bekemeyer;

[1]*German Aerospace Center – DLR, Institute of Aerodynamics and Flow Technology, Department $C^2A^2S^2E$*

**Abstract**

Physics-informed neural networks (PINNs) have emerged as a promising approach for solving partial differential equations in aerodynamics, leveraging neural networks as global ansatz functions trained via residual minimization and automatic differentiation. PINNs have demonstrated success in simulating subsonic and supersonic inviscid as well as laminar and incompressible flows. They show particular promise in inverse problems and parametric studies where traditional solvers would require repeated, computationally expensive simulations. Despite these advantages, the application of PINNs remains largely restricted to simple, two-dimensional geometries. A major challenge arises in complex three-dimensional configurations, where the neural network must approximate solutions across domains with highly non-uniform frequencies. Sharp gradients (e.g. in boundary layers and shocks) lead to poorly conditioned optimization problems and thus slow convergence and inaccurate predictions. To overcome these limitations, we introduce a CFD-informed neural network (CINN), which integrates the industrial-grade computational fluid dynamics (CFD) solver CODA for the residual evaluation in the loss function that is used to train the neural network. CODA's differentiability can be exploited to train a neural network using automatic differentiation. By combining the flexibility of a neural network ansatz function for inverse and parametric problems with the robustness and versatility of classical CFD solvers, CINNs aim to enhance accuracy and convergence for complex aerodynamic flows. CODA is the CFD software being developed as part of a collaboration between the French Aerospace Lab ONERA, the German Aerospace Center (DLR), Airbus, and their European research partners. CODA is jointly owned by ONERA, DLR and Airbus.

**Keywords: aerodynamics, physics-informed neural networks, computational fluid dynamics**

# Physics-informed learning of unsteady fluid systems via dynamical invariants

[1,2]Gulliver Van Essche*; [1,2]Gonçalo Granjal Cruz; [1,2]Tim De Troyer; [1,2]Mark Runacres; [1,2]Jan Decuyper;

[1]*Thermo and Fluid Dynamics (FLOW), Faculty of Engineering, Vrije Universiteit Brussel (VUB), Brussels, 1050, Belgium.*

[2]*Brussels Institute for Thermal-Fluid Systems and Clean Energy (BRITE), Vrije Universiteit Brussel (VUB) and Université libre de Bruxelles (ULB), Brussels, 1050, Belgium.*

## Abstract

Data-driven methods successfully provide fast surrogate models of fluid dynamical systems. Such models are trained solely on observational data. However, available data are often limited or constrained. As a result, data-driven methods frequently yield poor extrapolation properties [1] and error accumulation over time [2]. Hybrid data- and physics-driven approaches provide broader support and improved generalisability, while also enabling a reduction of the training data and its associated costs [3].

A common approach is to augment data-fitting objectives with additional loss terms that penalise deviations from assumed governing equations, or enforce them explicitly through the model structure, with physics-informed neural networks (PINNs) [3] as the archetypal example. However, in practice, implementing a loss term to account for deviations from the governing equations is not straightforward. Through relatively simple experiments, dynamical invariants of certain systems can be identified and subsequently incorporated more directly into the objective function.

We demonstrate this approach when identifying a dynamical model of the wake behind a cylinder undergoing imposed motion. At Re = 100, a stationary cylinder exhibits stable vortex shedding [4]. This behaviour is typically described by a limit cycle of the system, materialising in the neighbourhood of an unstable fixed point [5]. When the cylinder is forced to oscillate transversally to the incoming flow, other nonlinear effects (e.g. lock-in) appear. Models trained solely on such imposed-motion data, i.e. on observations of the system far away from the fixed point, often fail to capture the limit-cycle response. By adding physics-driven regularisation terms that promote the stability properties of fixed points, we constructed a model that correctly reproduces the stable limit cycle without having observed this regime during experiments.

**Keywords:** unsteady fluid dynamics; nonlinear system identification; physics-informed learning; dynamical regularisation.

[1] Brunton, S. L., Noack, B. R., and Koumoutsakos, P. (2020). "Machine learning for fluid mechanics." Annual Review of Fluid Mechanics, 52, 477–508.

[2] Vlachas, P. R., Pathak, J., Hunt, B., Sapsis, T. P., Girvan, M., Ott, E., and Koumoutsakos, P. (2018). "Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics." Neural Networks, 126, 191–217.

[3] Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." Journal of Computational Physics, 378, 686–707.

[4] Williamson, C. H. K. (1996). "Vortex dynamics in the cylinder wake." Annual Review of Fluid Mechanics, 28, 477–539.

[5] Noack, B. R., Afanasiev, K., Morzyński, M., Tadmor, G., and Thiele, F. (2003). "A hierarchy of low-dimensional models for the transient and post-transient cylinder wake." Journal of Fluid Mechanics, 497, 335–363.

# IGANets: Blending isogeometric analysis with operator learning

[1]Matthias Möller*

[1]*Department of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands*

## Abstract

Physics-informed machine learning has attracted growing interest as a potential alternative to classical numerical methods for solving initial and initial-boundary value problems. However, many neural network-based approaches, such as Physics-Informed Neural Networks (PINNs) and Deep Operator Networks (DeepONets), rely on formulations that are not naturally compatible with established numerical analysis workflows, including finite element and isogeometric analysis.

In this work, we introduce IGANets, a novel operator-learning framework that remains fully within the paradigm of classical isogeometric analysis. Rather than approximating the solution field directly, IGANets predict the coefficients of an a priori chosen spline basis. This structure-preserving design enables high-quality, near real-time inference and allows the network output to serve as an efficient initial guess for subsequent offline iterative refinement.

IGANets are trained across a broad range of problem configurations, including geometric and parametric variations, which are represented consistently through their spline coefficients. This makes the approach particularly attractive for design optimization scenarios, where the one-time offline training effort amortizes over many design evaluations.

We present original results for both linear and nonlinear elasticity problems, along with preliminary findings for fluid flow applications. Furthermore, we compare IGANets with alternative physics-informed learning approaches, in particular PINNs and DeepONets, with respect to computational efficiency and generalization performance. Finally, we discuss the integration of IGANets as initial-guess generators within a classical isogeometric analysis framework based on iterative solvers.

**Keywords: Isogeometric analysis, Operator learning, Scientific machine learning**

# Pareto-Optimal Generative Modeling with Physical Constraints via Physics-Based Flow Matching

[1]Giacomo Baldan*; [1]Qiang Liu; [1]Nils Thuerey;

[1]*Technical University of Munich (TUM), Garching, 85748, Germany*

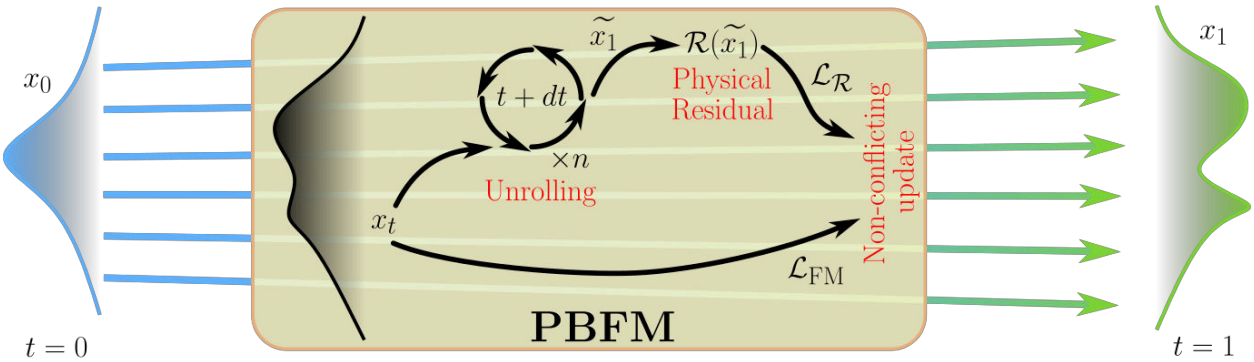
## Abstract

Physics-constrained generative modeling aims to generate high-dimensional samples that are both physically consistent and distributionally accurate. These objectives are often conflicting: enforcing physical constraints can degrade generative fidelity, while optimizing only data likelihood may violate governing equations. Recent advances in flow matching have enabled efficient generative modeling for scientific data, yet incorporating physics typically requires manual loss balancing or expensive inference-time corrections.

We address this challenge by explicitly formulating physics-constrained generative modeling as a Pareto optimization problem between physical accuracy and distributional fidelity. We introduce Physics-Based Flow Matching (PBFM), a training framework that integrates physical constraints directly into flow matching using conflict-free gradient updates. By aligning gradients of the generative and physics losses, PBFM eliminates the need for manual weighting and enables simultaneous optimization of both objectives. To further improve physical consistency, we mitigate Jensen's gap by unrolling the generative trajectory during training, allowing residuals to be evaluated on more accurate predictions of the final clean sample.

We evaluate the method across representative physics-driven learning problems of increasing complexity: Darcy flow, Kolmogorov flow, and unsteady aerodynamics in dynamic stall conditions. These problems span steady, stochastic, and highly nonlinear transient regimes, providing a challenging testbed for physics-constrained generative modeling. Results show that explicitly accounting for the Pareto structure enables models that maintain high sample quality while significantly improving physical consistency. The proposed framework provides a scalable path for generative modeling in scientific and engineering applications where both statistical accuracy and adherence to physical laws are essential.

**Keywords: Flow Matching, Physics, PDE, Diffusion Models**



# Physics-Informed ML for CFD Applications: Forward Modeling of Microchannel Heat Transfer and Inverse Identification in Hydrogen Injection

[1]Tamara Gammaidoni*; [1]Jacopo Zembi; [1]Michele Battistoni;

[1]*Università degli Studi di Perugia, Via G. Duranti n. 93, 06125, Perugia, Italy*

**Abstract**

Physics-Informed Neural Networks (PINNs) have emerged as a promising approach for solving Partial Differential Equations (PDEs) by embedding governing physical laws into the neural network training process. By enforcing the PDE residual within the loss function, PINNs can approximate physically consistent solutions even without large labeled datasets. However, their applicability to industrial thermo-fluid problems remains challenging, particularly for complex geometries, multi-physics coupling, and highly nonlinear flows.

This work investigates the capabilities of PINN-based approaches in two representative thermo-fluid applications relevant to engineering design and modeling: forward simulation of conjugate heat transfer in microchannel heat exchangers and inverse parameter identification in hydrogen injection processes.

In the first case-study, PINNs are used to model laminar flow and heat transfer in parametrized rectangular and wavy microchannel geometries, assessing their capability to predict coupled momentum and heat transport phenomena in varying geometries. Particular attention is devoted to training strategies that significantly influence convergence and accuracy, including collocation point distribution, the balance between physics-based and partially supervised training, and network architectures. Results are also compared with Physics-Informed Fourier Neural Operators (PINO), to evaluate their potential as surrogate models for parametric thermo-fluid simulations and rapid design exploration.

The second case-study focuses on an inverse problem related to hydrogen jets. PINNs are employed to infer unknown physical parameters directly from observed flow fields, enabling the reconstruction of quantities of interest for CFD simulations such as mass diffusivity. This problem introduces additional challenges due to the presence of transient and turbulent flow dynamics, which significantly increase the complexity of the governing equations and the associated training process.

The combined analysis of forward and inverse problems reveals the strengths and limitations of PINNs in complex thermo-fluid scenarios and their potential to complement traditional CFD workflows for parametric studies, reduced-order modeling, and parameter estimation.

**Keywords:**

Physics-Informed Neural Networks (PINNs); Computational Fluid Dynamics (CFD), Conjugate Heat Transfer (CHT), Inverse Problems, Hydrogen Injection

# A Two-Stage Physics-Informed Neural Network for Temperature Reconstruction from Lagrangian Particle Tracking Data

[1]H. Shin*; [1,2]A. Schröder

[1]*Inst. of Aerodynamics and Flow Technology, German Aerospace Center (DLR), Göttingen, Germany*

[2]*Inst. of Traffic Engineering, Dep. of Image Based Measurement Techniques, Brandenburgische Technische Universität, Cottbus, Germany*

## Abstract

Physics-Informed Neural Networks (PINNs) have emerged as a powerful framework for reconstructing flow fields from sparse experimental measurements by embedding governing physical equations into neural network training. Previous studies have demonstrated that PINNs can accurately reconstruct velocity and acceleration fields from Lagrangian Particle Tracking (LPT) data and can further infer pressure fields through the Navier–Stokes equations. However, reconstructing temperature fields from pure LPT data remains challenging in thermally driven flows, where temperature actively influences the flow dynamics through buoyancy coupling.

In this work, we investigate the feasibility of reconstructing temperature fields from LPT data using PINNs in a three-dimensional thermal convection system. Direct Numerical Simulation (DNS) data of Rayleigh–Bénard convection (Pr = 7, Ra = $10^6$, Γ=8) are used as the reference dataset. To emulate realistic LPT measurements, approximately 35,000 particles are advected in the convection cell, corresponding to a mean inter-particle spacing of approximately three times the local Kolmogorov length scale.

A two-stage PINN framework is proposed to improve the reconstruction accuracy of the temperature field from particle trajectory data. The results demonstrate that, when appropriate boundary conditions are imposed at the upper and lower plates, the proposed PINN framework can successfully reconstruct the temperature field close to upper and lower plates without requiring prior knowledge of the flow structure, such as the mean temperature profile.

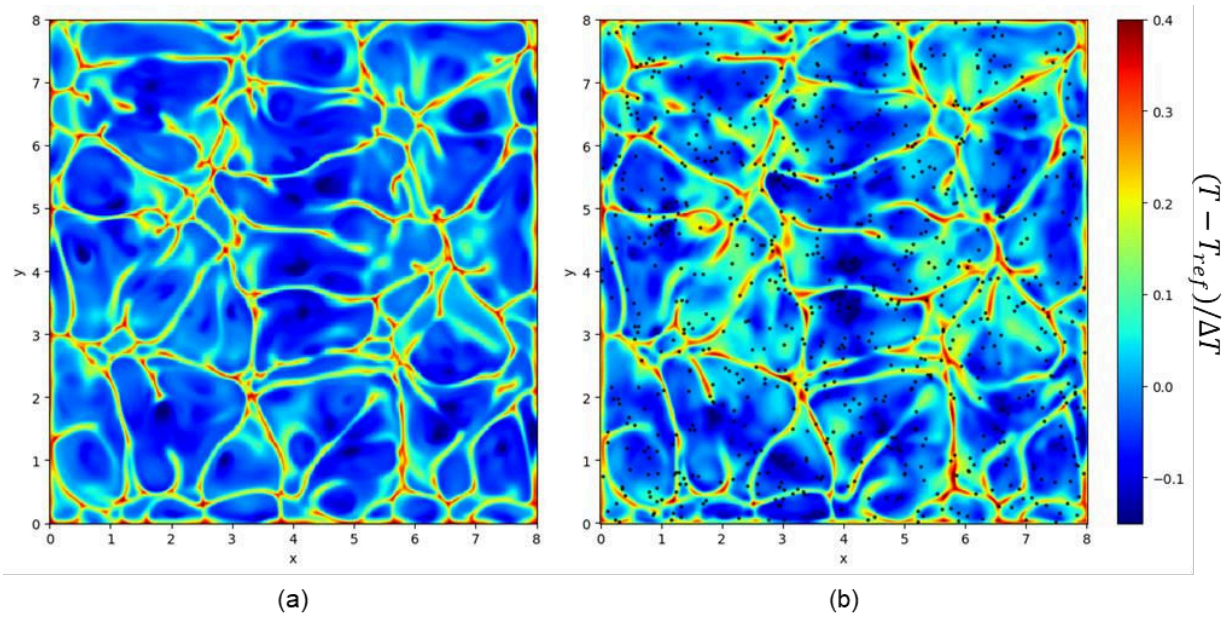



***Figure 1.*** *Comparison of temperature fields near the lower plate (z/H = 0.05): (a) DNS reference and (b) PINN reconstruction from synthetic LPT data. Black dots on the right-hand side figure indicate the locations of synthetic particles within a slab of thickness 0.01H centered at the plotted plane.*

**Keywords: physics informed neural network, data assimilation, particle tracking velocimetry, Rayleigh-Bénard convection**

# SIM-PINN: Segregated, Iterative, and Multi-Stage Physics-Informed Neural Network for Incompressible Flow Simulations

[1]Yunde Su*; [1,2]Zhenyu Zhang;

[1]*National Key Laboratory of Fundamental Algorithms and Models for Engineering Simulation, Sichuan University, Chengdu 610207, China*

[2]*School of Computer Science, Sichuan University, Chengdu 610207, China*

## Abstract

Physics-informed neural networks (PINNs) offer a mesh-free, continuously differentiable approach to fluid simulation by embedding governing equations into the training objective. However, for incompressible Navier–Stokes equations, standard monolithic PINNs often suffer from severe constraint coupling and spectral biases. This leads to difficulties in achieving deep convergence and high accuracy. To address these challenges, we propose a Segregated, Iterative, and Multi-stage PINN (SIM-PINN) inspired by classical projection methods in computational fluid dynamics. Instead of simultaneous optimization, SIM-PINN employs projection-style inner iterations in which the momentum and continuity equations are optimized in a segregated manner using different neural networks. In each iteration, the velocity field is firstly predicted from the momentum equations and then updated using the pressure-correction Poisson equation. Both the momentum and continuity equations are satisfied when interaction converges. To enhance convergence and solution accuracy, a multi-stage refinement network architecture is used for each equation. Each stage corresponds to an iteration step. A new network is introduced at each stage to learn the residual obtained from the previous stage and correct the solution variables within a targeted range of frequency. Numerical experiments on the lid-driven cavity and unsteady Kolmogorov flow simulations demonstrate that, compared with standard PINN, SIM-PINN consistently yields lower and more uniformly distributed PDE residuals and delivers improved accuracy and stability.

**Keywords:** Physics-Informed Neural Networks (PINNs); Incompressible Navier-Stokes Equations; Projection Method; Segregated iteration; Multi-stage Architecture; Frequency-aware Refinement

Proceedings of the 2nd international Symposium on AI and Fluid Mechanics
Paper No [leave blank]

# Solving Navier–Stokes equations using the feature-enhanced neural network

[1,2,3]Jiahao Song*; [1,2,3]Wenbo Cao; [1,2,3]Weiwei Zhang;

[1]*School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China;*

[2]*International Joint Institute of Artificial Intelligence on Fluid Mechanics, Northwestern Polytechnical University, Xi'an, 710072, China;*

[3]*National Key Laboratory of Aircraft Configuration Design, Xi'an 710072, China*

**Abstract**

Physics-informed neural networks (PINNs) have shown remarkable prospects in solving forward and inverse problems involving partial differential equations (PDEs). The method embeds PDEs into the neural network by calculating the PDE loss at a set of collocation points, providing advantages such as meshfree and convenience in solving inverse and parametric problems. However, PINNs still face challenges in solving strongly nonlinear PDEs involving fluid dynamics, particularly in terms of limited universality and high computational cost. Inspired by input design in surrogate modeling, we propose a feature-enhanced neural network (FENN) in this study. By introducing geometric features and physical features in the inputs of PINNs, FENN can learn the flow more easily, resulting in better performance in terms of both accuracy and efficiency. To avoid invalid PDE residuals caused by neglecting the partial derivatives of the features with respect to space-time coordinates, the feature networks are established offline in advance. Through three numerical experiments involving forward and inverse problems, we first verify that FENN generally reduces the computational cost of PINNs and advanced algorithms by approximately four times and two times, respectively. Furthermore, numerical experiment show that the proposed method can reduce the number of observed data required for the inverse problem. Finally, the capability of FENN to solve the compressible Navier–Stokes equations is validated. To the best of our knowledge, this is the first time that a PINN-like method has successfully solved these equations.

**Keywords**: Feature-enhanced, Physics-informed neural networks, Navier-Stokes equations, Inverse problems

## 1. Introduction

In recent years, physics-informed neural networks (PINNs) [1] have emerged as a promising tool for solving forward and inverse problems involving partial differential equations (PDEs). The central idea of PINNs is integrating the PDE residual into the loss function of the neural networks, ensuring that the networks minimize the residual while approaching the boundary conditions or observed data. Compared with traditional numerical methods, such as the finite volume method [2], finite difference method [3], and finite element method [4], PINNs offer several attractive advantages. First, PINNs employ automatic differentiation [5] to compute partial derivatives, thereby eliminating the need for mesh generation. Second, PINNs can readily integrate observed data, making them particularly suitable for inverse problems, such as reconstructing velocity and pressure fields from concentration measurements [6]. Owing to these advantages, PINNs have achieved remarkable success in computational science and engineering.

Despite these advantages, PINNs still suffer from slow convergence and even fail to obtain correct solutions, particularly in complex fluid mechanics problems. To address this issue, many studies have been devoted to improving the performance of PINNs. Representative approaches include Fourier feature embedding [7], adaptive activation functions [8] and weight normalization [9]. However, they mainly focus on modifications to network architectures, activation functions, or loss formulations, while the inputs of PINNs are generally limited to space-time coordinates. In contrast, researchers in surrogate modeling often design specialized input features to enhance modeling accuracy and generalization [10, 11]. Inspired by these studies, we propose a feature-enhanced neural network (FENN) to accelerate the convergence of PINNs in fluid dynamics by introducing specifically designed geometric or physical features into the inputs of PINNs.

## 2. Methodology

### *2.1 Feature-enhanced neural network*

In surrogate modeling, the choice of network inputs has a significant influence on model performance [10, 11]. Previous studies have shown that carefully designed features, such as the signed distance function and metrics, can

*Corresponding Author, Jiahao Song jiahaosong@mail.nwpu.edu.cn

substantially improve model accuracy and generalization capability. In contrast, the inputs of PINNs are typically limited to space-time coordinates $(\mathbf{x},t)$. Inspired by these studies, we propose a feature-enhanced neural network (FENN) for solving PDEs involving fluid mechanics. Specifically, we combine space-time coordinates with designed features $F(\mathbf{x},t)$ as inputs of the neural network, while other settings are kept the same as in PINNs. By introducing features in the network inputs, FENN can learn the flow more easily, leading to improved accuracy and computational efficiency. Moreover, since only a few additional neurons are introduced in the input layer, the extra computational cost is negligible. A schematic of FENN is shown in Figure 1. The feature design will be presented in detail in Section 2.2.

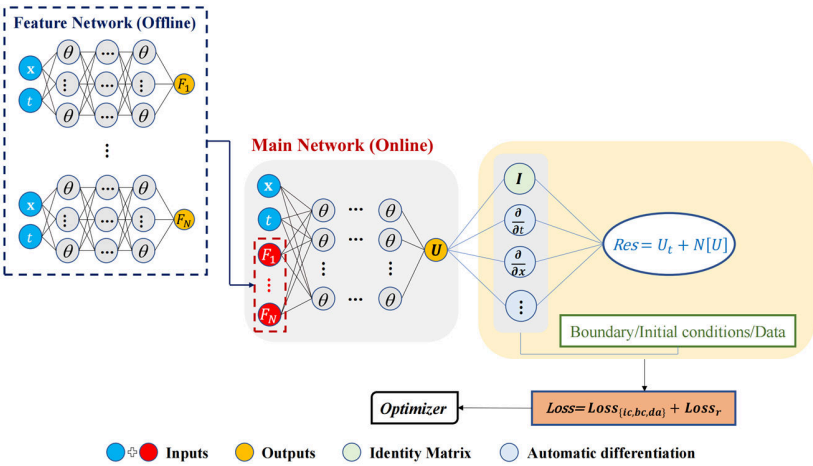



Figure 1 A schematic of FENN for solving partial differential equations.

The feature networks are established offline to address the incorrect computation of partial derivatives caused by directly introducing features into the inputs. We illustrate this issue by computing $\partial \boldsymbol{U}_\theta / \partial x$ ($\boldsymbol{U}_\theta$ represents the output of FENN, and $\theta$ denotes the network parameters). Since $F_i$ is a function of $x$, according to the chain rule:

$$\frac{\partial \boldsymbol{U}_\theta}{\partial x} = \left(\frac{\partial \boldsymbol{U}_\theta}{\partial x}\right)_s + \sum_{i=1}^{N} \frac{\partial \boldsymbol{U}_\theta}{\partial F_i}\frac{\partial F_i}{\partial x} \tag{1}$$

where $(\partial \boldsymbol{U}_\theta / \partial x)_s$ and $\partial \boldsymbol{U}_\theta / \partial F_i$ denote the partial derivatives of the output $\boldsymbol{U}_\theta$ with respect to the inputs $x$ and $F_i$, computed via automatic differentiation. Directly introducing the features into the inputs of PINNs implies that $x$ and $F_i$ are independent, which leads to $\partial F_i / \partial x = 0$. Consequently, an erroneous $\partial \boldsymbol{U}_\theta / \partial x$ is obtained. To resolve this problem, the feature networks are trained offline in a supervised manner to obtain regression models of $F_i$ with respect to $(\mathbf{x},t)$, which are subsequently employed as features in the inputs. During the computation of $\partial \boldsymbol{U}_\theta / \partial x$ in the main network, $\partial F_i / \partial x$ is extracted from the feature networks via automatic differentiation, thereby ensuring the correctness of $\partial \boldsymbol{U}_\theta / \partial x$.

Compared with PINNs, FENN introduces additional computational cost due to the training of the feature networks. However, the feature networks contain significantly fewer parameters than the main network and are easier to train because they are supervised. Therefore, the computational cost of feature networks is much lower than that of the main network. Detailed results from numerical experiments are provided in Section 3.

*2.2 Feature design*

We design both geometric and physical features in this study. First, the geometric features are designed based on [12], including the distance feature $D$ and the angle feature $\Phi$. Specifically, for a point $\mathbf{x}$, $D$ represents the shortest distance from $\mathbf{x}$ to the wall, where the corresponding point $\mathbf{x}_l$ on the wall is defined as the local point. The feature $\Phi$ is defined as the product of the angle $\varphi$ between the tangent direction at $\mathbf{x}_l$ and the wall $0°$ angle of attack, and

*Corresponding Author, Jiahao Song jiahaosong@mail.nwpu.edu.cn

the distance attenuation function $\delta(D) = 1 - D^3/(D^3 + c/10)$ . $c$ represents the characteristic length, such as the diameter of a cylinder or the chord length of an airfoil.

In fluid mechanics, the Navier–Stokes equations degenerate into the Euler equations under inviscid conditions, while the Euler equations further degenerate into the potential flow equation under irrotational conditions. Compared with space-time coordinates, potential flow includes fundamental flow characteristics and exhibits a simpler relationship with inviscid and viscous flows. Meanwhile, the difficulty and computational cost of solving the potential flow equation are significantly lower than those of the Euler and Navier–Stokes equations, and analytical solutions are available for certain problems. Therefore, the solution of the potential flow equation $\nabla^2\phi = 0$ ($\phi$ represents velocity potential) is employed to design the physical features. Specifically, the $x$-component $u^{pf}$ and the $y$-component $v^{pf}$ of the velocity vector, together with the pressure $p^{pf}$, are introduced into the inputs of FENN.

We refer to FENN using geometric features as G-FENN, and FENN using physical features as U-FENN. When both geometric and physical features are used, it is referred to as GU-FENN.

## 3. Results

We validate the effectiveness of FENN by solving two forward problems and one inverse problem. The forward problems include: viscous incompressible flow around the NACA0012 airfoil and viscous compressible flow around the NACA0012 airfoil. The inverse problem involves reconstructing high-resolution velocity and pressure fields of viscous incompressible flow around a circular cylinder using sparse velocity data. We compare the performance of the conventional PINNs, PINNs with Fourier feature embedding [7] (referred to as FF-PINNs in this study), and FENN. For the hyperparameter $\sigma$ in FF-PINNs, we search for the optimal value for testing at intervals of 1 within the range $\sigma \in [0,10]$ recommended by the authors. The tanh activation function is employed, and the network parameters are initialized using Xavier normal initialization. The limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) optimizer is adopted for training, with the maximum number of inner iterations per epoch set to 20. The feature networks contain 3 hidden layers with 20 neurons per layer and are trained for 1000 epochs. The study develops programs based on the PyTorch platform and executes the algorithm on an NVIDIA GeForce RTX 4090 GPU. Since neural network results are affected by random initialization, each experiment is repeated five times, and the mean and standard deviation are reported.

### *3.1 Viscous incompressible flow around the NACA0012 airfoil*

We first solve the steady viscous incompressible flow around the NACA0012 airfoil, which is governed by the two-dimensional incompressible Navier–Stokes equations. The Reynolds number is set to $Re = 400$, and the angle of attack is set to $\alpha = 5°$. PINNs, FF-PINNs, and FENN are independently applied in solving the problem. In FF-PINNs, the hyperparameter $\sigma = 10$. All the networks contain 7 hidden layers with 64 neurons per layer. PINNs, FF-PINNs, and FENN are trained for 20000, 15000, and 10000 epochs, respectively. For more details, please refer to [13].

Figure 2(a) and (b) show the comparison of the loss convergence among PINNs, FF-PINNs, and FENN. To evaluate the solution accuracy, the pressure coefficient $C_p = (p - p_\infty)/[0.5\rho_\infty(u_\infty^2 + v_\infty^2)]$ on the airfoil is employed, where the reference solution is obtained using the finite volume method (FVM). The convergence history of the relative $L_2$ errors of $C_p$ for PINNs, FF-PINNs and FENN is shown in Figure 2(c). The vertical axis is plotted on a logarithmic scale for clarity. We observe that G-FENN, U-FENN, and GU-FENN exhibit similar performance and converge faster than PINNs and FF-PINNs. Table 1 summarizes the relative $L_2$ errors and wall times of the three methods after being trained for epochs listed in the table. A comparison of the wall times shows that G-FENN is 3.60 times faster than PINNs and 1.93 times faster than FF-PINNs.

### *3.2 Inverse problem involving viscous incompressible flow around a circular cylinder*

Next, we test the performance of FENN in solving inverse problems. Solving inverse problems is a limitation of CFD methods, as they require extremely high computational cost [6]. In contrast, PINNs allow for convenient integration

*Corresponding Author, Jiahao Song jiahaosong@mail.nwpu.edu.cn

of observed data, demonstrating significant advantages over CFD methods in solving such problems. We consider reconstructing high-resolution velocity and pressure fields for steady viscous incompressible flow around a circular cylinder based on sparse velocity data when boundary conditions are unknown, which is a standard inverse problem in fluid mechanics. The Reynolds number is set to $Re = 40$. We consider two cases where the number of observed data points is $N_d = 20$ and $N_d = 100$. PINNs, FF-PINNs, and FENN are independently applied in solving the problem. In FF-PINNs, the hyperparameter $\sigma = 6$. All the networks contain 5 hidden layers with 64 neurons per layer. PINNs, FF-PINNs, and FENN are trained for 6000, 3000, and 2000 epochs, respectively.

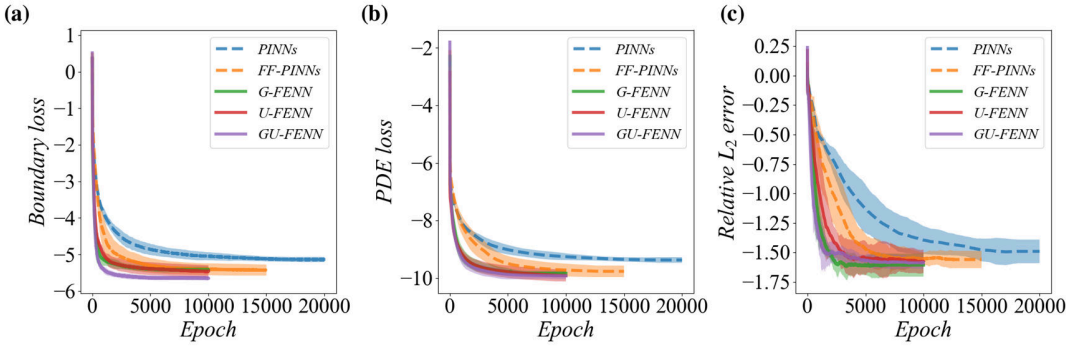



Figure 2 Comparison of the convergence history obtained by PINNs, FF-PINNs, and FENN for solving viscous incompressible flow around the NACA0012 airfoil. The curves and shaded regions represent the mean and two standard deviations.

Table 1 Comparison of relative $L_2$ error and wall time of PINNs, FF-PINNs, and FENN for solving viscous incompressible flow around the NACA0012 airfoil. The mean and one standard deviation are reported.

| | PINNs | FF-PINNs | G-FENN | U-FENN | GU-FENN |
|---|---|---|---|---|---|
| Epoch | 10000 | 5000 | 2500 | 2500 | 2500 |
| Error (%) | $4.03 \pm 0.59$ | $3.39 \pm 0.53$ | $2.55 \pm 0.29$ | $3.52 \pm 0.55$ | $3.12 \pm 0.23$ |
| Wall time (min) | $53.82 \pm 4.48$ | $28.79 \pm 2.77$ | $\mathbf{14.93 \pm 1.26}$ | $15.26 \pm 1.81$ | $16.17 \pm 0.92$ |

Figure 3 shows the convergence history of PINNs, FF-PINNs, and FENN. We observe that FENN achieves lower errors than PINNs and FF-PINNs under $N_d = 20$, demonstrating its capability to significantly improve the reconstruction accuracy of the flow field under extremely sparse data. In addition, G-FENN exhibits lower errors than U-FENN, possibly because the geometric features implicitly contain the boundary information of the cylinder, which is critical for reconstructing the flow field. As $N_d$ increases to 100, the errors of the three methods become nearly identical. This is expected, as more flow field constraints are incorporated. Table 2 and Table 3 summarize the relative $L_2$ errors and wall times of the three methods under $N_d = 20$ and $N_d = 100$, respectively. When $N_d = 100$, G-FENN achieves a speedup of 3.65 over PINNs and 1.97 over FF-PINNs in terms of wall time.

### *3.3 Viscous compressible flow around the NACA0012 airfoil*

In this section, we validate the effectiveness of physical features by solving steady viscous compressible flow around the NACA0012 airfoil, which is governed by the two-dimensional compressible Navier–Stokes equations. The Reynolds number is set to $Re = 400$, the Mach number is set to $Ma = 0.7$, and the angle of attack is set to $\alpha = 20°$. In this flow condition, the flow exhibits separation induced by the large angle of attack. We solve the flow using PINNs and G-FENN. All the networks are constructed with 7 hidden layers, each containing 64 neurons, and are trained for 6000 epochs.

Figure 4 illustrates the pressure coefficient on the airfoil obtained by PINNs and G-FENN. Clearly, the results of PINNs exhibit large discrepancies compared with the reference solution. In contrast, the results of the G-FENN show a small deviation from the reference solution. Furthermore, we plot the streamlines of the separated flow, as shown in Figure 5. We observe that the separation vortices obtained by the G-FENN are in good agreement with the reference

*Corresponding Author, Jiahao Song jiahaosong@mail.nwpu.edu.cn

solution. To the best of our knowledge, this is the first time that a PINN-like method has successfully solved forward problem involving compressible viscous flow.

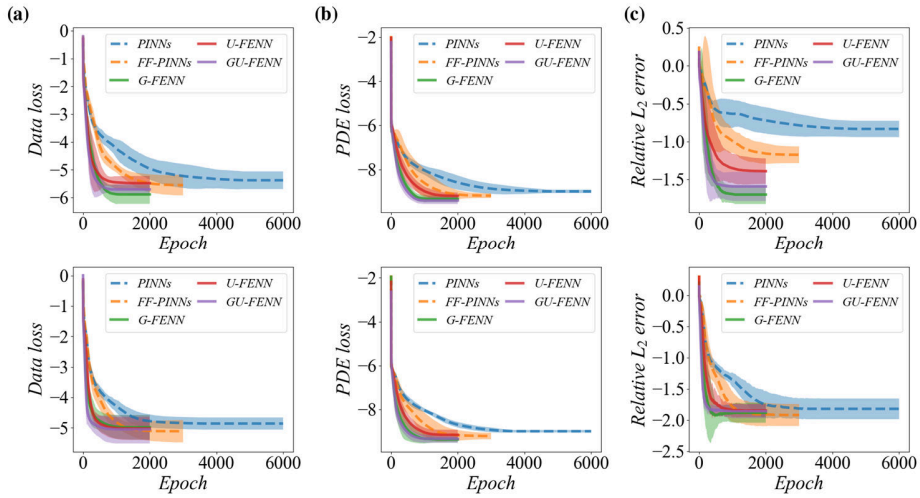



Figure 3 Comparison of the convergence history obtained by PINNs, FF-PINNs, and FENN for solving the inverse problem involving viscous incompressible flow around a circular cylinder at $N_d = 20$ (top) and $N_d = 100$ (bottom). The curves and shaded regions represent the mean and two standard deviations.

Table 2 Comparison of relative $L_2$ error and wall time of PINNs, FF-PINNs, and FENN for solving the inverse problem involving viscous incompressible flow around a circular cylinder at $N_d = 20$. The mean and one standard deviation are reported.

| | PINNs | FF-PINNs | G-FENN | U-FENN | GU-FENN |
|---|---|---|---|---|---|
| Epoch | 3000 | 1600 | 800 | 800 | 800 |
| Error (%) | 16.37 ± 3.04 | 7.46 ± 1.33 | **2.08 ± 0.32** | 4.84 ± 1.32 | 2.58 ± 0.60 |
| Wall time (min) | 18.51 ± 1.42 | 10.26 ± 0.98 | 5.07 ± 0.53 | 4.95 ± 0.35 | 5.10 ± 0.25 |

Table 3 Comparison of relative $L_2$ error and wall time of PINNs, FF-PINNs, and FENN for solving the inverse problem involving viscous incompressible flow around a circular cylinder at $N_d = 100$. The mean and one standard deviation are reported.

| | PINNs | FF-PINNs | G-FENN | U-FENN | GU-FENN |
|---|---|---|---|---|---|
| Epoch | 3000 | 1600 | 800 | 800 | 800 |
| Error (%) | 1.53 ± 0.32 | 1.25 ± 0.30 | 1.28 ± 0.22 | 1.52 ± 0.25 | 1.40 ± 0.17 |
| Wall time (min) | 17.24 ± 1.13 | 9.30 ± 0.69 | **4.72 ± 0.39** | 4.81 ± 0.30 | 4.89 ± 0.29 |

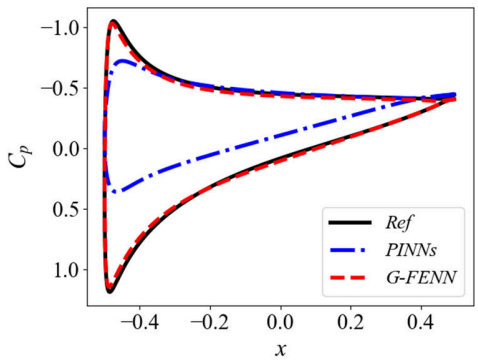



Figure 4 Comparison of the pressure coefficient on the NACA0012 airfoil obtained by various methods for solving viscous compressible flow over the NACA0012 airfoil. For clarity, we only plot the results of the first training.

*Corresponding Author, Jiahao Song jiahaosong@mail.nwpu.edu.cn

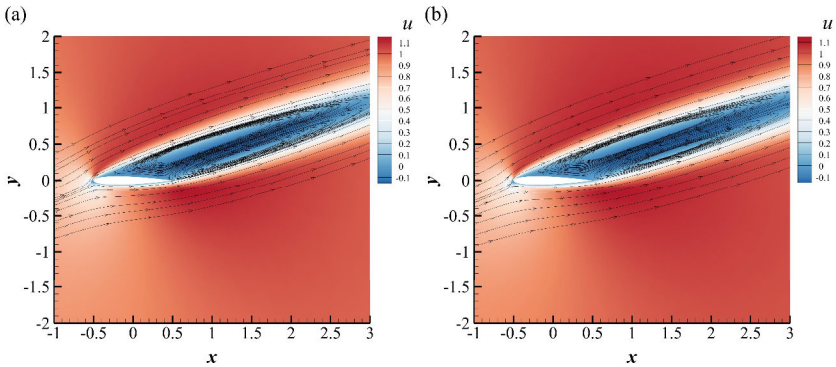



Figure 5 Streamlines near the airfoil obtained by (a) FVM and (b) G-FENN for solving the compressible viscous flow around the NACA0012 airfoil.

## 4. Conclusion

In this study, we propose the feature-enhanced neural network to solve PDEs involving fluid dynamics. Geometric features and physical features are introduced in the inputs of PINNs, enabling FENN to learn the flow more easily. Consequently, the accuracy and efficiency of solving PDEs involving fluid dynamics are improved. The designed features offer the advantages of low computational cost and high relevance to the problem being solved. To prevent incorrect partial derivatives in FENN due to neglecting the partial derivatives of the features with respect to space-time coordinates, we establish the feature networks to replace the features in the FENN inputs and train them offline in advance. The effectiveness of FENN is validated through solving forward and inverse problems involving different flow conditions.

## References

[1] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics, 378 (2019) 686-707.

[2] R. Courant, K. Friedrichs, H. Lewy, Über die partiellen Differenzengleichungen der mathematischen Physik, Mathematische annalen, 100 (1928) 32-74.

[3] R. Eymard, T. Gallouët, R. Herbin, Finite volume methods, Handbook of numerical analysis, 7 (2000) 713-1018.

[4] R. Courant, Variational methods for the solution of problems of equilibrium and vibrations, Bulletin of the American Mathematical Society, 49 (1943) 1-23.

[5] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, Journal of Marchine Learning Research, 18 (2018) 1-43.

[6] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations, Science, 367 (2020) 1026-1030.

[7] S. Wang, H. Wang, P. Perdikaris, On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks, Computer Methods in Applied Mechanics and Engineering, 384 (2021) 113938.

[8] A.D. Jagtap, K. Kawaguchi, G.E. Karniadakis, Adaptive activation functions accelerate convergence in deep and physics-informed neural networks, Journal of Computational Physics, 404 (2020) 109136.

[9] S. Wang, Y. Teng, P. Perdikaris, Understanding and mitigating gradient flow pathologies in physics-informed neural networks, SIAM Journal on Scientific Computing, 43 (2021) A3055-A3081.

[10] J.N. Kutz, Deep learning in fluid dynamics, Journal of Fluid Mechanics, 814 (2017) 1-4.

[11] C. Jiang, R. Vinuesa, R. Chen, J. Mi, S. Laima, H. Li, An interpretable framework of data-driven turbulence modeling using deep neural networks, Physics of Fluids, 33 (2021).

[12] J.-W. Hu, W.-W. Zhang, Mesh-Conv: Convolution operator with mesh resolution independence for flow field modeling, Journal of Computational Physics, 452 (2022) 110896.

[13] J. Song, W. Cao, W. Zhang, FENN: Feature-enhanced neural network for solving partial differential equations involving fluid mechanics, Journal of Computational Physics, 542 (2025) 114370.

*Corresponding Author, Jiahao Song jiahaosong@mail.nwpu.edu.cn

# Benchmarking PINN architectures for temperature-reconstruction by determining viscosity deviations

[1]Michael Mommert*; [1,2]Marie-Christine Volk; [1]Christian Bauer

*[1]German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology,*

*Department Ground Vehicles, Bunsenstr. 10, 37073 Göttingen, Germany;*

*[2]Technische Universität Ilmenau, Institute of Thermodynamics and Fluid*

*Mechanics, 98693 Ilmenau, Germany*

**Keywords:** PINN, temperature reconstruction, thermal convection, trainable viscosity

**Abstract.** In search of efficient PINN architectures to reconstruct temperature fields from velocity vector data, we explore the usage of the dimensionless viscosity  as trainable parameter of a PINN model.
The results show that it improves the temperature reconstruction results. The estimation of the viscosity through the training can further be used as performance indicator, when an expected value is known.

## 1. Introduction

Since their introduction [1], physics-informed neural networks (PINNs) have been brought to a great variety of applications [2]. The particular application investigated here is the reconstruction of temperature fields [3, 4], prospectively based on velocity measurements by particle image velocimetry [5] or particle tracking velocimetry [6, 7].
While this type of application is currently predominantly investigated for canonical flows, like Rayleigh-Bénard convection, future applications to non-canonical flow problems pose questions regarding the non-dimensionalization of the Navier-Stokes equations. In this context, the dimensionless numbers, and especially the Rayleigh number , which sets the convective forcing in relation to diffusive forces, are often not clearly defined.
Hence, we examine whether  can be unknown to reconstruct temperature and pressure fields from sparse velocity fields. This also raises the question whether  can be reconstructed from the dimensionless viscosity  of the Navier-Stokes equations with adequate accuracy within the same process.
Vice versa, if prior knowledge of  exists, its reconstruction accuracy can be used as a performance indicator for the field's reconstruction, which is investigated in the second part of the paper for different network architectures.

## 2. Use Case

The use case for the presented investigation is Rayleigh-Bénard convection inside a cubic enclosure. Its boundaries are no-slip walls. While the side walls are considered as adiabatic, the bottom and top faces are set to have constant temperatures  and , respectively.
Together with the respective quantities expressed in dimensional units (indicated by ), namely the gravitational acceleration , the thermal expansion coefficient , the thermal diffusivity , the kinematic viscosity , and the height of the domain , the system becomes fully described by the Prandtl number  and the Rayleigh number .
By applying the Oberbeck-Boussinesq approximation, the resulting three-dimensional flow is governed by the following set of partial differential equations, in which  is the pressure field:

(1)

(2)

(3)

In order to evaluate the PINN performance against ground truth data, a direct numerical simulation of the flow at and  were conducted (for details, see [4] or [5]). Figure 1 gives an impression of the resulting

turbulent flow, with hot and cold plumes detaching from the bottom and top plates, respectively, which is visualised through a volume rendering of the fluid temperature.

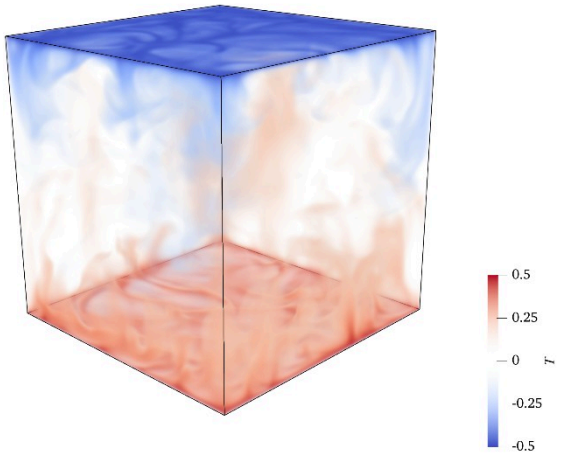



*Figure 1: Instantaneous volume rendering of the fluid temperatures in cubic Rayleigh-Bénard convection at Ra=$10^8$ and Pr=0.71.*

To emulate measured velocity () data, the data provided to the PINN is interpolated at randomly distributed points. Spatially, Poisson disk sampling [8] is employed for this task. In contrast to mimicking particle tracking data, these positions remain locked for the progression over the selected time steps, to avoid any leakage of spatial resolution. Related to the dimensionless parameters sample height $H$ and free-fall time , the here examined sampling densities are  in space and  in time. To account for statistical fluctuations of the PINN method, each case is tested for three different random samplings according to the aforementioned parameters, covering  at between  and  spatial locations, which is well within the limit achievable for a real experimental setup [5].

## 3. PINNs with trainable viscosity

PINNs are neural networks, which are set up to represent the function , where  is the totality of trainable parameters, including the trainable viscosity.

For the following investigation, we consider three different architecture types to pose as the function .

The first one is represented by sine-activated multi-layered perceptrons (MLPs) [9, 4]. They are a sequence of  dense layers with neurons each. Of these layers, the weights of the first one are initialised with a three-times wider distribution than the standard for sine-activated layers [9] to provide higher wavenumbers. This network type also includes a final linear dense layer which combines the  features of the of the penultimate layer to the 5 output fields [5].

The second architecture considered are modified Pirate-Nets (PNs) [10], which are an architecture based on residual blocks with gating operations referring to shallow latent features. The here examined networks differ from the original publication in the following aspects: All layers are sine-activated. Therefore, the first layer, which originally acts as a fixed Fourier embedding, is implemented as the same sine-activated layer, with a wider weight initialization like for the MLP. The outputs of these networks are also combined by an additional linear layer, as in the MLP. This way, we isolate the effect of the introduction of the shallow-latent-feature layers incorporated by gating operations in the first two layers of each of the three-layer-wide blocks, the residual operation for the last layer of these  blocks, and the usage of random weight factorization layers [10] throughout the blocks.

The third and last considered architecture type is the Kolmogorov-Arnold network with Chebychev polynomials (cKANs), which was e.g. used by [6]. For this type of architecture, each layer first performs a non-linear (Chebychev polynomial of 5th degree) transformation for each of its inputs, before they are combined by summation. These networks are also specified by their number of layers  and neurons .

For the present comparison, these networks were trained by an Adam optimizer with the learning rate , with the combined loss function

(4)

, (5)

, (6)

. (7)

The data, sampled as described in section 2, is divided into batches of size 4096 for each training step. The losses , , and are simultaneously evaluated at the same number of collocation points, random-uniformly sampled within the domain for each step. Regarding the trainable parameter , only gradients originating in are considered for its optimization. Also random-uniformly sampled, 512 boundary point are generated on the faces of the cube for each step. On all faces, a mean-squared error (MSE) loss is computed for the wall-normal () and wall-parallel () velocity components. On the top and bottom faces, the temperature is prescribed via an MSE loss (). For comparability, all investigated cases cover optimization steps.

## 4. Rayleigh number reconstruction

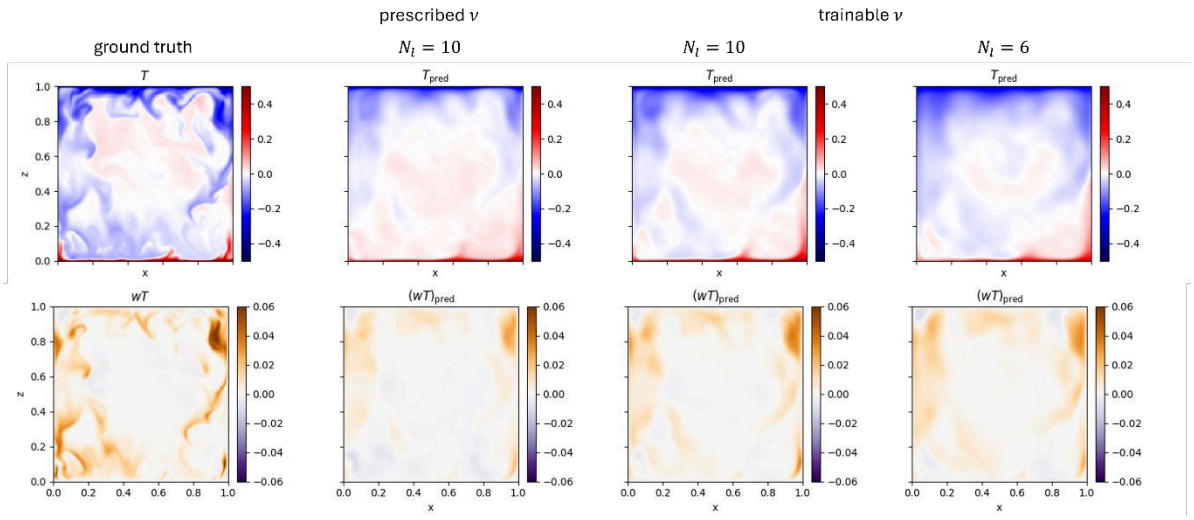



*Figure 2: Comparison of the fields of the temperatures T and convective heat fluxes wT for Rayleigh-Bénard convection at Ra=$10^8$. From left to right: Ground truth, PINN reconstruction with prescribed viscosity (MLP, $N_n$=256, $N_l$=10) and trainable viscosity for networks with 10 and 6 layers ($N_n$=256).*

To investigate reconstruction-capabilities of the Rayleigh number, related to the trainable viscosity by , we focus on two exemplary runs for the 'fine' temporal and spatial sampling of the case with an MLP with and . Additionally, we regard a respective case with without a trainable viscosity.
Figure 2 shows a central section of the respectively reconstructed temperature fields against the ground truth on the left. Overall, the PINNs are not able to reconstruct the features on all scales, due to the limited provided sampling density. However, with regard to the cases, making trainable actually improves the results from Pearson correlation coefficient of the temperature field from to . This improvement becomes especially pronounced in the field of the convective heat flux, for which the case with trainable viscosity produces a much better prediction in the region of the bottom left corner. With a reduced number of layers, see the case, this effect prevails, while the deficit in small scale reconstruction increases due the reduced expressivity.
This shows that introducing as a trainable parameter is beneficial, as it appears to open new training trajectories which lead to an improved reconstruction. Thereby, the predicted Rayleigh numbers of the

two cases are and for and , respectively. They are robust predictions, but indicate a systematic overprediction of the viscosity due to the limitations of the sampling resolution and network expressivity.

## 5. Benchmarking

In this section, we explore the performance of the different network types with a trainable on the sampling case described in section 2. For this, the metric of the Pearson correlation coefficient of the temperature field is used. First, we consider the pure performance of the different models with respect to this metric. Second, we investigate whether the relative deviation of can be used as a performance indicator when both and Pr are known.

### *5.1. Model complexity*

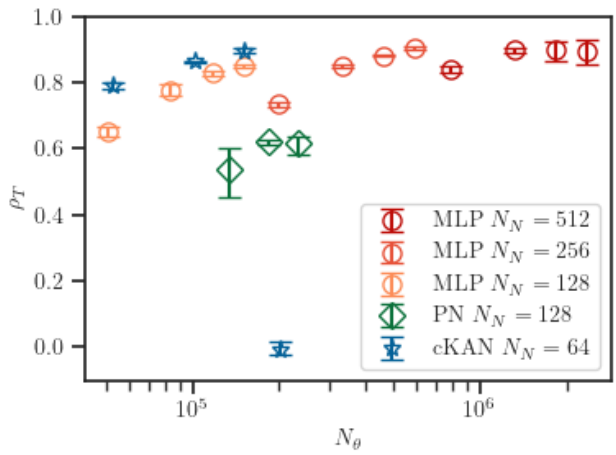



*Figure 3: Pearson correlation coefficient of the temperature field for different network architectures MLP, PN, and cKAN with varying numbers of layers $N_l$ and neurons per layer $N_N$, leading to different total numbers of trainable parameters $N_\theta$. The error bars indicate the spread between minimum and maximum temperature correlation for 3 runs with different random samplings.*

To investigate the effect of model complexity on performance, the architecture of the MLPs was varied in terms of the number of layers, , and neurons per layer, . For the PN, the number of blocks, , was varied for a prescribed layer width of . For the cKAN, the layer width was set to with as depths.

Figure 3 shows the results of these benchmarks plotted over the total number of trainable parameters . It reveals a trend for all network types, that increasing depth improves the results, however with diminishing returns. For the MLPs, figure 3 further shows that, for a similar total trainable parameter count, the option with more but smaller layers appears to be more efficient.

Comparing the different network types with each other renders the cKAN as the most efficient architecture considering total parameter count, equalling the performance of significantly larger MLPs. The here used PNs lack in performance, which contradicts other investigations [10], and poses the question whether this is related to the specific use case or the modifications applied here.

Another overall trend is that very large, and especially deep, networks become overall harder to train. This shows in an increased variation of the results of the largest MLP, a small decline in the mean performance of the largest PN, and a complete loss of trainability for the largest cKAN.

While the number of trainable parameters is relevant for the memory requirements, figure 4 shows the performance indicator of the different networks plotted over the expended wall time of the training on an Nvidia A100 GPU. In this regard, the multiplicative operations within PNs are especially expensive, which leads to them having the largest wall time requirements for training. The backpropagation for cKANs is also significantly less efficient compared to the one of MLPs, as their training requires wall-times similar to MLPs with widths of or , while having the parameter count equivalent to MLPs with a width of Hence, within this comparison, MLPs are the most efficient option regarding the required wall

time and therefore also energy. Within the MLP category, narrower yet deeper networks generally yield better results for a given wall time.

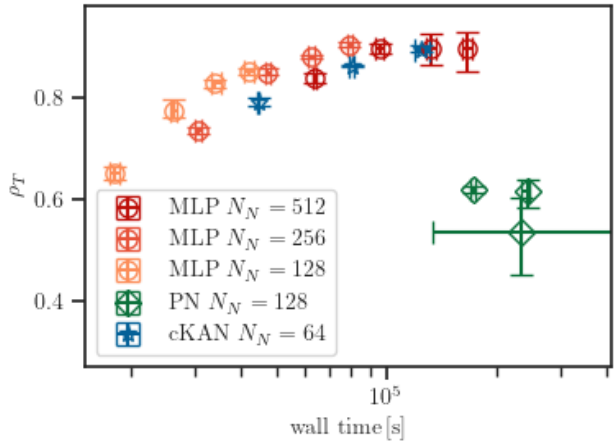



*Figure 4: Pearson correlation coefficient of the temperature field for different network architectures MLP, PN, and cKAN with varying numbers of trainable parameters plotted over the wall time expended for $10^6$ training steps (data point of one color vary in network depth). The error bars indicate the spread between minimum and maximum temperature correlation for 3 runs with different random samplings.*

### *5.2. Trainable viscosity as performance indicator*

For the cases considered in the previous section, we now also consider their trained viscosity value, in particular its relative deviation from the ground truth .

Figure 5 exhibits its relation to the Pearson correlation metric of the temperature reconstruction. It shows that there is a strong negative correlation between the two metrics for each individual network type. While the point clouds for MLPs and cKANs even overlap, the PN cases form a separate point cloud. This shows that the deviations of a trainable viscosity can be used as performance indicators, as long as no major changes on the network architecture occur. This can be explained by a better prediction of the viscosity requiring the network to render as many scales of the flow as possible correctly, which is also beneficial for the temperature reconstruction performance.

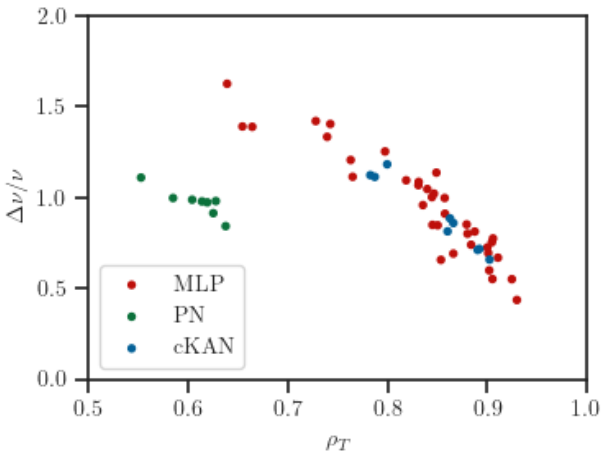



*Figure 5: Relative deviation of the trainable viscosity of the cases (single dots for every of the three random samplings) from section 5.1 plotted over their Pearson correlation coefficient for the temperature field. The different colors indicate the network type.*

## 6. Conclusion and Outlook

We have shown that allowing PINNs to train the dimensionless viscosity improves temperature reconstruction results and yields robust Rayleigh number estimates. The subsequent benchmarking of sine-activated multi-layered perceptrons, modified Pirate-Nets and Chebychev-Kolmogorov-Arnold networks revealed that, of the three, multi-layered perceptrons and Kolmogorov-Arnold networks are

the most efficient for the task at hand. The former is especially efficient regarding wall-time, while the latter is especially efficient when it comes to parameter count and therefore memory usage.
It was also found that the estimation of the viscosity through training can itself function as performance indicator when the Rayleigh and Prandtl numbers are known.
Future research directions include expanding to different temporal and spatial sampling densities, as well as incorporating additional network architectures.

## 7. Acknowledgments

The authors gratefully acknowledge the scientific support and HPC resources provided by the German Aerospace Center (DLR). The HPC system CARA is partially funded by "Saxon State Ministry for Economic Affairs, Labour, Energy and Climate Action" and "Federal Ministry of Research, Technology and Space".
M.M. gratefully acknowledges the financial support of the German Research Foundation (DFG) under grant MO 5240/1-1.

## 8. References

[1] M. Raissi, P. Perdikaris, G.E. Karniadakis 'Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations', J Comput Phys, https://doi.org/10.1016/j.jcp.2018.10.045
[2] S. Cai, Z. Mao, Z. Wang, M. Yin, G.E. Karniadakis 'Physics-informed neural networks (PINNs) for fluid mechanics: a review', Acta Mech Sin, https://doi.org/10.1007/s10409-021-01148-1
[3] D. Lucor, A. Agrawal, A. Sergent 'Simple computational strategies for more effective physics-informed neural networks modeling of turbulent natural convection', J Comput Phys, https://doi.org/10.1016/j.jcp.2022.111022
[4] M. Mommert, R. Barta, C. Bauer, M.-C. Volk, C. Wagner 'Periodically activated physics-informed neural networks for assimilation tasks for three-dimensional Rayleigh–Bénard convection', Comput & Fluids, https://doi.org/10.1016/j.compfluid.2024.106419
[5] M.-C. Volk, A. Sergent, D. Lucor, M. Mommert, C. Bauer, C. Wagner 'A PINN methodology for temperature field reconstruction in the PIV measurement plane: Case of Rayleigh–Bénard convection', Int Comm Heat Mass Transf, https://doi.org/10.1016/j.icheatmasstransfer.2025.109284
[6] J.D. Toscano, T. Käufer, Z. Wang, M. Maxey, C. Cierpka, G.E. Karniadakis 'AIVT: Inference of turbulent thermal convection from measured 3D velocity data by physics-informed Kolmogorov-Arnold networks', Sci Adv, https://doi.org/10.1126/sciadv.ads5236
[7] R. Barta, M.-C. Volk, C. Bauer, C. Wagner, M. Mommert 'Temperature and pressure reconstruction in turbulent Rayleigh–Bénard convection by Lagrangian velocities using PINN', Meas Sci Technol, https://doi.org/10.1088/1361-6501/adee38
[8] R.L. Cook 'Stochastic sampling in computer graphics', ACM Trans Graph, https://doi.org/10.1145/7529.8927
[9] V. Sitzmann, J. Martel, A. Bergmann, D. Lindell, G. Wetzstein 'Implicit Neural Representations with Periodic Activation Functions', NeurIPS 2020, Online, 6-12 December 2020
[10] S. Wang, B. Li, Y. Chen, P. Perdikaris 'PirateNets: Physics-informed Deep Learning with Residual Adaptive Networks', J Mach Learn Res, http://jmlr.org/papers/v25/24-0313.html

# Data-Driven and Physics-Informed Graph Neural Networks for Flow Field Prediction in Two-Dimensional Channel Flows Featuring Bluff Body Obstacles

[1]Alireza Rouhi*; [1]Adrian Krämer; [1]Carsten Mehring;

[1]*Institute of Mechanical Process Engineering, University of Stuttgart, Germany*

## Abstract

This study investigates the application of Graph Neural Network (GNN) models to predict pressure and velocity fields in laminar, steady channel flows containing obstacles of varying shapes, sizes, orientations and positions at different Reynolds numbers. The models are based on an EdgeConv message passing architecture with multilayer perceptrons (MLPs) and are trained using both data-driven and physics-informed loss functions. In the physics-informed formulation, the conservation equations for mass and momentum are used in the loss formulation. The dataset was generated using the computational fluid dynamics (CFD) solver STAR-CCM+. After adjusting key hyperparameters, the models were trained with the objective of achieving robust generalization across the considered configuration space. The results show that the purely data-driven approach predicts velocity fields with high accuracy but has difficulties in predicting pressure fields. These issues are found to primarily origin from the higher variance in the pressure distribution. By shifting the global minimum in the pressure field to $p = 0\ Pa$ in all cases, the prediction accuracy improves significantly. Further evaluation demonstrates that the data-driven model generalizes reliably to other Reynolds numbers in the laminar regime, although generalization across substantially different geometries remains a challenge. In the physics-informed model, two gradient approximation methods were employed instead of automatic differentiation, resulting in more accurate gradient predictions and significantly reduced memory usage. However, the physics-informed model did not show improved predictions. Instead, it increased computational cost, slowed training, and produced satisfactory results only on a simplified dataset, where the achieved accuracy was comparable to values reported in the literature. In contrast, the purely data-driven approach consistently outperformed the physics-informed model.

**Keywords:** Graph neural networks, Physics-informed neural networks, Data-driven, Computational fluid dynamics, Fluid mechanics, Channel flow, Bluff bodies, Generalization.

# Fidelity Scaling via Physics-Informed Graph Learning for Marine Propeller Blades

[1]Ian Hubbard*, [2]Evert-Jan Foeth
[1]*Hydrodynamics Research, Wärtsilä Netherlands B.V.*
[2]*R&D, Maritime Research Institute Netherlands (MARIN).*

**Keywords:** Physics-Informed Machine Learning, Graph Neural Networks, Multi-Fidelity Modelling, Geometric Deep Learning, Marine Propeller Cavitation

**Abstract.** In the design optimization of complex geometries, high-fidelity simulations are often required to accurately guide objectives and constraints, but their computational cost limits their use in large-scale design exploration. Lower-fidelity solvers and surrogate models enable broader exploration, but can lose accuracy for key performance quantities, particularly in cavitation-sensitive regimes. This work addresses this trade-off with a cross-resolution encoder-decoder graph neural network that learns the gap between low- and high-fidelity boundary element method (BEM) simulations of marine propeller pressure fields. The blade surface is represented as a graph using a UV parameterization derived from the structured BEM grid, which maps the 3D surface to a unit-square parameter domain shared across resolutions. A graph convolutional encoder maps a coarse, non-cavitating mesh to per-node latent embeddings. These embeddings are transferred to a finer cavitating mesh through a UV-distance-based cross-attention mechanism, replacing heuristic interpolation with a geometry-aware, end-to-end differentiable coarse-to-fine mapping. The model is trained and evaluated on approximately 366,000 graph samples spanning multiple propeller geometries and operating conditions, and accurately predicts cavitating pressure distributions on the fine mesh. A data-efficiency study comparing the physics-informed variant, which uses the low-fidelity wetted $C_{P_N}$ field as input, with a geometry-only baseline demonstrates that the physics-based prior improves both sample efficiency and prediction accuracy.

## 1 Introduction

Design optimization of marine propellers requires repeated flow evaluations to steer objectives and constraints. Surface pressure distributions are particularly important because they determine blade loading and indicate cavitation behaviour, directly relevant for performance, vibration, noise, and erosion risk. High-fidelity simulations provide accurate predictions but are computationally prohibitive for large-scale design exploration, while lower-fidelity solvers sacrifice accuracy in cavitation-sensitive regimes. This motivates learning the gap between fidelity levels rather than replacing the solver outright. The model presented here predicts the fine-resolution cavitating pressure coefficient $C_{P_N}^{\text{cav}}$ from the coarser wetted field $C_{P_N}^{\text{wet}}$, both computed with the BEM solver PROCAL [1]. Two gaps are bridged in a single forward pass: a physical-model gap (non-cavitating to cavitating BEM) and a resolution gap (coarse to fine panel mesh), whereby the fine-grid cavitating calculations acts as a proxy for high-fidelity flow solutions.

This problem is informed by several recent works spanning the two fidelity gaps. On the physical-model side, low-fidelity pressure prior from panel methods have proven valuable on surface meshes for lifting bodies: boundary GNNs for 2D airfoils [2] use inviscid pressure to substantially reduce model and

training-set size, VortexNet [3] applies related ideas to 3D Delta wings, and a similar prior has been applied to propeller cavitation impact-loading [4]. On the resolution side, multiscale GNN message passing [5] bridges coarse and fine meshes via graph pooling, but without a physical-model prior. Neural operator methods more broadly, such as DeepONet [6], and transformer-based operators [7], have established that decoders can be queried at arbitrary spatial locations independent of training resolution, a principle adopted here through a cross-attention formulation [8] tailored to the structured panel-mesh setting.

The proposed architecture combines a graph convolutional encoder [9] on the coarse panel mesh with a cross-attention decoder queryable at arbitrary UV coordinates on the fine mesh. Coarse nodes act as fixed physical anchors carrying $C_{P_N}^{\text{wet}}$ as a per-node physics prior, and the attention mechanism is given by a geometric kernel in UV space rather than a learned dot product. The shared UV parameterization enables decoder queries at arbitrary surface locations without heuristic interpolation, while the structured panel mesh provides the graph connectivity used by the encoder. The main aims are to assess whether such a cross-resolution GNN can accurately learn the BEM fidelity gap, and to quantify how the physics prior affects data efficiency.

## 2 Physical Modelling & Dataset

The data are obtained with PROCAL [1], a potential-flow BEM solver for marine propeller analysis. For each operating condition, two paired simulations define the input and target fields: an unsteady, non-cavitating solution on a coarse blade-surface grid ($51 \times 19 = 969$ nodes), and a cavitating solution on a finer grid ($81 \times 31 = 2{,}511$ nodes). The quantity of interest is the pressure coefficient,

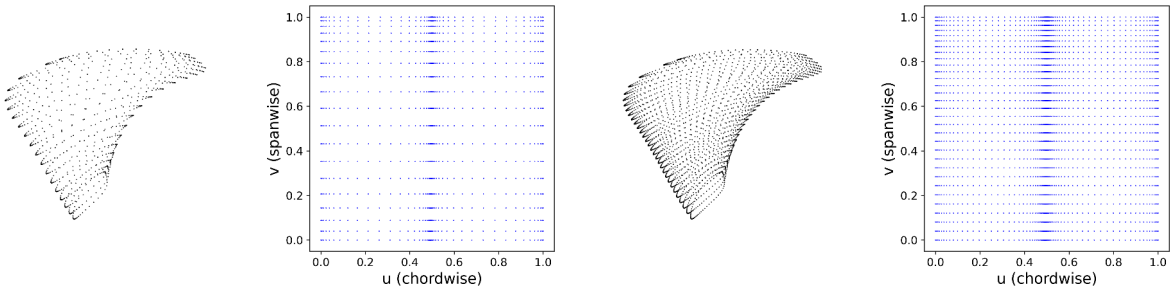
$$C_{p_N} = \frac{p - p_{ref}}{0.5\rho n^2 D^2}, \tag{1}$$

where $p$ is the local blade-surface pressure, $p_{ref}$ the static upstream pressure, $\rho$ is the fluid density, $n$ is the rotational speed, and $D$ is the propeller diameter. The wetted coefficient $C_{P_N}^{\text{wet}}$ serves as a physics-based input feature, while the cavitating coefficient $C_{P_N}^{\text{cav}}$ defines the target field.

A dataset is constructed using a nested Latin Hypercube Sampling (LHS) plan comprising 6000+ propeller variants reminiscent of the F-series propeller [10] with 30 geometric degrees of freedom. The propellers are evaluated at three advance coefficients $J = V_s(1 - w)/nD$, as a function of the mean pitch $P/D$, with $V_S$ as the ship speed and $w$ the effective wake factor due to an assumed non-uniform inflow. The latter follows from stock inflow wake fields, assigned to each propeller via a binning function in one LHS entry, with the $\sigma_N \in [1.1, 1.9]$ varied in five steps. Each operating condition is sampled over one full propeller revolution at $6^\circ$ angular increments, giving 60 instantaneous blade positions. The non-uniform ship wake causes the blade inflow to vary with angular position, so each snapshot is treated as an independent graph sample [4]. The dataset contains approximately 366,000 total graph samples spanning multiple propeller geometries and operating conditions, partitioned into fixed training (75%), test (15%), and holdout (10%) sets.

## 3 Model Architecture

The coarse blade surface is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose nodes are the wetted mesh points. The 3D surface is parameterised by a UV mapping derived from the structured BEM panel grid that unfolds the blade onto $[0, 1]^2$, as shown in Figure 1. Here, $i$ and $m$ index chordwise points, $j$ indexes spanwise sections, $I$ and $J$ denote the number of chordwise and spanwise points, and $\mathbf{p}_i^{(j)} = (X_{ij}, Y_{ij}, Z_{ij})$ is the corresponding 3D point position. The chordwise coordinate $u$ is defined by cumulative arc length along each spanwise section,




(a) Coarse, wetted grid and UV

(b) Fine, cavitating grid and UV

Figure 1: 3D blade geometries with shared UV parameterization, coarse encoder & fine decoder grids.

$$u_i^{(j)} = \frac{\sum_{m=1}^{i} \left\| \mathbf{p}_m^{(j)} - \mathbf{p}_{m-1}^{(j)} \right\|_2}{\sum_{m=1}^{I-1} \left\| \mathbf{p}_m^{(j)} - \mathbf{p}_{m-1}^{(j)} \right\|_2}, \qquad u_0^{(j)} = 0, \tag{2}$$

The spanwise coordinate $v$ is obtained from the mean radial position of each section,

$$v^{(j)} = \frac{\bar{r}_j - \bar{r}_0}{\bar{r}_{J-1} - \bar{r}_0}, \qquad \bar{r}_j = \frac{1}{I} \sum_{i=0}^{I-1} \sqrt{Y_{ij}^2 + Z_{ij}^2}, \tag{3}$$

where $\bar{r}_j$ is the mean radius of spanwise section $j$. The structured panel grid is exploited in two ways. The panel indices give graph connectivity via connected adjacency (chordwise, spanwise, and diagonal neighbours), and the arc-length parameterization defined by Eqs. 2 and 3 unfolds the blade into a 2D parameter domain shared across resolutions. The cross-attention kernel operates on this 2D domain, eliminating the through-blade shortcut that a 3D Euclidean kernel would introduce near the tightly spaced leading edge and thin trailing edges, where suction- and pressure-side panel points lie close in 3D but are separated along the surface.

The encoder consists of four graph convolutional layers with residual skip connections, progressively widening from the input dimension to the latent size. For node $i$ at layer $\ell$,

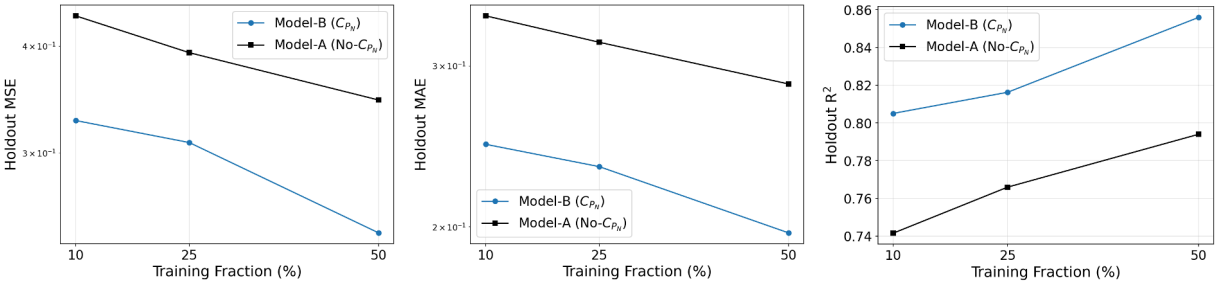
$$\mathbf{h}_i^{(\ell+1)} = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{\sqrt{\tilde{d}_i \tilde{d}_j}} \mathbf{W}^{(\ell)} \mathbf{h}_j^{(\ell)} + \mathbf{W}_{\text{skip}}^{(\ell)} \mathbf{h}_i^{(\ell)}, \tag{4}$$

where $\tilde{d}_i$ is the node degree. Graph normalisation, ReLU, and dropout are applied between layers. The model is additionally conditioned on global operating condition quantities $\mathbf{g} = [\sigma_n,\, J_A,\, J,\, J_R,\, s_{70}]^T$, being the cavitation number, apparent advance coefficient, advance coefficient (effective inflow), rotative advance coefficient, and slip ratio at 70% radius. These are mapped through a small MLP to the graph-convoluted embedding space,

$$\mathbf{h}_i' = \mathbf{h}_i + \mathbf{s}, \qquad \mathbf{s} = \text{MLP}_{\text{scalar}}(\mathbf{g}) \tag{5}$$

The decoder transfers information from the coarse encoder to the fine mesh through a modified cross-attention operation. In the standard transformer formulation [8], attention weights are derived from learned query and key projections. Here, we replace the learned score with a geometric distance kernel computed directly in UV space. For each decoder query point $\mathbf{u}_j^{\text{dec}}$ and encoder node $\mathbf{u}_k^{\text{enc}}$, the attention weight is,

$$\alpha_{jk} = \frac{\exp(-D_{jk}/\tau)}{\sum_{m=1}^{N_c} \exp(-D_{jm}/\tau)}, \qquad D_{jk} = \left\| \mathbf{u}_j^{\text{dec}} - \mathbf{u}_k^{\text{enc}} \right\|_2, \tag{6}$$




(a) MSE (b) MAE (c) $R^2$

Figure 2: Data efficiency: Model-B (with $C_{P_N}^{\text{wet}}$ prior) vs. Model-A (geometry only).

where $\tau$ controls the kernel's locality and $N_c$ is the number of coarse encoder nodes. Queries are decoder UV positions $\mathbf{u}_j^{\text{dec}}$, keys are encoder UV positions $\mathbf{u}_k^{\text{enc}}$, and values are the conditioned encoder embeddings $\mathbf{h}'_k$. With $\tau = 0.01$ held fixed during training, each query's effective support concentrates on its nearest coarse anchors in UV, tightening near the leading edge. The softmax retains nonzero weight on every anchor, giving each query access to the full coarse mesh rather than a fixed local neighbourhood. The result is a geometry-aware, bounded, differentiable interpolation mechanism whose locality is set by $\tau$ rather than learned projections, with context vector and final prediction given by,

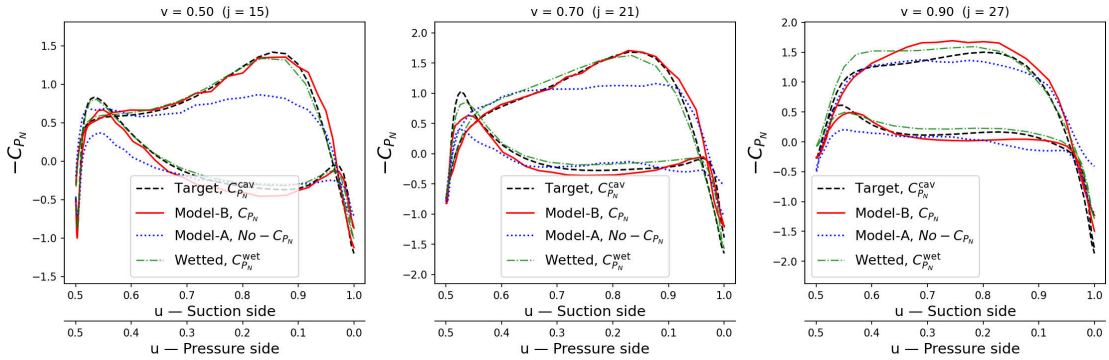
$$\hat{C}_{pN}^{\text{cav}}(u,v) = \text{MLP}_{\text{dec}}\left([\mathbf{c}(u,v) \mid\mid u,\ v]\right), \qquad \mathbf{c}(u,v) = \sum_{k=1}^{N_c} \alpha_k(u,v)\mathbf{h}'_k. \tag{7}$$

Since the attention weights are computed from UV coordinates during the forward pass rather than from a fixed graph topology, the decoder is not tied to the training mesh resolution. Queries can be placed at arbitrary surface locations, making the architecture discretisation independent on the output side while retaining the inductive bias of a structured graph encoder on the input side. The model has approximately 76k parameters and is trained with MSE loss using AdamW (LR=$10^{-3}$, weight decay $10^{-4}$, gradient clipping).

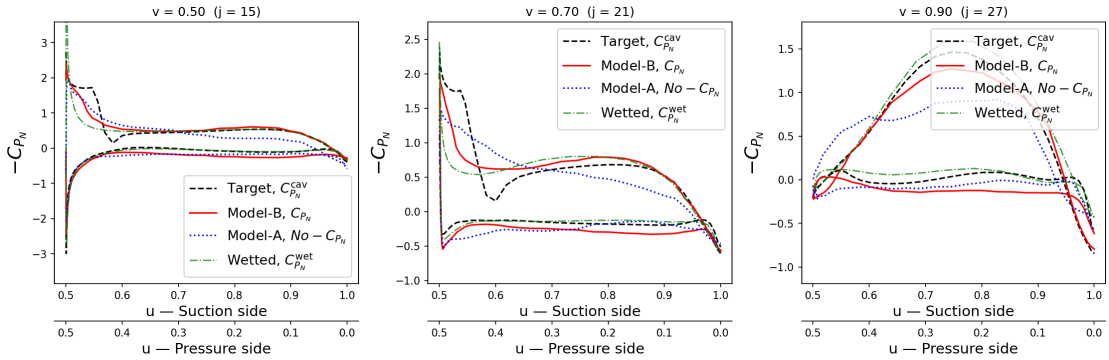
## 4 Results and Discussion

Two models are compared. Model-A is a geometry-only baseline using node positions and scalar conditioning, with no access to the low-fidelity pressure field. Model-B additionally receives $C_{P_N}^{\text{wet}}$ as a per-node input feature. Both share the same architecture and are trained at variable data fractions to assess efficiency. The benefit of the physics-informed variant is evident from Figure 2. The wetted field provides a spatially resolved hydrodynamic prior across the blade surface: in non-cavitating regions, $C_{P_N}^{\text{wet}}$ is essentially the target up to a small correction, so Model-B effectively learns the cavitation perturbation rather than the full pressure field. This reduces the data required to reach a given accuracy, in line with the broader finding that surface-pressure priors substantially improve data efficiency for graph-based pressure prediction [2, 4].

To test generalisation, both models, trained on 50% of the total training partition, are evaluated against the unseen holdout set. Figure 3 shows chordwise $C_{P_N}$ distributions at three radial sections for two holdout samples. Model-B predictions closely follow the target $C_{P_N}^{\text{cav}}$, while Model-A shows larger deviations, particularly in sensitive cavitation and pressure peak regions of the blade. Figure 4 shows suction-side views of the blade and the corresponding $C_{pN}$ surface distributions for the same holdout samples. In sample 2510, the distinction between Model-A and Model-B is particularly clear, where Model-B better preserves the low-$C_{pN}$ radial distribution across the blade leading edge, whereas Model-A smooths it over. This indicates that the wetted-pressure prior helps the model recover spatially localised pressure




(a) Holdout Sample 1100




(b) Holdout Sample 2510

Figure 3: Chordwise $C_{P_N}$ at three radial blade sections (0.5, 0.7 & 0.9 r/R) for Model-A, Model-B and $C_{P_N}^{\text{wet}}$ vs. target $C_{P_N}^{\text{cav}}$ on two individual holdout sample cases, shown in the UV reference frame.

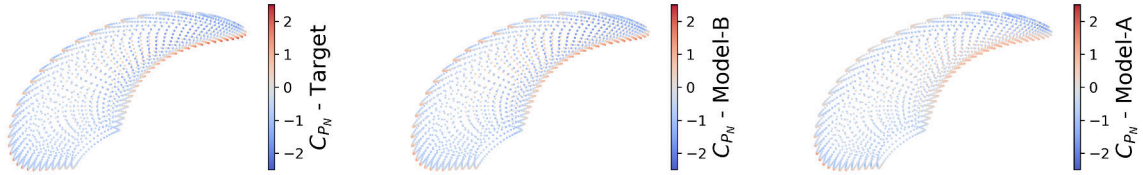
features. In addition, accuracy of predicted integral quantities, such as $K_T$ or $K_Q$, are assess on the hold-out set. The architecture does not incorporate a dedicated scalar prediction head for integral quantities, instead, direct numerical integration of the predicted pressure fields are performed, Figure 5, shows the integrated $K_T$ predicted by the models on a subsample of the holdout set, compared against the wetted and cavitating reference results obtained by the same surface integral. Based on $R^2$ and MAE, Model-B shows substantial improvements in accuracy.

## 5 Conclusion and Future Work

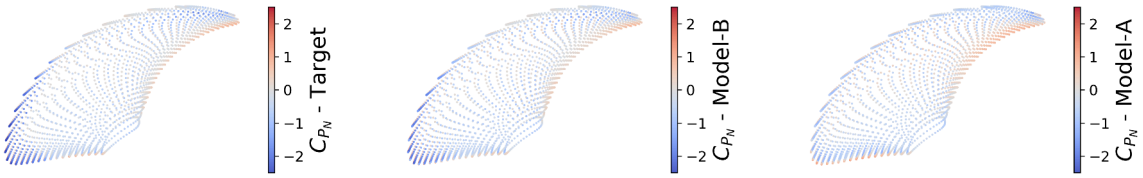
A cross-resolution graph neural network for fidelity-scaling across physical-model and resolution gaps in cavitating pressure prediction on marine propeller blades has been presented. The UV parameterization derived from the structured BEM panel grid supplies both a connected encoder graph and a surface-intrinsic metric for a cross-attention operator, enabling direct information transfer between coarse and fine resolutions without heuristic interpolation. The data-efficiency study confirms that physics-informed features (the low-fidelity wetted pressure) substantially improve accuracy and reduce training requirements compared to a geometry-only baseline. The approach extends an earlier feasibility study [4] by replacing k-NN-based upsampling with a UV cross-attention decoder, and by scaling to a larger dataset spanning multiple propeller geometries. The methodology presented here generalises to any panel-mesh BEM code whose output could serve as a physics-based prior. Future work aims to scale the method to higher-fidelity targets, such as URANS-derived pressure fields from full self-propulsion simulations.

## 6 Acknowledgements

This work was supported by the Cooperative Research Ships (CRS), https://crships.org/.

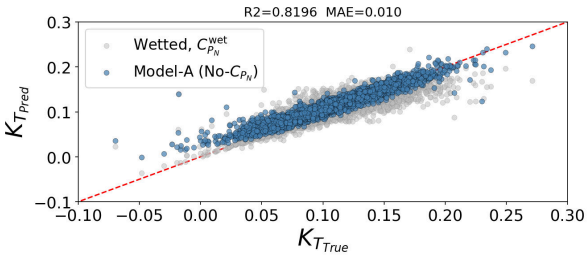



(a) Holdout Sample 1100

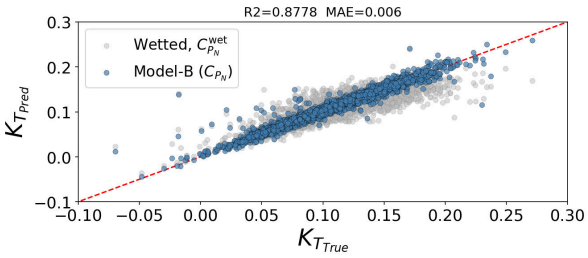



(b) Holdout Sample 2510

Figure 4: Suction side view of the $C_{P_N}$ surface distributions over the blade for two holdout set predictions. Left: Target $C_{P_N}^{\text{cav}}$, middle: Model-B predictions, right: Model-A predictions




(a) Model-A




(b) Model-B

Figure 5: Comparison of integrated $K_T$ from Model-A, Model-B, and the wetted reference against the target cavitating value $K_{T_{True}}$ on the hold-out subset.

## References

[1] G. Vaz and J. Bosschers, "Modelling three dimensional sheet cavitation on marine propellers using a boundary element method", Proceedings of the 6th Int. Symposium on Cavitation, (2006)

[2] S. Jena, G.D. Weymouth, A. Lidtke and A. Coraddu, "Predicting airfoil pressure distribution using boundary graph neural networks", Proceedings of MARINE, (2025)

[3] Y. Shen, J. T. Needels, and J.J Alonso, "VortexNet: A Graph Neural Network-Based Multi-Fidelity Surrogate Model for Field Predictions", AIAA SCITECH Forum, (2025).

[4] I. Hubbard and T. Melissaris, "Data & physics-driven graph convolutional networks for rapid cavitation energy predictions on marine propeller blades", Proceedings of the 1st Int. Symposium on AI and Fluid Mechanics, (2025)

[5] S. Barwey, P. Pal, S. Patel, R. Balin, B. Lusch, V. Vishwanath, R. Maulik, and R. Balakrishnan, "Mesh-based super-resolution of fluid flows with multiscale graph neural networks", Comp. Methods in Applied Mech. and Eng, (2025).

[6] L. Lu, P. Jin, G. Pang, Z. Zhang and G.E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators", Nature Machine Intelligence, Vol. 3, pp. 218–229, (2021)

[7] B. Alkin, A. Fürst, S. Schmid, L. Gruber, M. Holzleitner and J. Brandstetter, "Universal Physics Transformers: A framework for efficiently scaling neural operators", Advances in Neural Information Processing Systems, (2024)

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need", Advances in Neural Information Processing Systems, Vol. 30, (2017)

[9] T.N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks", arXiv, arXiv:1609.02907, (2017)

[10] J. Huisman, E.J. Foeth, J. Slot, A. Lampe, J. Moulijn and J. Dang, "Design of the Wageningen F-Series Propellers", Seventh Int. Symposium on Marine Propulsors, Wuxi, China. (2022)

# Solver-In-The-Loop Training with Multistep Inputs for Capturing Fine-Scale Turbulence

Amelie Lam; Héloïse Méheut; Sara El Bouch; André Ferrari

*Université Côte d'Azur, Observatoire de la Côte d'Azur, CNRS, Laboratoire Lagrange, Nice, France*

**Keywords:** 2D Turbulence, Differentiable Physics, Solver-In-The-Loop, Deep Learning

**Abstract.** Understanding turbulence is crucial for studying gas-dust dynamics in planet forming disks, yet resolving small-scale structures remains computationally expensive. To address this challenge, we employ a hybrid simulation framework that couples a coarse resolution fluid solver with a convolutional neural network to model unresolved dynamics during time integration. The network operates on solver outputs and is trained in a solver-in-the-loop setting, in which a differentiable fluid solver is integrated directly into optimisation, enabling end-to-end training against downsampled high-resolution direct numerical simulations. To improve long-term predictive performance, the model is trained on extended temporal rollouts. The framework has previously been evaluated on 2D homogeneous isotropic turbulence, where training on extended rollouts significantly improved long-term accuracy. Our main contribution is the introduction of a multi-step input framework in which the neural network receives both the post-solver state and velocity fields from previous timesteps. We show that this modification allows the model to be trained with shorter rollout lengths while surpassing the baseline model in terms of mean squared error. We further demonstrate that this modification, not only enables faster computation time and memory usage, but also preserves the ability to reproduce key statistical properties of the turbulence.

## 1 Introduction

Planet-forming disks are rotating thin disks around young stars, composed primarily of gas and a small fraction of dust particles that eventually coagulate and grow to form planets. Observations indicate that these disks are turbulent [1, 2], and this assumption is commonly incorporated into disk models, where turbulence drives angular momentum transport [3] and/or particle diffusion [4, 5]. Provided that these protoplanetary disks span astronomical length scales, direct resolution of all turbulent scales is computationally prohibitive. As a result, disk models rely on approximations. In particular, one often models disks in 2D, while local dynamics are often studied using a small box compared to the disk size [6]. However, integrating the complex behaviours captured in these local simulations into larger scale models remains challenging. Attempts to bridge this gap include implicit large eddy simulations (ILESs) [7], yet convergence to direct numerical simulations (DNSs) still requires high resolution. These results highlight the need for better models of small scale turbulence.

In parallel, data-driven methods have emerged as a promising approach for modelling fluid systems at reduced computational cost. However, turbulence modelling remains challenging due to its inherently multiscale nature, and existing approaches often struggle with generalisation and long-term stability. To address this, a growing body of work leverages differentiable physics frameworks [8] to incorporate physical structure into learning-based models [9–12], leading to improved accuracy and stability. Nevertheless, these approaches remain computationally expensive, particularly for 2D and 3D settings, due to the cost of gradient backpropagation through long rollout trajectories.

In this work, we build on prior approaches based on differentiable physics to develop a hybrid framework that combines machine learning with established numerical methods. Our goal is to construct efficient, data-driven turbulence models for planet-forming disks. While previous studies often used only the solver output as input of the neural network, we additionally provide velocity fields from previous timesteps. This provides more comprehensive temporal and dynamical information, enabling the use of shorter

rollout lengths during training while maintaining performance comparable to baseline models, and thus improving training efficiency.

Within this paper, section 2 presents the methodology used to construct our framework, section 3 discusses on the result obtained, and in section 4 we summarise our finding.

## 2 Methodology

### *2.1 Background*

Focusing on the small scales of a planet-forming disk, we assume the flow to be incompressible [13]. The flow is governed by the Navier-Stokes equations as follows,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u} + \mathbf{f}; \tag{2.1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2.2}$$

where $\mathbf{u} = [u, v]^T$ is the 2D velocity field of the fluid, and $\rho$, $p$ and $Re$ denote the fluid's uniform density, pressure, and Reynolds number, respectively. The additional forcing term $\mathbf{f}$ that sustains the fluid at a statistically steady state includes a term that inject energy at wavenumber $k_f = 4$ and a linear friction term with dissipation coefficient $\alpha_{\text{diss}} = 0.1$ that prevents energy accumulation at large scales.

We consider homogeneous, isotropic turbulence (HIT) in periodic boundary conditions, which serves as a simplified assumption for turbulence in planet-forming disk. Due to the direct cascade, the number of degrees of freedom required to fully resolve 2D turbulence scales linearly with the Reynolds number. In the context of planet-forming disk, where $Re \sim \mathcal{O}(10^{14})$, fully resolving all turbulent scales is computationally infeasible. Consequently, DNSs of planet-forming disk use an effective viscosity compatible with the grid resolution. In our experiment, we set $Re = 1000$, for which turbulence is fully resolved on a $2048 \times 2048$ grid.

### *2.2 Hybrid CNN-Solver Framework*

We define $\mathcal{P}$ as one time-step of the coarse differentiable solver of equations (2.1) and (2.2). A CNN $f_{\boldsymbol{\theta}}$, with trainable parameters $\boldsymbol{\theta}$, is used to predict a correction term accounting for unresolved subgrid effects. The corrected hybrid update of the velocity field at time $t+1$, $\mathbf{u}_{t+1}$, is computed from $\mathbf{u}_t$ and velocity fields from previous timesteps $\mathbf{u}_{t-1}, \cdots, \mathbf{u}_{t-p}$ as:

$$\mathbf{u}_{t+1} = \mathcal{P}(\mathbf{u}_t) + f_{\boldsymbol{\theta}}(\mathcal{P}(\mathbf{u}_t), \mathbf{u}_t, \mathbf{u}_{t-1}, \cdots, \mathbf{u}_{t-p}), \tag{2.3}$$

Unlike the baseline model [9] which uses only the post-solver state as input to the CNN, $\mathbf{u}_{t+1} = \mathcal{P}(\mathbf{u}_t) + f_\theta(\mathcal{P}(\mathbf{u}_t))$, our formulation (2.3) also includes $\mathbf{u}_t, \mathbf{u}_{t-1}, \cdots, \mathbf{u}_{t-p}$ in the network input. This augmented input is intended to provide temporal context through past states while retaining information about the solver dynamics through the post-solver state.

Key design choices of the framework are as follows:

**Numerical Solver.** We use the `jax-cfd` code, a finite-volume solver on a regular staggered grid, as the backbone $\mathcal{P}$ of our framework. A grid-based solver is chosen for consistency with most astrophysical fluid codes. The framework operates on a $64 \times 64$ grid, corresponding to a downsampling factor of 32 relative to the DNS reference simulation. Further details on the numerical schemes, see Appendix 1 of [9].

**Architecture of CNN Corrector.** The correction operator $f_{\boldsymbol{\theta}}$ is parameterized by a lightweight CNN with periodic padding, consistent with the solver step. In the baseline architecture [9], $f_{\boldsymbol{\theta}} : \mathbb{R}^{64\times64\times2} \to \mathbb{R}^{64\times64\times2}$ is a six-layer CNN acting on a two-channel velocity field. Each hidden layer consists of a convolution of channel size 64, batch normalization and a ReLU activation unit.

For the multi-step input variant, we stack the velocity fields from successive time-steps along the channel dimension, so that the input becomes $[\mathcal{P}(\mathbf{u}_t), \mathbf{u}_t, \mathbf{u}_{t-1}, \cdots, \mathbf{u}_{t-p}] \in \mathbb{R}^{64\times64\times2(p+2)}$. In some experiments, we keep the baseline architecture unchanged in order to isolate the effect of enlarging the temporal input alone and enable a direct comparison with [9]. In other experiments, we additionally increase the number of kernels in the first convolutional layer, using 128 kernels instead of 64, in order to increase the number of learnable parameters and provide the model with more capacity to exploit the richer multi-step input.

### *2.3 Solver-in-the-loop Training*

Building on the *solver-in-the-loop* training strategy proposed by [14], we integrate the differentiable fluid solver directly into the training process. This allows the neural network to learn over an extended temporal horizon, strongly improving its stability and long term predictive performance.

Let $\{\mathbf{u}^{\text{ref}}_{t,i}\}^{i=1,\cdots,M}_{t=1,\cdots N-T+1}$ denote a collection of $M$ independent high-resolution trajectories sampled at discrete times of temporal length $N$. We introduce a downsampling operator $\mathcal{D}$ mapping the DNS state to the coarse state space. The training loss is defined as an *a posteriori* Mean Square Error (MSE) between model predictions and the downsampled DNS states,

$$\mathcal{L}(\theta) = \sum_{i=1}^{M} \sum_{n=1}^{N-T+1} \sum_{t=n}^{n+T-1} \|\mathbf{u}^{\text{pred}}_{t,i} - \mathcal{D}(\mathbf{u}^{\text{ref}}_{t,i})\|^2_F, \tag{2.4}$$

where $T$ is the number of unrolled time-steps seen by the neural network per optimisation step and $\{\mathbf{u}^{\text{pred}}_{t,i}\}_{t=n\ldots n+T-1}$ are recursively computed from the initial state $\mathcal{D}(\mathbf{u}^{\text{pred}}_{t,i})$ using (2.3). The tight coupling between the fluid solver and the neural network necessitates the use of a differentiable solver.

We used a set of $M = 28$ simulations for training, and 4 simulations for validation. Each simulation is initialised from a random white-noise velocity field, evolved to $t_{\text{sim}} = 40$ to reach a statistically steady state, and then run for an additional $N = 500$ timesteps. Training samples are obtained using a sliding window of length $T + 1$ along the temporal dimension, where the first timestep provides the initial conditions and the remaining timesteps serve as targets. This adds up to around 13,000 samples. Model parameters are optimised using a standard Adam optimizer with a fixed learning rate of $10^{-4}$ and a batch size of $B = 64$. All models are trained for 400 epochs. We note that $T$ has to be increased progressively to maintain training stability.

## 3 Results and Discussion

To evaluate the framework, we performed rollouts over 3000 timesteps using 10 initial velocity fields not seen during training. We compare several models: a baseline CNN using only the post-solver state as input, trained with rollout lengths of $T = 32$ (denoted T32), same CNN trained over $T = 16$ (denoted T16) and our proposed framework, (2.3), with $p \in [0, 4]$, trained with $T = 16$ (denoted T16_p0-4). As previously mentioned, for this set of experiments we keep the baseline architecture unchanged to isolate the effect of incorporating more temporal information. Additionally, for $p = 3$ and $p = 4$, we compare the results with a larger model by increasing the number of kernels in the initial layer to 128, this configuration is denoted by an additional subscript $L$ (T16_p0-4_L). We also report results from

downsampled high resolution DNSs as reference, and corresponding uncorrected low resolution DNSs initialised with the same velocity fields.

### *3.1 Mean Squared Error*

As our training objective is to match the framework output to fully resolved DNS data, we compare the normalized MSEs. In Figure 1, we overlay the MSEs evaluated on training and validation sets at the last epoch 400 for the different models. We also compare MSEs across different time horizons between the models and the reference. The results are reported in Table 1.

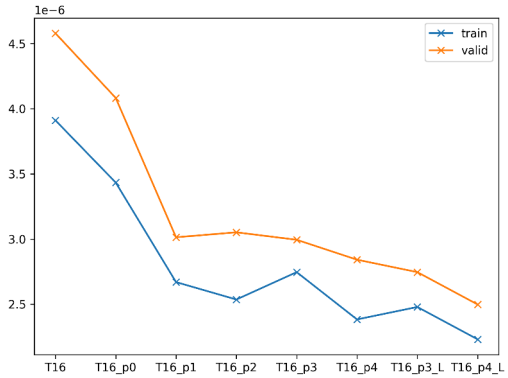



Figure 1: One-step normalized MSE loss at epoch 400 of the different models on training and validations sets.

| Model | 100 | 500 | 1000 |
|---|---|---|---|
| T16 | $1.85 \times 10^{-4}$ | 0.0117 | 0.374 |
| T32 | $2.14 \times 10^{-4}$ | 0.0159 | 0.48 |
| T16_p0 | $2.95 \times 10^{-4}$ | 0.0307 | 0.688 |
| T16_p1 | $1.40 \times 10^{-4}$ | 0.0073 | 0.216 |
| T16_p2 | $2.25 \times 10^{-4}$ | 0.0256 | 0.532 |
| T16_p3 | $1.44 \times 10^{-4}$ | 0.0078 | 0.254 |
| T16_p4 | $1.89 \times 10^{-4}$ | 0.0207 | 0.44 |
| T16_p3_L | $1.28 \times 10^{-4}$ | 0.0070 | 0.265 |
| T16_p4_L | $1.22 \times 10^{-4}$ | 0.0073 | 0.252 |
| No Corr | $9.39 \times 10^{-3}$ | 0.29 | 1.02 |

Table 1: MSE at selected timesteps, averaged over the test set.

Figure 1 also shows the one-step MSE loss at the end of training for each model trained with $T = 16$. The results indicate that increasing the number of velocity states from previous timesteps leads to a lower training loss after a fixed number of epochs. All hybrid frameworks outperform the uncorrected low resolution DNS by order of magnitude. We find that, compared to T32, the T16 model already provides a strong representation of the turbulent flow. The slight degradation observed for T32 may be due to the model primarily learning mean-flow dynamics rather than dynamics at more local scales. For our proposed framework with the baseline architecture, 2.3, we find that T16_p1 yields the best MSE performance and improves upon the baseline T16 model, suggesting that incorporating temporal information enhances the model's ability to capture temporal dynamics more effectively. We further find that, with larger number of previous states $p$ in the input, the larger model outperforms the baseline architecture, indicating that additional learnable parameters improve the representation of temporal patterns.

### *3.2 Turbulence statistics*

Maintaining pointwise correlation in a chaotic system such as turbulence is inherently unrealistic beyond short timescales. It is, therefore, more meaningful to evaluate statistical properties of the flow. We compute the energy spectrum, structure functions, and vorticity probability density functions (PDFs) over the regime where the framework has decorrelated from the reference DNS. Figure 2 shows the corresponding logarithmic errors in the energy spectrum and the fourth-order velocity structure function relative to the reference statistics, while Figure 3 presents the vorticity PDFs at different scales.

We find that power law behaviour of the energy spectrum and velocity structure function is already well captured with relatively simple models, and the T16 models are already sufficient in reproducing statistics comparable to reference DNSs. Furthermore, for applications where the structure functions are key,

including previous timesteps as additional inputs shows clear improvements. However, when more than 2 previous states are included, the baseline architecture lacks the complexity to fully represent the additional temporal information, leading to degraded performance. A larger CNN is hence needed for further improvement. For applications to planet formation, where the vorticity PDF is decisive, our framework shows a clear improvement when 2 previous timesteps are considered (T16_p2).

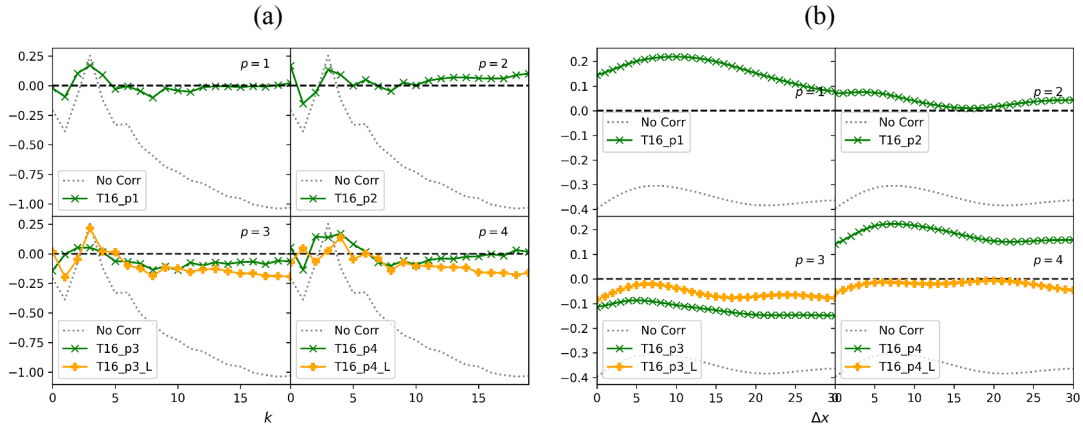



Figure 2: (a) Log difference in energy spectrum over wavenumber $k$ and (b) log difference in fourth-order velocity structure function over cell separation $\Delta x$ relative to the reference statistics averaged between 2500 and 3000 timesteps across 10 independent simulations.

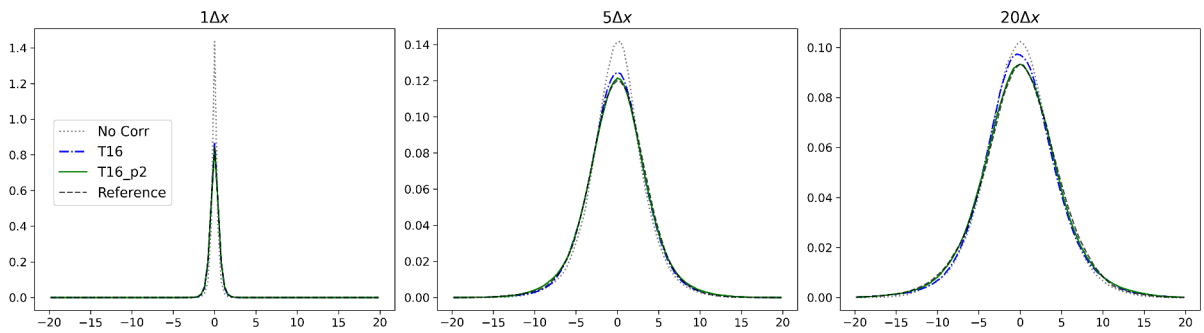



Figure 3: Vorticity PDFs at grid separation $1\Delta x$, $5\Delta x$ and $20\Delta x$ respecitvely, averaging between 2500 and 3000 simulation timesteps, over 10 independent simulations.

### *3.3 Computation Time*

Training with rollout lengths of $T = 32$ requires 12 hours over 400 epochs, whereas reducing the rollout length to $T = 16$ lowers the training time to 7 hours, while maintaining performance comparable to the T32 model. Moreover, by reducing the resolution and including the CNN, we achieve a speedup of $40\times$ compared to fully resolved DNS, despite a $4\times$ increase in computing time compared to the uncorrected low resolution simulation.

## 4 Conclusion

In conclusion, we build on prior work leveraging differentiable physics to develop a hybrid framework that combines a grid-based numerical solver with a convolutional neural network to model turbulence accurately at reduced computation cost. By providing the network with velocity fields from previous

timesteps, we incorporate additional temporal information that improves training efficiency, while reproducing statistical properties of turbulence comparable to reference data. We further show that the rollout length during training can be reduced by a factor of two without significant degradation in performance, substantially lowering the overall training cost. However, our results also suggest that larger network architectures may be required to fully exploit the increased complexity introduced by additional temporal inputs. These findings highlight the potential of hybrid solver-CNN frameworks for developing turbulence models in astrophysical fluid dynamics. Future work will investigate the dynamics of particles evolving within the predicted turbulent flow.

**Acknowledgement**

This work was supported by the French government through the France 2030 investment plan managed by the National Research Agency (ANR), as part of the Initiative of Excellence Université Côte d'Azur under reference number ANR-15-IDEX-01, and as part of the PEPR Origins under reference number ANR-22-EXOR-0001. This project was provided with computing HPC/AI and storage resources by GENCI at IDRIS thanks to the grant 2024-AD010415910 on the supercomputer Jean Zay's V100/A100 partition.

**References**

[1] Kevin M. Flaherty et al. "Weak Turbulence in the HD 163296 Protoplanetary Disk Revealed by ALMA CO Observations". In: *ApJ* 813.2, 99 (Nov. 2015), p. 99.

[2] Jinshi Sai et al. "Nonthermal Velocity Dispersion in the Outer Disk of HL Tau". In: *arXiv e-prints*, arXiv:2603.29147 (Mar. 2026).

[3] Steven A. Balbus and John F. Hawley. "Instability, turbulence, and enhanced transport in accretion disks". In: *Reviews of Modern Physics* 70.1 (Jan. 1998), pp. 1–53.

[4] Sebastien Fromang and John Papaloizou. "Dust settling in local simulations of turbulent protoplanetary disks". In: *A&A* 452.3 (June 2006), pp. 751–762.

[5] Andrew Youdin and Anders Johansen. "Protoplanetary Disk Turbulence Driven by the Streaming Instability: Linear Evolution and Numerical Methods". In: *ApJ* 662.1 (June 2007), pp. 613–626.

[6] John F. Hawley, Charles F. Gammie, and Steven A. Balbus. "Local Three-dimensional Magnetohydrodynamic Simulations of Accretion Disks". In: *ApJ* 440 (Feb. 1995), p. 742.

[7] Heloise Meheut et al. "Angular momentum transport and large eddy simulations in magnetorotational turbulence: the small Pm limit". In: *A&A* 579, A117 (July 2015), A117.

[8] Nils Thuerey et al. *Physics-based Deep Learning*. 2021. URL: `https://physicsbaseddeeplearning.org`.

[9] Dmitrii Kochkov et al. "Machine learning-accelerated computational fluid dynamics". In: *Proceedings of the National Academy of Science* 118.21 (May 2021).

[10] Björn List, Li-Wei Chen, and Nils Thuerey. "Learned turbulence modelling with differentiable fluid solvers: physics-based loss functions and optimisation horizons". In: *Journal of Fluid Mechanics* 949, A25 (Oct. 2022), A25.

[11] Hugo Frezat et al. "A Posteriori Learning for Quasi-Geostrophic Turbulence Parametrization". In: *Journal of Advances in Modeling Earth Systems* 14.11 (Nov. 2022).

[12] André Freitas et al. "Solver-in-the-loop approach to closure of shell models of turbulence". In: *Physical Review Fluids* 10.4 (Apr. 2025).

[13] Oded Regev and Orkan M. Umurhan. "On the viability of the shearing box approximation for numerical studies of MHD turbulence in accretion disks". In: *A&A* 481.1 (Apr. 2008), pp. 21–32.

[14] Kiwon Um et al. "Solver-in-the-Loop: Learning from Differentiable Physics to Interact with Iterative PDE-Solvers". In: *Advances in Neural Information Processing Systems* (2020).

# Stochastic reconstruction of wall shear stress fluctuations via Physics-Informed Diffusion Models

Roberto Dal Monte[1], Michele Cogo[2], and Francesco Picano[1,2]

[1]Centro di Ateneo di Studi e Attività Spaziali 'Giuseppe Colombo', University of Padova, Via Venezia 1, 35131 Padova, Italy
[2]Department of Industrial Engineering, University of Padova, Via Venezia 1, 35131 Padova, Italy

**Abstract.** Near-wall turbulence in boundary layers is governed by extremely small-scale eddies that are primarily responsible for shear stresses exerted on the wall. Directly resolving these structures at high Reynolds numbers is computationally intractable, making near-wall modeling essential for practical simulations. Among the various modeling approaches, the Equilibrium Wall Model (EWM) is widely used due to its simplicity, but it relies on restrictive assumptions such as local equilibrium and statistical stationarity, providing only mean flow quantities and failing to reproduce the correct spatial organization and statistical distribution of the wall stress, leading to poor prediction of extreme events. In this work, we leverage Denoising Diffusion Probabilistic Model (DDPM) trained on Direct Numerical Simulation (DNS) data to augment EWM predictions by reconstructing the unresolved near-wall dynamics. A physics-guided strategy is employed to enforce consistency with DNS cross-correlation coefficients, ensuring physically consistent predictions. Unlike conventional deterministic approaches, the proposed enhanced wall model captures fluctuations, intermittency and rare extreme events, providing a more realistic and comprehensive description of near-wall turbulence dynamics.

**Keywords:** wall turbulence, Diffusion Model, turbulent boundary layer

## 1. Introduction

Accurate simulation of wall-bounded turbulent flows is a cornerstone of aerospace engineering, where reliable prediction of wall-shear stress and heat flux is essential for assessing drag, thermal loads, and structural integrity. At high Reynolds numbers, however, resolving near-wall turbulence through Direct Numerical Simulation (DNS) becomes computationally intractable due to the extreme spatial and temporal resolution required [1]. Large Eddy Simulation (LES) alleviates this burden by modeling the smallest turbulent scales, but the near-wall region still remains prohibitively expensive to resolve. As a result, Wall-Modeled LES (WMLES) has emerged as a practical approach for simulations at engineering-relevant conditions [2].

Among wall-modeling strategies, Equilibrium Wall Models (EWM) are widely adopted because of their simplicity and computational efficiency. These models estimate wall-shear stress from outer-flow velocity using simplified assumptions such as local equilibrium and statistical stationarity. While effective for predicting mean quantities, they fail to capture the complex,

multi-scale, and stochastic nature of near-wall turbulence when applied to instantaneous fields. Consequently, EWM predictions exhibit unrealistic spatial organization, damped fluctuations, and poor representation of intermittent extreme events, which are often critical in engineering applications [3].

In recent years, the growing availability of high-fidelity datasets has motivated the application of machine learning (ML) techniques to turbulence modeling. Supervised ML approaches have been successfully employed for wall modeling, often improving predictive accuracy over classical methods. However, these approaches are typically formulated as deterministic input–output mappings, which limits their ability to represent the intrinsic variability, intermittency, and multi-scale dynamics of turbulence. Furthermore, they generally lack mechanisms to enforce physical consistency, thereby reducing their robustness and generalizability [4].

On the contrary, generative models offer a fundamentally different paradigm by learning full probability distributions rather than deterministic mappings. In particular, Denoising Diffusion Probabilistic Models (DDPMs) [5] have recently demonstrated remarkable capabilities in modeling complex, high-dimensional data and have shown promising results in various fluid-dynamics applications [6]. In this work, we propose a stochastic, physics-guided wall-modeling framework based on DDPMs trained exclusively on high-fidelity DNS data. The model is employed to augment EWM predictions by reconstructing unresolved near-wall dynamics, including small-scale fluctuations and intermittent extreme events. Physical consistency is enforced during generation through guidance based on DNS-derived cross-correlation statistics between wall-shear stress and outer-flow velocity fields. Further details of the proposed methodology are provided in the following section.

## 2. Methodology

First, a high-fidelity dataset composed of 100 uncorrelated two-dimensional snapshots of wall-shear stress was generated from Direct Numerical Simulation (DNS) of a turbulent channel flow at friction Reynolds number $Re_\tau = 1000$ using the fluid solver CaNS [7]. The corresponding low-fidelity dataset was then obtained using the Equilibrium Wall Model (EWM), where wall-shear stress is reconstructed from streamwise velocity fields sampled at wall-normal locations in the range $y^+ = [10, 200]$. Here, $y^+ = y/\delta_v$ denotes the wall-normal distance in wall-units, with $\delta_v = \nu/u_\tau$ the viscous length scale, $u_\tau$ the friction velocity and $\nu$ the kinematic viscosity. Although computationally efficient, EWM is intrinsically designed to predict mean quantities only. When applied to instantaneous fields, its deterministic formulation, $\tau_w = f(u)$, enforces a direct mapping between the sampled outer-flow velocity and the wall response. As a consequence, the predicted wall-shear stress inherits the spatial organization of the outer flow rather than reconstructing the true near-wall dynamics, as shown in Figure 1a. This results in strongly damped fluctuations and thus the inability to capture rare but critical extreme events (Figure 1b). Moreover, EWM imposes unrealistically strong correlations between wall-shear stress and velocity fluctuations, in contrast with DNS observations where such correlations progressively weaken with increasing wall-normal distance, reflecting the stochastic and multi-scale nature of turbulence (Figure 1c).

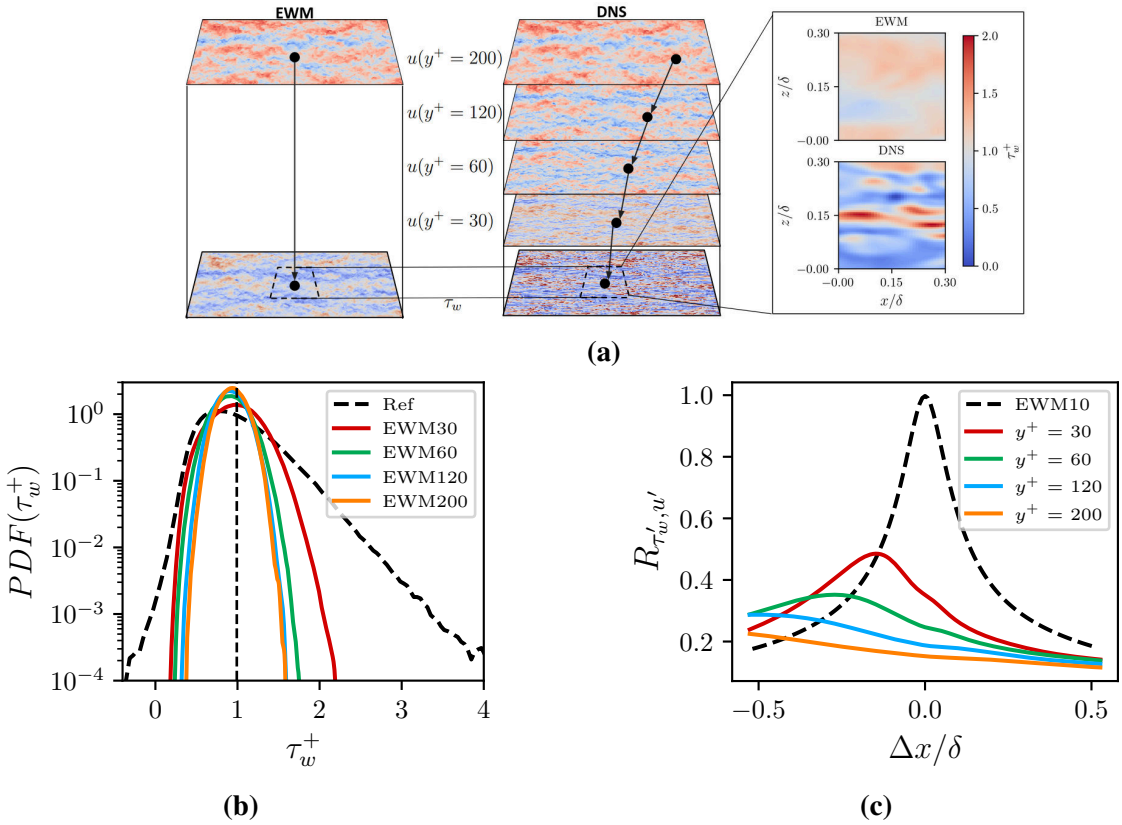



**Figure 1.** $(a)$: qualitative comparison between wall shear stress prediction provided by EWM and DNS. Highlighted the points which have the max correlations with the local wall-shear stress represented. $(b)$ Probability Density Functions (PDFs) of wall shear stress by DNS (black dashed line) and EWM predictions computed from $y^+ = 30$ (red), 60 (green), 120 (blue), 200 (orange). $(c)$ Cross-correlation coefficient between $\tau_w'$ and $u'$ sampled at different matching location as in $(b)$. Black-dashed line represent the perfect cross-correlation of EWM. $\tau_w^+ = \tau_w/\langle\tau_w\rangle$

A natural strategy to augment EWM predictions would consist in training a Denoising Diffusion Probabilistic Model (DDPM) to directly map low-fidelity EWM inputs to statistically equivalent high-fidelity DNS outputs. However, such an approach is highly sensitive to distribution shifts. Indeed, EWM predictions computed from different wall-normal locations exhibit significantly different statistical distributions (Fig. 1b); consequently, a model trained on low-fidelity inputs obtained at a specific sampling location may generalize poorly when applied to EWM predictions generated from substantially different wall-normal positions. To overcome this limitation, we follow the strategy proposed by Shu et al. [6] and train the DDPM exclusively on high-fidelity DNS data. In this configuration, the model learns the probability distribution of wall-shear stress fields and generates statistically consistent samples through a Markovian diffusion process starting from Gaussian noise $x_T \sim \mathcal{N}(0, \boldsymbol{I})$.
The reconstruction of high-fidelity fields from low-fidelity EWM inputs is then achieved through a guided generation procedure. First, a controlled amount of Gaussian noise is added to the EWM samples using the same forward diffusion process adopted during DDPM training. This preprocessing step shifts the different EWM distributions toward the distribution of an intermediate diffusion state $x_t$ with $0 < t < T$, reducing the mismatch between inputs associated

with different wall-normal sampling locations. The resulting partially noised EWM sample is then used as the initial condition for a truncated reverse diffusion process, which progressively reconstructs the denoised field $x_0$. In this way, residual information from the original EWM prediction is retained and used to guide the generation. Although EWM fails to reproduce small-scale fluctuations and intermittency, it still provides a reasonable estimate of the mean wall-shear stress, which acts as a physically meaningful constraint during reconstruction.
To further enforce physical consistency, a Bayesian posterior guidance strategy is introduced during the reverse diffusion process [8]. Specifically, depending on the wall-normal sampling location associated with the EWM input, the generation is constrained to reproduce the spatial cross-correlation coefficients between wall-shear stress fluctuations, $\tau_w'$, and velocity fluctuations, $u'$, extracted from DNS data (Equation 1).

$$R_{\tau',u'}(\Delta x) = \frac{\langle \tau_w'(x,z), u'(x-\Delta x, z)\rangle}{\sigma_\tau \sigma_u} \tag{1}$$

This guidance ensures that the generated samples maintain physically realistic correlations with the outer-flow velocity field, overcoming the artificial perfect correlation imposed by deterministic wall models. As a result, the proposed framework reconstructs wall-shear stress fields with statistically consistent distributions, including intermittent and localized extreme events, while preserving physical consistency with the underlying turbulent dynamics.
It is worth mentioning that, to address memory constraints and enable scalable deployment in future *a posteriori* CFD simulations, each snapshot was decomposed into smaller patches prior to training and inference. The patch size was selected as a trade-off between retaining sufficient spatial information within each sample and limiting the computational cost associated with training and generation. To preserve model generalizability, the patch dimensions were defined in wall units as $l_x^+ \times l_z^+ \approx 523 \times 295$ . However, as shown in Figure 1c, the streamwise shift $\delta_x$ associated with the maximum cross-correlation increases with the wall-normal sampling location due to the inclination of coherent turbulent structures. Beyond a certain wall-normal distance, this shift becomes larger than the streamwise patch length $l_x^+$, thereby reducing the effectiveness of the proposed physics-guided constraint. To overcome this limitation, the velocity fields were shifted in the streamwise direction prior to computing both the EWM predictions and the reference cross-correlations, such that the maximum correlation is aligned at $\delta_x = 0$. This preprocessing step ensures that the footprint of the relevant coherent structures remains fully contained within each patch, thereby improving the effectiveness of the physics-guided generation procedure.

## 3. Results

The proposed Physics-Guided DDPM (PG-DDPM) framework was assessed through *a priori* analyses by comparing the reconstructed wall-shear stress fields against the DNS reference and the corresponding EWM predictions. Particular attention was devoted to the capability of the model to recover the statistical behavior of near-wall turbulence, including intermittency and rare extreme events. Figure 2a reports the probability density functions (PDFs) of the wall-shear stress fluctuations obtained from DNS, EWM, and PG-DDPM reconstructions for different matching locations. For consistency, the statistics were computed using the same number of patches, corresponding to 100 instantaneous snapshots. As shown PG-DDPM reconstructions accurately recover the DNS statistics over the entire distribution range, including the tails associated with rare and intermittent events. In contrast, the EWM predictions exhibit strongly

damped fluctuations and significantly underestimate extreme events. These results demonstrate the capability of the proposed generative framework to reconstruct physically realistic fluctuations that are entirely absent in deterministic EWM predictions.

The spatial consistency of the reconstructed fields was further evaluated through the cross-correlation coefficients between wall-shear stress fluctuations and outer-flow velocity fluctuations. Figure 2b shows that the PG-DDPM predictions closely reproduce the DNS reference correlations, correctly recovering the progressive decorrelation observed as the sampling location moves away from the wall. This behavior overcomes the deterministic EWM limitation, which imposes unrealistically strong correlations between wall-shear stress and sampled velocity fields, leading to perfect correlation independently of the matching location. This result highlights the effectiveness of the proposed physics-guided sampling strategy in enforcing physically meaningful relations between outer-flow information and reconstructed near-wall dynamics. Qualitative comparisons between instantaneous DNS, EWM, and PG-DDPM wall-shear stress patches are reported in Figure 2c. As previously discussed, EWM fields inherit the large-scale organization of the sampled velocity field while lacking the small-scale intermittent structures characteristic of near-wall turbulence. In contrast, PG-DDPM reconstructions recover realistic multi-scale dynamics, generating coherent small-scale fluctuations while preserving the large-scale information provided by the EWM input. Overall, the reconstructed fields show significantly improved agreement with DNS in terms of both spatial organization and fluctuation intensity.

Overall, the results demonstrate that the proposed framework successfully augments deterministic wall-model predictions by restoring the stochastic and intermittent dynamics of wall turbulence, while maintaining physical consistency through the cross-correlation guidance.

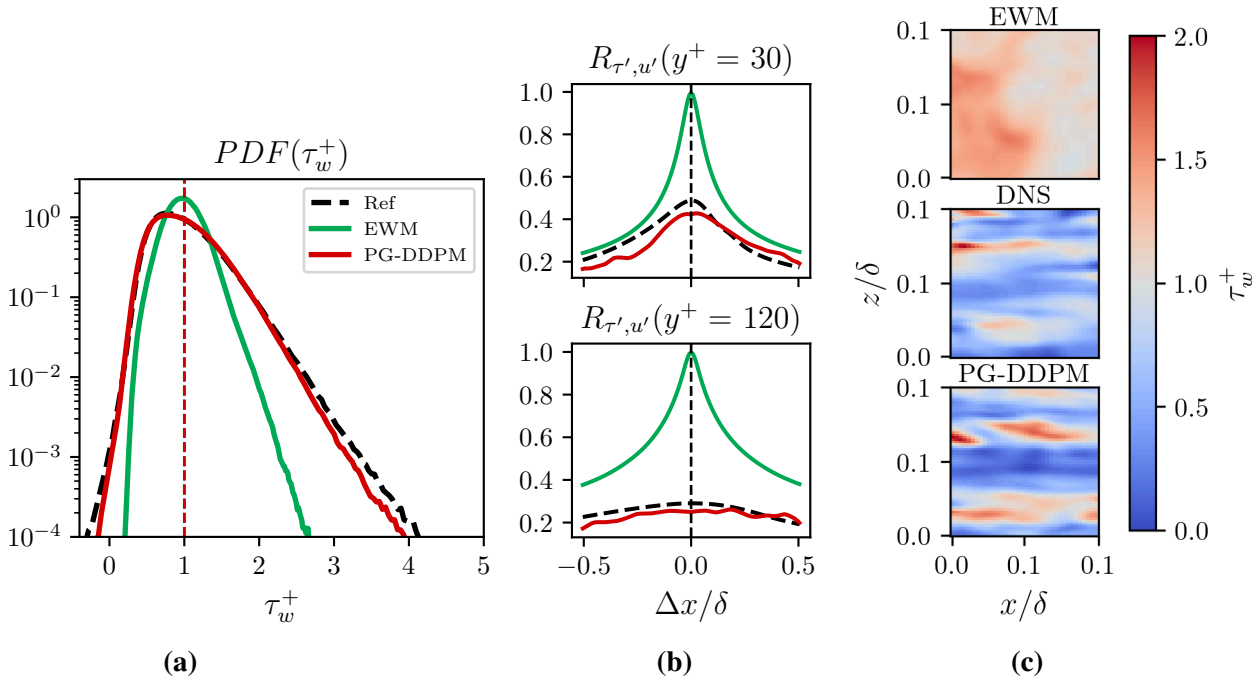



**Figure 2.** $(a)$ Instantaneous wall-shear stress distribution provided by DNS, EWM, PG-DDPMs. $(b)$ Cross-correlation coefficients between $\tau_w'$ and $u'$ for matching locations $y^+ = 30$ and $y^+ = 120$. $(c)$ Qualitative comparison among samples. EWM sample is computed from $y^+ = 120$

## 4. Conclusions

In this work, a Physics-Guided generative framework based on Denoising Diffusion Probabilistic Models (DDPMs) was proposed for the reconstruction of instantaneous wall-shear stress fields in turbulent channel flow. The model was trained exclusively on high-fidelity DNS data and employed to augment low-fidelity Equilibrium Wall Model (EWM) predictions through a physics-guided reverse diffusion process. The proposed approach addresses one of the main limitations of classical deterministic wall models, namely the inability to reproduce the stochastic and multi-scale nature of near-wall turbulence. By learning the underlying probability distribution of DNS wall-shear stress fields, the DDPM is able to reconstruct realistic fluctuations, intermittency, and extreme events that are entirely missed by EWM predictions. In addition, the introduction of a Bayesian physics-guided sampling strategy based on DNS-derived cross-correlations ensures physical consistency between the reconstructed wall dynamics and the outer-flow velocity field.

*A priori* analyses demonstrated excellent agreement between the generated and DNS statistics, both in terms of probability distributions and spatial cross-correlations, while qualitative comparisons confirmed the recovery of realistic near-wall coherent structures. Furthermore, the framework showed promising generalization capabilities when applied to conditions different from those used during training (not shown in this paper).

These results highlight the potential of diffusion-based generative models as a new class of stochastic wall-modeling tools for WMLES applications, capable of combining physical consistency, statistical accuracy, and computational efficiency.

## References


[1] Parviz Moin and Krishnan Mahesh. "Direct numerical simulation: a tool in turbulence research". In: *Annual review of fluid mechanics* 30.1 (1998), pp. 539–578.

[2] Harish Gopalan, Stefan Heinz, and Michael K Stöllinger. "A unified RANS–LES model: Computational development, accuracy and cost". In: *Journal of Computational Physics* 249 (2013), pp. 249–274.

[3] Sanjeeb T Bose and George Ilhwan Park. "Wall-modeled large-eddy simulation for complex turbulent flows". In: *Annual review of fluid mechanics* 50 (2018), pp. 535–561.

[4] Aurélien Vadrot, Xiang IA Yang, and Mahdi Abkar. "Survey of machine-learning wall models for large-eddy simulation". In: *Physical Review Fluids* 8.6 (2023), p. 064603.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[6] Dule Shu, Zijie Li, and Amir Barati Farimani. "A physics-informed diffusion model for high-fidelity flow field reconstruction". In: *Journal of Computational Physics* 478 (2023), p. 111972.

[7] Pedro Costa et al. "GPU acceleration of CaNS for massively-parallel direct numerical simulations of canonical fluid flows". In: *Computers & Mathematics with Applications* 81 (2021), pp. 502–511.

[8] Hyungjin Chung et al. "Diffusion posterior sampling for general noisy inverse problems". In: *arXiv preprint arXiv:2209.14687* (2022).

# Physics-Informed Latent Neural Operator for Three-Dimensional Compressor Cascade Flow Prediction

[1]Yuling Han; [1]Zhihui Li*; [1]Zhibin Yu;

[1]*School of Engineering, University of Liverpool, UK;*

**Abstract**

A physics-informed latent neural operator framework is proposed for three-dimensional compressor cascade flow prediction. The proposed method combines a Transolver-based latent encoder, which extracts compact global latent representations of flow conditions from point-cloud CFD data, with a coordinate-based PINNs decoder to reconstruct continuous flow fields while incorporating physics-informed residual constraints during training. The results show that the proposed framework can accurately reconstruct pressure and velocity distributions of complex three-dimensional cascade flows and maintains good prediction capability under previously unseen operating condition. The present method demonstrates promising potential for efficient CFD surrogate modelling and aerodynamic prediction in turbomachinery applications.

**Keywords**: Physics-informed neural operator; Transolver; Compressor cascade; Turbomachinery aerodynamics

## 1 Introduction

High-fidelity computational fluid dynamics (CFD) simulations are widely used in turbomachinery aerodynamic analysis and compressor cascade design. However, accurate three-dimensional CFD simulations usually require large-scale numerical discretisation and iterative solvers, resulting in substantial computational cost and long simulation times. These limitations become increasingly significant in applications involving aerodynamic optimisation and multi-condition flow evaluations.

To reduce the computational expense of conventional CFD simulations, various deep learning-based surrogate modelling approaches have been developed in recent years. Early studies mainly employed convolutional neural networks (CNNs) for flow-field prediction on structured grids [1–3]. To improve geometric flexibility, graph neural networks (GNNs) and neural operator methods were subsequently introduced for PDE learning on irregular domains [4–6]. In particular, neural operator frameworks provide a promising direction for learning mappings between physical fields under different operating conditions.

More recently, transformer-based architectures have attracted increasing attention in scientific machine learning and CFD applications due to their capability for capturing long-range spatial correlations. Among these approaches, Transolver [7] introduced a transformer-based PDE learning framework for general physical domains using slice attention mechanisms. However, existing Transolver-based approaches remain primarily data-driven and mainly perform discrete point-wise flow-field regression without explicitly constructing continuous differentiable field representations. As a result, incorporating physics-informed PDE residual constraints into transformer-based CFD surrogate models remains challenging, and the extrapolation capability of purely data-driven transformer models is still relatively limited for complex three-dimensional turbomachinery flows.

Physics-informed neural networks (PINNs) [8] provide an effective approach for incorporating governing equations into deep learning frameworks through PDE residual constraints. However, conventional PINNs frameworks often suffer from optimization difficulty and slow convergence when applied to large-scale three-dimensional CFD problems. In addition, directly solving complex turbulent flow fields using purely physics-driven PINNs models remains computationally challenging for practical turbomachinery applications.

To address these challenges, the present study proposes a physics-informed latent neural operator framework for three-dimensional compressor cascade flow prediction on CFD point clouds. The proposed framework combines a Transolver-based latent encoder with a coordinate-based PINNs decoder to reconstruct continuous flow fields from latent aerodynamic representations. Physics-informed residual constraints based on the continuity and momentum equations are further incorporated during training to improve physical consistency and extrapolation capability. In addition, prediction task under unseen attack-angle condition is investigated to evaluate the generalization performance of the proposed framework.

## 2 Numerical Method

### 2.1 CFD Simulation and Validation

The CFD dataset used in the present study was generated from three-dimensional compressor cascade simulations. The computational domain consists of a three-dimensional compressor cascade geometry, and all flow-field data were

*Corresponding Author, Zhihui Li zhihui.li@liverpool.ac.uk

obtained using ANSYS Fluent. The simulations were performed on structured meshes in order to accurately capture the complex geometric boundaries and three-dimensional flow structures within the cascade passage [9].

The computational mesh contains approximately 476,736 cells and 501,168 mesh points. The regular mesh distribution enables refined spatial resolution near blade surfaces and regions with strong flow gradients, while maintaining computational efficiency in relatively smooth flow regions. The resulting CFD solutions were subsequently converted into regular point-cloud representations for deep learning-based flow-field prediction.

To verify the reliability of the CFD simulations, the predicted static pressure coefficient distribution on the blade surface was compared with available experimental data. The static pressure coefficient is defined as:

$$C_p = \frac{p - p_{ref}}{\frac{1}{2}\rho U_{ref}^2} \tag{1}$$

where $p$ denotes the local static pressure, $p_{ref}$ represents the reference pressure, $\rho$ is the fluid density, and $U_{ref}$ is the reference inlet velocity.

Figures 1(a) and 1(b) present the comparisons of the static pressure coefficient distributions at the 5.4% and 50% spanwise sections, respectively. Experimental measurements are represented by open-circle markers, while CFD results are plotted using solid lines. Results corresponding to the same attack angle are shown using the same colour.

Overall, the CFD predictions show good agreement with the experimental data on both the suction side and pressure side of the blade surface for different attack angles. Compared with the 50% spanwise location, relatively larger discrepancies are observed near the 5.4% spanwise section due to the more complex near-wall flow structures close to the endwall region. Nevertheless, the major pressure variation trends are still accurately captured at both spanwise locations, indicating that the present CFD setup provides reliable flow-field data for subsequent deep learning model training.

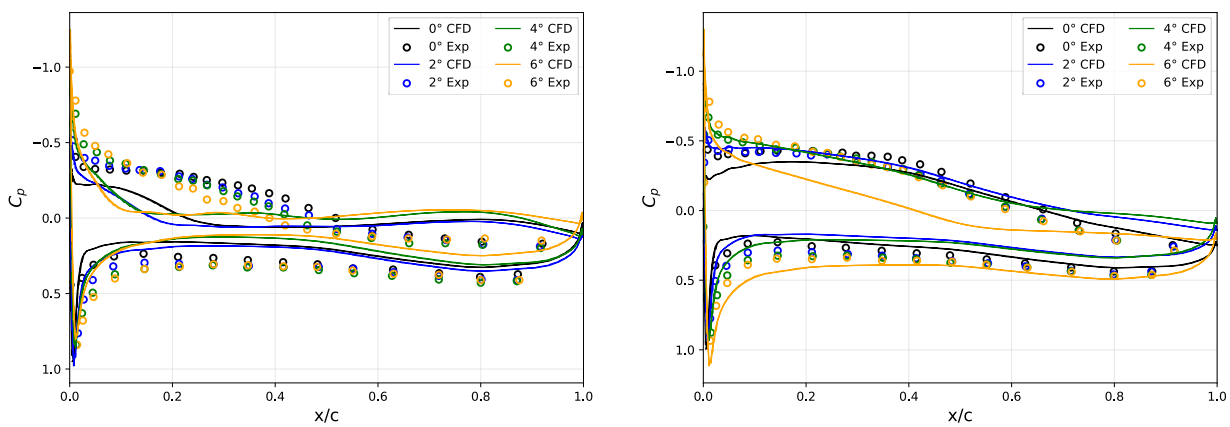



*Figure 1. Comparison of static pressure coefficient distributions between CFD predictions and experimental measurements at different attack angles: (a) 5.4% spanwise section and (b) 50% spanwise section.*

### 2.2 CFD Point-Cloud Representation

The CFD solutions obtained from *ANSYS Fluent* were converted into point-cloud representations for deep learning model training. Specifically, the flow-field data were exported from *ParaView* by directly saving the spatial point information and corresponding flow variables from the CFD results.

Each point in the computational domain contains the three-dimensional spatial coordinates together with the wall-distance information and operating-condition features. Therefore, the input feature vector of the $i$-th point can be represented as:

$$\mathbf{x}_i = [x_i, y_i, z_i, d_i, \sin(\alpha), \cos(\alpha)] \tag{2}$$

where $x_i$, $y_i$, and $z_i$ denote the spatial coordinates of the $i$-th point, $d_i$ represents the wall distance, and $\alpha$ denotes the attack angle. The corresponding target flow variables are defined as:

$$\mathbf{q}_i = [u_i, v_i, w_i, p_i, \nu_{eff,i}] \tag{3}$$

where $u_i$, $v_i$, and $w_i$ denote the velocity components, $p_i$ is the static pressure, and $\nu_{eff,i}$ represents the effective viscosity. All input and output variables were normalized using the statistical properties of the training dataset before model training.

### 2.3 Framework Architecture

*Corresponding Author, Zhihui Li zhihui.li@liverpool.ac.uk

Figure 2 illustrates the overall architecture of the proposed physics-informed latent neural operator framework. The framework consists of a Transolver-based latent encoder and a coordinate-based PINNs decoder for continuous three dimensional flow field reconstruction.

The Transolver encoder receives geometric point features and operating-condition information as input and maps the large scale CFD point-cloud data into a compact latent aerodynamic representation. The encoder input consists of normalized spatial coordinates, wall-distance information, and attack angle encoding features. Through multiple transformer layers with slice attention mechanisms, the encoder captures long-range spatial correlations and global aerodynamic characteristics within the compressor cascade flow field.

The latent representation generated by the encoder can be expressed as:

$$\mathbf{z} = \mathcal{E}(\mathrm{x}, \mathrm{f}) \tag{4}$$

where $\mathcal{E}$ denotes the Transolver encoder, x represents the geometric point features, f denotes the attack-angle encoding, and $\mathbf{z}$ is the latent aerodynamic representation.

To reconstruct continuous flow fields from the latent representation, a coordinate-based PINNs decoder was employed. The decoder receives query-point coordinates together with the latent representation as input and predicts the corresponding flow variables at arbitrary spatial locations. The decoder mapping can be written as:

$$\hat{\mathbf{q}} = \mathcal{D}(\mathbf{x}_i, \mathbf{z}) \tag{5}$$

where $\mathcal{D}$ denotes the coordinate-based decoder and $\hat{\mathbf{q}}$ represents the predicted flow variables.

The decoder predicts the corresponding flow variables at query point locations. The predicted point-wise variables are then post-processed for flow-field visualization and comparison with CFD reference results.

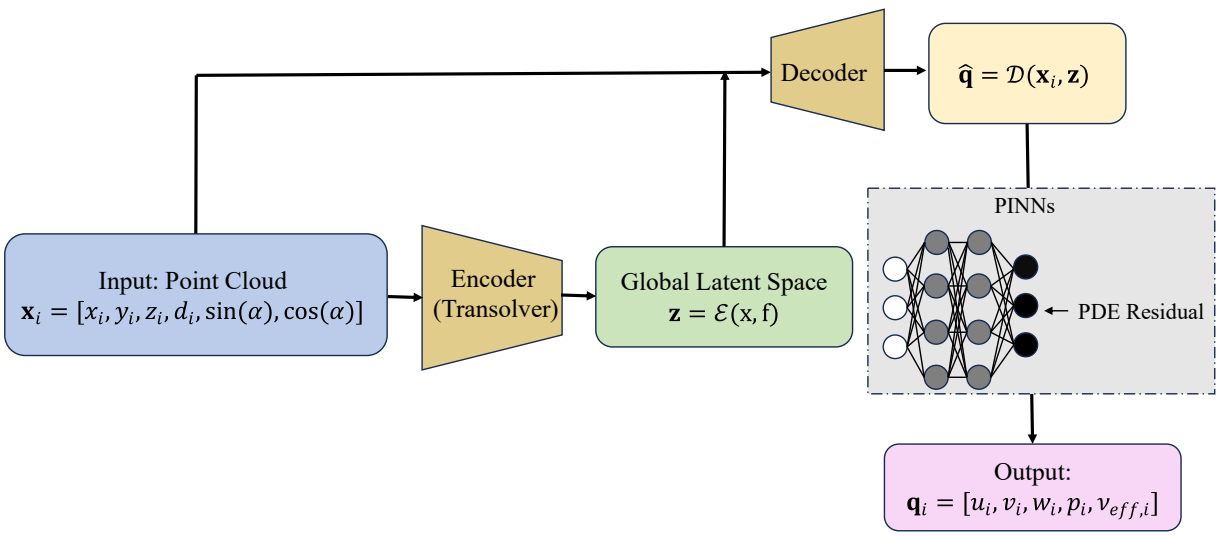



*Figure 2. Framework architecture for CFD field prediction*

### 2.4 Physics-Informed Training Strategy

To improve the physical consistency and extrapolation capability of the proposed framework, physics-informed residual constraints were incorporated during training. The continuity equation is defined as:

$$\nabla \cdot \mathbf{u} = 0 \tag{6}$$

The momentum residual equations can be written as:

$$(\mathrm{u} \cdot \nabla)\mathbf{u} + \nabla p - \nabla \cdot \left[\nu_{eff}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)\right] = 0 \tag{7}$$

During training, automatic differentiation was employed to compute PDE residuals with respect to the spatial coordinates. The overall loss function consists of the data reconstruction loss and physics-informed residual losses:

$$\mathcal{L} = \mathcal{L}_{data} + \lambda_c \mathcal{L}_{cont} + \lambda_m \mathcal{L}_{mom} \tag{8}$$

where $\mathcal{L}_{data}$, $\mathcal{L}_{cont}$, and $\mathcal{L}_{mom}$ denote the data loss, continuity residual loss, and momentum residual loss, respectively. The weighting coefficients $\lambda_c$ and $\lambda_m$ control the contributions of the physics-informed constraints during training. To improve computational efficiency for large-scale CFD point clouds, random point subsampling was employed during training. The model parameters were optimized using the *AdamW* optimizer, and all training procedures were implemented using the *PyTorch* deep learning framework.

### 2.5 Training Datasets and Task Construction

*Corresponding Author, Zhihui Li zhihui.li@liverpool.ac.uk

To investigate the capability of the proposed framework for predicting compressor cascade flow fields under unknown operating condition, CFD datasets with multiple attack angles were constructed. The training dataset consisted of flow-field samples with attack angles of 1°, 2°, 3°, 4°, 6°, 7°, 8°, and 9°. The flow field corresponding to the unseen 5°attack-angle condition was employed as the testing dataset.

All CFD datasets were converted into point-cloud representations containing spatial coordinates, wall-distance information, and corresponding flow variables. During training and evaluation, the same data preprocessing and normalization procedures were consistently applied across all attack-angle conditions.

### 2.6 Computing Facility

All training and testing procedures in this study were carried out on the high-performance computing (HPC) platform Barkla at the University of Liverpool with *Nvidia* H100 GPU card.

## 3 Results

The capability of the proposed framework was evaluated using the unseen 5°attack-angle condition, which lies within the range of the training dataset. Fig.3 presents the comparison of the three-dimensional flow-field predictions between the CFD reference solution and the proposed framework.

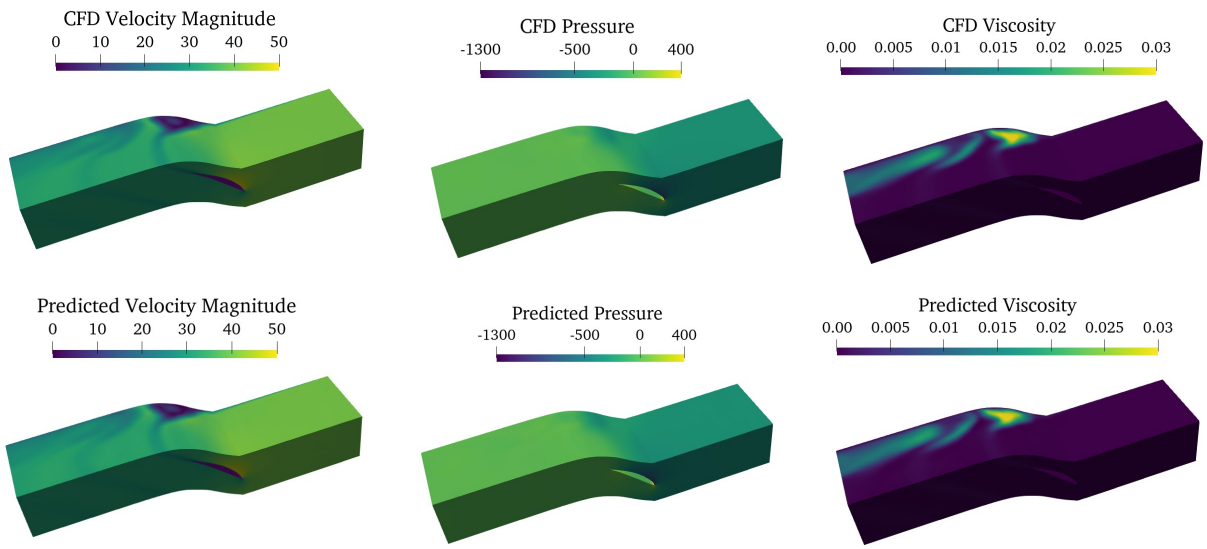



*Figure 3. Comparison of three-dimensional prediction results at the 5°attack-angle condition between the CFD reference solution and the proposed framework: (a) velocity magnitude, (b) pressure, and (c) effective viscosity*

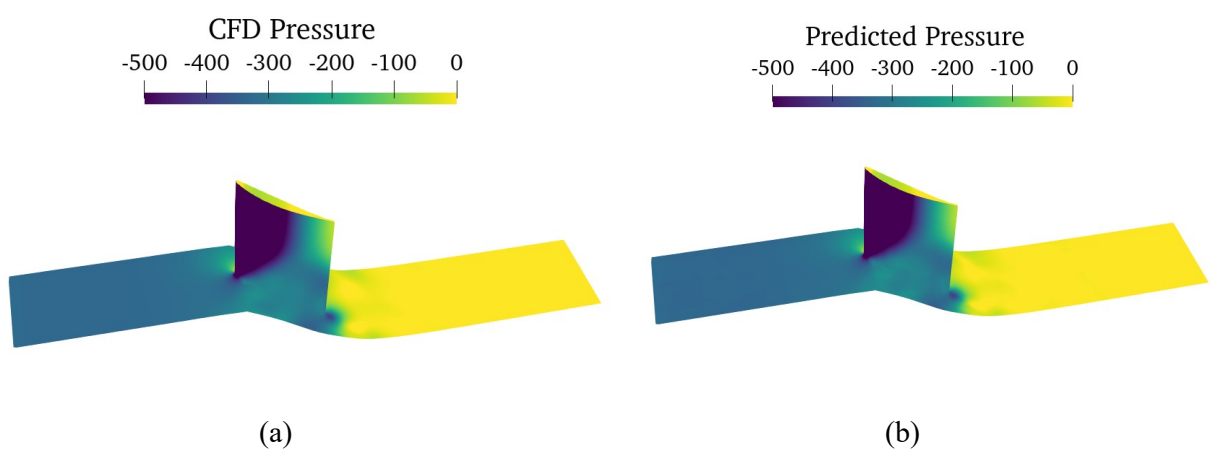



*Figure 4. Comparison of blade-surface pressure distributions for the 5°condition between the CFD reference solution and the proposed framework*

Overall, the predicted velocity magnitude, pressure, and effective viscosity distributions show good agreement with the CFD results. The proposed framework successfully reconstructs the major aerodynamic characteristics of the compressor cascade flow, including the flow acceleration within the blade passage, the low-pressure region near the

*Corresponding Author, Zhihui Li zhihui.li@liverpool.ac.uk

suction side, and the high viscosity regions associated with boundary-layer development. In addition, the predicted flow fields remain smooth and physically consistent without obvious nonphysical oscillations.

To further evaluate the near-wall prediction performance, Fig. 4 compares the blade-surface pressure distributions between the CFD reference solution and the predicted results. Good agreement can be observed on both the pressure side and suction side of the blade surface. The proposed framework accurately captures the pressure variation along the blade surface, including regions with relatively strong pressure gradients near the leading-edge and trailing-edge regions. Although minor discrepancies can still be observed in localized regions with complex three-dimensional flow interactions, the overall prediction accuracy remains satisfactory for the unseen condition. The errors are mainly concentrated near regions with strong secondary-flow structures and relatively large pressure gradients.

To quantitatively evaluate the prediction performance, the relative root mean square error (RMSE) values of different flow variables are summarized in Table 1. The results indicate that the proposed framework achieves low prediction errors for all flow variables under the unseen 5°attack-angle condition, demonstrating its capability for accurate three-dimensional compressor cascade flow construction on CFD point clouds.

**Table 1. Relative RMSE comparison of different flow variables for the 5° prediction case**

| RMSE_$u$ | RMSE_$v$ | RMSE_$w$ | RMSE_$V_m$ | RMSE_$p$ | RMSE_$\nu_{eff}$ |
|---|---|---|---|---|---|
| 2.67% | 0.83% | 1.06% | 0.74% | 1.05% | 3.39% |

## 4 Conclusion

In the present study, a physics-informed latent neural operator framework was proposed for three-dimensional compressor cascade flow prediction on CFD point clouds. The proposed method combines a Transolver-based latent encoder with a coordinate-based PINNs decoder to reconstruct continuous flow fields while incorporating physics-informed residual constraints during training. The prediction performance of the proposed framework was evaluated using the previously unseen 5°attack-angle condition.

The results demonstrate that the proposed method can accurately reconstruct velocity magnitude, pressure, and effective viscosity distributions for complex three-dimensional cascade flows. Good agreement was obtained between the predicted results and CFD reference solutions for both global three-dimensional flow structures and blade-surface flow distributions. In addition, the proposed framework achieved low prediction errors for different flow variables under the unseen operating condition.

Overall, the present study indicates that the proposed physics-informed latent neural operator framework provides a promising approach for efficient CFD surrogate modelling and aerodynamic prediction in turbomachinery applications.

## References

1. X. Guo, W. Li, and F. Iorio (2016). Convolutional neural networks for steady flow approximation. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 481–490.

2. O. Ronneberger, P. Fischer, and T. Brox (2015). U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 234–241.

3. K. He, X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

4. T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. Battaglia (2021). Learning mesh-based simulation with graph networks. *International Conference on Learning Representations (ICLR)*.

5. N. Brandstetter, D. Worrall, and M. Welling (2022). Message passing neural PDE solvers. *International Conference on Learning Representations (ICLR)*.

6. Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar (2021). Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations (ICLR)*.

7. J. Wu, W. Chen, H. Wang, et al. (2024). Transolver: A fast transformer solver for PDEs on general geometries. *Advances in Neural Information Processing Systems (NeurIPS)*.

8. M. Raissi, P. Perdikaris, and G. E. Karniadakis (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

9. W. Ma (2012). Experimental investigation of corner stall in a linear compressor cascade. *PhD thesis, École Centrale de Lyon.*

*Corresponding Author, Zhihui Li zhihui.li@liverpool.ac.uk

# Real-time control of multiphase processes with learned operators

Paolo Guida*

*Clean Energy Research Center (CERP), Physical Science and Engineering (PSE) Division*

*King Abdullah University of Science and Technology*

## Abstract

Real-time feedback control of multiphase flows remains challenging because the governing dynamics are strongly nonlinear, exhibit regime transitions, and are typically accessible only through computationally expensive high-fidelity CFD. This work proposes a surrogate-assisted model predictive control (MPC) framework for the regulation of multiphase processes using learned operators. A Fourier Neural Operator (FNO) is trained to forecast the spatiotemporal evolution of a phase-indicator field (the volume fraction) over a finite horizon from a short history of recent states and a candidate actuation signal. The predicted fields are mapped to a task-relevant scalar observable (the liquid level), and a receding-horizon objective is minimised under input bounds and smoothness penalties. Because the resulting cost can be nonconvex and nonsmooth due to the level-extraction operation, we employ Bayesian optimisation to select the constant-hold inlet velocity that minimises the finite-horizon tracking error with few surrogate rollouts. The approach is demonstrated by solving an optimal control problem (OCP) on a two-phase Eulerian bubble column, where the controller tracks piecewise-constant level setpoints while respecting actuator limits. Results indicate that field-level forecasting with FNOs enables closed-loop performance at a fraction of the cost of embedded CFD, providing a practical route toward MPC for fast multiphase unit operations and a foundation for future extensions to partial observability and physics-informed operator learning.

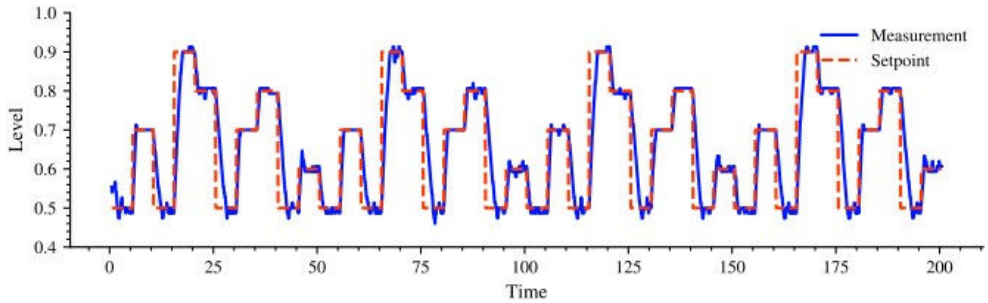



*Figure 1: Closed-loop setpoint tracking performance for the bubble-column liquid level. The measured level (blue) is regulated to a piecewise-constant reference (red dashed) across repeated setpoint changes, with short transients at switching times and small steady-state offsets near the operating extremes.*

**Keywords:** Neural Operators, Model Predictive Control, Multiphase Flows

**Proceedings of the 2nd International Symposium on AI and Fluid Mechanics**

Paper No: [leave blank]

# Physics-Informed Neural Networks for Thermohydrodynamic Modeling of Fuel Injection Pumps with Fuel-Dependent Properties

**Faras Brumand-Poor[1], Georg Michael Puntigam[1], Marius Hofmeister[1], Tim Kotte[1], Katharina Schmitz[1]**

[1]RWTH Aachen University, Institute for Fluid Power Drives and Systems (ifas), Campus-Boulevard 30, 52074 Aachen, Germany

**Keywords:** AI; Thermohydrodynamic Lubrication; Reynolds and Energy Equation; Physics-Informed Machine Learning

**Abstract.** A significant share of the energy consumed in diesel engines is required to operate the fuel injection pump. Within the Excellence Cluster The Integrated Fuel & Chemical Science Center (FSC$^2$) at RWTH Aachen University, methods are being developed to design novel fuels. While research has largely focused on combustion characteristics, the fuel design also plays a significant role in improving injection pump efficiency and, in turn, overall system efficiency. Investigating fuel design requires assessing numerous pure fuels and blend combinations across operating conditions, resulting in substantial computational cost. To address this, we substituted classical thermohydrodynamic (THD) simulations with physics-informed neural networks (PINNs) to accelerate high-dimensional analyses. Building on prior THD-PINNs, we introduce a parameter-dependent PINN that captures multiple operating conditions and fluid compositions within a unified surrogate model. Fluid blends are modeled using simplified mixture rules that incorporate temperature- and pressure-dependent thermal conductivity, along with established density and viscosity dependencies. The model is evaluated on different mixtures of n-heptane/ethanol aligned with ongoing FSC$^2$ research. The surrogate enables efficient determination of THD behavior across the whole solution domain, supporting multi-objective optimization of injection pump performance and trade-offs between ignition delay, pump efficiency, and environmental impact.

## 1. Introduction

Within the Fuel Science Center, numerous approaches have already been developed to optimize bio-hybrid fuels. These efforts address not only combustion-related properties, such as ignition delay, efficiency, and emissions, but also environmental and economic performance indicators. Building on these optimization efforts, the efficiency of the injection pump becomes a key factor, as several kilowatts of power may be required to operate it in passenger-car applications. To estimate the main losses, primarily at the piston-bushing contact, where the pump piston moves within the cylinder sleeve,

thermohydrodynamic (THD) models account for fluid flow, pressure, and temperature, while thermo-elastohydrodynamic (TEHD) models also account for elastic deformation. These models allow straightforward variation of fuel properties and, through appropriate mixing rules, predict changes in blend properties. Despite advances in these simulation models, classical simulation of such tribological systems remains both time-consuming and computationally expensive. The large number of potential fuel candidates, combined with the virtually unlimited number of possible blend combinations, makes the search for an optimal mixture composition highly demanding computationally. To address these computational challenges, physics-informed neural networks (PINNs) represent a highly promising alternative. In previous work, PINNs were successfully applied to simulate a simplified piston–bushing contact [4]. It was demonstrated that PINNs can also resolve the strong pressure- and temperature-dependent variations in material properties characteristic of fuel injection pumps. Extending this approach, whereas previous work by the authors [4] focused on establishing a thermohydrodynamic physics-informed neural network (THD-PINN) framework to train PINNs without incorporating boundary conditions and initial conditions as input parameters, this work now presents parameter-dependent PINNs (also known as multi-case PINNs), where boundary conditions, initial conditions, and fluid mixture fractions are all included as input parameters. In the following, Section 2 introduces the investigated system, the injection pump, the governing physical equations, the investigated test cases, the fuel mixture rules, the THD simulation, and the THD-PINN. Afterward, Section 3 presents the results of the THD-PINN compared to the conventional THD simulation, and eventually Section 4 provides a conclusion and an outlook.

## 2. Methodology

The efficiency of an injection system, as shown in Figure 1, is heavily impacted by losses within the injection pump. Beyond bearing and shaft–cam friction, pump efficiency depends chiefly on volumetric losses at the piston-bushing interface. Consequently, this study centers exclusively on this contact. As the main objective of this work is to develop a multi-case PINN for predicting the impact of fuel composition on volumetric losses at this contact, a simplified computational domain (Figure 1, left) is used. Although the actual contact is cylindrical and has surface roughness, the simplified model is represented as a one-dimensional gap with constant height. Other effects, such as solid deformation and thermal interactions between fluid and solid, are also omitted.

To describe the pressure and temperature distribution for the presented domain, the one-dimensional Reynolds equation (2.1a) and the energy equation (2.1b) are solved.

$$\underbrace{\frac{\partial}{\partial x}\left(-\frac{\rho h^3}{12\eta}\frac{\partial p}{\partial x}\right)}_{Poiseuille}+\underbrace{\frac{u}{2}\frac{\partial(\rho h)}{\partial x}}_{Couette}+\underbrace{\frac{\partial(\rho h)}{\partial t}}_{Trans.}=0 \quad (2.1a) \qquad \underbrace{\rho c_p\left(\frac{\partial\vartheta}{\partial t}+u\frac{\partial\vartheta}{\partial x}\right)}_{Convection}-\underbrace{\lambda\left(\frac{\partial^2\vartheta}{\partial x^2}\right)}_{Diff.}=\underbrace{\eta\left(\frac{\partial u}{\partial z}\right)^2}_{Diss.} \quad (2.1b)$$

High pressures (up to 3000 bar) at the piston–bushing contact raise the fluid temperature to 250 °C. Therefore, fuel properties are modeled as functions of both pressure and temperature. The relations used are shown in Equations (2.2a)–(2.2c) [6]. Building on the modeling approach introduced above, the dynamic viscosity $\eta$ is calculated using a Vogel–Barus equation of state. The temperature dependence of the density $\rho$ is modeled by a linear relation, while its pressure and temperature dependence are represented by a modified Tait equation. In both cases, polynomial functions (2.2d) are employed to improve the accuracy of the pressure dependence. Furthermore, the thermal conductivity $\lambda$ is modeled using an exponential function based on the same polynomial formulation. For the heat capacity $c_p$, the influence of pressure and temperature is neglected, and a constant value is assumed. [1]

$$\eta(p,T)=A\cdot\exp\left(\frac{B}{C+T}+p\cdot P_\eta\right) \quad (2.2a) \qquad \lambda(p,T)=\exp(P_\lambda) \quad (2.2b)$$

$$\rho(p,T)=A_0(1-A_1T)\left(\frac{p+P_{\rho1}}{P_{\rho1}}\right)^{\frac{1}{P_{\rho2}}} \quad (2.2c) \qquad P_k=\sum_{i=0}^{n}\sum_{i=0}^{n}p_{kij}T^ip^j \quad (2.2d)$$

Based on discrete data points obtained using CoolProp and the proposed fluid models, a regression analysis was performed. These results show good agreement between the model predictions and the reference data [2]. Specifically, the coefficient of determination, ($R^2$), exceeds 0.98 for all fluid properties and all fuels considered. Furthermore, no polynomial of degree higher than two was required, $n \leq 2$. The fluid properties of fuel mixtures are defined using the simplified mixing rules in Equations (2.3a)–(2.3c). Specifically, the viscosity $\eta_{mix}$ is calculated with a logarithmic rule, while the density $\rho_{mix}$ uses the harmonic mean of the components. Furthermore, the thermal conductivity $\lambda_{mix}$ and heat capacity $c_{pmix}$ are determined with an arithmetic rule. [1]

$$\ln(\eta_{mix})=\sum_{i=0}^{n}\ln(\eta_i)\cdot x_i \quad (2.3a) \qquad \frac{1}{\rho_{mix}}=\sum_{i=0}^{n}\frac{x_i}{\rho_i} \quad (2.3b) \qquad \phi_{mix}=\sum_{i=0}^{n}\phi_i\cdot x_i \quad (2.3c)$$

The presented equations describing the pressure and temperature fields, as well as the fluid properties, are applied consistently in both the THD simulation and the THD PINN.

### *2.1. The Thermohydrodynamic Simulation*

To solve governing equations, the simulation model described in [4], based on the finite volume method (FVM), was used. Central differencing schemes were applied for the pressure and temperature gradients, while all convective terms were treated using an upwind scheme to ensure numerical stability. Where required, harmonic averaging was employed for the evaluation of interfacial properties. The simulation stopped when the residuals fell below $1\times10^{-8}$. A time step size of 10 µs was used, and the computational domain was discretized into 500 uniformly sized control volumes. For a more detailed description of the simulation model and its numerical implementation, the reader is referred to [4, 7].

### 2.2. The Thermohydrodynamic PINN

The THD PINN receives spatial and temporal grid points from the simulation domain $(x,t)$ as well as the values of the boundary conditions and the fluid fractions as inputs, as listed in Table 1 and illustrated in Figure 2. For the sum of all fluid fractions, it holds that $\sum_{i=1}^{m} X_i = 1$. Between input ranges, either five or nine points were sampled, denoted as "N" in table 2. Several PINNs were trained with a combination of $x_{\text{mix}}$, $p$ and $T$ as input parameters for the multicase PINN, denoted as "MC" in table 2.

Table 1: The variables and their range provided to the THD PINN as inputs.

| Input | Description | Range | Input | Description | Range |
|---|---|---|---|---|---|
| $x$ | spatial coordinate | $[0, 0.25]$ | $\vartheta_{x=0}$ | temperature at left boundary in °C | $[20, 60]$ |
| $t$ | temporal coordinate | $[0, 0.01]$ | $p_{x=0}$ | pressure at left boundary in bar | $[100, 3000]$ |
| | | | $X_{fl}$ | vol. fraction $X$ of fluid $fl$ | $[0, 1]$ |

Boundary and initial conditions were enforced to reduce losses during balancing. Subsequently, the previously described equations are used. The NN consists of 3 layers with 32 neurons each, and *GELU* [5] activation functions were chosen. Furthermore, a Bayesian optimizer [10] was used for hyperparameter tuning, but it did not improve the NN model's performance after 80 iterations, each with $10^3$ epochs. For the training of the PINN model, the *SPAM* variant [8] of the *ADAM* optimizer was used. Finally, for loss balancing, the *ReLoBRaLo* algorithm [3] and the *conFIG* algorithm [9] were used. For a more detailed description of the PINN, the reader is referred to [4].

## 3. Results

Table 2: Comparison of results with varying number of sampling points during training and input parameters.

| MC | N | $\max(e_{\text{rel, p}})$ [%] | $\text{mean}(e_{\text{rel, p}})$ [%] | $\max(e_{\text{rel},\vartheta})$ [%] | $\text{mean}(e_{\text{rel},\vartheta})$[%] | $n_{\text{train}}$ [1k] | $t_{\text{train}}$ [min] |
|---|---|---|---|---|---|---|---|
| $x_{\text{mix}}$ | 5 | 3.83 | 1.10 | 3.32 | 0.58 | 100 | 6.931 |
| $x_{\text{mix}}$ | 9 | 5.14 | 0.35 | 4.29 | 0.45 | 100 | 10.07 |
| $x_{\text{mix}}, p$ | 5 | 15.60 | 1.32 | 6.15 | 0.50 | 10 | 25.48 |
| $x_{\text{mix}}, p$ | 9 | 14.98 | 1.09 | 6.06 | 0.35 | 10 | 79.07 |
| $x_{\text{mix}}, \vartheta$ | 5 | 7.18 | 0.83 | 4.26 | 0.79 | 10 | 25.32 |
| $x_{\text{mix}}, \vartheta$ | 9 | 10.51 | 1.97 | 11.87 | 3.94 | 10 | 78.78 |
| $x_{\text{mix}}, p, \vartheta$ | 5 | 35.58 | 2.55 | 20.75 | 1.45 | 1 | 16.8 |
| $x_{\text{mix}}, p, \vartheta$ | 9 | 22.80 | 1.36 | 14.95 | 0.93 | 1 | 70.26 |

The simulation was conducted over the full factorial design of linearly spaced parameter values, comprising 5 points for the left-boundary temperature $\vartheta_{x=0} \in [20, 60]\,°C$, 5 points for the left-boundary pressure $p_{x=0} \in [100, 3000]bar$, and 9 points for the volume fraction $X_i \in [0, 1]$. Table 2 shows maximal and mean relative errors for both pressure $p$ and Temperature $\vartheta$ compared to simulation results. One simulation ran for about 6 minutes, indicating that all PINNs achieve a significant speedup, even when training time is included.

Figures 3, 4a, 4b show a statistical evaluation of relative errors across one input parameter of a single $x_{\text{mix}}, p, \vartheta$ multicase PINN. Outliers were omitted for visibility as maximal errors are already shown in table 2.

## 4. Conclusion & Outlook

This work introduces a parameterized THD-PINN, which is effectively applied in a multi-case scenario to characterize the pressure and temperature dynamics of an injection pump operating with an ethanol/n-heptane blend. The THD-PINN yields slightly less accurate results than conventional THD simulations while significantly reducing computational time. Although initial training involves greater computational resources, the trained PINN enables rapid evaluation of numerous test scenarios, accommodating variations in fuel mixtures, pressures, and temperatures. Conversely, traditional THD simulations require individual computation for each scenario, leading to substantially higher computational demands for broad parameter studies. Therefore, the THD-PINN offers an efficient solution for quickly analyzing diverse operating conditions and fuel blends. Future work will expand the method to additional fuels and more complex mixing models. Furthermore, upcoming research will examine more complex geometries and kinematics in conjunction with thermal and mechanical interactions within solid bodies.


## Acknowledgments

The Fuel & Chemical Science Center (FSC$^2$) is funded by the German Council of Science and Humanities (Wissenschaftsrat, WR) and by Deutsche Forschungsgemeinschaft (DFG).


## References


[1] Dirk Bartel. *Simulation von Tribosystemen: Grundlagen und Anwendungen*. Vieweg+Teubner Verlag, Wiesbaden, Germany, 2010.

[2] Ian H. Bell, Jorrit Wronski, Sylvain Quoilin, and Vincent Lemort. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolprop. *Industrial & Engineering Chemistry Research*, 53(6):2498–2508, 2014.

[3] Rafael Bischof and Michael A. Kraus. Multi-objective loss balancing for physics-informed deep learning. *Computer Methods in Applied Mechanics and Engineering*, 439:117914, May 2025.

[4] Faras Brumand-Poor, Georg Michael Puntigam, Marius Hofmeister, and Katharina Schmitz. Advancing ml-based thermal hydrodynamic lubrication: A data-free physics-informed deep learning framework solving temperature-dependent lubricated contact simulations. *Lubricants*, 14(2), 2026.

[5] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023.

[6] John B. Heywood. *Internal Combustion Engine Fundamentals*. McGraw-Hill, New York, 1988.

[7] Marius Hofmeister and Katharina Schmitz. Increasing efficiency of diesel injection pumps by targeting blending of bio-hybrid fuels. In *66. Tribologie-Fachtagung*, pages 292–295. Gesellschaft für Tribologie e.V., 2025.

[8] Tianjin Huang, Ziquan Zhu, Gaojie Jin, Lu Liu, Zhangyang Wang, and Shiwei Liu. Spam: Spike-aware adam with momentum reset for stable llm training, 2025.

[9] Qiang Liu, Mengyu Chu, and Nils Thuerey. Config: Towards conflict-free training of physics-informed neural networks. *arXiv preprint arXiv:2408.11104*, 2024.

[10] Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, and Luca Invernizzi. Kerastuner. `https://github.com/keras-team/keras-tuner`, 2019.

aifluids/figures/Injection system.png

Figure 1: Schematic sketch of a state-of-the-art common-rail injection system and the simplified computational domain for the piston/bushing contact.

aifluids/figures/PINN_adjusted.png

Figure 2: Illustration of information flow during the PINN training.

aifluids/figures/figure_1_x_mix_errors_short.pdf

Figure 3: Relative errors for varying ethanol fraction in fluid

aifluids/figures/figure_2_pressure_errors_half_ds.pdf

(a) Varying left pressure boundary condition

aifluids/figures/figure_3_temperature_errors_half_

(b) Varying left temperature boundary condition

Figure 4: Relative errors for varying left boundary conditions

# Physics-informed Neural Networks and Differentiable Chemistry for Solving the Unsteady Flamelet Equation

**Milos Babic[1,2], Franz Rohrhofer[3], Clemens Gößnitzer[4], Ricardo Novella[5], José Maria García-Oliver[5], Stefan Posch[1,2]**

[1] *CD Laboratory for Physics-driven Machine Learning in Industrial Applications, Graz, Austria*
[2] *Institute of Thermodynamics and Sustainable Propulsion Systems, Graz University of Technology, Austria*
[3] *Know-Center Research GmbH, Austria*
[4] *Large Engines Competence Center GmbH, Austria*
[5] *CMT – Clean Mobility & Thermofluids, Universitat Politècnica de València, Spain*
*Corresponding author: Stefan Posch (st.posch@tugraz.at)*

**Keywords:** PINN, combustion, flamelet, differentiable chemistry

**Abstract.** The numerical simulation of turbulent combustion is faced by a trade-off between modeling depth and affordable computation cost. A well-established approach for non-premixed combustion is the flamelet model which represents a turbulent flame as an ensemble of laminar diffusion flamelets and couples CFD to an *a priori* tabulated library of solutions of the unsteady flamelet equation. While this method is accurate, it comes with large memory footprints to keep discretization errors small, and an interpolation step that adds runtime to the final CFD simulation. To overcome this drawbacks, a physics-informed neural network (PINN) is proposed that acts as a PDE solver for the unsteady flamelet equation in forward mode. A flexible differential chemistry solver is integrated into the physics-based loss function enabling backpropagation through the full reaction mechanism while enforcing the governing PDE together with boundary and initial constraints. Thus, the strong diffusion-reaction nature of the unsteady flamelet equation and the chemical stiffness is handled. The PINN is validated on hydrogen diffusion flamelets which is an ideal stress test due to pronounced kinetic stiffness. Additional results on ammonia combustion confirm that the proposed PINN is able to reconstruct flamelets with high accuracy by using hard constraints for boundary and initial conditions and consequently without the need of data generated by numerical simulation. The work also discusses the ability of PINNs to generalize across flamelet parameters and thus, replacing flamelet tables.

## 1. Introduction

Turbulent combustion is one of the most challenging tasks in computational fluid dynamics (CFD). The combination of solving turbulent flows, which requires highly accurate modeling approaches and significant knowledge by itself, and complex chemistry, leads to high demands on the modeling approaches [1]. In addition to the requirements to accurately model turbulent flames and their underlying physics, especially turbulence chemistry interaction, the computational demand of turbulent combustion models should be kept in reasonable limits. A well-established approach, especially in case of non-premixed flames, is the flamelet model. Here, the flame is represented by an ensemble of finite laminar flames, the so-called flamelets. The flamelet assumption is valid in flame regimes where the interactions between turbulence and combustion are limited since small eddies remain larger than the reaction zone [2]. However, this assumption holds true for the majority of technical relevant combustion appliances. For non-premixed combustion, the flamelets are modeled dependent on the mixture fraction $Z$, which represents the fuel mass fraction. This dependence implies that the combustion is essentially governed by the mixing of fuel and oxidizer [3]. In addition of the flame representation by flamelets, the three-dimensional turbulent flow can be separated from the one-dimensional (1D) reaction/diffusion process in the laminar flame. Thus, the flamelets can be calculated independently of the turbulent flow field simulation in a

preprocessing step, in which the flamelet results are stored in look-up tables. While this approach is in principle computationally cheap for CFD since the chemistry solution is outsourced, some drawbacks still exist: Each parameter for the pre-generated flamelet adds a dimension to the lookup table, which increases the size of the tables to unmanageable limits if multi-fuel or non-isobaric conditions are considered. The use of machine learning (ML) methods for the memory efficient representation of the flamelet data has been investigated. Rohrhofer et al. [4] introduced a neural network to learn species mass fractions of flamelet tables. Their approach incorporates a network architecture that automatically conserves mass and employs a particular loss weighting based on species depletion. Owoyele et al. [5] compared three different training strategies namely one ANN for each flamelet output, one artificial neural network (ANN) for all flamelet output and one ANN for data groups based on similar species behaviour. They implemented the latter two variants in an LES of the engine combustion network (ECN) spray A dodecane spray combustion case as well as an LES of an internal combustion engine. The authors reported a significant memory reduction while keeping accuracy in similar ranges as the tabulated case. Li et al. [6] compared different ML methods such as ANN and gradient boosted trees to represent tabulated flamelet data. They reported that data preprocessing influences the model performance significantly. In contrast to the aforementioned purely data-driven ML approaches, the present work investigates the application of physics-informed ML models, in particular physics-informed neural networks (PINNs). The main idea of using PINNs is to combine the flamelet data generation process and the efficient data storage capabilities of neural networks in a single step. Thus, a novel PINN model [7] is used which solves the unsteady flamelet equation by implementing a differentiable chemistry solver enabling end-to-end backpropagation through the chemistry solver during training. The model is validated on hydrogen ($H_2$) and ammonia ($NH_3$) systems, showing that the PINN is able to flamelets with high accuracy.

# 2. Methodology

## *2.1. Flamelet model*

The laminar flamelet solution used for the CFD look-up tables is usually determined assuming a structure similar to a laminar stagnation point flame given by a planar opposed jet configuration [2]. Assuming that the mixture fraction $Z$ is a conserved scalar and unity Lewis number, the 1D mass conservation equation for each species $k$ in $Z$-space can be written by [2]

$$\frac{\partial Y_k}{\partial t} = \frac{\chi}{2}\frac{\partial^2 Y_k}{\partial Z^2} + \dot{\omega}_k, \tag{1}$$

where $\dot{\omega}_k$ denotes the reaction term of the relevant species $k$ and $\chi$ is the scalar dissipation rate. This formulation represents a 1D reaction/diffusion partial differential equation (PDE) in the most simple form. The scalar dissipation rate can be expressed as [2]

$$\chi^{\infty}(a, Z) = \frac{a}{\pi}\exp\left[-2\left(\mathrm{erfc}^{-1}(2Z)\right)^2\right], \tag{2}$$

with $a$ is the strain rate as flamelet parameter. To solve the unsteady flamelet equation, $Y_k(Z=0)$ and $Y_k(Z=1)$ are given at the oxidizer and fuel boundary, respectively. At $t=0$, a linear dependency of $Y_k$ over Z is assumed. The total enthalpy of the mixture is approximated by $h = Z \cdot h_{fuel} + (1-Z) \cdot h_{oxidizer}$, which is again a function of composition and temperature.

## *2.2. Differentiable chemistry*

Looking at Eq. 1, the chemical source term $\dot{\omega}_k$ of species $k$ depends on the mass fractions of the full set of species $\boldsymbol{Y}$ resulting in a tightly coupled, nonlinear system across all species. In numerical simulations, chemical kinetics are usually solved by suitable chemistry solvers such as *Cantera* [8]. However, the coupling of $\dot{\omega}_k$ with the PINN output vector $\boldsymbol{Y}$ requires the integration of the source term calculation

to be part of the computational graph and thus, the chemistry solver has to be differentiable. In the present PINN framework, the differentiable chemistry backend *reactorch* [9] is used. *reactorch* enables the determination of the thermodynamic variables and chemistry by using cantera-like input. Since the thermodynamic state definition in *reactorch* requires temperature $T$, pressure $p$ and species mass fractions $\boldsymbol{Y}$, an additional differentiable root finding algorithm was implemented to determine $T$ from $h$ based on the NASA 7-coefficient polynomials.

### *2.3. Physics-informed neural network*

PINNs are a deep-learning framework for solving forward and inverse problems described by ordinary or partial differential equations [10]. In forward mode, PINNs can be used to solve differential equations without the need of a computational mesh. By approximating the solution by a neural network with parameters $\theta$, the corresponding physics residual for the flamelet equation 1 can be written as

$$r_k^\theta(t, Z) = -\frac{\partial Y_k^\theta}{\partial t} + \frac{\chi}{2}\frac{\partial^2 Y_k^\theta}{\partial Z^2} + \dot{\omega}_k. \tag{3}$$

The corresponding physics loss to enforce the governing equations at collocation points sampled in the interior of the domain is

$$\mathcal{L}_\mathrm{f} = \sum_{i=1}^{N_f}\sum_{k=1}^{N_s}\left|r_k^\theta(t^i, Z^i)\right|^2, \tag{4}$$

where $N_f$ denotes the number of collocation points and $N_s$ denotes the number of species. Since the boundary and initial conditions of the present PINN model to solve the flamelet equation in forward mode are implemented via hard constraints, the network is solely trained on the physics loss $\mathcal{L}_\mathrm{f}$. In order to solve stiff reaction systems such as the flamelet equation, an adapted form of residual weighting was used given by [4]

$$\rho_k^i = \frac{1}{1 + |\omega_k^i(\boldsymbol{Y}^i)|}, \tag{5}$$

where $\omega_k^i(\boldsymbol{Y}^i)$ denotes the reaction rate of species $k$ evaluated for the mass-fraction vector $\boldsymbol{Y}^i$. This weighting reduces the contribution of regions with very high reaction rates to the loss function, thereby reducing train instabilities commonly encountered in stiff reactive systems. Furthermore, mass conservation is enforced in the outlet layer of the network. Figure 1 shows the schematic illustration of the PINN model.

## 3. Result

In the following, the results of the PINN model are presented for $H_2$ and $NH_3$ global reactions. The numerical results for the validation are generated with the flamelet solver *ZLFLAM* [11].

### *3.1. Hydrogen*

Hydrogen combustion is the first reactions on which the PINN model is tested. Due to its fast kinetics, it represents major challenges for the PINN in solving the resulting problem stiffness. The chemistry is modeled by a single-step global mechanism $2H_2 + O_2 \rightleftharpoons 2H_2O$ with the Arrhenius constants $A = 1.8\mathrm{e}19$, $\beta = 0$ and $E_a = 17614$. The thermodynamic properties are taken from the well-known San Diego hydrogen mechanism [12]. Nitrogen as an inert species is implemented as hard constraint to the network.

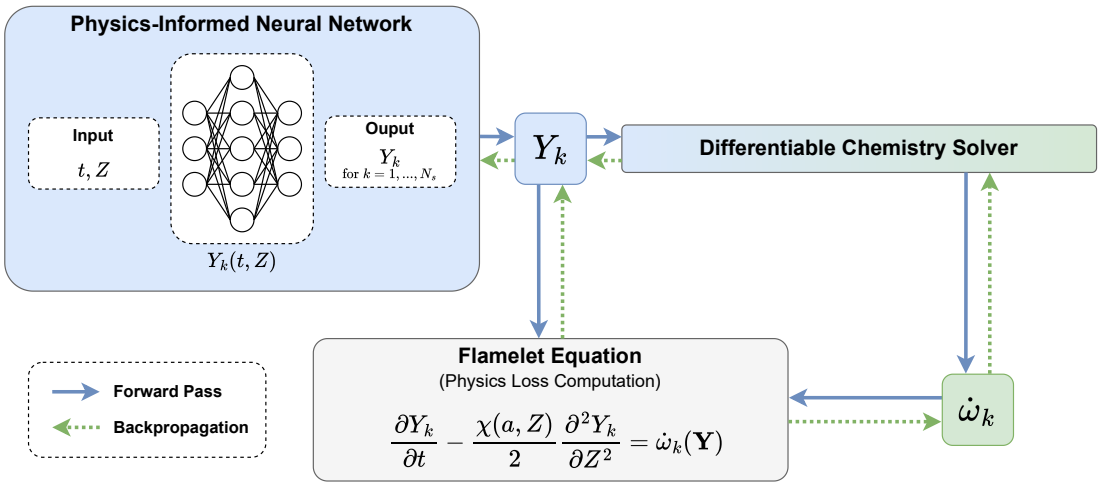



Figure 1: **Schematic illustration of the PINN model.** The PINN approximates the species mass fractions $Y_i$. These predictions are evaluated by a differentiable chemistry solver to compute the species reaction rates $\dot{\omega}_i$. The resulting mass fractions and reaction rates enter the physics-informed loss, enabling end-to-end backpropagation through the chemistry solver during training.

Figure 2(a) shows the solution of the flamelet equation for a fixed strain rate $a = 1\text{s}^{-1}$. The PINN accurately reproduces the species evolution across the full spatiotemporal domain without showing instabilities nor oscillations. Additionally, the PINN achieves stable convergence in a regime characterized by extremely stiff and sharply varying reaction dynamics. These results strongly underline the ability of the present PINN to model reaction/diffusion PDEs such as the flamelet equation by incorporating a differentiable chemistry solver enabling the end-to-end backpropagation while training.

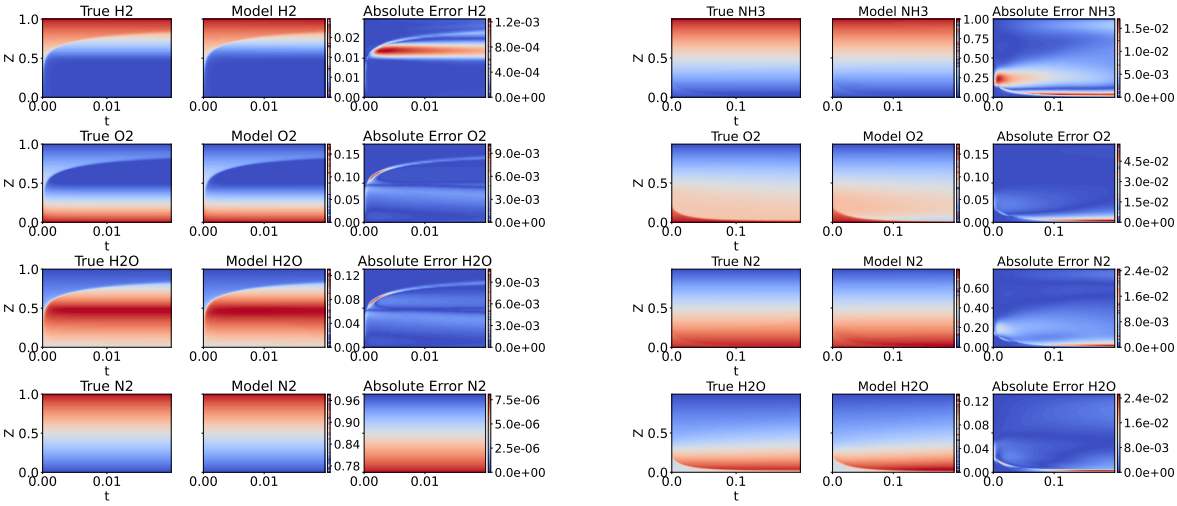
### *3.2. Ammonia*

For the second test, the PINN is used to solve the flamelet equation in forward mode for $NH_3$ combustion. The chemistry of the $NH_3$ reaction is again modeled by a single-step global mechanism $NH_3 + 3/4O_2 \rightleftharpoons 1/2N_2 + 3/2H_2O$ with the Arrhenius constants $A = 2.51\text{e}16$, $\beta = 0$ and $E_a = 75000$. The thermodynamic properties are taken from the GRI-Mech 3.0 [13].

Figure 2(b) shows the solution for a fixed strain rate of $a = 1\text{s}^{-1}$, identical to the hydrogen case. The PINN accurately captures the evolution of the species mass fractions across the entire spatiotemporal domain. Compared to hydrogen combustion, the ammonia reaction exhibits slower chemical kinetics, resulting in less pronounced stiffness of the governing equations. However, unlike the hydrogen case, no inert species are present, increasing the dimensionality of the PINN output space from three to four species. These results further confirm the capability of the proposed PINN framework to accurately solve coupled reaction-diffusion flamelet equations across different combustion regimes.

## 4. Summary and conclusions

The present work presents a general framework for integrating differentiable chemistry solvers into PINNs to solve the unsteady flamelet equation. By further measures such as residual weighting based on




(a) $H_2$ flamelet solution for strain rate $a = 1\,\mathrm{s}^{-1}$.

(b) $NH_3$ flamelet solution for strain rate $a = 1\,\mathrm{s}^{-1}$.

Figure 2: Results of the flamelet solutions for strain rate $a = 1\,\mathrm{s}^{-1}$. The left columns show the reference species mass fractions, the middle columns the PINN predictions using the proposed framework, and the right columns the absolute errors.

the chemical source term, enforcing mass conservation and using hard constraints for initial and boundary conditions, the PINN is able to solve even pronounced kinetic stiffness such as is the case for hydrogen reaction. This work also builds the base for further PINN extensions for flamelet applications. Using one PINN instance for multiple strain rates or other flamelet parameters such as pressure, a potential pathway toward integrating PINN-based chemistry models into high-fidelity CFD simulations, where detailed chemical kinetics are traditionally represented through discrete precomputed lookup tables, would be achieved. Thus, a more flexible and memory-efficient alternative for modeling combustion chemistry across parameter spaces could be offered.

However, the current PINN model is limited to global one-step Arrhenius reaction mechanisms. While the differentiable chemistry solver is directly applicable to multi-step and detailed chemical kinetics, the adaption of the PINN is challenging due to increased stiffness and system dimensionality. These extensions are part of future investigations.

## Acknowledgements

The financial support by the Austrian Federal Ministry of Economy, Energy and Tourism, and the Christian Doppler Research Association is gratefully acknowledged.

## References

[1] S. Posch, C. Gößnitzer, M. Lang, R. Novella, H. Steiner, and A. Wimmer, "Turbulent combustion modeling for internal combustion engine cfd: A review," *Progress in Energy and Combustion Science*, vol. 106, p. 101200, 2025.

[2] N. Peters, *Turbulent Combustion*. Cambridge Monographs on Mechanics, 2004.

[3] G. Stiesch, *Modeling Engine Spray and Combustion Processes*. 01 2003.

[4] F. M. Rohrhofer, S. Posch, C. Gößnitzer, J. M. García-Oliver, and B. C. Geiger, "Utilizing neural networks to supplant chemical kinetics tabulation through mass conservation and weighting of species depletion," *Energy and AI*, vol. 16, p. 100341, 2024.

[5] O. Owoyele, P. Kundu, M. M. Ameen, T. Echekki, and S. Som, "Application of deep artificial neural networks to multi-dimensional flamelet libraries and spray flames," *International Journal of Engine Research*, vol. 21, no. 1, pp. 151–168, 2020.

[6] K. Li, P. Rahnama, R. Novella, and B. Somers, "Combining flamelet-generated manifold and machine learning models in simulation of a non-premixed diffusion flame," *Energy and AI*, vol. 14, p. 100266, 2023.

[7] M. Babić, F. M. Rohrhofer, and S. Posch, "Differentiable chemistry in pinns for solving parameterized and stiff reaction systems," 2026.

[8] D. G. Goodwin, H. K. Moffat, I. Schoegl, R. L. Speth, and B. W. Weber, "Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes," 2025.

[9] W. Ji and S. Deng, "Reactorch: A differentiable reacting flow simulation package in pytorch." `https://github.com/DENG-MIT/reactorch`, 2020.

[10] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[11] B. Naud, R. Novella, J. M. Pastor, and J. F. Winklinger, "RANS modelling of a lifted H2/N2 flame using an unsteady flamelet progress variable approach with presumed PDF," *Combustion and Flame*, vol. 162, no. 4, pp. 893–906, 2015.

[12] P. Saxena and F. A. Williams, "Testing a small detailed chemical-kinetic mechanism for the combustion of hydrogen and carbon monoxide," *Combustion and Flame*, vol. 145, no. 1, pp. 316–323, 2006.

[13] G. P. Smith, D. M. Golden, M. Frenklach, N. W. Moriarty, B. Eiteneer, M. Goldenberg, C. T. Bowman, R. K. Hanson, S. Song, W. C. Gardiner, J. V. V. Lissianski, and Z. Qin, "Gri-mech 3.0."

# Appendix

*Training Settings*

The solution function is approximated using a fully connected neural network with four hidden layers and 50 neurons per layer. All hidden layers use the hyperbolic tangent activation function, while an exponential activation is applied to the output layer to ensure non-negativity of the predicted mass fractions and also to allow applying constraint for the mass fraction sum. Input variables are scaled to the interval $[-1, 1]$. Collocation points in time ($t$) and mixture fraction ($Z$) are sampled using Latin hypercube sampling, with $1024$ collocation points per epoch.

Hard constraints for initial and boundary conditions are implemented as

$$\tilde{\boldsymbol{Y}}_{\text{react}}(t, Z) = Z\boldsymbol{Y}_{\text{fuel}} + (1 - Z)\boldsymbol{Y}_{\text{coflow}} + \boldsymbol{f}_{\theta}(t, Z)(t/t_{\max})Z(1 - Z). \tag{6}$$

In the case of hydrogen, inert specie mass fraction is hard constrained as:

$$Y_{\text{N}_2}(t) = ZY_{\text{fuelN}_2} + (1 - Z)Y_{\text{coflowN}_2}. \tag{7}$$

In both setups, mass conservation is enforced by normalizing the reactive species such that

$$\boldsymbol{Y}_{\text{react}}(t) = \frac{\tilde{\boldsymbol{Y}}_{\text{react}}(t)}{\sum_i \tilde{Y}_{\text{react},i}(t)} \left(1 - Y_{\text{N}_2}(0)\right). \tag{8}$$

*Combustion Settings*

*Hydrogen* The flamelet formulation is defined by two opposing streams corresponding to the fuel and coflow (oxidizer) states. The fuel and coflow states for the $H_2$ system are specified by their temperature and composition according to [11] as

$$\begin{aligned} \text{Fuel:} \quad & T = 305\ \text{K}, \quad \boldsymbol{X}_{\text{fuel}} = \{X_{\text{H}_2} = 0.25,\ X_{\text{N}_2} = 0.75\}, \\ \text{Coflow:} \quad & T = 1045\ \text{K}, \quad \boldsymbol{X}_{\text{coflow}} = \{X_{\text{O}_2} = 0.14744,\ X_{\text{N}_2} = 0.75363,\ X_{\text{H}_2\text{O}} = 0.09893\}. \end{aligned}$$

Using *Cantera*, the exact thermochemical states of the fuel and coflow streams can be computed. The resulting species mass fractions define Dirichlet boundary conditions in $Z$ space,

$$\boldsymbol{Y}(Z=0) = \boldsymbol{Y}_{\text{coflow}}, \qquad \boldsymbol{Y}(Z=1) = \boldsymbol{Y}_{\text{fuel}}. \tag{9}$$

The corresponding mass-fraction values obtained from the equilibrium evaluation are

$$\begin{aligned} Y_{\text{H}_2}(Z=0) &= 0.0000, & Y_{\text{H}_2}(Z=1) &= 0.0234, \\ Y_{\text{O}_2}(Z=0) &= 0.1709, & Y_{\text{O}_2}(Z=1) &= 0.0000, \\ Y_{\text{H}_2\text{O}}(Z=0) &= 0.0645, & Y_{\text{H}_2\text{O}}(Z=1) &= 0.0000, \\ Y_{\text{N}_2}(Z=0) &= 0.7646, & Y_{\text{N}_2}(Z=1) &= 0.9766. \end{aligned}$$

*Ammonia* For the $NH_3$ system, the relevant parameters are:

$$\begin{aligned} \text{Fuel:} \quad & T = 2000\ \text{K}, \quad \boldsymbol{X}_{\text{fuel}} = \{X_{\text{NH}_3} = 1.0\} \\ \text{Coflow:} \quad & T = 1045\ \text{K}, \quad \boldsymbol{X}_{\text{coflow}} = \{X_{\text{O}_2} = 0.14744,\ X_{\text{N}_2} = 0.75363,\ X_{\text{H}_2\text{O}} = 0.09893\}. \end{aligned}$$

The corresponding mass-fraction values are

$$\begin{aligned} Y_{\text{NH}_3}(Z=0) &= 0.0000, & Y_{\text{H}_2}(Z=1) &= 1.0000, \\ Y_{\text{O}_2}(Z=0) &= 0.1709, & Y_{\text{O}_2}(Z=1) &= 0.0000, \\ Y_{\text{N}_2}(Z=0) &= 0.7646, & Y_{\text{H}_2\text{O}}(Z=1) &= 0.0000, \\ Y_{\text{H}_2\text{O}}(Z=0) &= 0.0645, & Y_{\text{N}_2}(Z=1) &= 0.0000. \end{aligned}$$