Data-Driven Hybrid Boundary XPINNs for Scalable PDE Learning

Hamze EL HALABI, Yusuke TAKAHASHI

halabihamze.el.k3@elms.hokudai.ac.jp; ytakahashi@eng.hokudai.ac.jp

北海道大学 HOKKAIDO UNIVERSITY

北海道大学大学院 機械·宇宙航空工学部門

宇宙輸送工学研究室

Space Transportation System Laboratory

SCA 2026 Supercomputing Asia — Gathering the Best of HPC in Asia

HPC Asia 2026

Introduction

- **Motivation:** High-fidelity CFD/FEM is accurate but expensive for large sweeps and 3D cases.
- **Background:** PINNs solve PDEs by enforcing physics in the loss, reducing reliance on dense labels.
- **Challenge:** Training is unstable/slow near boundaries and interfaces due to loss imbalance and continuity constraints.
- **Goal:** Improve stability and time-to-accuracy while keeping a clear path to HPC scaling.
- **Contribution:** DHB-XPINN = XPINN + data-anchored boundary/interface loss (physics residual in the interior).
- **HPC angle:** Subdomains can train in parallel (multi-GPU/node) with lightweight interface synchronization.

Methods

- **Governing Equations**

$$\nabla \cdot u = 0; \ \frac{\partial u}{\partial t} + (u \cdot \nabla)u + \frac{1}{\rho}\nabla p - \nu \nabla^2 u = 0$$

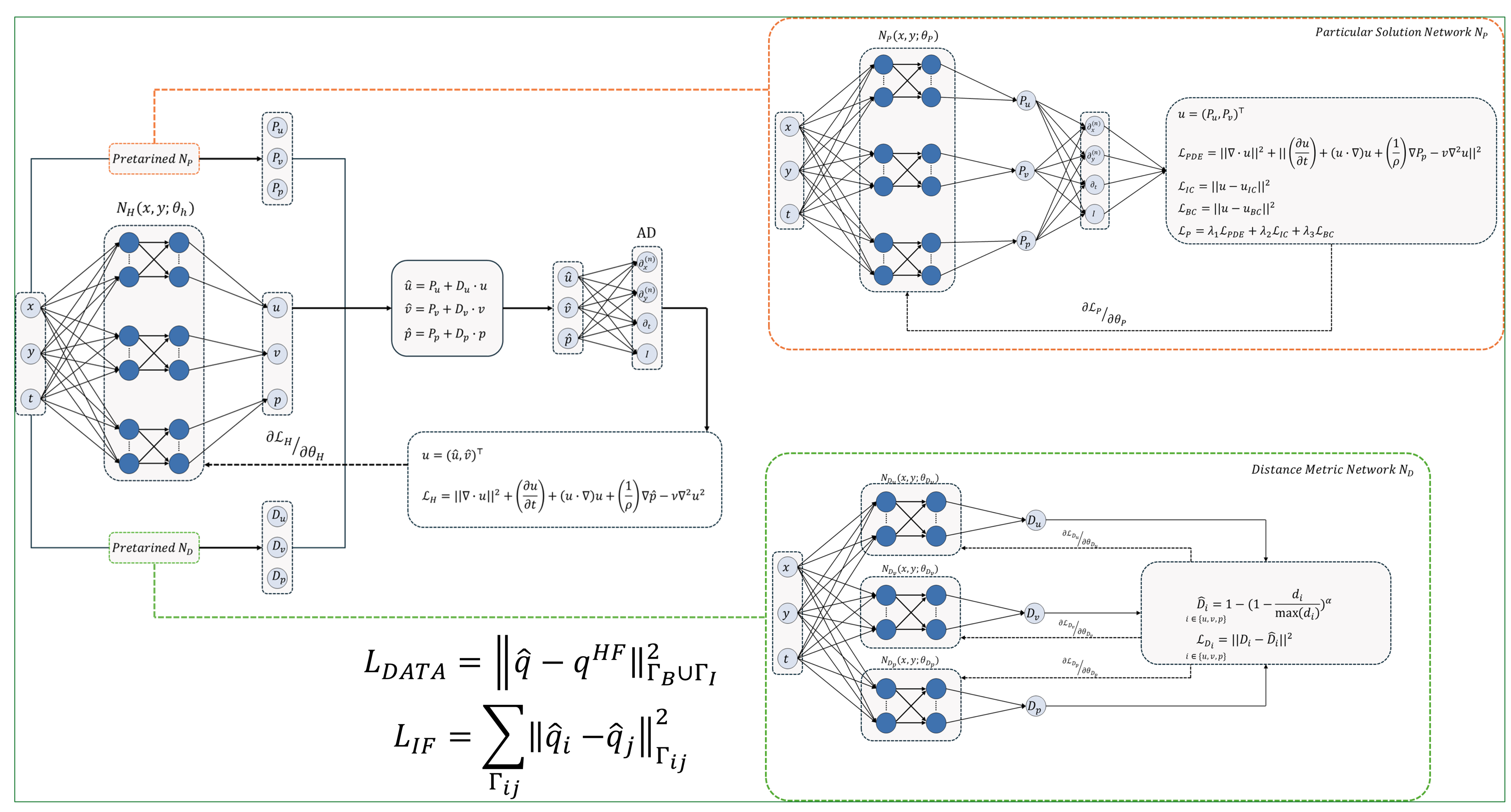
- **Hybrid Boundary Form**

$$\hat{u} = P_u + D_u u, \qquad \hat{v} = P_v + D_v v, \qquad \hat{p} = P_p + D_p p$$

Here u, v, p denote primary-network outputs not the physical ground truth.

- **Final objective**

$$L = \sum_{k=1}^{K} \omega_r L_{PDE}^{(k)} + \omega_{if} L_{IF} + \omega_d L_{DATA}, \qquad \hat{q} = (\hat{u}, \hat{v}, \hat{p})$$



Results

Case 1: Inlet $u = 1, v = 0$; **Outlet** $p = 0$; Walls no-slip

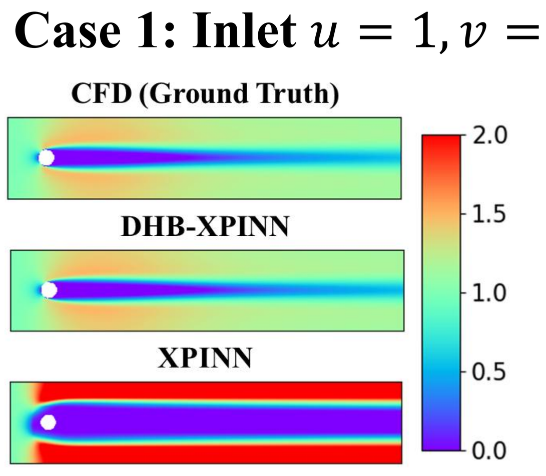


Fig 1:Velocity distributions for the flow around a cylinder

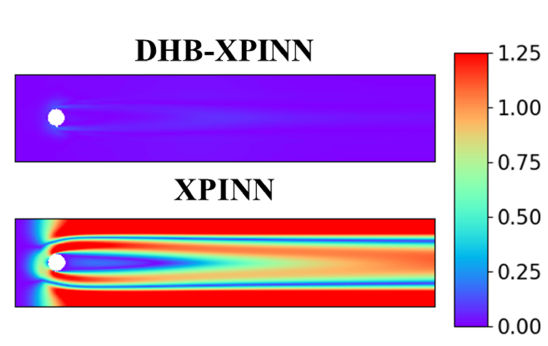


Fig 2:Residuals compared to GT (case 1)

Case 2: Inlet $u = 0.5, v = 0$; **Outlet:** $p = 0$; Walls no-slip

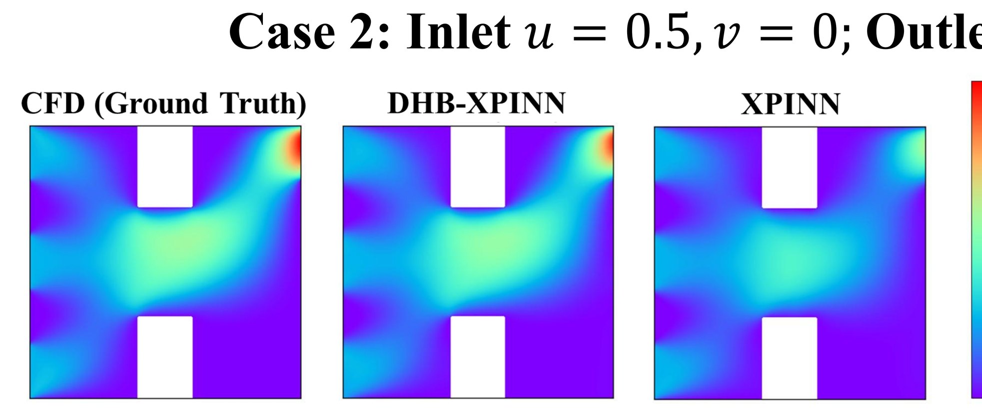


Fig 4: Velocity distributions for Case 2

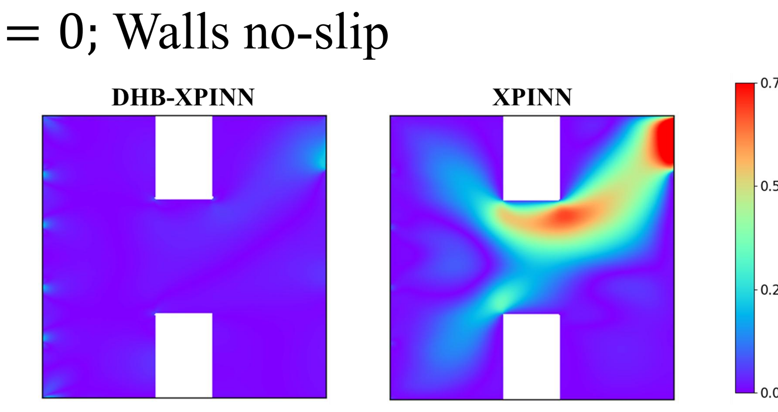


Fig 5:Residuals compared to GT (case 2)

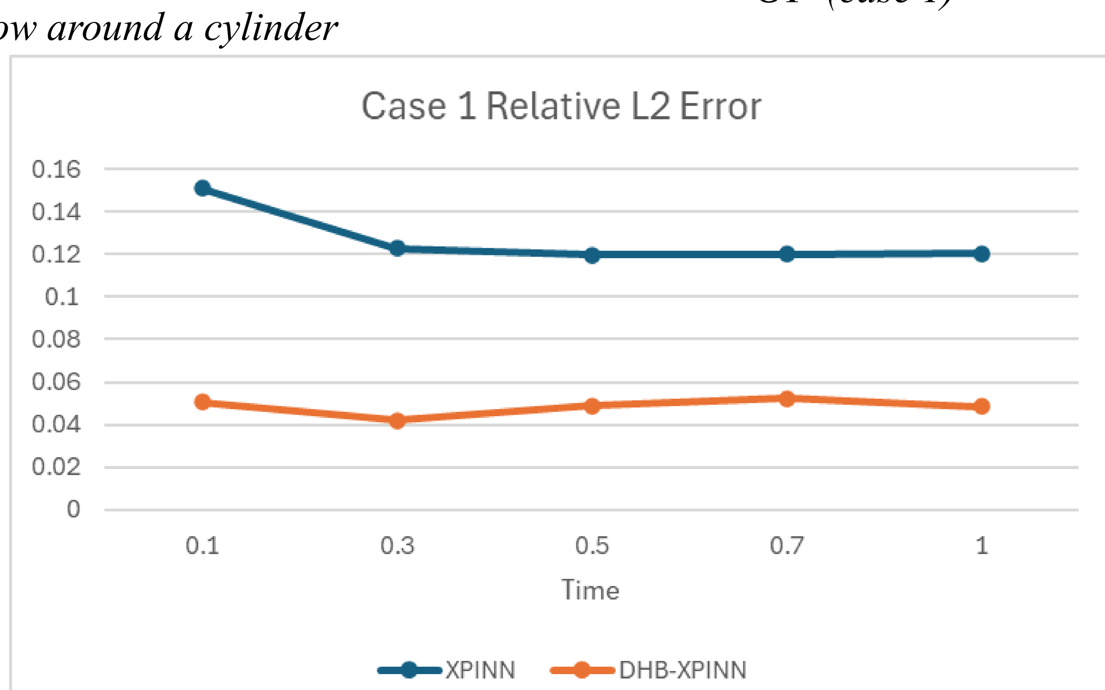


Fig 3:Case 1 Relative L2 Error

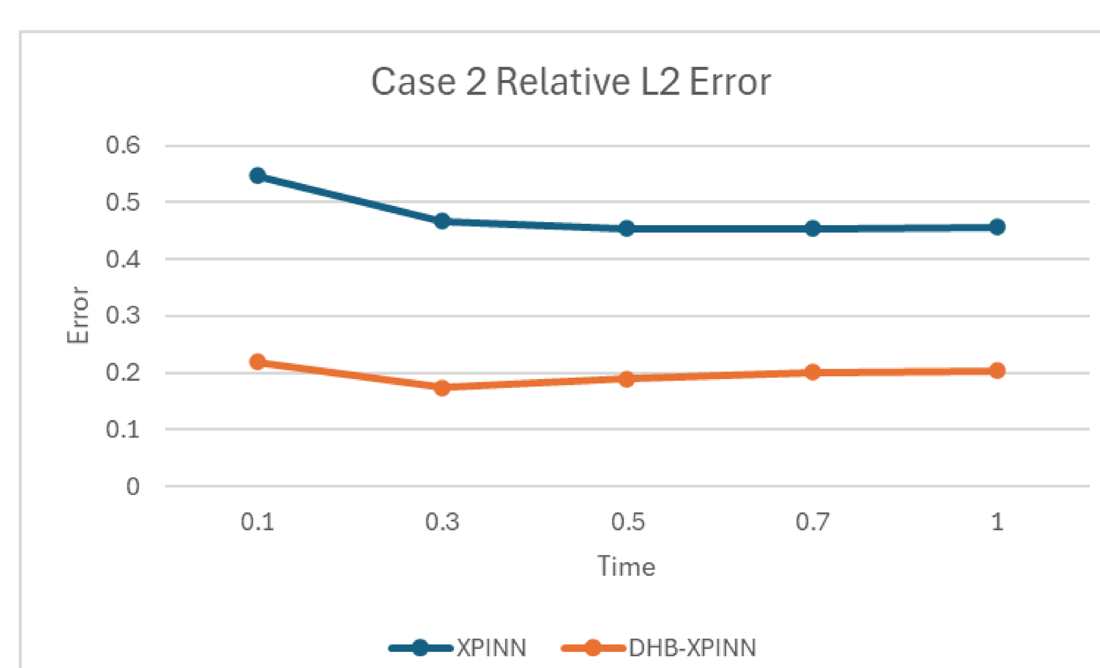


Fig 6:Case 2 Relative L2 Error

Training time:

- 2D: 2–4 h on [GPU/CPU model].
- 3D: projected 8–12 h (scaling study ongoing)

Speed up:

- Up to ~10 × vs PINN (same settings, same hardware).

Discussion

- The Hybrid Boundary form separates a boundary/IC component P from a distance-weighted correction $D \cdot H$, improving stability near boundaries.
- XPINN domain decomposition increases scalability, but accuracy depends on strong interface coupling to avoid "seams."
- Sparse boundary/interface data anchors difficult regions and improves time-to-accuracy versus purely physics-only training.
- Key sensitivities remain in loss-weight tuning, partition choice, and sampling density near sharp gradients.

Future Work

- Apply to external aerodynamics: NACA 0012 (2D) → ONERA M6 (3D).
- Extend to compressible flows and add stronger interface constraints (e.g., flux/gradient continuity).
- Use adaptive sampling/domain refinement near interfaces and high-gradient regions.
- Implement multi-GPU/node subdomain-parallel training and report scaling and wall-time results.

References

[1] M. Raissi, P. Perdikaris, and 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, vol. 378, pp. 686–707, Feb. 2019, doi: 10.1016/j.jcp.2018.10.045.

[2] A. D. Jagtap and G. E. Karniadakis, "Extended Physics-Informed Neural Networks (XPINNs): A Generalized Space-Time Domain Decomposition Based Deep Learning Framework for Nonlinear Partial Differential Equations," Communications in Computational Physics, vol. 28, no. 5, pp. 2002–2041, Nov. 2020, doi: 10.4208/cicp.OA-2020-0164.