Residual-guided AI-CFD automated framework can accelerate 3D fluid flows: A natural circulation case study

Shilaj Baral [a], Youngkyu Lee [b], Sangam Khanal [c], Sangseung Lee[d], Joongoo Jeon [a*]
[a]*Division of Advanced Nuclear Engineering, Pohang University of Science and Technology, Pohang-si, Gyeongsangbuk-do, Korea 37673*
[b]*Division of Applied Mathematics, Brown University, Providence, Rhode Island, USA 02912*
[c]*Graduate School of Integrated Energy-AI, Jeonbuk National University, Jeonju-si, Jeollabuk-do, Korea 54896*
[d]*Department of Mechanical Engineering, Inha University, Incheon-si, Michuhol-gu, Korea 22212*
[*]*Corresponding author: jgjeon41@postech.ac.kr*

Keywords : artificial intelligence, computational fluid dynamics, hybrid framework, 3D, passive-control

1. Introduction

High-fidelity computational fluid dynamics (CFD) is an indispensable tool for analyzing complex thermal-hydraulic phenomena, such as natural circulation in advanced nuclear reactor designs and passive safety systems. However, traditional CFD relies on computationally expensive spatial and temporal discretizations. For long-term 3D transient simulations, which are often required to accurately capture the slow-evolving dynamics of buoyancy-driven flows, the computational burden becomes a critical bottleneck, severely limiting iterative design optimization and real-time control applications for emerging nuclear systems like small modular reactors (SMRs) [1] and micro modular reactors (MMRs).

To circumvent these computational limits, data-driven surrogates powered by deep learning have been widely explored [2,3,4]. By leveraging GPU parallelism, these models can rapidly forecast flow fields using autoregressive predictions, where the output of one timestep becomes the input for the next. Despite their speed, purely autoregressive neural networks share a fatal flaw: they inevitably accumulate numerical errors over long rollouts, eventually drifting into non-physical states and causing catastrophic simulation failure.

Hybrid AI-CFD strategies offer a pragmatic solution by combining the rapid inference of neural networks with the rigorous physics of traditional solvers. Timestep-coupled approaches (where AI-CFD handover in each hybrid cycle happens on timesteps basis [5] rather than numerical iteration basis [6]), in particular, allow an AI model to advance the simulation while periodically falling back to a CFD solver to correct accumulated errors. However, early proof-of-concept implementations of this strategy heavily relied on manual data exchange and were mostly restricted to 2D benchmark problems, hindering their practical deployment for complex engineering tasks.

To bridge this gap, this study introduces XRePIT (eXtensible Residual-based Physics-Informed Transfer learning), a fully automated hybrid framework built around OpenFOAM [7]. XRePIT continuously monitors the mass-conservation residual of the AI's predictions; once a predefined threshold is breached, the framework automatically triggers the CFD solver to restore physical consistency and performs physics-aware corrections along with lightweight online transfer learning. In this paper, we deploy the XRePIT framework to simulate a 3D natural circulation flow with 4.98 times acceleration. We demonstrate that this residual-guided paradigm effectively prevents long-term error drift in higher dimensions, delivering substantial wall-clock acceleration without sacrificing the fidelity required for nuclear thermal-hydraulic analysis.

2. Methods and Results

This section details the computational setup, the proposed hybrid methodology, and the subsequent evaluation of its performance. We first outline the automated architecture of the eXtensible Residual-based Physics-Informed Transfer learning (XRePIT) framework. Following this, we evaluate the framework's capacity to overcome autoregressive error accumulation in baseline two-dimensional natural circulation scenarios, before ultimately demonstrating its robust scalability and acceleration capabilities in three-dimensional buoyancy-driven flow.

2.1 The XRePIT Framework Architecture

The XRePIT framework is designed to seamlessly bridge the operational gap between rapid, Python-based machine learning environments and high-fidelity, C++-based traditional numerical solvers. By adopting a timestep-coupled approach, the framework preserves the integrity of the underlying computational fluid dynamics (CFD) solver while offloading the bulk of the transient temporal progression to a neural network surrogate.

At the core of this methodology is an adaptive control loop governed by a dynamic, physics-based switching logic. During the rapid autoregressive prediction phase, the machine learning module continuously calculates a normalized mass conservation (continuity) residual directly from the newly predicted velocity fields. As illustrated in Fig. 1, if this calculated residual exceeds a strictly predefined tolerance threshold, the framework

automatically halts the neural network rollout to prevent the further drift from the true results.

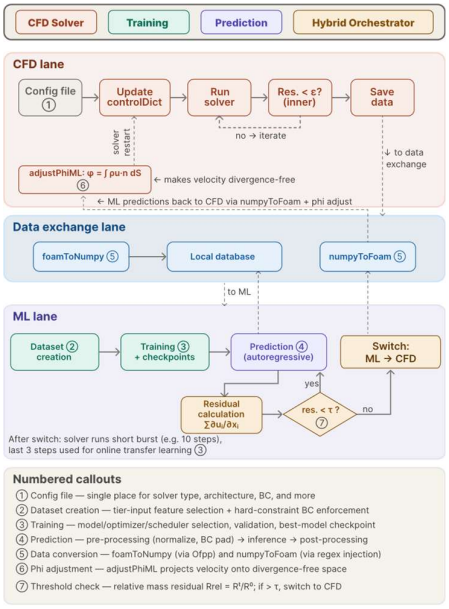

Fig. 1. Schematic of the automated XRePIT workflow showing the dataset creation, prediction logic, residual checking, and CFD correction loop.

Upon this threshold breach, the orchestrator initiates a rigorous physical correction sequence. Crucially, before the traditional CFD solver is re-invoked, *a posteriori* mass flux adjustment routine is applied to the latest machine learning predictions, projecting the velocity field onto a divergence-free space to ensure strict mass conservation. Following this vital physical consistency correction, OpenFOAM computes a short burst of high-fidelity timesteps. These newly generated ground-truth snapshots are then immediately utilized to perform online transfer learning. This targeted retraining updates the surrogate's weights, adapting it to the evolving flow dynamics before resuming the accelerated prediction phase.

2.2 Natural Circulation Testbed and Drift Stabilization

To evaluate the framework, a buoyancy-driven natural circulation flow was selected as a representative testbed, simulating phenomena critical to passive heat removal systems. The baseline two-dimensional configuration utilized a 200×200 computational mesh with adiabatic top and bottom boundaries, while the side walls were subjected to a temperature differential to drive the convective loop.

The primary limitation of standalone, data-driven surrogates is their susceptibility to catastrophic failure during extended autoregressive forecasting because of error accumulation. When a neural network is trained offline and deployed without physical constraints (denoted as "Single Training") numerical errors accumulate exponentially. As evidenced in Fig. 2, the velocity probe measurements at the top and bottom walls for the single training model diverge completely from the physical bounds within the first few hundred timesteps, leading to a total breakdown of the simulated flow field. In contrast, the physically corrected automated hybrid loop achieved through XRePIT, successfully contains this error growth. By enforcing on-demand CFD corrections triggered by the residual threshold, the framework continuously re-grounds the surrogate in rigorous physics. The probe data in Fig. 2 clearly demonstrates that the XRePIT-guided predictions remain tightly bounded. The hybrid model accurately tracks the cyclical velocity fluctuations for over 10,000 timesteps without exhibiting the non-physical drift inherent to purely data-driven methods.

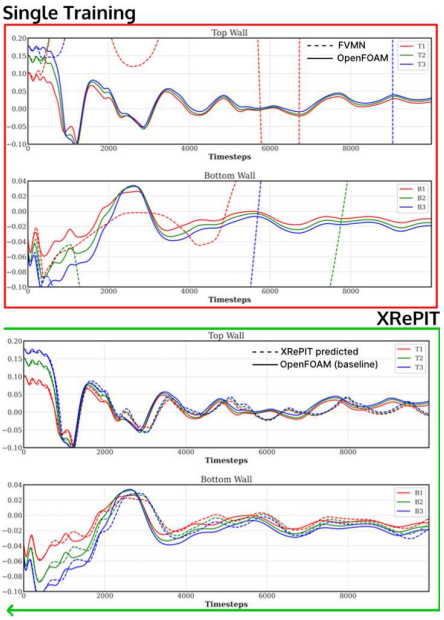

Fig. 2. Comparison of velocity probe measurements at the top and bottom walls between a diverging Single Training model and the stable XRePIT framework.

2.3 Tunable performance and boundary-condition generalization

The operational efficiency of the timestep-coupled hybrid method is governed by a fundamental trade-off between simulation speed and physical fidelity. This balance is controlled by two key hyperparameters within the XRePIT logic: the relative residual threshold, which dictates the frequency of CFD solver interventions, and the number of transfer-learning epochs, which determines the extent of online model adaptation.

As illustrated in Fig. 3, increasing the residual threshold allows the neural network to execute longer uninterrupted rollouts, which directly boosts the acceleration factor (ψ). However, this speedup comes at the expense of a quantifiable increase in the Mean

Squared Error (MSE) for the temperature field. Furthermore, the hyperparameter analysis reveals that configuring the online update to 10 epochs provides no significant accuracy benefit over a 2-epoch configuration at higher thresholds, but incurs a substantial computational penalty. Consequently, the 2-epoch, 5-residual threshold setup is identified as the optimal balance for maximizing acceleration while maintaining strict physical fidelity.

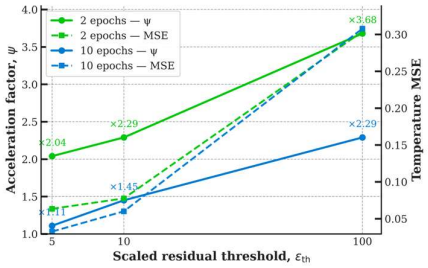

Fig. 3. The relationship between the acceleration factor, temperature MSE, and hyperparameter configurations.

Beyond tunable acceleration, the framework also demonstrates robust generalization across unseen physical setups without the need for offline retraining. The hybrid loop was deployed on two additional natural circulation cases (Case 2 and Case 3) featuring altered thermal boundary conditions at the heated and cooled walls. The network adapts to these new conditions entirely through the framework's intrinsic online transfer-learning cycles.

Fig. 4 confirms that physics-based correction through CFD fallback successfully stabilizes the rollout while the surrogate adapts online. Although a characteristic "adaptation lag" causes an initial error transient in response to the highly dynamic startup physics, the residual-guided switching mechanism forces frequent CFD corrections to realign the model weights. Consequently, the relative L_2 errors for temperature and velocity magnitude stabilize rapidly and remain bounded at minimal levels over the entire 10,000-timestep rollout.

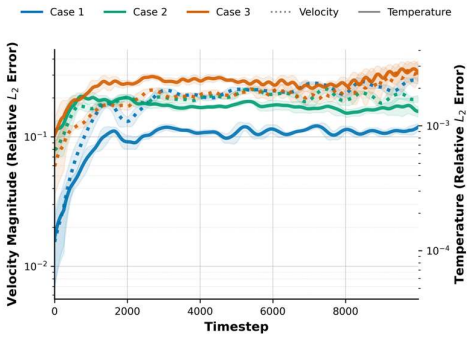

Fig. 4. Domain-wide relative L_2 error showing bounded, stable predictions across three different thermal boundary conditions: Caes 1 (blue), Case 2 (green), Case 3 (orange) for both velocity magnitude (dotted lines) and temperature (solid lines).

2.4 Extension to 3D Flow

Three-dimensional simulations play an important role in quantifying the efficacy of any CFD acceleration strategy in several fields of engineering including the nuclear engineering. Here, computational costs become prohibitively high and flow physics complexity increases dramatically. To rigorously assess the method's scalability, the XRePIT framework was extended to a 3D natural circulation case within a 1 m x 1 m x 1 m spatial domain. A uniform grid of 34 x 34 x 34 cells was utilized, providing approximately 40,000 degrees of freedom.

Quantitative analysis confirms that continuously monitoring the residual value effectively prevents error accumulation in higher dimensions. Fig. 5 demonstrates the temporal evolution of the velocity magnitude at semi-randomly selected 3D probe locations. The hybrid predictions closely track the ground truth values over time. Correspondingly, Fig. 6 illustrates the domain-wide relative L_2 error for both velocity magnitude and temperature. Similar to the 2D cases, after a brief initial adaptation transient, the errors stabilize at remarkably low values over the entire 10,000-timestep simulation, serving as a crucial indicator of long-term reliability.

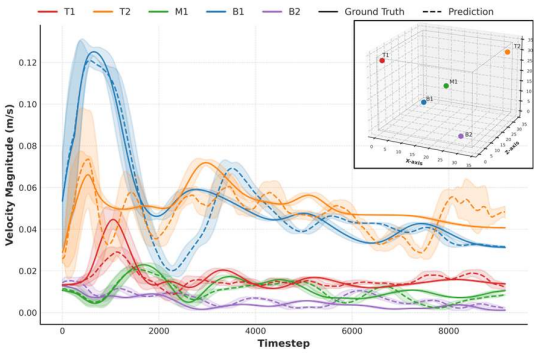

Fig. 5. Temporal evolution of velocity magnitude at 3D probe locations comparing XRePIT and OpenFOAM baseline.

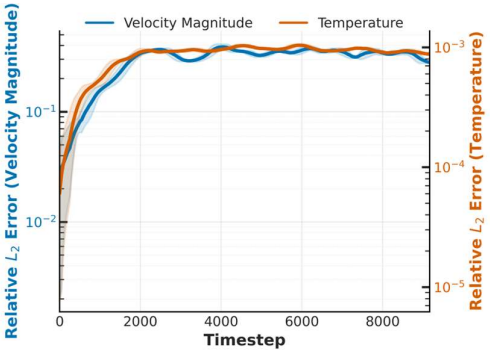

Fig. 6. Domain-wide relative L_2 error over 10,000 timesteps showing error stabilization in 3D natural circulation case too.

Beyond quantitative metrics, reproducing complex, large-scale flow structures is critical. Fig. 7 (b) presents a volumetric

rendering of the temperature field overlaid with surface-glyph velocity vectors, confirming that XRePIT accurately reconstructs the dominant macro-scale physics. The hybrid solver successfully captures the characteristic mushroom-shaped thermal plume rising from the heated wall and properly reproduces the primary recirculation zones.

Furthermore, to scrutinize the capture of shear layers, Q-criterion iso-surfaces at a value of 0.05 were computed to isolate coherent vortex cores (Fig. 7(a)). While the AI method introduces minor spatial discontinuities or "jitter" at the shear interfaces due to a lack of explicit smoothness constraints inherent to numerical differential operators, the macro-topology and orientation of the vortex cores remain highly consistent with the ground truth. Ultimately, this successful extension to 3D achieved a substantial 4.98x wall-clock speedup relative to the traditional solver. This definitive demonstration elevates the XRePIT methodology to a practical, scalable strategy for accelerating high-fidelity transient simulations

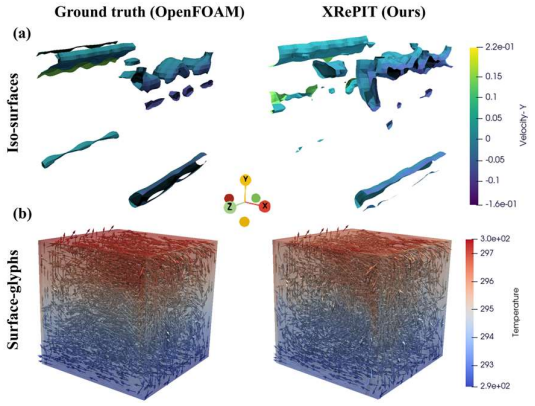

Fig. 7. 3D structural fidelity at t = 110 s. (a) Q-criterion iso-surfaces highlighting vortical cores. (b) Temperature field with surface-glyph velocity vectors illustrating agreement in the macro-scale plume structure.

3. Conclusions

This study demonstrates the efficacy of the XRePIT framework as a robust, automated methodology for accelerating multi-dimensional unsteady computational fluid dynamics (CFD) simulations. While purely autoregressive neural networks inherently suffer from error accumulation and non-physical drift during long-term rollouts, the timestep-coupled hybrid approach mitigates these issues through residual-guided, on-demand physical corrections. By systematizing the data exchange between a machine learning surrogate and the OpenFOAM solver, the framework guarantees that the simulation remains physically grounded over extended prediction horizons.

The application of this methodology to a 3D buoyancy-driven natural circulation flow validates its scalability to higher-dimensional problems characterized by complex spatial topologies. The framework successfully captured critical macro-scale flow structures, including thermal plumes and primary recirculation zones, while maintaining relative L_2 errors within $O(10^{-3})$. Crucially, the hybrid cycle achieved a substantial $4.98\times$ wall-clock acceleration relative to a standalone, single-core CFD simulation.

For the nuclear engineering sector, particularly in the development and safety analysis of SMRs relying on passive heat removal systems, computational cost remains a prohibitive bottleneck for design optimization and real-time control. The results presented here confirm that integrating residual monitoring, physics-informed switching, and lightweight online transfer learning into a single automated pipeline provides a highly reliable and computationally efficient alternative. Future work will focus on expanding this generalization capability across diverse nuclear thermal-hydraulic regimes and more complex reactor geometries.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MIST) (RS-2025-02634798) and by the National Research Council of Science & Technology (NST) grant by the Korea government (MIST) (No. GTL24031-000).

REFERENCES

[1] G. Locatelli, C. Bingham, and M. Mancini, “Small modular reactors: A comprehensive overview of their economics and strategic aspects,” *Progress in Nuclear Energy*, vol. 73, pp. 75–85, 2014.

[2] R. Vinuesa and S. L. Brunton, “Enhancing computational fluid dynamics with machine learning,” *Nature Computational Science*, vol. 2, no. 6, pp. 358–366, 2022.

[3] S. Khanal, S. Baral, and J. Jeon, “Comparison of CNN-based deep learning architectures for unsteady CFD acceleration on small datasets,” *arXiv preprint arXiv:2502.06837*, 2025.

[4] J. Jeon, J. Rabault, J. Vasanth, F. Alcántara-Ávila, S. Baral, and R. Vinuesa, “Advanced deep-reinforcement-learning methods for flow control: group-invariant and positional-encoding networks improve learning speed and quality,” *arXiv preprint arXiv:2407.17822*, 2024.

[5] J. Jeon, J. Lee, R. Vinuesa, and S. J. Kim, “Residual-based physics-informed transfer learning: A hybrid method for accelerating long-term CFD simulations via deep learning,” *International Journal of Heat and Mass Transfer*, vol. 220, p. 124900, 2024.

[6] E. Zhang *et al.*, “Blending neural operators and relaxation methods in PDE numerical solvers,” *Nature Machine Intelligence*, vol. 6, no. 11, pp. 1303–1313, 2024.

[7] H. Jasak, A. Jemcov, Z. Tukovic, and others, “OpenFOAM: A C++ library for complex physics simulations,” in *International workshop on coupled methods in numerical dynamics*, Dubrovnik, Croatia), 2007, pp. 1–20.