Preliminary Performance Comparison of CNN and GNN Based Reduced-Order Models for Large-Scale CFD Simulations

Seongmin Oh[a], Minseo Lee[b], Minseop Song [c], Bumjin Cho[c], Chaehyeon Song [c], Joongoo Jeon [b]*
[a] Department of Quantum System Engineering, Jeonbuk National University,567, Baekje-daero, Deokjin-gu, Jeonju-si, Republic of Korea
[b] Division of Advanced Nuclear Engineering, POSTECH, 77 Cheongam-ro, Nam-gu, Pohang-si, Gyeongsangbuk-do, Republic of Korea
[c] Department of Nuclear Engineering, Hanyang University, Wangsimni-ro, Seongdong-gu, Seoul, Republic of Korea
[*] Corresponding author: jgjeon41@postech.ac.kr

***Keywords :** computational fluid dynamics, reduced-order model , deep learning, small modular reactor

1. Introduction

Computational fluid dynamics (CFD) has become an essential tool for accurately analyzing complex fluid phenomena across various industrial applications. However, high-resolution CFD simulations require substantial computational resources and long execution times, which limit their applicability in tasks such as real-time prediction or design optimization. To overcome these constraints, the integration of deep learning with reduced-order models (ROMs) has emerged as a powerful strategy. [1, 3] Rather than directly learning high-dimensional CFD data which often leads to prohibitive training costs and reduced accuracy—ROMs facilitate acceleration by mapping complex flow fields into a low-dimensional latent space.

As demonstrated in recent frameworks such as Latent DeepONet (L-DeepONet) [4, 5], utilizing a latent representation allows neural operators to learn the underlying physics more efficiently than in the original high-dimensional space. By capturing the essential features of the flow via an autoencoder-based ROM, the subsequent learning process achieves superior convergence and predictive fidelity[7, 8]. This hierarchical approach—compressing data through ROMs before performing operator learning—is key to enabling practical CFD acceleration for large-scale industrial problems.

Among the deep learning architectures employed for ROM construction, the convolutional neural network (CNN) and the graph neural network (GNN) are two representative approaches[2].

However, CNN-based ROMs depend on structured grids, which limits their applicability to reactor geometries or engineering domains that are inherently represented by unstructured meshes. In addition, existing studies seldom provide a consistent comparison between structured- and unstructured-grid ROMs, leaving a gap in understanding how latent representations differ across grid formats. This gap highlights the need for models capable of handling both grid types within a unified latent-space framework.

While autoencoder-based ROMs have been widely explored in CFD applications, many existing studies have been limited to two-dimensional flow fields or relatively small-scale meshes with orders of magnitude fewer degrees of freedom. Systematic comparisons between CNN- and GNN-based architectures on three-dimensional, reactor-scale unstructured meshes comprising millions of nodes remain largely unexplored. This study addresses this gap by conducting a direct performance comparison under matched latent-space dimensionality on a full three-dimensional domain with approximately 15 million cells, providing practical insights into the scalability and feasibility of each approach for industrial-scale thermal-hydraulic analysis.

Motivated by these limitations, this study develops two complementary ROM architectures designed to extract comparable latent representations from structured and unstructured CFD fields. The proposed methodology is evaluated using high-fidelity CFD data from the entrance region of the system integrated modular advanced reactor (SMART) reactor primary loop, a location characterized by complex three-dimensional flow redistribution. The flow non-uniformity at the reactor inlet directly influences the thermal performance of the downstream helical coil steam generator (HCSG), as non-uniform velocity distributions can lead to localized thermal imbalances across the steam generator tubes. Accurate and rapid prediction of the inlet flow field is therefore essential not only for design optimization but also for safety assessment of SMR systems.

2. Methods

In this section, the model architecture, training methodology, and data preprocessing techniques used in this study are described in detail.

2.1 Numerical Methodology and Data Generation

In this study, high fidelity CFD results obtained from the entrance region of the SMART [9] reactor primary loop were used to construct and validate the ROMs. The entrance region, located downstream of the pressurizer and upstream of the helical coil steam generator (HCSG), is characterized by complex flow redistribution that significantly influences the downstream thermal hydraulic behavior.

The computational mesh was generated using SALOME and consisted of approximately 15 million unstructured tetrahedral cells. All simulations were performed using the same geometry, mesh, and thermo-

physical conditions. The k–ω SST turbulence model was adopted. The outlet boundary was set to a gauge pressure of 0 Pa, while the inlet boundary condition was defined by varying the mass flow rate from 20% to 100% of the nominal value of 396.25 kg/s, in increments of 2%. This resulted in a total of 41 steady state CFD cases, each corresponding to a different flow-rate condition. Thermophysical properties were taken from the NIST Chemistry WebBook, with density and dynamic viscosity set to 710.78 kg/m³ and 8.56×10^{-5} Pa·s, respectively.

To obtain steady-state solutions, a pseudo-time stepping strategy was applied, gradually increasing the pseudo-time step from $\Delta t = 1.0$ to 1.5, 2.0, and 3.0 to ensure stable convergence. For all operating conditions, the residuals were reduced to the order of $10^{-4} \sim 10^{-5}$.

Across this operational range, which corresponds to the full operating envelope of the SMART reactor, as the mass flow rate decreases, the Reynolds number in the entrance region is correspondingly reduced, which leads to weaker momentum-driven mixing and greater spatial non-uniformity in the velocity distribution across the cross-section. Conversely, at higher flow rates, the increased inertial forces promote more uniform flow redistribution. Although the overall flow structure remains qualitatively similar across the operating conditions, the physical quantities such as local velocity magnitude, velocity gradients, and the degree of spatial non-uniformity vary substantially with the flow rate. This continuous variation in the physical quantities across the full operational range provides the ROM with a physically meaningful and representative training dataset.

The resulting velocity and turbulence fields from the 41 cases were used as training and validation datasets for the CNN-based and GNN-based autoencoder ROMs.

2.2 Data Preprocessing

In this study, the unstructured CFD data obtained from the entrance region of the SMART reactor were preprocessed to match the input requirements of the CNN-based and GNN-based autoencoder models. The same physical variable (velocity magnitude) and identical preprocessing procedures were applied to all 41 mass-flow-rate conditions (20%–100%, in increments of 2%) to ensure a fair comparison between the two ROM approaches.

2.2.1 CNN Data Preprocessing

For the CNN-based model, the input must be provided as a structured three-dimensional grid. Therefore, the unstructured CFD fields were converted into a structured tensor using inverse distance weighting (IDW) interpolation. The power parameter of the IDW scheme was set to 3.8, and the value at each structured-grid point was computed using its six nearest neighbors (k = 6). The structured grid was generated with a resolution of 192 × 464 × 144. Since the complex geometry of the entrance region introduces void spaces where fluid does not exist, a masking procedure was applied. Based on the average spacing of the original unstructured mesh (approximately 6.3 mm), a threshold distance of t = 13 mm was used structured-grid points located more than 13 mm away from the fluid domain were treated as invalid and masked out to prevent non-physical learning.

2.2.2 GNN Data Preprocessing

For the GNN-based model, the original unstructured mesh was directly utilized by representing the CFD domain as a graph. The entrance region mesh contains approximately 5.17 million nodes, each corresponding to a cell center in the CFD mesh. The spatial coordinates of each node were stored as pos = (x, y, z), and the velocity magnitude was used as a scalar node feature. To capture local spatial connectivity, each node was connected to its 16 nearest neighbors (k = 16) using a k-nearest-neighbors algorithm. All edge connectivity information was stored in a single, shared edge-index file, while node-feature files were generated separately for each of the 41 operating conditions. Thus, the full GNN dataset consisted of 41 node-feature files and one common edge-index file.

2.3 ROM Architecture

In this study, an autoencoder-based ROM [6] was developed to efficiently compress and reconstruct high-dimensional CFD data obtained from an unstructured mesh. The autoencoder framework consists of an encoder that maps the original flow field into a low-dimensional latent space and a decoder that reconstructs the field back to its original resolution. To enable a fair comparison between the CNN-based and GNN-based ROMs, the latent-space dimensionality of both models was matched to a similar scale (approximately 1×10^5).

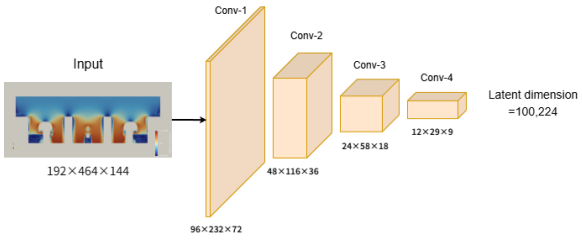

Figure 1. CNN Autoencoder Architecture

The CNN-based autoencoder takes the structured-grid representation of the CFD data as its input, as illustrated in Figure 1. The encoder consists of four down sampling stages, each employing a 3D convolution with a kernel size of 3 and a stride of 2. The number of feature channels increases progressively from 4 to 8, 16, and 32 across the four stages. After the final down sampling operation, the encoder produces a latent representation with a total dimensionality of 100,224. The decoder mirrors the encoder architecture and reconstructs the flow field to its

original spatial resolution through a sequence of transpose convolution layers.

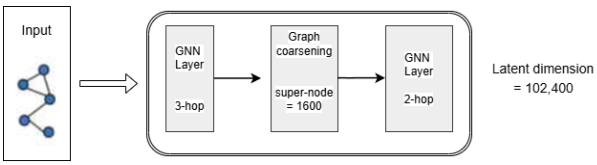

Figure 2. GNN Autoencoder Architecture

The GNN-based autoencoder directly utilizes the unstructured CFD mesh, preserving its geometric and topological characteristics, as illustrated in Figure 2. The encoder first applies 3-hop message passing based on a k-nearest neighbors scheme with k = 10, enabling each node to aggregate information from multi-level neighboring nodes. The graph is then coarsened from approximately 5.17 million nodes to 1,600 super-nodes, providing a compact representation of the domain. On the coarsened graph, an additional message-passing layer with k = 8 and a 2-hop neighborhood is applied to enhance local feature interactions.

The resulting latent representation consists of 1,600 nodes with 64 channels, yielding a latent-space dimensionality of 102,400, which is comparable to that of the CNN-based model. For decoding, the latent graph is mapped back onto the original ~5.17 million-node mesh using a distance-weighted interpolation scheme, where each original node is reconstructed using its k = 8 nearest latent neighbors. This enables the compressed super-node representation to be smoothly expanded over the full unstructured domain.

2.4 Training method

The training of both the CNN- and GNN-based autoencoder ROMs was formulated as a supervised learning problem aimed at minimizing the reconstruction error. The dataset, comprising 41 steady-state snapshots, was partitioned into 33 cases for training and 8 for validation. For both architectures, the Mean Squared Error (MSE) between the ground-truth CFD velocity fields and the reconstructed outputs served as the loss function. Optimization was performed using the Adam optimizer with a constant learning rate of 10^{-4}. All computational procedures were executed on an NVIDIA A100 GPU utilizing the PyTorch framework.

Table I: Hyperparameter Used for Training

	Epoch	**Batch size**	**Learning rate**	**Loss**
CNN	5000	41	10^{-4}	MSE
GNN	10	4096	10^{-4}	MSE

Table I summarizes the primary hyperparameters. Notably, the batch size and epoch settings differ significantly between the two models to accommodate their distinct data structures. Because the CNN-based model utilizes structured 3D tensors, the entire dataset of 41 snapshots could be processed simultaneously as a single full batch. Consequently, the model was trained for 5,000 epochs to ensure stable convergence.

In contrast, the GNN-based model operates directly on the massive unstructured mesh, which contains approximately 5.17 million nodes per snapshot. Since loading the entire graph geometry into GPU memory simultaneously is computationally prohibitive, a node-level mini-batching strategy with a batch size of 4,096 was implemented. In this node-level paradigm, a single epoch encompasses multiple iterations processing millions of nodes.

Consequently, these structural and methodological differences resulted in highly distinct computational costs. The CNN-based model required approximately 5.26 hours to complete the entire 5,000 epochs. Conversely, the GNN-based model took approximately 2 hours and 7 minutes per epoch. Due to this immense computational burden and the practical limitations of available computing resources, the maximum number of training epochs for the GNN was constrained to 10, culminating in a total training time of approximately 21.17 hours. However, this limitation does not imply that the model has reached full convergence. In this study, the epoch count was restricted purely due to computational resource constraints, and future work will analyze the convergence behavior under extended training epochs as additional resources become available. This substantial disparity in computational overhead highlights the critical challenges of processing large-scale unstructured graphs in realistic hardware environments

3. Results

To evaluate the training stability and learning dynamics of the employed architectures, the convergence trajectories of both models were analyzed. Figure 3 and Figure 4 illustrate the training and validation loss curves over the learning steps for the baseline CNN and the proposed GNN, respectively.

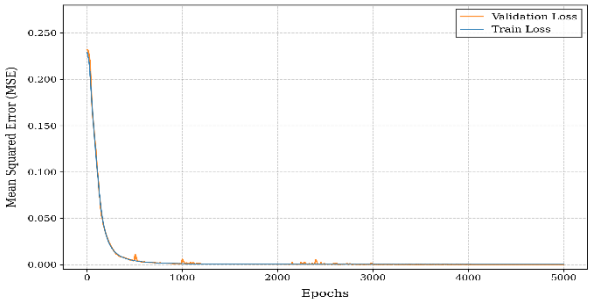

Figure 3. Convergence curves of the CNN model

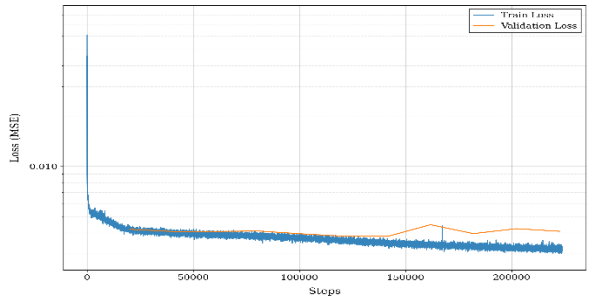

Figure 4. Convergence curves of the GNN model

The terminal loss metrics for both models are summarized in Table 2. The CNN model achieved a lower global Mean Squared Error (MSE) than the GNN model. It should be noted, however, that the CNN requires a grid-transformation process to resample unstructured CFD data into a structured format. In contrast, the GNN operates directly on the raw unstructured mesh, maintaining the original spatial connectivity without interpolation.

Table II: Summary of training and validation MSE for CNN and GNN models.

	Data	Train MSE	Val MSE
CNN	Structured Grid	2.62×10^{-4}	1.96×10^{-4}
GNN	Unstructured Mesh	3.31×10^{-3}	3.55×10^{-3}

To further evaluate the predictive performance, the reconstructed velocity fields are examined from two different perspectives. Figure 5 shows the velocity magnitude distributions on a mid-plane longitudinal slice, providing a frontal view of the internal flow. Figure 6 presents the bottom-surface view, capturing the flow redistribution near the boundaries.

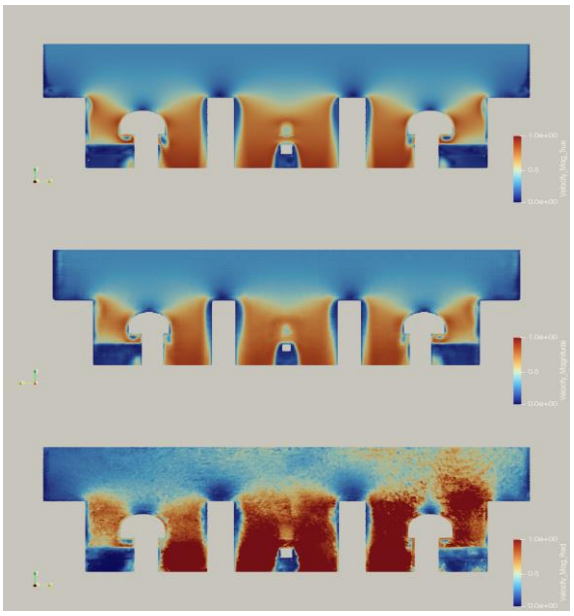

Figure 5. Comparison of velocity magnitude fields on the mid-plane.
(Top: Ground Truth Middle: CNN prediction Bottom: GNN prediction)

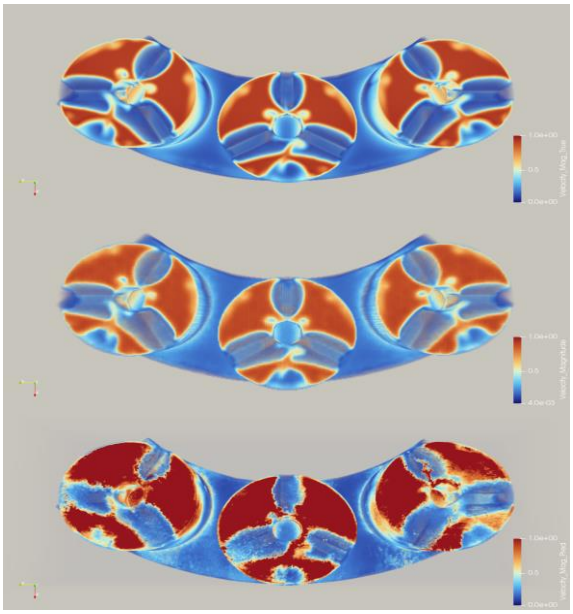

Figure 6. Comparison of velocity magnitude fields on the bottom surface.
(Top: Ground Truth Middle: CNN prediction Bottom: GNN prediction)

As observed in both Figure 5 and Figure 6, the CNN-based model demonstrates superior reconstruction fidelity, nearly identical to the Ground Truth. In the mid-plane slice (Figure 5), the CNN accurately restores the high-velocity gradients, whereas the GNN exhibits significant numerical noise and fails to capture the sharp transitions. This trend continues in the bottom view (Figure 6), where the CNN clearly outperforms the GNN in representing the complex flow patterns near the reactor primary loop's entrance. The visual evidence confirms that the CNN-based approach is significantly more robust and precise than the GNN in this specific numerical benchmark.

4. Conclusions

In this study, a comparative analysis of CNN-based and GNN-based autoencoder reduced-order models (ROMs) was conducted to compress large-scale, high-fidelity CFD data from the entrance region of the SMART reactor. To ensure a fair evaluation, both models compressed the flow fields into a latent space of comparable dimensionality across 41 steady-state snapshots. The dataset was partitioned into 33 cases for training and 8 cases for validation, ensuring a consistent basis for evaluating both ROM architectures.

The results demonstrated the overwhelming superiority of the CNN-based approach under the current hardware constraints. By converting the unstructured

mesh into a structured 3D tensor, the CNN model fully leveraged highly optimized GPU operations. It efficiently processed the entire dataset as a single batch, completing 5,000 epochs in approximately 5.26 hours. This exceptional computational efficiency allowed the CNN to achieve stable convergence and highly accurate flow field reconstructions, proving it to be a highly practical and powerful tool for large-scale CFD compression. Although explicit quantitative error metrics are not reported in this study, visual inspection of the reconstructed fields consistently showed that the CNN converged to accurate solutions during validation, whereas the GNN exhibited noticeable reconstruction deviations due to its limited number of effective training epochs.

Conversely, the GNN-based ROM faced severe computational bottlenecks. Because it directly processed the original unstructured mesh of over 5.17 million nodes, the massive graph size necessitated computationally expensive node-level mini-batching. Consequently, the training was restricted to merely 10 epochs over 21 hours. Due to this strict limitation, the GNN did not reach its convergence regime, and its reconstruction quality remained inferior to that of the CNN, as qualitatively observed in Figures 5 and 6.

Nevertheless, the GNN approach remains a highly promising methodology worth pursuing. Its intrinsic ability to perfectly preserve complex geometric and topological characteristics without the numerical smoothing and boundary loss caused by grid interpolation is a significant theoretical advantage.
Future work will extend the number of GNN training epochs and incorporate quantitative error metrics (e.g., L2 norm, MAE, and relative reconstruction error) to rigorously evaluate convergence behavior once additional computational resources become available.

Furthermore, while the present study focused on the entrance region of the SMART reactor, the developed ROM framework is intended to be extended to the full SMR primary loop geometry in future work. By progressively expanding the computational domain to encompass the complete reactor system, the proposed approach aims to enable efficient full-system thermal-hydraulic predictions that support both design optimization and real-time monitoring of SMR operations.

ACKNOWLEDGMENT

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

REFERENCES

[1] Yadav, Vaibhav, et al. "Regulatory Considerations for Nuclear Energy Applications of Digital Twin Technologies." No. INL/RPT--22-67630. Idaho National Laboratory (INL), 2022.
[2] Srinivasan, Prem A., et al. "Predictions of turbulent shear flows using deep neural networks." Physical Review Fluids 4.5 (2019): 054603.
[3] Khanal, Sangam, Shilaj Baral, and Joongoo Jeon. "Comparison of CNN-based deep learning architectures for unsteady CFD acceleration on small datasets." Nuclear Engineering and Technology 57.10 (2025): 103703.
[4] Lu, Lu, et al. "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators." Nature Machine Intelligence 3.3 (2021): 218-229.
[5] Kontolati, Katiana, et al. "Learning nonlinear operators in latent spaces for real-time predictions of complex dynamics in physical systems." Nature Communications 15.1 (2024): 5101.
[6] Bank, Dor, Noam Koenigstein, and Raja Giryes. "Autoencoders." Machine Learning for Data Science Handbook (2023): 353-374.
[7] Rahaman, Nasim, et al. "On the spectral bias of neural networks." International Conference on Machine Learning. PMLR, 2019.
[8] Wang, Bo, Lizuo Liu, and Wei Cai. "Multi-scale deeponet (mscale-deeponet) for mitigating spectral bias in learning high frequency operators of oscillatory functions." arXiv preprint arXiv:2504.10932 (2025).
[9] Kim, Keung-Koo, et al. "The first licensed advanced integral reactor." Journal of Nuclear Science and Technology 51.4 (2014): 553-565.