Proceedings of the Thirteenth International Conference on
Engineering Computational Technology
Edited by: P. Iványi, J. Kruis and B.H.V. Topping
Civil-Comp Conferences, Volume 13, Paper 1.1
Civil-Comp Press, Edinburgh, United Kingdom, 2026
ISSN: 2753-3239, doi: 10.4203/ccc.13.1.1

Propagation-Gated Graph Neural Operators for Steady-State CFD Surrogate Modelling on Unstructured Meshes

M. J. Kong[1], J. W. Jeoung[1], S. K. Yeom[1], D. H. Jung[1], T. H. Hong[1], K. B. Jeong[2] and Y. W. Hwang[2]

[1]Yonsei University, Department of Architecture and Architectural Engineering, Seoul, South Korea
[2]Sejong University, Deep Learning Architecture Research Center, Department of Architectural Engineering, Seoul, South Korea

Abstract

This study proposes a propagation-gated graph neural operator (PG-GNO) for steady-state computational fluid dynamics (CFD) surrogate modeling on unstructured meshes. The approach is motivated by the observation that steady-state CFD solutions are typically obtained through iterative fixed-point solvers converging to a stationary flow field. Instead of directly predicting the solution in a single step, PG-GNO explicitly mimics this iterative convergence process through recurrent neural operator updates. Unstructured CFD meshes are represented as graphs, and a graph neural operator is employed to achieve discretization invariance, enabling flexible application across different mesh resolutions and sensor configurations. To reconstruct global flow fields from limited inlet and sensor information, a propagation gate mechanism is introduced to regulate information transmission across the graph. This mechanism allows inlet and sensor information to be gradually propagated throughout the computational domain while suppressing contributions from uninformed regions. Validation experiments conducted on untrained inlet boundary conditions and sensor placements demonstrate that the proposed model achieves prediction accuracy comparable to that of existing CFD surrogate models. Overall, the PG-GNO framework offers a practical and generalizable surrogate modeling approach for steady-state CFD problems involving unstructured meshes and realistic, sparse measurements.

Keywords: computational fluid dynamics, graph neural operator, surrogate model, flow field reconstruction, sparse measurement, propagation gate

1 Introduction

The quantitative prediction of climatic and environmental variables is indispensable across a spectrum of engineering domains, ranging from the pipeline flow [1], indoor environment [2] to urban air quality management [3]. Reliable estimation of the distributions of physical quantities such as temperature, humidity, wind direction, and wind speed has therefore long been a major subject of research. Consequently, various numerical attempts have been made to quantify these complex environmental variables.

Central to these numerical efforts is the advancement of computational fluid dynamics (CFD), where physics-based simulations that directly solve governing equations have been widely adopted to quantify complex environmental variables (i.e., flow field). CFD simulations have established themselves as a powerful tool capable of accurately examining airflow distributions and heat transfer across a wide range of spatial scales [4]. This capability enables quantitative comparison and evaluation of various design and operational scenarios even prior to physical experimentation or construction, thereby enhancing the reliability of engineering decision-making. Nevertheless, CFD-based analyses still suffer from structural limitations, including extremely high computational cost and sensitivity to boundary condition specification [5]. In particular, applications that demand both rapid response and high accuracy over large spatial domains face intrinsic constraints when relying solely on conventional CFD approaches.

Against this background, rapidly advancing artificial intelligence (AI) and data-driven modeling techniques have emerged as promising alternatives capable of complementing or partially replacing CFD simulations. Machine learning (ML)-based CFD surrogate models offer the potential to dramatically reduce computational cost by approximating repeated numerical computations with data-driven operations [6]. Vinuesa and Brunton summarized how ML is used to speed up CFD through data-driven surrogates, improved turbulence modeling, and hybrid ML-CFD strategies [7]. They argued that near-term impact is most likely from methods that complement solvers rather than fully replacing them. Wei and Ooka demonstrated that physics-informed training can improve indoor airflow field reconstruction and enhance robustness under limited data [8]. Their results suggest adding physical constraints can reduce non-physical predictions in data-driven surrogates. Shao et al. proposed a physics-informed graph neural network (GNN) to rapidly predict urban wind fields on unstructured meshes. They reported stable, scalable field predictions while enforcing physical constraints during learning [9]. Jeon et al. presented a residual-based physics-informed transfer learning framework that alternates ML predictions with CFD to control long-horizon error growth. Their results support hybrid ML-CFD loops as a practical route to faster yet reliable simulations [10].

However, existing ML-based CFD surrogate models still exhibit several important limitations. First, convolutional neural network (CNN)-based grid-structured models inherently assume regular grids, and therefore interpolation is required when dealing with complex geometries represented by unstructured meshes, which may lead to the loss of local flow characteristics. Second, although GNNs can naturally represent unstructured meshes, their computational cost increase rapidly with graph size, making it difficult to scale to large three-dimensional problems [11]. Third, deep learning-based models generally do not guarantee discretization invariance; since they depend on specific sampling locations, sensors must remain fixed, limiting applicability in realistic scenarios where sensors may be added, removed, or repositioned [12,13]. Furthermore, most existing studies focus on predicting the next time-step flow field in transient CFD simulations [14,15], whereas relatively few studies directly address steady-state CFD analysis.

Steady-state CFD simulations typically solve nonlinear residual systems through iterative schemes (e.g., pressure-velocity coupling) [16], which can be interpreted as fixed-point iterations converging to a steady solution. Additionally, CFD simulation can be viewed as a mapping from a geometry representation and its associated boundary conditions to a resulting climate field.

Motivated by this perspective, this study introduces the propagation gate (PG) concept as a learning framework that explicitly approximates the fixed-point iterative structure. To simultaneously address challenges associated with unstructured meshes and discretization invariance, a graph neural operator (GNO) is adopted as the underlying representation model that maps input fields directly to output fields [17]. In particular, only the geometry and inlet conditions are provided as initial inputs, and through repeated application of the neural operator, inlet information is progressively propagated throughout the computational domain until convergence to a steady-state solution is achieved. The proposed PG-GNO model is designed to approximate both the physical process of flow information transport and the numerical convergence process inherent in steady-state CFD simulations.

2 Materials and Methodology

2.1 Dataset

In this study, the CFD dataset was generated using a steady-state finite volume method solver (i.e., ANSYS Fluent) applied to a fixed T-junction pipe geometry with an unstructured mesh (see Figure 1). The fixed T-junction pipe geometry comprises a main pipe and a branch pipe with different diameters. The main pipe has a diameter of 0.153 m at the large inlet, while the branch pipe has a diameter of 0.102 m at the small inlet. The main pipe spans 0.711 m along the x-direction, and the branch pipe extends 0.451 m along the z-direction. The mesh topology is identical across all cases, comprising a number of control volumes (i.e., total 12,537 control volumes). Variability was introduced solely through the inlet boundary conditions, specifically designed to change systematically within the x-z plane (u_{in}, w_{in}) while the vertical component (v_{in}) remained zero across all cases. Specifically, each case was created by combining four large inlet boundary conditions (1, 2, 3, 4 m/s) and five small inlet boundary conditions (1, 2, 3, 4, 5 m/s).

For each simulation case except one for the validation purpose, 20 distinct samples were created by randomly selecting sparse sensor locations, with the number of sensors uniformly sampled from the range of 3 to 5. Consequently, samples belonging to the same case share identical ground-truth flow fields and differ only in the spatial configuration of the sensors. The final dataset consists of 19 cases, yielding a total of 380 samples utilized in the experiments.

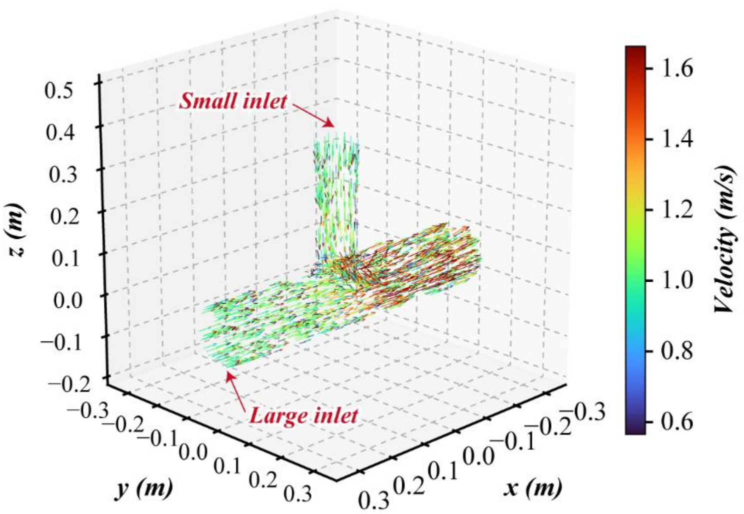

Figure 1: Visualization of the T-junction pipe geometry and unstructured mesh used in the CFD simulations

2.2 Data Preprocessing

In this study, to utilize CFD dataset for training a surrogate model (i.e., PG-GNO), a preprocessing pipeline was established to convert unstructured mesh data into a graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The preprocessing pipeline follows four steps: (i) node definition, (ii) edge construction, (iii) feature engineering, and (iv) ground truth assignment (see Figure 2).

For node definition, each control volume is defined as a node ($v \in \mathcal{V}$) in the graph, and its spatial coordinates are derived. Given the irregular shapes and sizes of control volumes inherent in unstructured meshes, the cell center coordinates were calculated by arithmetically averaging the centroids of all faces constituting the cell to ensure the spatial representativeness of each node. These three-dimensional coordinates serve as the reference points for defining graph connectivity and extracting physical features described subsequently.

For edge construction, k-nearest neighbors (k-NN) algorithm based on spatial coordinates was introduced to generate edges ($e_{ij} \in \mathcal{E}$) to effectively aggregate local flow information in complex geometries. For each node, directed edges were formed by identifying the k-nearest neighboring nodes based on Euclidean distance. In this study, k was set to 24 to secure a sufficient receptive field.

For feature engineering, node features with 16 channels and edge features with 4 channels were designed. Specifically, the node features are categorized into three distinct groups: (i) geometric features (8 channels), (ii) boundary condition features (4 channels), and (iii) sparse sensor features (4 channels). First, regarding geometric features, to facilitate the learning of complex flow geometries and boundary layer effects, the input vector included three-dimensional coordinates (x, y, z), a binary wall adjacency mask, the shortest distance to the nearest solid boundary, and the surface normal vector components (n_x, n_y, n_z). These features enable the model to recognize spatial orientation relative to boundaries, facilitating the prediction of near-wall flow characteristics. Second, regarding boundary condition features, an inlet region indicator was assigned to relevant nodes. Furthermore, the prescribed inlet velocity vector components (u_{in}, v_{in}, w_{in}) derived from the simulation boundary conditions were explicitly mapped to these nodes. For nodes located outside the inlet region, these components are set to zero, effectively distinguishing the active inflow boundary from the rest of the domain. Third, regarding sparse sensor features, a binary sensor indicator was assigned and the corresponding measured velocity vector components $(u_{obs}, v_{obs}, w_{obs})$ at sampled locations. These sparse inputs function as internal constraints (or anchor points), allowing the model to propagate ground-truth information from specific locations to the entire domain, thereby facilitating the accurate reconstruction of the global flow field from limited observations. Simultaneously, regarding edge features, to encode the spatial topology of the unstructured mesh, a 4-channel feature vector was constructed for each connected edge. This vector comprises the relative position vector components $(\Delta x, \Delta y, \Delta z)$ and the Euclidean distance between the nodes. By incorporating these geometric relations, the model can learn spatial directionality, enabling the adaptive modulation of message-passing weights to accommodate the irregularities of the non-uniform mesh.

For ground truth assignment, the steady-state velocity field obtained from the converged CFD simulations was utilized as the target variable. Specifically, the three-dimensional velocity vector components (u, v, w) at each cell center were extracted and explicitly assigned to the corresponding graph nodes.

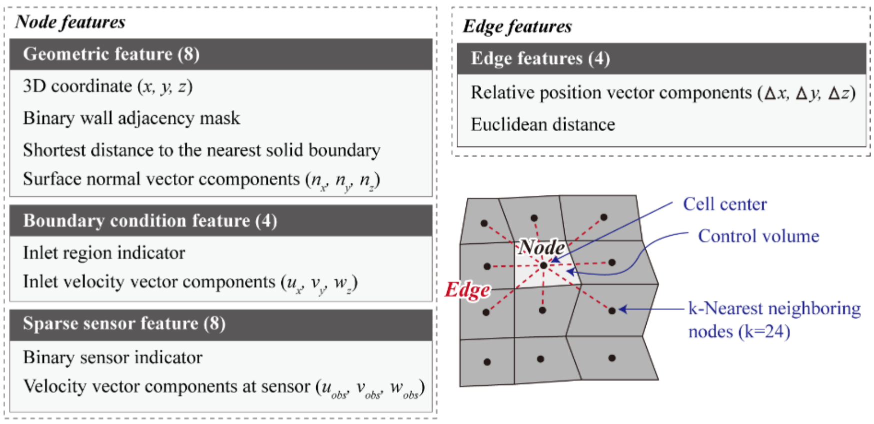

Figure 2: Illustration of the graph construction process, highlighting node and edge feature definitions

2.3 Propagation-Gated Graph Neural Operator

In this study, to reconstruct global flow fields from sparse sensor observations, the PG-GNO is proposed. Unlike conventional GNNs that propagate information uniformly across all edges, PG-GNO models the physical transport of flow information. The architecture consists of three components: (i) a probabilistic propagation module that infers valid information pathways, (ii) a gated message passing mechanism that filters noise from uninformative regions, and (iii) an implicit fixed-point solver that iteratively refines the solution to reach a steady state (see Figure 3).

Firstly, to effectively propagate sparse sensor data across an unstructured mesh, the probabilistic propagation module determines which edges serve as valid pathways for information transmission. For this purpose, a learnable edge transmission probability π_{ij} is introduced, which represents the likelihood of flow information traveling from node j to node i. This probability is predicted by a multi-layer perceptron (MLP) taking node features ($\mathbf{a_i}, \mathbf{a_j}$) and edge features ($\mathbf{e_{ij}}$) as inputs (see Equation (1)).

$$\pi_{ij} = \sigma\left(\mathrm{MLP}_{\pi}\left(a_i, a_j, e_{ij}\right)\right) \tag{1}$$

where σ is the sigmoid activation function. Using π_{ij}, the information availability probability $p_i^{(t)}$ at each node is updated iteratively. The sensors serve as the source of information ($p = 1$), and this certainty propagates through the graph according to the following probabilistic update rule (see Equation (2)).

$$p_i^{(t+1)} = 1 - \prod_{j\in\mathcal{N}(i)}\left(1 - \beta \cdot \pi_{ji} \cdot p_j^{(t)}\right) \tag{2}$$

Here, $\beta \in (0,1]$ is a decay factor that controls the range of propagation. This mechanism mimics physical diffusion, creating a time-evolving probability map that highlights regions where valid flow information has reached.

Subsequently, the inferred transmission probability is utilized as a soft gate within the gated message passing mechanism. A gated GNO layer is employed where the aggregation of messages is modulated by an edge gate g_{ij}. The gate is defined proportional to the source node's probability ($g_{ij} \approx p_j$), ensuring that nodes only aggregate messages from neighbors that possess valid information. The node update rule at layer l is defined as Equation (3) and Equation (4).

$$m_i^{(l)} = \sum_{j\in\mathcal{N}(i)} g_{ij}^{(t)} \cdot \phi\left(h_i^{(l)}, h_j^{(l)}, e_{ij}\right) \tag{3}$$

$$h_i^{(l+1)} = \mathrm{LayerNorm}\left(h_i^{(l)} + \psi\left(h_i^{(l)}, m_i^{(l)}\right)\right) \tag{4}$$

where ϕ and ψ are message and update MLPs, respectively. This gating mechanism effectively blocks noise propagation from uncorrelated (uninformed) regions, stabilizing the learning process on sparse inputs.

Finally, the implicit fixed-point solver is adopted to address the fundamental nature of steady-state CFD simulations, which aim to find a solution where physical residuals vanish. Inspired by this, PG-GNO utilizes a recurrent fixed-point iteration scheme rather than a direct feed-forward mapping. Starting from an initial state $u^{(0)}$ (initialized to zero or boundary conditions), the model iteratively updates the latent flow field u using the gated GNO layer: (see Equation (5)).

$$u^{(t+1)} = u^{(t)} + \alpha \cdot \Delta u^{(t)} \tag{5}$$

where α is the step size. In this study, the solver runs for a fixed number of iterations (e.g., $T = 35$). This recurrent formulation allows the model to progressively refine the flow field predictions, mimicking the convergence process of numerical solvers and ensuring physical consistency across the domain.

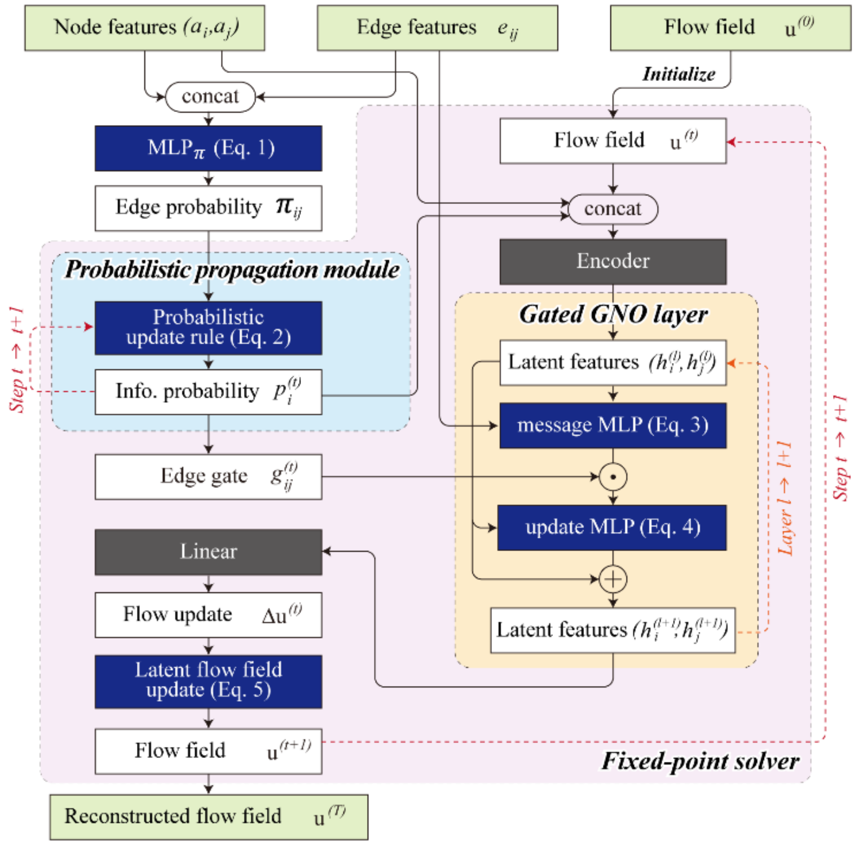

Figure 3: Model architecture of PG-GNO

2.4 Loss function

To train the proposed PG-GNO, a composite objective function is employed that combines data-driven supervision with physical constraints. The total loss is defined as a weighted sum of the (i) reconstruction loss, (ii) boundary condition loss, and (iii) physics-informed regularization loss.

First, the reconstruction loss ensures that the predicted flow field matches the ground truth. It consists of a global mean squared error (MSE) $\mathcal{L}_{recon}^{global}$ over the entire domain and a localized MSE $\mathcal{L}_{recon}^{sensor}$ specifically at the sensor locations. To enforce strict adherence to the sparse observations, a higher weight is assigned to the sensor loss compared to the global loss (see Equation (6)).

Second, physics-informed regularization is incorporated to improve the validity of the solution. This includes a divergence matching loss $\mathcal{L}_{phys}^{div}$ to impose mass conservation and a smoothness loss $\mathcal{L}_{phys}^{smooth}$ to penalize abrupt non-physical variations between neighboring nodes (see Equation (7)).

Third, boundary condition losses are applied to enforce consistency at the domain boundaries. These comprise an inlet matching loss $\mathcal{L}_{BC}^{inlet}$ for the prescribed velocity profile and a wall no-penetration loss $\mathcal{L}_{BC}^{wall}$ that penalizes velocity components normal to the solid boundaries (see Equation (8)).

Finally, a curriculum learning strategy is adopted for the physical and boundary constraints. While the data reconstruction weights are fixed, the weights for physical constraints are initialized at zero and gradually ramped up to their maximum values. Consequently, the final objective function minimized at the end of the training process is expressed as Equation (9):

$$\mathcal{L}_{recon} = 10 \cdot \mathcal{L}_{recon}^{global} + 15 \cdot \mathcal{L}_{recon}^{sensor} \tag{6}$$
$$\mathcal{L}_{phys} = 0.005 \cdot \mathcal{L}_{phys}^{div} + 0.02 \cdot \mathcal{L}_{phys}^{smooth} \tag{7}$$
$$\mathcal{L}_{BC} = 0.05 \cdot \mathcal{L}_{BC}^{inlet} + 0.05 \cdot \mathcal{L}_{BC}^{wall} \tag{8}$$
$$\mathcal{L}_{total} = \mathcal{L}_{recon} + \mathcal{L}_{phys} + \mathcal{L}_{BC} \tag{9}$$

Notably, the specific weighting coefficients assigned to each loss term were determined empirically through preliminary experiments to ensure a balanced optimization between data fidelity and physical consistency.

2.5 Evaluation methods

The evaluation metrics differ between the velocity magnitude (i.e., speed) and the velocity components (i.e., u, v, w). Since the velocity magnitude allows for direct comparison of physical quantities, the evaluation is conducted using root mean square error (RMSE), normalized mean square error (NMSE), fraction of the prediction within a factor of two of the observations (FAC2), and fractional bias (FB). For the individual velocity components, the evaluation is performed using RMSE and R^2 (see Equation (10), Equation (11), Equation (12), Equation (13), and Equation (14)).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_i - O_i)^2} \tag{10}$$

$$NMSE = \frac{\frac{1}{N}\sum_{i=1}^{N}(P_i - O_i)^2}{\left(\frac{1}{N}\sum_{i=1}^{N} O_i\right)\left(\frac{1}{N}\sum_{i=1}^{N} P_i\right)} \tag{11}$$

$$FAC2 = \frac{1}{N}\sum_{i=1}^{N} n_i \text{ with } n_i = \begin{cases} 1, \text{ for } 0.5 \leq \frac{P_i}{O_i} \leq 2.0 \\ 0, \text{ else} \end{cases} \tag{12}$$

$$FB = \frac{[O]-[P]}{0.5([O]+[P])} \tag{13}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(O_i - P_i)^2}{\sum_{i=1}^{N}(O_i - [O])^2} \tag{14}$$

Here, P_i stands for predicted value of node i; O_i stands for observed value of node i.

3 Results

In this study, the PG-GNO model was trained using 380 samples with different inlet conditions and sensor locations. The results of model training and validation are presented as follows.

3.1 Training results

To examine the prediction performance within the inlet conditions learned by the model, one sample was randomly selected for each of 19 cases (from case 002 to case 020), and the predicted values were compared with the observed values. Figure 4 presents a visual comparison between the predicted and observed values by illustrating the wind speed and direction on the x-z plane using quiver plots for three samples (004-010, 008-013, 016-010) out of the 19 samples.

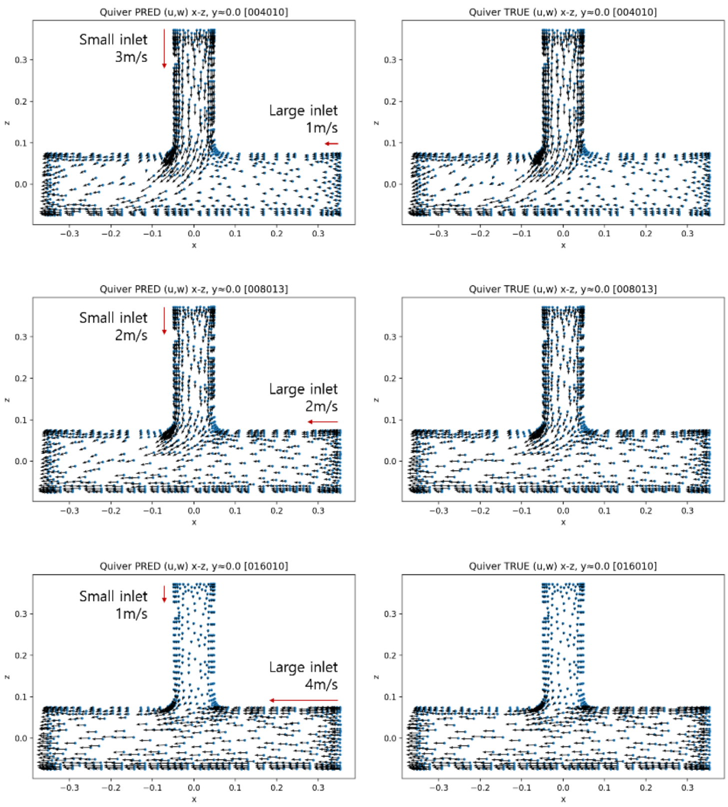

Figure 4: Comparison between predicted and observed value for test samples

Table 1 summarizes the overall prediction performance of the velocity magnitude for the three test samples shown in Figure 4 as well as for all 19 test samples. For the overall samples, the evaluation metrics were calculated as RMSE = 0.200, NMSE = 0.0061, FAC2 = 0.9986, and FB = 0.006. The large inlet boundary condition (u_{in}) and small inlet boundary condition (w_{in}) of test sample 008-010 are both 2 m/s, which can be regarded as close to the center of all cases in terms of boundary conditions. Consistently, the evaluation metrics for test sample 008-010 are observed to be similar to those of the overall samples.

Test sample	RMSE	NMSE	FAC2	FB
004-010	0.179	0.0084	0.9985	0.031
008-013	0.179	0.0064	0.9985	0.007
016-010	0.224	0.0048	0.9978	-0.021
Overall	0.200	0.0061	0.9986	0.006

Table 1: Evaluation metrics for velocity magnitude on trained model.

3.2 Validation results

This study aims to examine the model performance under inlet boundary conditions that were not included in the training process. To this end, case 001, which was excluded from training among the 20 cases, was used to construct three validation samples with different sensor locations (001-001, 001-002, and 001-003), and validation was subsequently performed.

To compare prediction performance on trained inlet boundary conditions and untrained inlet boundary conditions, RMSE, NMSE, FAC2, and FB for the velocity magnitude on each validation sample were computed. Also, the prediction performance of the proposed model was compared with existing models. Subsequently, to assess the prediction performance of the proposed model for each velocity component (u, v, w), RMSE and R^2 were calculated.

3.2.1 Velocity magnitude evaluation results

Figure 5 presents a visual comparison between the predicted and observed values for the three validation samples by visualizing the wind speed and direction on the x–z plane using quiver plots.

Table 2 summarizes the wind speed performance under new, untrained inlet conditions and sensor locations. The RMSE values were calculated as 0.362, 0.410, and 0.308, respectively. Compared to the RMSE of 0.200 obtained under the trained conditions, it is observed that the RMSE under the untrained conditions increased. In contrast, the NMSE under the untrained conditions ranged from 0.0054 to 0.0099, and the FAC2 ranged from 0.9986 to 0.9989, which are nearly identical to the NMSE and FAC2 values obtained under the trained conditions. This indicates a very high level of agreement between the predicted and observed values. The FB values were calculated in the range of -0.020 to -0.051. Compared to the near-zero FB observed under the trained conditions, this indicates the presence of a small negative bias under the untrained conditions.

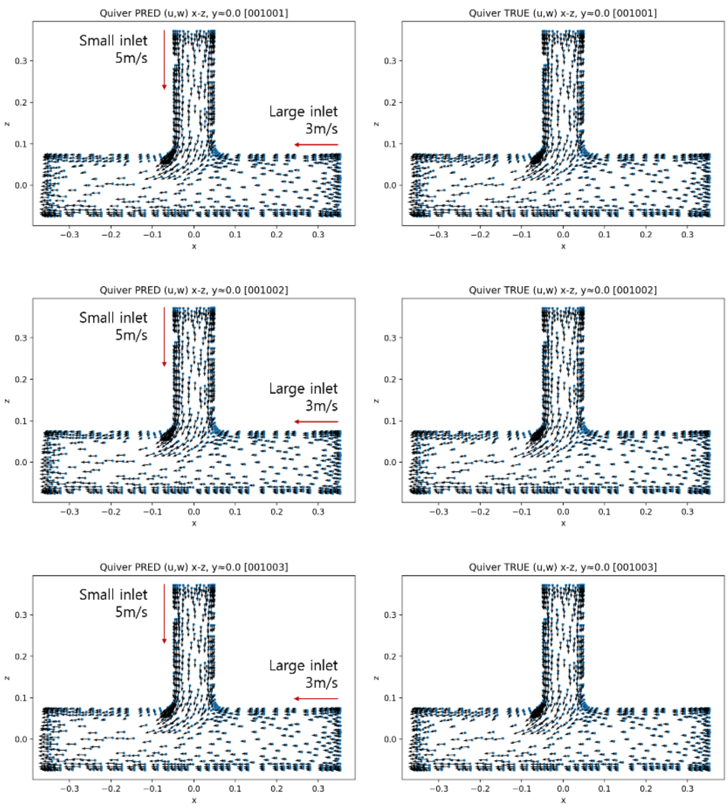

Figure 5: Comparison between predicted and observed value for validation samples

The performance of models proposed in previous studies has been reported at levels of NMSE = 0.016, FAC2 = 0.97, FB = 0.016 [18]; relative error = 2%, R2 = 0.9959 [14]; relative error = 5% [19]; and MAPE = 4.7% [20]. Compared with the PG-GNO model proposed in this study, the FB and some R^2 metrics are inferior, whereas other metrics such as NMSE and FAC2 are improved. Unlike existing models, the proposed PG-GNO model ensures discretization invariance and enables the use of realistic sensor data. Considering these aspects, the proposed PG-GNO model can be regarded as exhibiting higher practicality and generalization potential for real-world engineering applications.

Validation sample	RMSE	NMSE	FAC2	FB
001-001	0.362	0.0077	0.9989	-0.049
001-002	0.410	0.0099	0.9986	-0.051
001-003	0.308	0.0054	0.9988	-0.020

Table 2: Evaluation metrics for velocity magnitude on validation case.

3.2.2 Velocity component evaluation results

Table 3 summarizes the performance for the individual velocity components (u, v, w). The u and w component exhibit high R^2 values ranging from 0.96 to 0.99, while the v component, although relatively lower, still demonstrates stable generalization performance with R^2 values between 0.85 and 0.90. In particular, the w component shows the best and most consistent performance across all cases, with R^2 values ranging from 0.991 to 0.993. This can be attributed to the fact that the wind enters in the u direction through the large inlet and in the w direction through the small inlet, whereas the prediction of the v components, which is generated solely by the inflow conditions and the geometry associated with the u and w directions, is relatively more challenging.

Validation sample	RMSE			R^2		
	u	v	w	u	v	w
001-001	0.336	0.156	0.189	0.969	0.895	0.993
001-002	0.381	0.187	0.215	0.960	0.849	0.991
001-003	0.285	0.179	0.214	0.978	0.863	0.991

Table 3: Evaluation metrics for velocity components on validation case.

4 Conclusions and Contributions

This study presented a propagation-gated graph neural operator (PG-GNO) framework for steady-state CFD surrogate modeling. The proposed approach is motivated by the observation that steady CFD simulations are commonly solved through iterative numerical procedures that converge to a stationary solution. Rather than directly regressing flow fields in a single step, the PG-GNO model adopts an iterative update structure, in which a neural operator is repeatedly applied until a stable flow field is obtained.

The PG-GNO framework operates on unstructured meshes by employing a Graph Neural Operator as its core representation, thereby avoiding reliance on mesh-specific discretization schemes. Given geometric information and inlet boundary conditions, the model iteratively updates the flow field while a propagation gate regulates the spatial influence of inlet information. This mechanism provides a compact way to control how boundary information is distributed across the computational domain during the iterative process.

Comprehensive evaluations showed that the PG-GNO model achieves high predictive accuracy under trained inlet conditions and retains strong generalization performance when validated on previously unseen, i.e., untrained inlet conditions and sensor locations. Although an increase in RMSE was observed in untrained condition, normalized error metrics and agreement factors remained comparable to those obtained under trained conditions, indicating stable relative accuracy.

Compared with existing data-driven CFD surrogate models, the proposed PG-GNO framework demonstrates competitive or improved performance in key evaluation metrics such as NMSE and FAC2. More importantly, the model is inherently discretization-invariant and can naturally accommodate changes in sensor placement and density.

Future work will focus on further aligning the PG-GNO framework with physical modeling principles. In particular, additional fluid-dynamic variables such as pressure and temperature will be incorporated, together with physics-based loss terms, to enforce stronger physical consistency. These extensions are expected to enable the development of a more comprehensive surrogate model that more closely approximates full CFD solvers while retaining the computational efficiency of neural operator-based approaches.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MIST; Ministry of Science and ICT) (NRF-2021R1A3B1076769).

References

[1] N.M.C. Martins, N.J.G. Carriço, H.M. Ramos, D.I.C. Covas, “Velocity-distribution in pressurized pipe flow using CFD: Accuracy and mesh analysis”, Computers & Fluids, 105, 218-230, 2014, DOI:10.1016/j.compfluid.2014.09.031.

[2] J. Liu, J. Niu, “CFD simulation of the wind environment around an isolated high-rise building: An evaluation of SRANS, LES and DES models”, Building and Environment, 96, 91-106, 2016, DOI:10.1016/j.buildenv.2015.11.007.

[3] JY. Toparlar, B. Blocken, B. Maiheu, G.J.F. van Heijst, “A review on the CFD analysis of urban microclimate”, Renewable and Sustainable Energy Reviews, 80, 1613-1640, 2017, DOI:10.1016/j.rser.2017.05.248.

[4] A. Stamou, I. Katsiris, “Verification of a CFD model for indoor airflow and heat transfer”, Building and Environment, 41, 1171-1181, 2006, DOI:10.1016/j.buildenv.2005.06.029.

[5] M. Hajdukiewicz, F.J. González Gallero, P. Mannion, M.G.L.C. Loomans, M.M. Keane, “A narrative review to credible computational fluid dynamics models of naturally ventilated built environments”, Renewable and Sustainable Energy Reviews, 198, 114404, 2024, DOI:10.1016/j.rser.2024.114404.

[6] C. Caron, P. Lauret, A. Bastide, “Machine Learning to speed up Computational Fluid Dynamics engineering simulations for built environments: A review”, Building and Environment, 267, 112229, 2025, DOI:10.1016/j.buildenv.2024.112229.

[7] R. Vinuesa, S.L. Brunton, “Enhancing computational fluid dynamics with machine learning”, Nature Computational Science, 2, 358-366, 2022, DOI:10.1038/s43588-022-00264-7.

[8] C. Wei, R. Ooka, “Indoor airflow field reconstruction using physics-informed neural network”, Building and Environment, 242, 110563, 2023, DOI:10.1016/j.buildenv.2023.110563.

[9] X. Shao, Z. Liu, S. Zhang, Z. Zhao, C. Hu, "PIGNN-CFD: A physics-informed graph neural network for rapid predicting urban wind field defined on unstructured mesh", Building and Environment, 232, 110056, 2023, DOI:10.1016/j.buildenv.2023.110056.
[10] J. Jeon, J. Lee, R. Vinuesa, 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, 220, 124900, 2024, DOI:10.1016/j.ijheatmasstransfer.2023.124900.
[11] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, "Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs", Journal of Machine Learning Research, 24, 1-97, 2023.
[12] X. Liu, S. Yang, H. Sun, Z. Wang, X. Guan, Y. Gu, Y. Wang, "Review of deep learning-based aerodynamic shape surrogate models and optimization for airfoils and blade profiles", Physics of Fluids, 37, 041304, 2025, DOI:10.1063/5.0268466.
[13] B. Bahmani, S. Goswami, I.G. Kevrekidis, M.D. Shields, "A resolution independent neural operator", Computer Methods in Applied Mechanics and Engineering, 444, 118113, 2025, DOI:10.1016/j.cma.2025.118113.
[14] Z. Huang, T. Li, K. Huang, H. Ke, M. Lin, Q. Wang, "Predictions of flow and temperature fields in a T-junction based on dynamic mode decomposition and deep learning", Energy, 261, 125228, 2022, DOI:10.1016/j.energy.2022.125228.
[15] S. Wu, H. Wang, K.H. Luo, "A robust autoregressive long-term spatiotemporal forecasting framework for surrogate-based turbulent combustion modeling via deep learning", Energy and AI, 15, 100333, 2024, DOI:10.1016/j.egyai.2023.100333.
[16] H. Wang, H. Wang, F. Gao, P. Zhou, Z. (John) Zhai, "Literature review on pressure-velocity decoupling algorithms applied to built-environment CFD simulation", Building and Environment, 143, 671-678, 2018, DOI:10.1016/j.buildenv.2018.07.046.
[17] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, "Neural Operator: Graph Kernel Network for Partial Differential Equations", arXiv, 2003.03485, 2020, DOI:10.48550/arXiv.2003.03485.
[18] Z. Liu, S. Zhang, X. Shao, Z. Wu, "Accurate and efficient urban wind prediction at city-scale with memory-scalable graph neural network", Sustainable Cities and Society, 99, 104935, 2023, DOI:10.1016/j.scs.2023.104935.
[19] Y.-J. Wang, E. Baglietto, K. Shirvan, "A two-level machine learning approach for predicting thermal striping in T-junctions with upstream elbow", Numerical Heat Transfer, Part B: Fundamentals, 85, 662-682, 2024, DOI:10.1080/10407790.2023.2253362.
[20] M.R. Vandewiel, D.D. Eneyew, A.D. Awol, M.A.M. Capretz, G.T. Bitsuamlak, "Approximating CFD simulations of natural ventilation: A deep surrogate model with spatial attention mechanism", Journal of Building Engineering, 105, 112425, 2025, DOI:10.1016/j.jobe.2025.112425.