Predicting airfoil pressure distribution using boundary graph neural networks

Sankalp Jena[1,*], Gabriel D. Weymouth [1], Artur K. Lidtke [2], Andrea Coraddu [1]

[1] Delft University of Technology, The Netherlands,

[2] Maritime Research Institute Netherlands, The Netherlands

[*] s.jena@tudelft.nl

ABSTRACT

Surrogate models are essential for fast and accurate surface pressure and friction predictions during design optimization of complex lifting surfaces. This study focuses on predicting pressure distribution over two-dimensional airfoils using graph neural networks (GNNs), leveraging their ability to process non-parametric geometries. We introduce boundary graph neural networks (B-GNNs) that operate *exclusively* on surface meshes and compare these to previous work on volumetric GNNs operating on volume meshes. All of the training and evaluation is done using the `airfRANS` (Reynolds-averaged Navier-Stokes) database. We demonstrate the importance of all-to-all communication in GNNs to enforce the global incompressible flow constraint and ensure accurate predictions. We show that supplying the B-GNNs with local physics-based input-features, such as an approximate local Reynolds number Re_x and the inviscid pressure distribution from a panel method code, enables a 83% reduction of model size and 93% of training set size relative to models using purely geometric inputs to achieve the same in-distribution prediction accuracy. We investigate the generalization capabilities of the B-GNNs to out-of-distribution predictions on the `S809/27` wind turbine blade section and find that incorporating inviscid pressure distribution as a feature reduces error by up to 88% relative to purely geometry-based inputs. Finally, we find that the physics-based model reduces error by 85% compared to the state-of-the-art volumetric model INFINITY.

Keywords: Graph Neural Networks, Data-Driven Surrogate Model, Pressure Distribution, Airfoils

1 INTRODUCTION

Cavitation on ship propellers produces underwater radiated noise that adversely affects marine life (Basan et al., 2024) and erodes propeller surface (Dular and Petkovšek, 2015). Design exploration and optimization are performed to address the above concerns while striving for high performance. Scale-resolved computational fluid dynamics (CFD) simulations of cavitating propellers are computationally prohibitive for even a modest number of cases, making fast surrogate models an essential tool for design studies. While panel methods like XFOIL (Drela, 1989) can simulate flow over 2D airfoils using only a surface mesh, and similar methods can be applied to 3D lifting surfaces, these tools require empirical corrections for viscous and cavitation effects which are based on small data sets and their 3D generalization error is poorly documented. In this work, we focus on surrogate modeling methodologies with the possiblity to scale to complex 3D propeller flow starting with 2D airfoil shapes as a stepping stone on the way to address the full propeller geometry.

Although Gaussian-process regression is the most popular approach for surrogate modeling, they apply to low-dimensional and fixed parametrization of the geometry. Deep learning approaches are suitable for high-dimensional learning such as NeuralFoil (Sharpe, 2024) which employs multilayer perceptrons (MLPs) to predict pressure distribution over 2D airfoils. While NeuralFoil operates on nonparametric 2D foil mesh, it maps the geometry to a fixed shape parameterization in its pipeline, limiting it's ability to make out-of-sample predictions. Alternatively, Graph Neural Network (GNNs) are specialized neural networks designed to operate on graph-structured data (Scarselli et al., 2009; Bronstein et al., 2021). This makes them well-suited to predicting fluid flows where computational meshes are already in common use, such as Immordino et al. (2025) that uses GNNs to predict pressure distributions on a 3D aircraft. While that work and many others actually use geometric parameters as inputs, a fully nonparametric GNN enables predictions on shapes completely outside the sample distribution, as we will show in this paper.

Most GNN airfoil applications employ a *volumetric* model; predicting flow quantities, such as velocity and pressure, over the entire numerical domain. Notable examples include the U-Net convolutional neural network (CNN) architecture by Thuerey et al. (2020), a GNN-based approach by Bonnet et al. (2023), and an implicit neural fields model (INFINITY) by Serrano et al. (2023) for predicting Reynolds-averaged Navier-Stokes (RANS) flow around 2D airfoils. However, a volumetric approach not only increases the computational cost but also often fails to generalize well to out-of-distribution geometries. Therefore, in the current study, we focus exclusively on the use of a *boundary* model; as used in Durasov et al. (2023) who propose a GNN architecture for 2D airfoils inspired by iterative CFD solvers.

A clear gap in the literature is the limited work that relates the fundamentals of fluid mechanics to the architecture and input set for GNNs. For example, the governing fluid equations are elliptic due to the incompressibility constraint. This means that every point on the boundary will affect the pressure at every other point; requiring all-to-all communication in the prediction pipeline. The scaling of the network with the size of the boundary graph is therefore a serious concern, and while the iterative model by Durasov et al. (2023) proposes one solution, they do not investigate the complexity of the architecture required for accurate pressure predictions and do not consider the scaling of the architecture when extending to handle 3D geometries. In addition, since our target function is physical, the use of physics-based features (Weymouth and Yue, 2013) could greatly enhance our data independence and generalization performance. In this work, we address this gap by developing a systematic framework for boundary-graph neural networks (B-GNNs) that demonstrates the importance of all-to-all communication and provides the scaling of the B-GNNs with the size of the geometry. We show that boundary graph U-Nets have optimal all-to-all scaling and physics-based features improve generalization by 7× compared to the state of the art and enable extrapolation to completely out-of-sample geometries.

2 METHODOLOGY

2.1 Problem definition

The surrogate model operates on a discretized airfoil geometry, which is represented as a mesh containing N nodes. Each node in the mesh has spatial coordinates (x, y) and associated flow information $\mathbf{q}_i$. The input and output features at the i^{th}-node is defined as:

$$\mathbf{x}_i = [x, y, \mathbf{q}_i]^{\mathrm{T}}; \qquad \mathbf{y}_i = [c_p], \tag{1}$$

where c_p is the viscous pressure coefficient and $\mathbf{q}_i$ refers to the flow information that is provided to some of the discussed models. The airfoils are rotated by the freestream angle of attack (α), embedding the information into the geometry. This design allows the model to infer the angle of attack implicitly through the node coordinates.

The airfoil mesh can be represented as a graph, where the nodes correspond to discrete points on the airfoil surface and the edges define connectivity between them. The adjacency matrix $\mathbf{A}$ encodes the graph structure (Bronstein et al., 2021). For a two-dimensional airfoil, the connectivity follows a natural order, running from the trailing edge to the leading edge and back, forming a closed-loop structure. As a result, the airfoil mesh is a ring graph (Bronstein et al., 2021), as illustrated in Figure 1. The input node embedding matrix $\mathbf{X}$ contains the node features $\mathbf{x}_i$. A convolution layer of GCN is a function $\mathbf{F}$ with parameters $\mathbf{\Phi}$. It takes node embeddings $\mathbf{X}$ and the adjacency matrix $\mathbf{A}$ as the input to generate new node embeddings. The node embedding matrix $\mathbf{H}$ after K convolutions is given by

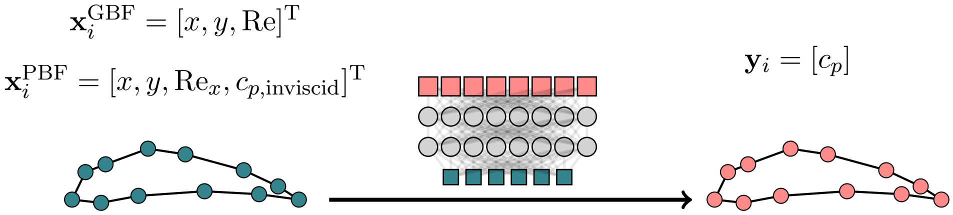

$\mathbf{x}_i^{\text{GBF}} = [x, y, \text{Re}]^{\text{T}}$
$\mathbf{x}_i^{\text{PBF}} = [x, y, \text{Re}_x, c_{p,\text{inviscid}}]^{\text{T}}$
$\mathbf{y}_i = [c_p]$

Figure 1: B-GNN framework used to learn the pressure distribution of an airfoil in steady viscous flow. i^{th} node with geometry-based feature (GBF), $\mathbf{x}_i^{\text{GBF}}$, containing node coordinates (x, y) and Reynolds number (Re), is mapped to node prediction, $\mathbf{y}_i$, with coefficient of pressure, c_p. Models with physics-based features (PBFs), $\mathbf{x}_i^{\text{PBF}}$, additionally use local Reynolds number Re_x and inviscid coefficient of pressure $c_{p,\text{inviscid}}$ as input feature.

$$\mathbf{H}_1 = \mathbf{F}[\mathbf{X}, \mathbf{A}, \mathbf{\Phi}_0] \tag{2}$$
$$\vdots$$
$$\mathbf{H}_K = \mathbf{F}[\mathbf{H}_{K-1}, \mathbf{A}, \mathbf{\Phi}_{K-1}] \tag{3}$$

Initially, each node's embedding only contains self-information. As the GCN operates, node embeddings are updated by passing messages from neighboring nodes. This is done using edge convolution (Wang et al., 2019). The node embedding at K-th convolution is given by

$$\mathbf{h}_{i,K} = \max_{j:(i,j)\in\mathcal{E}} \bar{h}_\Theta(\mathbf{x}_i, \mathbf{x}_j - \mathbf{x}_i) \tag{4}$$

where $\bar{h}_\Theta : \mathbb{R}^{|\mathbf{x}_i|} \times \mathbb{R}^{|\mathbf{x}_j|} \to \mathbb{R}^{|\mathbf{h}_{i,K}|}$ is an MLP with a set of learnable parameters Θ and exponential linear unit (ELU) (Clevert et al., 2016) activation function, and $\mathcal{E}$ is the set of edges. Messages are aggregated using a symmetric operation $\max$. The central node feature $\mathbf{x}_i$ captures the global shape information with respect to the entire graph, and local neighborhood information with respect to its neighbors accounted by the difference with neighborhood $\mathbf{x}_j - \mathbf{x}_i$.

2.2 Dataset

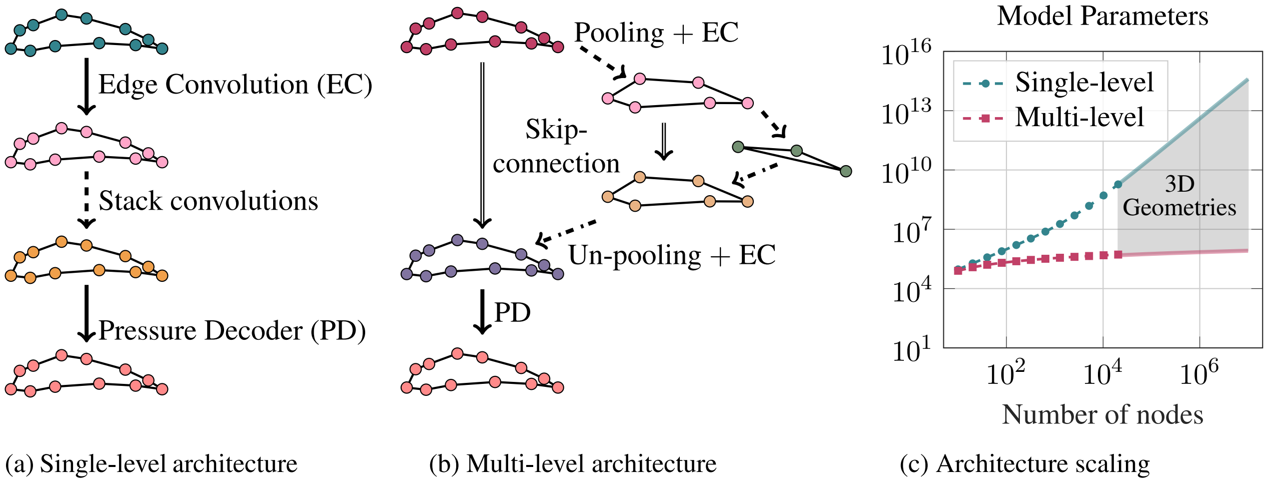
We utilize the `airfRANS` dataset (Bonnet et al., 2023), which provides steady-state turbulent flow solutions over two-dimensional `NACA4/5` airfoils computed using Reynolds-averaged Navier–Stokes equations. To construct airfoil graphs, each airfoil is discretized by random sampling between 120 and 160 nodes from the surface, with a higher density of points near the leading and trailing edges to capture steep pressure gradients accurately. Having varying numbers of nodes across airfoil graphs serves as a test of the ability of B-GNNs to deal with inconsistent geometry descriptions in the dataset. The pressure coefficient, c_p, is computed by normalizing pressure values with the dynamic pressure, $q_\infty = 0.5\rho U_\infty^2$, where ρ is the density of the fluid and U_∞ is the freestream velocity. The dataset consists of predefined 800 training samples and 200 test samples, with the training set further split into 90% for training and 10% for validation. All features and labels are normalized to the range $[0, 1]$ using min-max normalization based on the training set statistics.

2.3 Input features

The model arguments are the spatial coordinates (x, y) and the Reynolds number (Re). The geometry-based feature (GBF) is given by

$$\mathbf{x}_i^{\text{GBF}} = [x, y, \text{Re}]^{\text{T}} \tag{5}$$

Physics-based features



(a) Single-level architecture (b) Multi-level architecture (c) Architecture scaling

Figure 2: (a) Single-level B-GCN needs recursive convolutions, whereas, (b) multi-level B-GUN coarsens the input graph to capture far-node influence. (c) For 3D geometries, multi-level B-GUN is feasible while single-level B-GCN becomes infeasible.

To investigate the influence of *a priori* knowledge of the problem, B-GNNs are supplied with physics-based features (PBFs) (Weymouth and Yue, 2013) derived from potential flow solutions obtained using XFOIL (Drela, 1989). These features include the approximate local Reynolds Number with respect to the stagnation point (x_{stag}), $\text{Re}_x = (x - x_{\text{stag}})U_\infty/\nu$, where ν is the kinematic viscosity of the fluid, and the inviscid pressure coefficient, $c_{p,\text{inviscid}}$. These features encode fundamental physics, such as the growth of the boundary layer and non-viscous pressure distributions, which could enable B-GNNs models to learn viscous pressure distribution. The physics-based feature (PBF) is given by

$$\mathbf{x}_i^{\text{PBF}} = [x, y, \text{Re}_x, c_{p,\text{inviscid}}]^{\text{T}} \tag{6}$$

2.4 Model architectures

From the global incompressibility constraint, it follows that each node of the discretized airfoil geometry affects all the other nodes. Therefore, we evaluate two architectures that aggregate far-away neighborhood information with different complexities: single-level boundary graph convolutional neural network (B-GCN) and multi-level boundary graph-U-Net (B-GUN). The level refers to the number of neighborhood nodes from which messages are aggregated in one convolution layer.

Single-level B-GCN

In a B-GCN (Figure 2a), a single edge convolution K gathers information from neighborhood nodes that are 1-hop away. Recursive convolutions are applied to accumulate information from far-away nodes. Node features from every convolution layer are stacked and passed through as pressure decoder MLP to predict the viscous pressure coefficient. The number of convolutions K determines the communication range. In a ring graph with N nodes, full communication requires $K = N/2$ convolutions. Therefore, stacking node embeddings of dimension H from K layers in the pressure decoder MLP leads to quadratic growth in model size (Figure 2c), as the MLP requires $\{\,\mathcal{O}(K \cdot H), \mathcal{O}(K \cdot H), 1\}$ neurons. While this might not be expensive for 2D geometries, for 3D geometries with hundreds of thousands of nodes, the model would be inefficient. This limitation motivates the need for a more efficient communication mechanism: the multi-level B-GUN.

Multi-level B-GUN

In contrast to B-GCN, an edge convolution in B-GUN (Figure 2b) gathers information from nodes more than

1-hop away. This is achieved by coarsening the graph using a binary fusion pooling operation. In binary pooling, two nodes are merged into one, and their features are averaged. This process continues until the graph is reduced to three nodes, as at this level, a single edge convolution aggregates messages from all neighbors. To reconstruct the original graph structure, un-pooling operations are performed from the coarsened graph with three nodes. During un-pooling, the graph is upscaled to match the number of nodes at the corresponding coarse level. The node features are copied from the coarser graph and passed to the finer level with an addition operation, creating skip-connections. After $\log_2(N/3)$ un-pooling operations, the original graph structure is restored. At this point, the node embeddings (H) are passed through a pressure decoder MLP with $\{H, H, 1\}$ neurons to predict the pressure. Unlike B-GCN, the node embedding size for the pressure decoder MLP in B-GUN does not scale with N, making it computationally efficient and suitable for 3D geometries with hundreds of thousands of nodes (Figure 2c).

2.5 Model training details

Four models are analyzed based on two design choices — architecture and input features — $\{$B-GCN, B-GUN$\}$ $\times$ $\{$Geometry-based features, Physics-based features$\}$. The mean squared error [REF] between the predicted and ground truth coefficient of pressure (c_p) at each node is used as the loss function

$$\mathcal{L} := \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2 \tag{7}$$

where $\mathcal{B}$ is the set of indices of boundary nodes, $\mathbf{y}_i$ is the RANS ground truth and $\hat{\mathbf{y}}$ is the model prediction. Adam optimizer with a learning rate of 10^{-4} is employed, alongside a batch size of 32 for the training. All models are trained for 32000 epochs. From the model checkpoints, the model with minimum validation loss is used for inference. In both architectures, the key hyperparameters are from the edge convolution function viz. the node embedding size (H), edge convolution MLP width (W), and edge convolution MLP layers (L). The optimal hyperparameter combination obtained at 20 convolutions is used for the single-level model (B-GCN) and at full depth is used for the multi-level model (B-GUN). The criterion is minimizing validation loss. To reduce computational effort models are trained for 1000 epochs.

Sensitivity to training dataset size

To analyze the influence of physics-based features (PBFs) on the training data requirement, we study the scaling of generalization error with shards of the training set. Following Hestness et al. (2017), we create shards of sizes $\{50, 100, 200, 400\}$ such that training data are added to the smaller shard following a power-law. Models are trained on 5 randomly sampled shards to provide statistics. A predefined test set of 200 samples is used for evaluation.

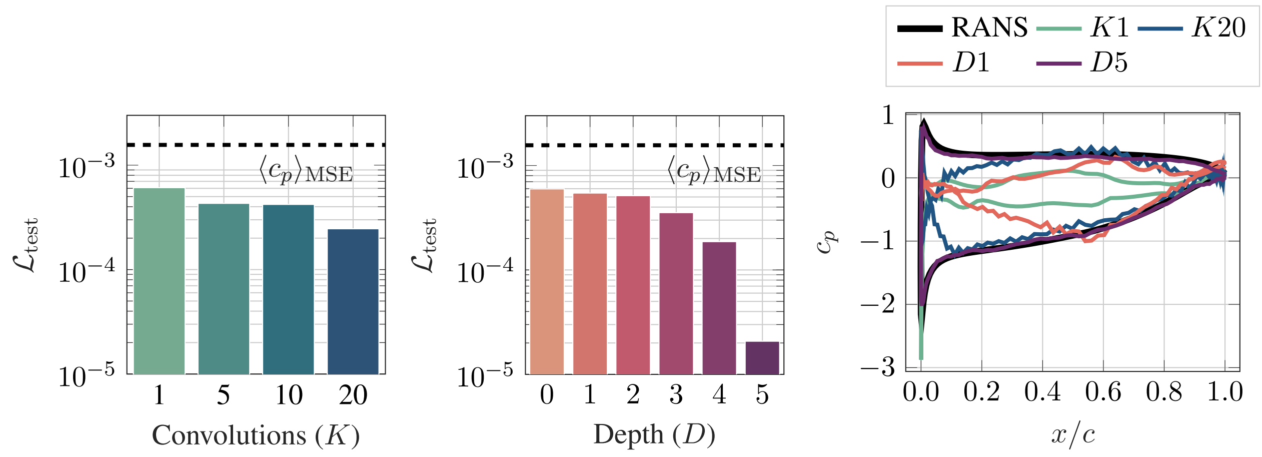
Baselines

The predictions by four B-GNNs, $\{$single-level geometry-based (GBF-B-GCN), multi-level geometry-based (GBF-B-GUN), single-level physics-based (PBF-B-GCN), multi-level physics-based (PBF-B-GUN)$\}$, are compared with two volume-based baselines – implicit neural fields model INFINITY (Serrano et al., 2023) and volumetric Graph-U-Net (Bonnet et al., 2023).

3 RESULTS and DISCUSSIONS

3.1 All-to-all node communication improves predictions

To test the hypothesis that incompressibility constraints require all-to-all node communication for accurate pressure prediction, we vary the parameters that control message passing among nodes in the single-level and multi-level architectures with geometry-based features, GBF-B-GCN and GBF-B-GUN respectively. In the single-level architecture, the pressure decoder MLP scales quadratically with K. To balance computational



(a) Convolution study

(b) Depth study

(c) Inference on a test case

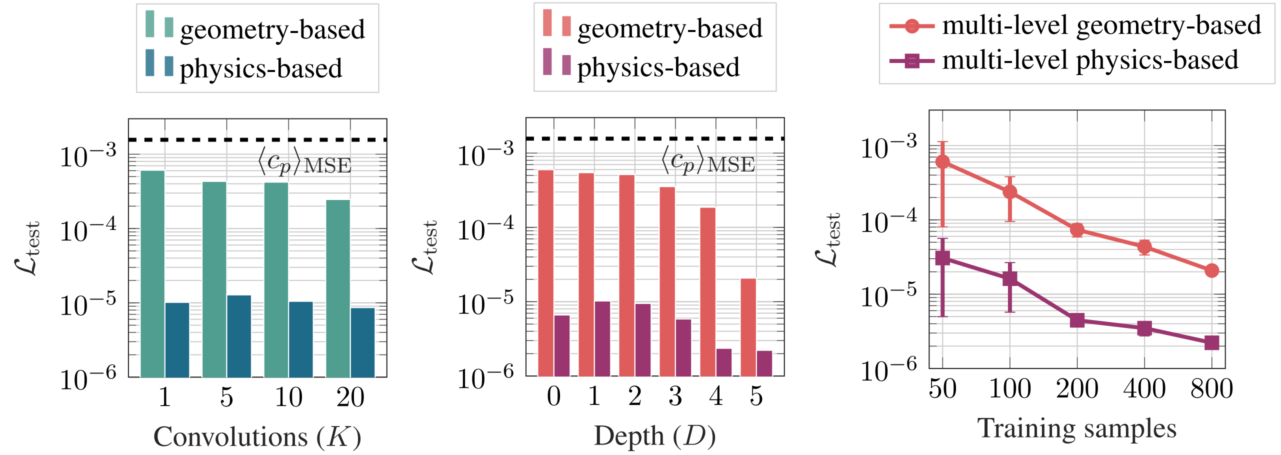
Figure 3: (a) Test loss decreases with recursive convolutions in the single-level model with the errors being lower than the reference $\langle c_p \rangle_{\text{MSE}}$. (b) Similarly, test loss decreases with depth in the multi-level model. (c) The multi-level model at full depth $D = 5$ agrees the most with the ground truth RANS.

effort and performance, we test $K = \{1, 5, 10, 20\}$ convolutions in this study. Unlike the single-level architecture, the multi-level architecture does not require stacking embeddings, and its pressure decoder MLP remains compact. This allows us to test the model up to full depth $D = \{0, 1, 2, 3, 4, 5\}$.

Figure 3a shows the test loss for the single-level models with increasing convolutions, demonstrating that accuracy improves with greater node communication. The reference error, $\langle c_p \rangle_{\text{MSE}}$, represents the loss if the model simply predicted the mean pressure distribution from the training set at every inference. Since the model achieves lower error, it is not merely regressing to the mean pressure distribution. This is further confirmed in Figure 3c, which illustrates inference on a test geometry: `NACA (6.914, 5.245, 8.861)` at $\text{Re} = 2.76 \times 10^6$, $\alpha = 6°$. While predictions at $K = 20$ are closer to the ground truth, they remain inaccurate and noisy, failing to capture the ground truth pressure distribution. This highlights that without all-to-all communication, the single-level model cannot produce reliable predictions. Similarly, the multi-level geometry-based model achieves higher accuracy as node communication increases with depth, as shown in Figure 3b. Moreover, the predictions at full depth ($D = 5$) closely match the ground truth (Figure 3c). This improvement is due to all-to-all communication, which enables the model to capture pressure distribution effectively.

3.2 Physics-based features enable reduction of model size and training data

Providing physics-based features improves accuracy by an order of magnitude, Figure 4, with physics-based B-GUN at full depth ($D = 5$) achieving the best performance. A closer look reveals that the physics-based models at their lowest model size (of $K = 1$ and $D = 0$ for B-GCN and B-GUN respectively) outperform the geometry-based counterparts at their highest. Thus, physics-based features reduce model size by 83% relative to models using purely geometric inputs. Figure 4c shows that with added training samples the generalization error improves. At every shard of full training samples including the full training set, the error by physics-based model is an order lower. This means about 93% fewer training samples are needed to get to a similar loss by the geometry-based model. Moreover, physics-based features serve as supplement datasets as they shift the requirement of having more samples to improve generalization in the case of the model without physics-based features.



(a) Physics improves accuracy of single-level B-GCN.

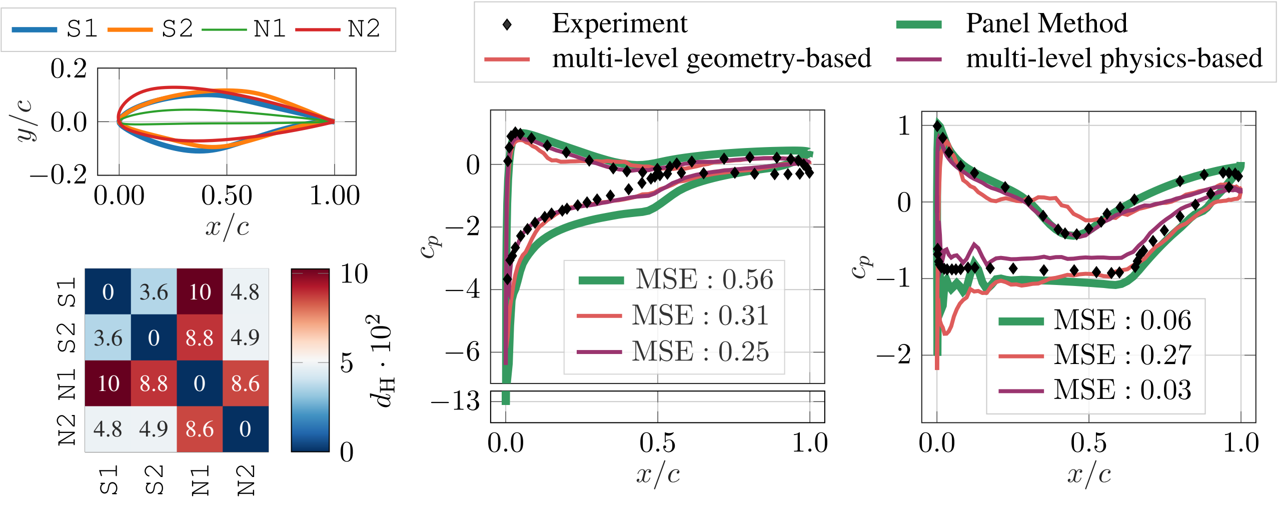
(b) Physics improves accuracy of multi-level B-GUN.

(c) Scaling of generalization error with training dataset size.

Figure 4: Physics-based features improve accuracy of both (a) single-level B-GCN and (b) multi-level B-GUN and (c) act as supplement data.

3.3 Physics-based features are essential for meaningful extrapolation

To analyze whether enforcing all-to-all communication in B-GNNs compels them to emulate potential flow solvers, we evaluate inference on out-of-distribution geometries commonly used in wind turbine blades: `S809` (Somers, 1997) and `S827` (Somers, 2005). Hausdorff distance (Rote, 1991), defined as the maximum distance between a point in one set and the nearest point in the other, is used to quantify geometric differences between the `S809/27` airfoils and the `NACA4/5` airfoils in the training set. While airfoil thickness provides a general measure of shape variation, it does not fully capture geometric differences. For instance, both `N2` and `S809/27` airfoils have similar maximum thicknesses—20% and 21% of chord length, respectively—but differ topologically. This distinction is quantified by the Hausdorff distance in Figure 5a in which `S809/27` airfoils are geometrically closer to each other than to the `NACA` airfoils. Figures 5b and 5c compare prediction by full depth geometry-based and physics-based multi-level models with the experiment for two cases: `S809` at $\text{Re} = 2 \times 10^6$ and $\alpha = 12°$ (Somers, 1997) and `S827` at $\text{Re} = 3 \times 10^6$ and $\alpha = 4°$ (Somers, 2005). At a high angle of attack $\alpha = 12°$, the pressure distributions of `NACA` and `S809/27` airfoils are similar, making the prediction task less challenging. In this case, both models capture the overall pressure distribution trend; however, the geometry-based model struggles to accurately predict the pressure drop at the leading edge, while the physics-based model aligns much more closely with the experimental data, particularly downstream on the suction surface. Conversely, at a lower angle of attack $\alpha = 4°$, where the pressure distributions of `NACA` and `S809/27` airfoils differ significantly, the extrapolation task becomes more difficult. Here, the geometry-based model overpredicts the pressure drop at the leading edge, leading to inaccurate results. In contrast, the physics-based model uses the inviscid pressure information from the Panel Method to predict a reasonable estimate of the viscous pressure distribution. Note that the Panel Method performs worse than the physics-based model in both cases. These results confirm that all-to-all-communication alone is insufficient for extrapolation – it requires the help of physics-based features. Given the relatively low computational overhead of obtaining physics-based features and substantial accuracy improvements, incorporating it into surrogate models is an effective strategy to ensure reliable physical predictions in out-of-distribution scenarios.



(a) NACA v/s S8 airfoils.

(b) S809 (S1) at $\alpha = 12°$

(c) S827 (S2) at $\alpha = 4°$

Figure 5: (a) Hausdorff distance, d_{H}, quantifies the difference between the profiles: S809 (S1), S827 (S2), NACA (1.541, 6.943, 0.0, 5.203) (N1), NACA (3.475, 3.252, 0.0, 19.801) (N2). (b) and (c) Physics-based features help the multi-level model to agree with the experiment better than the corresponding geometry-based model.

3.4 Boundary-GNN improves on baseline model performance

The baselines predict pressure, p, instead of the coefficient of pressure, c_p. To enable a direct comparison, the predictions by the geometry-based and physics-based multi-level models are converted to their respective pressure values, p, and are then normalized using the mean and standard deviation of the training set before comparison. Additionally, the coefficient of lift, C_l, is evaluated from the physical pressures.

Table 1 highlights the superior performance of the physics-based model over other models by having the lowest MSE for pressure and coefficient of lift. It has 94%, 85%, and 97% lower $(p|_{\mathcal{B}})$ error compared to the geometry-based model, INFINITY, and volumetric Graph-U-Net respectively. Unlike the geometry-based model, the volumetric models: INFINITY and volumetric Graph-U-Net use surface normals as additional input features, along with spatial coordinates and flow information. This allows them to differentiate between the upper and lower surfaces of the airfoil, which could explain INFINITY's improved performance over the geometry-based model. Further investigation is needed to determine whether incorporating surface normals into the geometry-based model could enhance its accuracy.

4 CONCLUSIONS

This study demonstrates that all-to-all node communication is essential for B-GNNs to accurately learn the mapping of airfoil geometries to pressure distributions. Incorporating physics-based features significantly enhances B-GUN performance by 83%. More importantly, they improve generalization to out-of-distribution

Table 1: Comparison of prediction error (MSE) by the geometry-based and physics-based multi-level models with that of INFINITY (Serrano et al., 2023) and volumetric Graph-U-Net (Bonnet et al., 2023).

	Multi-level physics-based	Multi-level geometry-based	INFINITY	Graph-U-Net
$p\|_{\mathcal{B}}$	$\mathbf{0.001 \pm 0.0001}$	0.020 ± 0.0043	0.007 ± 0.001	0.039 ± 0.007
c_p	$\mathbf{0.001 \pm 0.0002}$	0.020 ± 0.0043	-	-
C_l	$\mathbf{0.017 \pm 0.003}$	0.194 ± 0.047	0.081 ± 0.007	0.489 ± 0.105

geometries by up to 88% over the geometry-based model, addressing a major challenge in surrogate modeling. However, reliance on potential flow input-features may bias solutions away from the target flows when they feature strong nonlinear effects such as massive flow separation. Additionally, B-GNNs can produce noisy pressure distributions, which could be problematic for future applications such as cavitation modeling. Future work could focus on integrating additional physical features, improving noisy predictions, and extending these methods to three-dimensional unstructured propeller grids.

ACKNOWLEDGEMENTS

This research is supported by the granting authority European Research Executive Agency (REA) under the Marie Skłodowska-Curie Actions funded project, SCALE (HORIZON-MSCA-2022 DN-01-0).

APPENDIX

A.1 Model hyperparameter tuning

The optimal hyperparameters of each B-GNN model are found in the parameter space mentioned in Table 2.

Table 2: Hyperparameter design space used for the Edge Convolution MLP.

Hyperparameter	Values
Node Embeddings H	$\{4, 8\}$
Edge conv. MLP Width W	$\{8, 16, 32, 64, 128\}$
Edge conv. MLP Layers L	$\{1, 2\}$

Table 3: Edge Convolution MLP hyperparameters for different B-GNNs.

Model	Hyperparameters
GBF-B-GNN	$H = 4$, $W = 128$, $L = 2$
GBF-B-GUN	$H = 8$, $W = 128$, $L = 2$
PBF-B-GNN	$H = 4$, $W = 128$, $L = 1$
PBF-B-GUN	$H = 8$, $W = 128$, $L = 2$

References

F. Basan, J.-G. Fischer, R. Putland, J. Brinkkemper, C. De Jong, B. Binnerts, A. Norro, D. Kühnel, L.-A. Ødegaard, M. Andersson, E. Lalander, J. Tougaard, E. Griffiths, M. Kosecka, E. Edwards, N. Merchant, K. De Jong, S. Robinson, L. Wang, and N. Kinneging. The underwater soundscape of the north sea. *Marine Pollution Bulletin*, 198:115891, Jan. 2024. ISSN 0025326X. doi: 10.1016/j.marpolbul.2023.115891.

F. Bonnet, A. J. Mazari, P. Cinnella, and P. Gallinari. Airfrans: High fidelity computational fluid dynamics dataset for approximating reynolds-averaged navier-stokes solutions. (arXiv:2212.07564), June 2023. URL `http://arxiv.org/abs/2212.07564`. arXiv:2212.07564 [physics].

M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. (arXiv:2104.13478), May 2021. doi: 10.48550/arXiv.2104.13478. URL `http://arxiv.org/abs/2104.13478`. arXiv:2104.13478 [cs].

D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). (arXiv:1511.07289), Feb. 2016. doi: 10.48550/arXiv.1511.07289. URL `http://arxiv.org/abs/1511.07289`. arXiv:1511.07289 [cs].

M. Drela. Xfoil: An analysis and design system for low reynolds number airfoils. In T. J. Mueller, editor, *Low*

Reynolds Number Aerodynamics, page 1–12, Berlin, Heidelberg, 1989. Springer. ISBN 978-3-642-84010-4. doi: 10.1007/978-3-642-84010-4_1.

M. Dular and M. Petkovšek. On the mechanisms of cavitation erosion – coupling high speed videos to damage patterns. *Experimental Thermal and Fluid Science*, 68:359–370, Nov. 2015. ISSN 08941777. doi: 10.1016/j.expthermflusci.2015.06.001.

N. Durasov, A. Lukoyanov, J. Donier, and P. Fua. Debosh: Deep bayesian shape optimization. (arXiv:2109.13337), Oct. 2023. doi: 10.48550/arXiv.2109.13337. URL `http://arxiv.org/abs/2109.13337`.

J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou. Deep learning scaling is predictable, empirically. (arXiv:1712.00409), Dec. 2017. doi: 10.48550/arXiv.1712.00409. URL `http://arxiv.org/abs/1712.00409`. arXiv:1712.00409 [cs].

G. Immordino, A. Vaiuso, A. Da Ronch, and M. Righi. Predicting transonic flowfields in non–homogeneous unstructured grids using autoencoder graph convolutional networks. *Journal of Computational Physics*, 524:113708, Mar. 2025. ISSN 00219991. doi: 10.1016/j.jcp.2024.113708.

G. Rote. Computing the minimum hausdorff distance between two point sets on a line under translation. *Information Processing Letters*, 38(3):123–127, May 1991. ISSN 0020-0190. doi: 10.1016/0020-0190(91)90233-8.

F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(11):61–80, Jan. 2009. ISSN 1941-0093. doi: 10.1109/TNN.2008.2005605.

L. Serrano, L. Migus, Y. Yin, J. A. Mazari, and P. Gallinari. Infinity: Neural field modeling for reynolds-averaged navier-stokes equations. (arXiv:2307.13538), July 2023. doi: 10.48550/arXiv.2307.13538. URL `http://arxiv.org/abs/2307.13538`. arXiv:2307.13538 [cs].

P. D. Sharpe. Accelerating practical engineering design optimization with computational graph transformations. 2024.

D. M. Somers. *Design and Experimental Results for the S809 Airfoil*. 1997. URL `https://www.nrel.gov/docs/legosti/old/6918.pdf`.

D. M. Somers. Design and experimental results for the s827 airfoil: 1998-1999. 2005. doi: https://doi.org/10.2172/15011672.

N. Thuerey, K. Weißenow, L. Prantl, and X. Hu. Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows. *AIAA Journal*, 58(11):25–36, 2020. ISSN 0001-1452. doi: 10.2514/1.J058291.

Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. (arXiv:1801.07829arXiv:1801.07829), June 2019. doi: 10.48550/arXiv.1801.07829. URL `http://arxiv.org/abs/1801.07829`. arXiv:1801.07829 [cs].

G. D. Weymouth and D. K. Yue. Physics-based learning models for ship hydrodynamics. *Journal of Ship Research*, 57(1):1–12, Mar. 2013. ISSN 00224502, 15420604. doi: 10.5957/JOSR.57.1.120005.