WINDMIL: EQUIVARIANT GRAPH LEARNING FOR BUILDING WIND LOADS

Anonymous authors
Paper under double-blind review

ABSTRACT

Wind loading prediction is safety-critical for building design, yet it is largely absent from ML benchmarks because high-fidelity aerodynamic labels are expensive to generate and defined on irregular building surfaces. Current practice relies on wind tunnel testing, design codes, and computational fluid dynamics with large-eddy simulation (LES), where each case requires more than 24 hours, making parametric exploration infeasible. Here we introduce WINDMIL, a graph learning framework that advances both ML and wind engineering: it provides a realistic LES-based benchmark for studying physical inductive bias, extrapolation, and symmetry, while enabling LES-based surrogate prediction of pressure loads on buildings. We generate 462 LES simulations of low-rise buildings under realistic atmospheric boundary layer flow, spanning roof shapes and wind directions. The central challenge is not only accuracy, but also respecting the physical constraints: reflecting a building–wind configuration should reflect its predicted pressure field. However, training on reflected examples does not guarantee this consistency. WINDMIL encodes this order-2 symmetry directly in the architecture, creating a testbed for equivariance versus data augmentation. We compare it with a pointwise MLP, GRAPHSAGE, and reflection-augmented GRAPHSAGE at different data splits. Across random, barycentric extrapolation, and edge-extrapolation splits, WINDMIL achieves the highest overall accuracy for surface pressure coefficients while maintaining exact reflection symmetry. In the OOD edge extrapolation, WINDMIL reduces mean C_p RMSE by 7.9% relative to reflection-augmented GRAPHSAGE and improves hitrate relative to non-augmented GRAPHSAGE, by approximately 5%. These gains also translate to integrated drag and lift coefficients, with best hitrates and exact symmetry. These results show how a task-specific symmetry can improve generalization while enforcing a physical constraint that data augmentation only approximates. This study establishes wind loading prediction as a benchmark for geometric deep learning and ML-based computational wind engineering.

1 INTRODUCTION

Scientific machine learning has made rapid progress on molecular dynamics, weather forecasting, and canonical fluid flows (Noé et al., 2020; Zhang et al., 2025; Fukami et al., 2020). Wind loading on buildings poses a different combination of challenges: high-fidelity labels are expensive, outputs are defined on irregular surfaces, separated turbulence creates sharp pressure gradients, and predictions inform safety decisions for building design. This combination makes the problem a demanding testbed for geometric deep learning as well as a candidate for faster early-stage engineering analysis.

For wind engineering, accurate pressure predictions are essential for evaluating cladding loads, global forces, and design safety. Current workflows rely on building codes, wind tunnel experiments, full-scale measurements, and high-fidelity computational fluid dynamics (CFD) (Blocken et al., 2016; Tieleman, 2003; Phillips et al., 2019; Plate & Kiefer, 2001; Alrawashdeh & Stathopoulos, 2015; Richards & Hoxey, 2012; Vargiemezis & Gorle, 2024; Vargiemezis, 2024). Large-eddy simulation (LES) can resolve the turbulent flow structures that govern wind pressure distributions on buildings, but it remains too expensive for routine parametric exploration (Potsis et al., 2023; Blocken, 2015; Vargiemezis & Gorlé, 2025c;a). In our dataset, a single LES case requires more than 24 hours on 64 CPU cores. This computational cost limits the use of high-fidelity aerodynamic data during design and motivates fast surrogate models that remain consistent with the underlying physics.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Recent advances in geometric deep learning provide a promising foundation for such surrogates (Atz et al., 2021; Cao et al., 2020; 2022; Bronstein et al., 2021; Xu et al., 2019; Kanatsoulis & Ribeiro, 2024). Graph neural networks (GNNs) (Kipf & Welling, 2016; Hamilton et al., 2017) naturally represent irregular surface discretizations, with nodes corresponding to surface samples and edges encoding local geometric neighborhoods. Models such as MeshGraphNets (Pfaff et al., 2020; Song et al., 2020; Nabian et al., 2024) and Graph Network Simulators (Sanchez-Gonzalez et al., 2020; Allen et al., 2023; Wu et al., 2022; Rubanova et al., 2021) have demonstrated the value of graph learning for physical simulation, while related work in 3D building reconstruction shows the relevance of graph representations for built-environment geometry (Chen et al., 2024; Gurnani et al., 2023). For wind loading, the challenge is to predict pressure at each point on a building surface while preserving the spatial patterns that determine integrated forces.

A major barrier to progress for both ML and wind engineering is the lack of open, high-fidelity datasets designed for learning. Existing aerodynamic databases such as the TPU database (Quan et al., 2007) are invaluable for wind engineering, but they are primarily designed for engineering lookup, code calibration, and comparison across canonical cases. They are not designed as large-scale graph learning benchmarks with controlled interpolation, extrapolation, and symmetry-consistency evaluations. By contrast, automotive aerodynamics has recently benefited from systematic geometry generation paired with CFD datasets (Benjamin & Iaccarino, 2025; He et al., 2025; Hu et al., 2025; Qi et al., 2025; Vatani et al., 2025; Gu et al., 2026). No comparable LES-based benchmark exists for wind loading on low-rise buildings, despite the practical importance of this problem and its suitability for geometric deep learning.

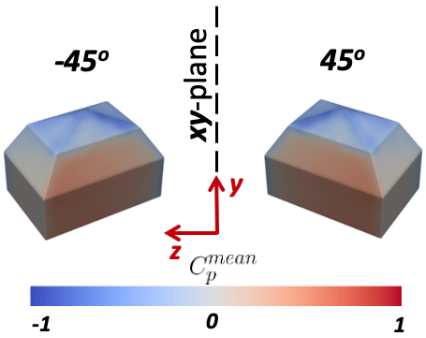

Figure 1: Reflection symmetry in wind loading. Mean pressure coefficient (C_p) distributions for opposite wind incidences are related by a lateral reflection of the building–wind configuration. Non-equivariant models can treat these cases as unrelated, while WINDMIL encodes this symmetry directly.

This setting raises a central question for scientific ML: when physical labels are expensive and symmetries are known, is it sufficient to expose a model to symmetric data through augmentation, or should the symmetry be encoded directly into the architecture? Wind loading on buildings provides a controlled instance of this question. The ground plane is fixed and the wind speed varies with height in the atmospheric boundary layer. Consequently, arbitrary three-dimensional rotations do not preserve the physical setup. Rotating a building while holding the inflow fixed also changes its wind incidence and, consequently, its pressure distribution. Enforcing full $E(3)$ equivariance on the building geometry alone would therefore constrain predictions under transformations that are not symmetries of this task. We instead encode lateral reflection of the complete building-wind configuration: when the building geometry is reflected with respect to the inflow wind direction, the predicted surface pressure field must reflect accordingly. Figure 1 illustrates this relationship for opposite wind incidences. For example, when the flow is aligned with $x-$direction, y is the vertical, and z is the spanwise, the pressure distribution of a building at $+45^\circ$ wind incidence should be the reflected version of a building at -45° wind incidence with respect to the $xy-$ plane.

We introduce WINDMIL, a reflection-equivariant GNN for predicting surface pressure coefficients and integrated wind loads on building graphs. The model applies a shared graph encoder to original and reflected inputs and aggregates the corresponding embeddings, guaranteeing consistency under the valid reflection transformation. We compare against a pointwise MLP, GRAPHSAGE, and GRAPHSAGE with reflection data augmentation. This allows us to isolate the effect of architectural equivariance from simple exposure to reflected samples during training.

Our contributions are:

1. **A high-fidelity graph learning benchmark for wind loading.** We generate, to our knowledge, the first large-scale LES-based dataset for wind loading prediction on low-rise buildings under realistic atmospheric boundary layer flow. The dataset contains 462 simulations spanning continuous roof-shape variations and wind directions.

2. **A physically targeted equivariant model.** We develop a reflection-equivariant GNN that encodes the only valid symmetry for wind loading predictions on buildings, avoiding spurious invariances that would arise from enforcing full Euclidean equivariance.
3. **A controlled comparison of equivariance and data augmentation.** We evaluate WINDMIL against non-equivariant and reflection-augmented baselines across different tasks that include interpolation and OOD extrapolation settings.
4. **Node-level and graph-level wind-load prediction.** We evaluate both surface pressure coefficients and integrated drag and lift coefficients, showing that improvements in pressure prediction translate to structural load quantities.

The benchmark tests whether enforcing this specific valid symmetry improves prediction and consistency in the sampled building family.

2 DATASET GENERATION

2.1 GEOMETRY PARAMETERIZATION

In this work, we adapt the Signed Distance Function (SDF)-based interpolation method, originally developed for automotive geometries by Benjamin & Iaccarino (2025); Zhang et al. (2026); Jacob et al. (2021), to systematically generate diverse building shapes for wind loading analysis. We begin with three basis geometries derived from the TPU dataset (Quan et al., 2007; Wang et al., 2020): a flat roof, a gable roof, and a hip roof. All buildings share the same footprint of 18 m length × 12 m width in full scale, while their heights differ: 8 m for the flat roof, 16 m for the gable roof, and 24 m for the hip roof.

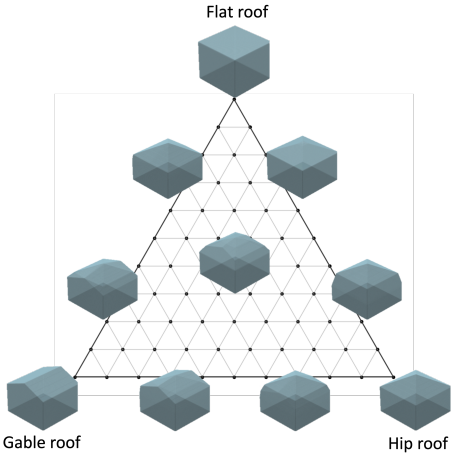

Figure 2: Convex hull of the three basis geometries. Points inside the convex hull correspond to interpolated buildings.

Each geometry is first converted into a binary grid using ray-tracing, where cells are marked as occupied or empty based on ray–surface intersections. The binary grids are then transformed into signed distance functions (SDFs), which assign to each grid point the signed distance to the building surface. By convention, the SDF equals zero on the surface, is positive outside, and negative inside. This continuous representation enables consistent shapes. To generate new buildings, we construct a convex hull in the SDF space spanned by the three basis geometries, as shown in Fig. 2. Barycentric sampling within this convex hull defines a continuous family of SDF fields that morph between the flat, gable, and hip roof cases. For the wind incidence at $0°$, we sample this shape space to obtain 66 unique building geometries. To build the complete dataset, each geometry is evaluated at wind directions from $0°$ to $90°$ in $15°$ increments, resulting in $66 \times 7 = 462$ LES cases.

Formally, new geometries are generated by taking a convex combination of two SDF fields, $\phi_1(\vec{x}_i)$ and $\phi_2(\vec{x}_i)$, defined on the same structured grid with spatial coordinates $\vec{x}_i$:

$$\phi_3((\vec{x_i})) = \alpha\phi_1((\vec{x_i})) + (1-\alpha)\phi_2((\vec{x_i})), \tag{1}$$

where α is a scalar weight. Varying α yields intermediate geometries between the basis cases. Applying Eq. (1) to a unit sphere and unit cube, one can produce intermediate shapes by varying α, an example of which is shown in Fig. 3.

Finally, the SDFs are reconstructed into surface meshes using the marching cubes algorithm (Benjamin & Iaccarino, 2025), followed by Laplacian smoothing to remove voxelization artifacts and improve surface quality. To verify that this step does not alter the building geometry, we quantify the deviation between the reconstructed and target surfaces and find a maximum deviation of approximately 1% of H_{ref}, comparable to the LES near-wall surface resolution ($\Delta/H_{ref} = 0.0098$). The smoothing

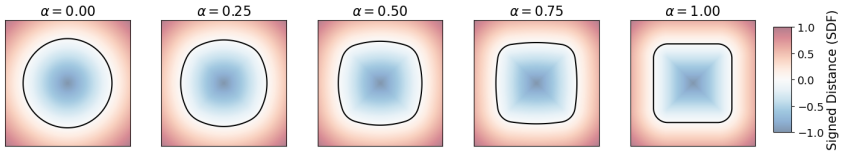

Figure 3: SDF shape morphing between a sphere and a cube, per Eq. (1) for different α.

therefore, removes grid artifacts without introducing geometry changes larger than the resolved near-wall scale. By combining SDF-based shape morphing with systematic rotations, the method enables the creation of a controlled graph dataset for wind loading prediction.

2.2 Large-eddy Simulations

To generate the dataset, LES simulations are performed using the CharLES code, a low-Mach, isentropic solver developed by Cadence Design Systems (2022). CharLES implements a finite volume approach with an automated body-fitted meshing technique based on 3D-clipped Voronoi diagrams, and uses numerical schemes with second-order accuracy in space and time to solve the governing equations (Ambo et al., 2020). In addition, the Vreman turbulence model is used to model the unresolved part of the stress tensor (Vreman, 2004). CharLES has been extensively validated against experimental data in a range of wind engineering applications. For instance, it has been validated against wind tunnel measurements of wind-induced pressure loads on high- and low-rise buildings (Ciarlatani et al., 2023; Vargiemezis & Gorlé, 2025a;c), as well as against measurements of wind-driven natural ventilation (Hwang & Gorlé, 2022). Comparisons against field measurements have also been performed for wind pressures on the Space Needle (Hochschild & Gorlé, 2024) and for natural ventilation flow in a dense urban environment (Hwang & Gorlé, 2023). In addition, CharLES has been used to create datasets for predicting wind fields in urban areas using a U-net-based model (Vargiemezis & Gorlé, 2025b).

The mesh uses 2.1×10^6 control volumes, with refinement near the no-slip surfaces of the building, as shown in Fig. 4. The resolution on the building surfaces is $\Delta/H_{ref} = 0.0098$, where Δ is the mesh size locally, and H_{ref} is the characteristic height of the building. The domain extents $5H_{ref}$ upstream, $15H_{ref}$ downstream, and $5H_{ref}$ in the lateral and vertical directions, following the proposed guidelines (Franke et al., 2011). At the inlet, a logarithmic mean velocity profile is prescribed, and artificially generated turbulent fluctuations are superimposed. The turbulent velocity field is generated using the divergence-free digital filter method proposed by (Kim et al., 2013). At the outlet, a zero gradient condition is applied, while the two lateral boundaries are periodic, and a slip condition is applied at the top boundary. Finally, at the ground, a rough wall function for a neutral atmospheric boundary layer with a fixed roughness of 0.0027 m is specified. More details regarding the setup can be found in the study of a different isolated building, since the same setup is used (Vargiemezis & Gorlé, 2025c).

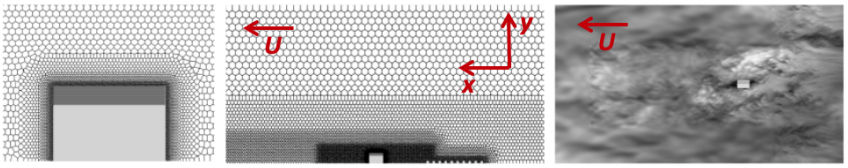

Figure 4: View of the mesh used for the large-eddy simulations, with side views of refinement zones shown: (left) near the building and (middle) in the surrounding area. (right) Top view of instantaneous velocity magnitude contours showing the turbulent flow around the building.

The simulation is run for a total time of 20 seconds, and statistics are collected over the last 15 seconds, after an initial burn-in period of 5 seconds. Each LES case requires approximately 24 hours on 64 CPU cores. The primary quantities of interest (QoIs) are the time-averaged mean and standard deviation of the surface pressure coefficient. The pressure coefficient at a surface node i is defined as

$$C_{p,i} = \frac{p_i - p_\infty}{\frac{1}{2}\rho U_\infty^2}, \tag{2}$$

where p_i is the local surface pressure at location i on the building surface, p_∞ is the freestream pressure, ρ is the density of air, and U_∞ is the reference velocity.

In addition to node-level pressure statistics, we compute graph-level integrated wind load quantities from the surface pressure field. Specifically, we evaluate the drag and lift coefficients, C_D and C_L, by integrating the pressure force over the building surface and normalizing by the dynamic pressure and reference area:

$$C_D = \frac{F_D}{\frac{1}{2}\rho U_\infty^2 A_{\text{ref}}}, \qquad C_L = \frac{F_L}{\frac{1}{2}\rho U_\infty^2 A_{\text{ref}}}, \tag{3}$$

where F_D and F_L are the pressure-induced force components in the drag and lift directions, respectively, and A_{ref} is the reference area. These integrated coefficients provide graph-level targets, complementing the node-level prediction of C_p.

3 Model Architecture

3.1 WindMiL: Reflection-equivariant model

As shown in Fig. 5, we represent each building surface as a point cloud, from which we construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where nodes $v \in \mathcal{V}$ correspond to sampled surface points and edges $e \in \mathcal{E}$ are defined based on spatial proximity. Each node is associated with a set of geometric features $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$. In particular, each node v has features $\mathbf{x}_v \in \mathbb{R}^d$, with $d = 6$: normalized coordinates (x, y, z) with respect to building height H_{ref} and surface normal components (n_x, n_y, n_z), which are unit vectors oriented outward of building surface. The learning task is to predict node-level mean or standard deviation of the surface pressure coefficient C_p, as well as graph-level integrated drag and lift coefficients, C_D and C_L.

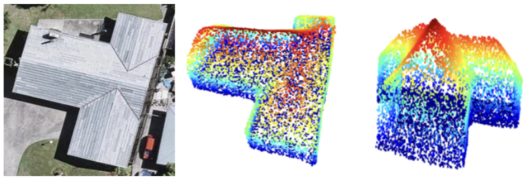

Figure 5: Illustration of building representation as pointcloud data. Left is the building, and on the right side, the pointcloud at different perspectives.

To process the graph structure and the geometrical features of each building, we employ a message-passing GNN $f_\theta(\mathcal{G}, \mathbf{X})$, (Kipf & Welling, 2016; Xu et al., 2019; Hamilton et al., 2017), defined by the following recursive formula:

$$\mathbf{x}_v^{(l)} = h^{(l-1)}\left(\mathbf{x}_v^{(l-1)}, g^{(l-1)}\left(\left\{\mathbf{x}_u^{(l-1)} : u \in \mathcal{N}(v)\right\}\right)\right), \tag{4}$$

where $\mathcal{N}(v)$ represents the 1-hop neighborhood of vertex v, and g, h are permutation equivariant operators. In our implementation, $f_\theta(\mathcal{G}, \mathbf{X})$ is modeled as a stack of GraphSAGE layers (Hamilton et al., 2017) with residual connections, layer normalization, and a global skip from the input projection. More details on the model specifics are given in Appendix A.

Wind loading exhibits a lateral reflection symmetry with respect to the streamwise-vertical (xy) plane, as shown in Fig. 1. To encode this symmetry, as shown in Fig. 6, we define a reflection operator $\mathcal{R}$ that flips the spanwise coordinate and normal component:

$$\mathcal{R}: \ (z, n_z) \mapsto (-z, -n_z),$$

while leaving (x, y, n_x, n_y) unchanged. The model is designed such that predictions are consistent under this transformation. To enforce symmetry, both the original and reflected features are passed through the same encoder f_θ. The resulting embeddings are then averaged:

$$\mathbf{z}_v = \tfrac{1}{2}\Big(f_\theta(\mathcal{G}, \mathbf{X})[v] + f_\theta(\mathcal{G}, \mathcal{R}(\mathbf{X}))[v]\Big).$$

The overall model $F(\cdot)$, which maps inputs to predictions, then satisfies the reflection property $F(\mathcal{R}(\mathbf{x})) = F(\mathbf{x})$. Finally, embeddings $\mathbf{z}_v$ are passed through a feed-forward predictor g_ϕ, providing the node-level of graph-level outputs $C_{p,v} = g_\phi(\mathbf{z}_v)$. This architecture ensures predictions that respect the underlying physical symmetry of wind loading on buildings.

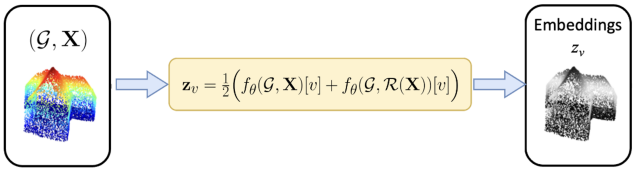

Figure 6: WindMiL model architecture. Input is the graph structure and the geometrical features. The embeddings $\mathbf{z}_v$ are passed through a feed-forward predictor g_ϕ, to provide the predictions $C_{p,v} = g_\phi(\mathbf{z}_v)$.

3.2 Baselines

To evaluate the separate effects of graph structure, data augmentation, and architectural equivariance, we compare WindMiL against three baselines:

- **MLP**: A pointwise multi-layer perceptron that treats each surface node independently. It takes node-level geometric features (x, y, z, n_x, n_y, n_z) and the wind angle as input.
- **GraphSAGE (GS)**: A non-equivariant GNN that uses message passing over the surface graph but does not explicitly encode reflection symmetry.
- **GraphSAGE with data augmentation (GSDAG)**: A GraphSAGE model trained with reflected geometries. This baseline tests whether exposure to both orientations is sufficient, or whether exact architectural equivariance is needed.

These baselines are chosen to isolate the core scientific question of the paper: whether the correct physical symmetry should be learned from augmented data or enforced by construction.

4 Experiments

We evaluate WindMiL and the baselines on three data splits of increasing difficulty, shown in Fig. 7. The *random* split assigns generated geometries inside the convex hull to train, validation, and test sets. The *barycentric extrapolation* split holds out geometries on the convex-hull boundary. The *edge extrapolation* split is stricter: models are trained on regions associated with two roof archetypes and tested near the third. Together, these splits evaluate in-distribution prediction, boundary generalization, and extrapolation across the roof-shape space.

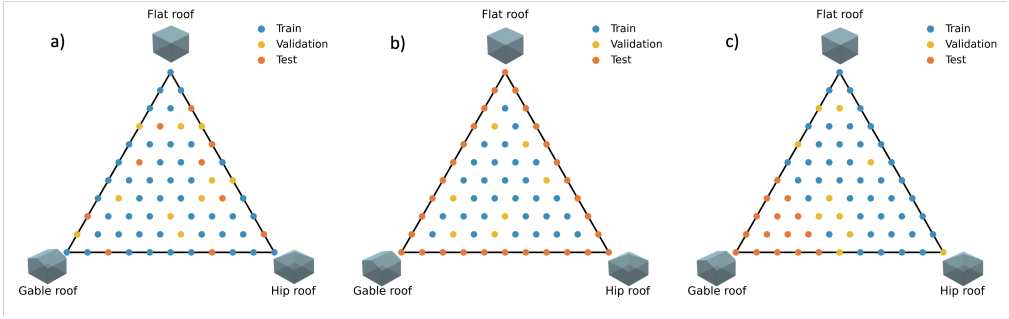

Figure 7: Visualization of the three dataset splits. (a) Random split: train/dev/test are assigned randomly. (b) Barycentric extrapolation split: geometries on the convex-hull boundary are held out for testing. (c) Edge extrapolation split: geometries near the gable-roof corner are held out for testing.

For each split, we evaluate the original test set and an extended reflected-test set denoted by +*Sym*, where held-out geometries are augmented with their reflected counterparts. Training is performed on the original geometries (without their reflected counterparts) unless otherwise specified. The +*Sym* evaluation tests whether models satisfy the physical reflection consistency illustrated in Fig. 1 by adding the reflected geometries in the test set. More details on the training and model specifics are given in Appendix A.

We evaluate node-level predictions of the mean and standard deviation of the pressure coefficient C_p, and graph-level integrated drag and lift coefficients (C_D, C_L). We report root mean squared

error (RMSE), mean absolute error (MAE), coefficient of determination (R^2), and hitrate. The hitrate is the percentage of predictions within a tolerance of ± 0.10 for mean C_p and ± 0.05 for std C_p, corresponding to approximately 10% of the maximum observed values.

4.1 C_p PREDICTION ACROSS RANDOM AND EXTRAPOLATION SPLITS

Table 1 summarizes the C_p prediction results across the *random*, *barycentric extrapolation*, and *edge extrapolation* splits. In the *random* split, all models are evaluated on geometries drawn from the same shape distribution as the training set. Both graph-based models achieve high accuracy, confirming that message passing over the building surface captures geometric context that is missing from the pointwise MLP. For mean C_p, WINDMIL achieves the lowest RMSE (0.048) and highest hitrate (95.19%), followed closely by GRAPHSAGE (RMSE $= 0.051$, hitrate $= 94.11\%$). The MLP baseline performs worse, with RMSE $= 0.069$ and hitrate $= 89.68\%$, indicating that local node features alone are insufficient to fully capture the spatial structure of the pressure field.

Table 1: C_p prediction across random, barycentric extrapolation, and edge extrapolation splits. *+Sym* indicates evaluation on a test set extended with reflected geometries. DAG denotes reflection data augmentation and WM denotes WINDMIL.

Split	Method	Mean C_p				Std C_p			
		RMSE	MAE	R^2	HR (%)	RMSE	MAE	R^2	HR (%)
Random	MLP	0.069	0.047	0.953	89.68	0.024	0.016	0.917	94.78
	MLP + Sym	0.105	0.069	0.892	79.59	0.034	0.022	0.838	89.47
	GS	0.051	0.037	0.974	94.11	0.019	0.013	0.947	97.03
	GS + Sym	0.067	0.048	0.956	88.36	0.026	0.018	0.901	93.23
	GSDAG	0.058	0.042	0.967	93.83	0.022	0.015	0.928	94.56
	GSDAG + Sym	0.060	0.044	0.965	93.50	0.023	0.016	0.925	94.00
	WM	0.048	0.034	0.978	95.19	0.021	0.015	0.940	96.65
	WM + Sym	0.048	0.034	0.978	95.19	0.021	0.015	0.940	96.65
Bary.	MLP	0.089	0.060	0.921	83.40	0.030	0.020	0.874	91.17
	MLP + Sym	0.120	0.080	0.855	74.76	0.037	0.025	0.817	87.23
	GS	0.070	0.051	0.951	87.37	0.026	0.018	0.905	93.60
	GS + Sym	0.086	0.062	0.926	81.15	0.032	0.022	0.863	89.86
	GSDAG	0.066	0.048	0.955	90.73	0.025	0.017	0.910	93.92
	GSDAG + Sym	0.070	0.050	0.952	90.59	0.027	0.019	0.902	93.81
	WM	0.063	0.047	0.960	92.83	0.025	0.017	0.913	94.82
	WM + Sym	0.063	0.047	0.960	92.83	0.025	0.017	0.913	94.82
Edge	MLP	0.089	0.060	0.920	84.50	0.029	0.020	0.880	91.00
	MLP + Sym	0.105	0.071	0.890	78.20	0.033	0.023	0.850	88.50
	GS	0.078	0.052	0.939	86.76	0.026	0.018	0.908	93.37
	GS + Sym	0.091	0.063	0.918	80.68	0.030	0.021	0.879	90.63
	GSDAG	0.076	0.050	0.942	87.50	0.025	0.017	0.915	94.00
	GSDAG + Sym	0.079	0.053	0.938	85.10	0.026	0.018	0.908	93.10
	WM	0.070	0.047	0.951	90.22	0.023	0.016	0.929	94.50
	WM + Sym	0.070	0.047	0.951	90.22	0.023	0.016	0.929	94.50

The reflected-test evaluation (*+Sym*) reveals a more important difference between models. When the random test set is extended with reflected geometries, the non-equivariant baselines degrade substantially: for mean C_p, MLP RMSE increases from 0.069 to 0.105 ($+52\%$), and GRAPHSAGE RMSE increases from 0.051 to 0.067 ($+31\%$). This shows that standard models treat reflected configurations as different cases, even though they are physically equivalent under the valid symmetry. GSDAG improves robustness by exposing GRAPHSAGE to reflected samples during training, but its metrics still change between the original and *+Sym* evaluations. Thus, data augmentation reduces symmetry error but only learns the symmetry approximately. In contrast, WINDMIL has identical metrics for the original and reflected test sets, as expected from the architecture.

The same trend becomes more evident in the barycentric extrapolation split, where models are tested on geometries located on the boundary of the convex hull. This setting evaluates whether the learned mapping generalizes beyond the interior of the sampled shape space. WINDMIL achieves

the best mean C_p performance, with RMSE $= 0.063$ and hitrate $= 92.83\%$, compared with 90.73% for GSDAG and 87.37% for GRAPHSAGE. The data-augmented baseline significantly improves over non-augmented GRAPHSAGE, confirming that reflected examples provide useful information. However, GSDAG still changes under *+Sym* evaluation, while WINDMIL remains unchanged. This supports the central claim that model equivariance is a mechanism for enforcing physically consistent extrapolation.

The edge extrapolation split provides the strictest OOD test. Here, models are trained on regions associated with two roof archetypes and tested near the third, making the test geometries more distinct from the training distribution. WINDMIL again achieves the highest hitrates for both mean and std C_p, reaching 90.22% and 94.50%, respectively. GSDAG follows with 87.50% and 94.00%, while GRAPHSAGE reaches 86.76% and 93.37%. These results show that the benefit of the equivariant inductive bias persists in the most challenging generalization regime.

Overall, the results support the main thesis of the paper. 1) Graph structure improves pressure prediction relative to pointwise learning; 2) reflection data augmentation improves robustness but does not guarantee consistency; and 3) encoding the physically valid reflection symmetry directly into the model gives exact reflected-test consistency while maintaining the best accuracy across all splits. This distinction is important for wind loading prediction because reflected building configurations should produce reflected pressure fields exactly, and not by statistical approximation.

4.2 INTEGRATED DRAG AND LIFT COEFFICIENTS C_D AND C_L

Beyond node-level pressure coefficients, structural design often depends on surface-integrated forces, such as drag and lift. We therefore evaluate drag and lift coefficients, C_D and C_L, across the random, barycentric extrapolation, and edge extrapolation splits. Since these are graph-level quantities derived from the surface pressure field, they provide a test of whether node-level pressure accuracy translates to integrated loads.

Table 2: Hitrate (HR) performance for integrated drag and lift coefficients. *+Sym* indicates evaluation on a test set extended with reflected geometries. DAG denotes reflection data augmentation and WM denotes WINDMIL.

Target	Method	Random Split		Barycentric Extrap.		Edge Extrap.	
		Mean HR (%)	Std HR (%)	Mean HR (%)	Std HR (%)	Mean HR (%)	Std HR (%)
C_D	MLP	93.43	93.55	93.71	95.43	91.11	92.28
	MLP + Sym	89.31	92.43	93.43	95.43	90.56	91.92
	GS	94.78	95.28	95.57	94.29	92.22	93.31
	GS + Sym	92.48	93.72	94.57	94.86	91.11	92.25
	GSDAG	95.93	96.82	95.00	94.29	90.00	94.48
	GSDAG + Sym	95.51	95.64	94.98	95.14	89.44	93.52
	WM	98.28	99.14	96.00	95.43	92.22	95.56
	WM + Sym	98.28	99.14	96.00	95.43	92.22	95.56
C_L	MLP	89.66	94.83	83.86	88.57	90.00	70.00
	MLP + Sym	88.79	94.83	82.71	87.71	88.32	68.34
	GS	96.55	96.55	90.14	94.86	91.13	89.46
	GS + Sym	85.34	91.38	88.14	91.71	86.17	82.82
	GSDAG	98.28	98.28	91.00	95.43	91.19	92.49
	GSDAG + Sym	83.62	93.10	88.71	94.29	86.67	91.13
	WM	96.55	98.28	91.86	96.00	91.67	93.33
	WM + Sym	96.55	98.28	91.86	96.00	91.67	93.33

Table 2 reports hitrates for mean and std C_D and C_L. The trends are consistent with the C_p results. WINDMIL achieves the best or comparable performance across the integrated load quantities while preserving exact reflected-test consistency. For drag, WINDMIL obtains the highest hitrates in the random split (98.28% for mean C_D and 99.14% for std C_D), the highest mean C_D hitrate in barycentric extrapolation (96.00%), and the highest std C_D hitrate in edge extrapolation (95.56%). For lift, WINDMIL achieves the highest barycentric extrapolation hitrates (91.86% for mean C_L and 96.00% for std C_L) and the highest edge extrapolation hitrates (91.67% and 93.33%).

As with the node-level pressure results, data augmentation improves some baselines but does not enforce exact symmetry. In several cases, GSDAG performs accurately on the original test set but changes under *+Sym* evaluation, especially for C_L. In contrast, WINDMIL gives identical original and reflected-test hitrates by construction. These results show that the benefits of symmetry-consistent pressure prediction extend to integrated wind loading quantities relevant for structural design.

4.3 QUALITATIVE ANALYSIS

Figure 8 shows representative mean and std C_p contours for the barycentric extrapolation split, for both original and reflected test geometries. Both graph-based models recover the main windward and leeward pressure patterns on original geometries, but the largest differences appear on reflected cases: GRAPHSAGE visibly deviates from LES and underpredicts high-variability regions near roof edges and corners, where separated turbulent structures produce sharp gradients. WINDMIL remains closely aligned with the LES contours for both original and reflected configurations, showing that equivariance improves not only global metrics but also spatial accuracy and exact symmetry consistency in aerodynamically critical regions. Additional random-split contours and discussion are provided in Appendix B.

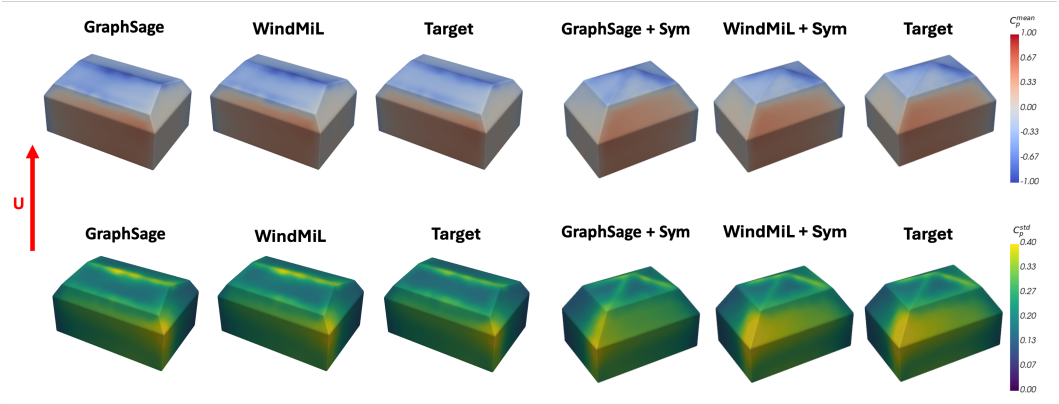

Figure 8: Qualitative comparison for the extrapolation split. Columns show GRAPHSAGE, WINDMIL, and LES ground truth for original and reflected test geometries; rows show mean and std C_p.

5 CONCLUSIONS

We introduced WINDMIL, a high-fidelity graph learning framework for wind loading prediction on buildings. The work advances both communities: for ML and graph learning, it introduces a realistic LES-based benchmark for studying data scarcity, graph geometry, extrapolation, and exact symmetry consistency; for wind engineering, it provides a first LES-based, physically consistent surrogate aligned with pressure coefficients and integrated drag/lift quantities. The dataset contains 462 LES simulations of low-rise buildings with continuous roof-shape variation and multiple wind directions, with each case requiring more than 24 hours on 64 CPU cores with LES.

Reflection-augmented GRAPHSAGE improves over the non-augmented baseline, but it does not guarantee symmetry and still changes under reflected testing. WINDMIL preserves identical predictions under the valid reflection transformation by construction and achieves the best performance for node-level pressure coefficients and graph-level integrated loads across random, barycentric extrapolation, and edge-extrapolation splits. Thus, a physically correct inductive bias can improve extrapolation and guarantee consistency.

This work is a first step, not a replacement for established wind engineering workflows. The dataset is limited to isolated low-rise buildings from three roof archetypes, a fixed footprint family, and a single atmospheric boundary layer profile. Because wind loading is safety-critical engineering task, surrogates should not replace building codes, wind tunnel testing, or validated CFD, but should be used alongside. Their intended role is early-stage exploration, sensitivity analysis, and ML benchmarking, with human oversight and domain-specific validation.

AI USE STATEMENT

Generative AI tools were used to edit, restructure this manuscript and improve readability. We have not used Generative AI to run the reported simulations or model experiments. We take responsibility for the final content of this work,

ETHICS STATEMENT

The study uses simulated building geometries and does not involve people or personal data. The principal risk is misuse of a learned surrogate in safety-critical structural design. The model is intended for research and early-stage exploration, not as a replacement for applicable codes, wind-tunnel testing, or validated CFD. Deployment outside the documented geometry and atmospheric regimes requires independent validation and qualified engineering oversight.

REPRODUCIBILITY STATEMENT

Sections 2–4 describe the data generation, architecture, evaluation protocol, and results. Appendix A gives the available training and metric details. The dataset, the model, and the code will be available and open-source upon acceptance of the manuscript.

REFERENCES

Kelsey R Allen, Tatiana Lopez Guevara, Yulia Rubanova, Kim Stachenfeld, Alvaro Sanchez-Gonzalez, Peter Battaglia, and Tobias Pfaff. Graph network simulators can learn discontinuous, rigid contact dynamics. In *Conference on Robot Learning*, pp. 1157–1167. PMLR, 2023.

Hatem Alrawashdeh and Ted Stathopoulos. Wind pressures on large roofs of low buildings and wind codes and standards. *Journal of Wind Engineering and Industrial Aerodynamics*, 147:212–225, 2015.

Kei Ambo, Hiroaki Nagaoka, David A Philips, Chris Ivey, Guillaume A Brès, and Sanjeeb T Bose. Aerodynamic force prediction of the laminar to turbulent flow transition around the front bumper of the vehicle using dynamic-slip wall model les. In *AIAA Scitech 2020 Forum*, pp. 0036, 2020.

Kenneth Atz, Francesca Grisoni, and Gisbert Schneider. Geometric deep learning on molecular representations. *Nature Machine Intelligence*, 3(12):1023–1032, 2021.

Mark Benjamin and Gianluca Iaccarino. A systematic dataset generation technique applied to data-driven automotive aerodynamics. *APL Machine Learning*, 3(1), 2025.

Bert Blocken. Computational fluid dynamics for urban physics: Importance, scales, possibilities, limitations and ten tips and tricks towards accurate and reliable simulations. *Building and Environment*, 91:219–245, 2015.

Bert Blocken, Ted Stathopoulos, and JPAJ Van Beeck. Pedestrian-level wind conditions around buildings: Review of wind-tunnel and cfd techniques and their accuracy for wind comfort assessment. *Building and Environment*, 100:50–81, 2016.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Cadence Design Systems. Fidelity charles, 2022. URL `https://community.cadence.com/`. Accessed: April 14, 2025.

Wenming Cao, Zhiyue Yan, Zhiquan He, and Zhihai He. A comprehensive survey on geometric deep learning. *IEEE Access*, 8:35929–35949, 2020.

Wenming Cao, Canta Zheng, Zhiyue Yan, and Weixin Xie. Geometric deep learning: progress, applications and challenges. *Science China. Information Sciences*, 65(2):126101, 2022.

Zhaiyu Chen, Yilei Shi, Liangliang Nan, Zhitong Xiong, and Xiao Xiang Zhu. Polygnn: Polyhedron-based graph neural network for 3d building reconstruction from point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 218:693–706, 2024.

Mattia Fabrizio Ciarlatani, Zhu Huang, David Philips, and Catherine Gorlé. Investigation of peak wind loading on a high-rise building in the atmospheric boundary layer using large-eddy simulations. *Journal of Wind Engineering and Industrial Aerodynamics*, 236:105408, 2023.

Jorg Franke, Antti Hellsten, K Heinke Schlunzen, and Bertrand Carissimo. The cost 732 best practice guideline for cfd simulation of flows in the urban environment: a summary. *International Journal of Environment and Pollution*, 44(1-4):419–427, 2011.

Kai Fukami, Koji Fukagata, and Kunihiko Taira. Assessment of supervised machine learning methods for fluid flows: K. fukami et al. *Theoretical and Computational Fluid Dynamics*, 34(4):497–519, 2020.

Jianghang Gu, Yuntian Chen, Yuanwei Bin, and Shiyi Chen. Geoformer: Mesh-free geometry-to-flow alignment framework for real-time aerodynamics on non-watertight vehicle geometries. *Physics of Fluids*, 38(1), 2026.

Rishi Gurnani, Christopher Kuenneth, Aubrey Toland, and Rampi Ramprasad. Polymer informatics at scale with multitask graph neural networks. *Chemistry of Materials*, 35(4):1560–1567, 2023.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1025–1035, 2017.

Jiaqi He, Xiangwen Luo, and Yiping Wang. Drivaer transformer: A high-precision and fast prediction method for vehicle aerodynamic drag coefficient based on the drivaernet++ dataset. *arXiv preprint arXiv:2504.08217*, 2025.

John Hochschild and Catherine Gorlé. Comparison of measured and les-predicted wind pressures on the space needle. *Journal of Wind Engineering and Industrial Aerodynamics*, 249:105749, 2024. ISSN 0167-6105. doi: https://doi.org/10.1016/j.jweia.2024.105749. URL https://www.sciencedirect.com/science/article/pii/S0167610524001120.

Sien Hu, Qi Jin, Chenyu Gao, Xijun Zhang, Mingcheng Lu, Yan He, Dianming Chu, and Wenjuan Bai. The new paradigm of computational fluid dynamics: Empowering computational fluid dynamics with machine learning. *Physics of Fluids*, 37(8), 2025.

Yunjae Hwang and Catherine Gorlé. Large-eddy simulations of wind-driven cross ventilation, part1: validation and sensitivity study. *arXiv preprint arXiv:2204.00786*, 2022.

Yunjae Hwang and Catherine Gorlé. Large-eddy simulations to define building-specific similarity relationships for natural ventilation flow rates. *Flow*, 3:E10, 2023.

Sam Jacob Jacob, Markus Mrosek, Carsten Othmer, and Harald Köstler. Deep learning for real-time aerodynamic evaluations of arbitrary vehicle shapes. *arXiv preprint arXiv:2108.05798*, 2021.

Charilaos I Kanatsoulis and Alejandro Ribeiro. Graph neural networks are more powerful than we think. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7550–7554. IEEE, 2024.

Yusik Kim, Ian P Castro, and Zheng-Tong Xie. Divergence-free turbulence inflow conditions for large-eddy simulations with incompressible flow solvers. *Computers & Fluids*, 84:56–68, 2013.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Mohammad Amin Nabian, Chang Liu, Rishikesh Ranade, and Sanjay Choudhry. X-meshgraphnet: Scalable multi-scale graph neural networks for physics simulation. *arXiv preprint arXiv:2411.17164*, 2024.

Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine learning for molecular simulation. *Annual review of physical chemistry*, 71(1):361–390, 2020.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International conference on learning representations*, 2020.

Duncan A Phillips, Michael J Soligo, et al. Will cfd ever replace wind tunnels for building wind simulations? *International journal of high-rise buildings*, 8(2):107–116, 2019.

Erich J Plate and Harald Kiefer. Wind loads in urban areas. *Journal of Wind Engineering and Industrial Aerodynamics*, 89(14-15):1233–1256, 2001.

Theodore Potsis, Yoshihide Tominaga, and Ted Stathopoulos. Computational wind engineering: 30 years of research progress in building structures and environment. *Journal of Wind Engineering and Industrial Aerodynamics*, 234:105346, 2023.

Xiaoyu Qi, Ming Yang, and Guang Tong. Predicting drag coefficients of vehicle geometries: A pointnet++ point cloud surrogate model approach. *AIP Advances*, 15(10), 2025.

Yinggeer Quan, Yukio Tamura, M. Matsui, S.Y. Cao, and A. Yoshida. Tpu aerodynamic database for low-rise buildings. *Proceedings of 12th International Conference on Wind Engineering*, pp. 1615–1622, 01 2007.

PJ Richards and RP Hoxey. Pressures on a cubic building—part 1: Full-scale results. *Journal of Wind Engineering and Industrial Aerodynamics*, 102:72–86, 2012.

Yulia Rubanova, Alvaro Sanchez-Gonzalez, Tobias Pfaff, and Peter Battaglia. Constraint-based graph network simulator. *arXiv preprint arXiv:2112.09161*, 2021.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pp. 8459–8468. PMLR, 2020.

An Ping Song, Xin Yi Di, Xiao Kang Xu, and Zi Heng Song. Meshgraphnet: An effective 3d polygon mesh recognition with topology reconstruction. *IEEE Access*, 8:205181–205189, 2020.

Henry W Tieleman. Wind tunnel simulation of wind loading on low-rise structures: a review. *Journal of wind engineering and industrial aerodynamics*, 91(12-15):1627–1649, 2003.

Themistoklis Vargiemezis. *Modeling Wind Loading on Low-Rise Buildings in Urban Environments: Leveraging Large-Eddy Simulations and Deep Neural Networks to Quantify Interference Effects*. Stanford University, 2024.

Themistoklis Vargiemezis and Catherine Gorle. Wind tunnel experiments on realistic low-rise building in urban area, 2024. URL `https://www.designsafe-ci.org/data/browser/public/designsafe.storage.published/PRJ-2571/#detail-4070145275427230191-242ac118-0001-012/?version=2`.

Themistoklis Vargiemezis and Catherine Gorlé. Predicting wind-induced interference effects on a low-rise building in a realistic urban area using large-eddy simulations. *Journal of Wind Engineering and Industrial Aerodynamics*, 2025a.

Themistoklis Vargiemezis and Catherine Gorlé. From large-eddy simulations to deep learning: A u-net model for fast urban canopy flow predictions. *arXiv preprint arXiv:2507.06533*, 2025b.

Themistoklis Vargiemezis and Catherine Gorlé. A predictive large-eddy simulation framework for the analysis of wind loads on a realistic low-rise building geometry. *Journal of Wind Engineering and Industrial Aerodynamics*, 256:105950, 2025c.

Parsa Vatani, Mohamed Elrefaie, Farhad Nazarpour, and Faez Ahmed. Tripoptimizer: Generative three-dimensional shape optimization and drag prediction using triplane variational autoencoder networks. *Physics of Fluids*, 37(12), 2025.

AW Vreman. An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of fluids*, 16(10):3670–3681, 2004.

XJ Wang, QS Li, and BW Yan. Full-scale measurements of wind pressures on a low-rise building during typhoons and comparison with wind tunnel test results and aerodynamic database. *Journal of Structural Engineering*, 146(10):04020196, 2020.

Tailin Wu, Qinchen Wang, Yinan Zhang, Rex Ying, Kaidi Cao, Rok Sosic, Ridwan Jalali, Hassan Hamam, Marko Maucec, and Jure Leskovec. Learning large-scale subsurface simulations with a hybrid graph network simulator. In *proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4184–4194, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Huijun Zhang, Yaxin Liu, Chongyu Zhang, and Ningyun Li. Machine learning methods for weather forecasting: A survey. *Atmosphere*, 16(1):82, 2025.

Zebin Zhang, Chuanrui Wang, Shizhao Jing, and Wenhao Wang. A survey on generative methods in automotive aerodynamic shape optimization. *Journal of Computational Design and Engineering*, pp. qwaf145, 2026.

A EXPERIMENTAL SETUP AND TRAINING DETAILS

This appendix provides additional details on the experimental setup used to train and evaluate all models. The same training protocol is used for WINDMIL and the baseline models unless otherwise stated. All reported results use the same fixed data splits, evaluation metrics, and reflected-test protocol described in the main text.

A.1 DATA SPLITS

We evaluate all models on three splits of increasing difficulty, defined in the barycentric roof-shape space shown in Fig. 7: random, barycentric extrapolation, and edge extrapolation.

Random split. The random split assigns generated building geometries inside the convex hull to training, validation, and test sets at random. This split evaluates in-distribution generalization across the sampled shape space. The same split definition is used for all wind directions so that the evaluation consistently tests generalization across the generated building family.

Barycentric extrapolation split. The barycentric extrapolation split holds out building geometries located on the boundary of the convex hull in the SDF-based shape space. The remaining interior geometries are used for training and validation. This split evaluates whether the models can generalize from interior roof-shape combinations to geometries near the boundary of the generated shape family.

Edge extrapolation split. The edge extrapolation split provides the strictest out-of-distribution evaluation. Models are trained on geometries sampled from the region spanned primarily by the flat and hip roof archetypes and tested on geometries located near the gable-roof edge of the convex hull. The same split is applied for all seven wind directions, resulting in 361 training/validation cases and 105 test cases. This split evaluates whether models can extrapolate toward a roof archetype that is underrepresented or absent from the training region.

A.2 REFLECTED-TEST EVALUATION

For each split, we evaluate models on both the original test set and an extended reflected-test set denoted by *+Sym*. The *+Sym* setting augments the held-out test geometries with their reflected counterparts. Training is performed on the original geometries unless otherwise specified.

For non-equivariant models, differences between the original and *+Sym* evaluations indicate reflection inconsistency. For WINDMIL, predictions are identical under the valid reflection transformation by construction. This protocol allows us to distinguish between approximate symmetry learned from data augmentation and exact symmetry enforced in the model architecture.

A.3 MODELS AND BASELINES

We compare the following models:

- **MLP**: A pointwise multi-layer perceptron that treats each surface node independently. It predicts local pressure quantities from node-level geometric features and wind direction, without using graph connectivity.
- **GraphSAGE (GS)**: A non-equivariant graph neural network that uses message passing over the building surface graph. This baseline tests the effect of graph structure without explicit reflection symmetry.
- **GraphSAGE with data augmentation (GSDAG)**: A GRAPHSAGE model trained with reflected geometries included in the training data. This baseline tests whether exposure to both orientations is sufficient to learn the reflection symmetry.
- **WINDMIL**: The proposed reflection-equivariant graph neural network. WINDMIL processes original and reflected inputs with a shared graph encoder and averages the resulting embeddings to enforce reflection consistency by construction.

All graph-based models use the same graph construction and node features. Each building surface is represented as a graph whose nodes correspond to sampled surface points and whose edges are defined by spatial proximity. The node features are the normalized coordinates and outward unit normal components,

$$(x, y, z, n_x, n_y, n_z).$$

A.4 Training hyperparameters

All models are trained using the Adam optimizer with a learning rate of 10^{-3}. Unless otherwise stated, the graph models use a hidden dimension of 64 and are trained for 100 epochs. The same optimizer settings and training schedule are used across baselines to isolate the effect of graph structure, data augmentation, and architectural equivariance.

The training objective is mean squared error between predicted and LES-computed targets. For node-level prediction, the loss is computed over surface-node pressure coefficients. For graph-level prediction, node embeddings are aggregated to predict integrated drag and lift coefficients. Separate models are trained for the mean and standard deviation targets.

Table 3: Training hyperparameters used for all models unless otherwise stated.

Hyperparameter	Value
Optimizer	Adam
Learning rate	10^{-3}
Hidden dimension	64
Number of epochs	100
Loss function	Mean squared error
Node features	(x, y, z, n_x, n_y, n_z)
Node-level targets	Mean/std C_p
Graph-level targets	Mean/std C_D, mean/std C_L

A.5 Compute resources

The LES data generation dominates the computational cost of this study. As described in the main text, each LES case requires approximately 24 hours on 64 CPU cores. In contrast, the ML training cost is small. All experiments were run on an NVIDIA Tesla V100S GPU with 32 GB of VRAM. The models use lightweight architectures with hidden dimension 64, batch size 8, and 100 training epochs. Training typically requires 3-5 GB of GPU memory and approximately 10-15 minutes per model, while inference on the full test set takes less than 10 seconds.

A.6 Evaluation metrics

We report root mean squared error (RMSE), mean absolute error (MAE), coefficient of determination (R^2), and hitrate. For a set of predictions $\hat{y}_i$ and LES targets y_i, RMSE and MAE are defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}, \qquad \mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|. \tag{5}$$

The coefficient of determination is computed as

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \tag{6}$$

where $\bar{y}$ is the mean of the LES targets.

The hitrate measures the fraction of predictions within a prescribed absolute tolerance of the LES target:

$$\mathrm{HR} = \frac{100}{N}\sum_{i=1}^{N}\mathbb{I}\left(|\hat{y}_i - y_i| \le \epsilon\right). \tag{7}$$

For pressure coefficients, we use $\epsilon = 0.10$ for mean C_p and $\epsilon = 0.05$ for std C_p, corresponding to approximately 10% of the maximum observed values. The same thresholding logic is used for integrated drag and lift coefficients.

A.7 Reproducibility

All reported models are evaluated on the same train/validation/test splits and the same reflected-test protocol. Full reproducible scripts, data split files, preprocessing routines, model definitions, and environment configuration files will be provided in the supplemental code after acceptance. These scripts reproduce the random split, barycentric extrapolation split, edge extrapolation split, reflected-test evaluations, and all reported metrics for the proposed model and baselines.

B Qualitative contours

To complement the barycentric extrapolation contours shown in the main text, we provide additional qualitative results for the random split in Fig. 9. Predictions from the baseline GraphSAGE and the proposed WindMiL are compared against the LES ground truth. For brevity, we omit MLP visualizations, since the graph-based models provide stronger baselines in the quantitative experiments.

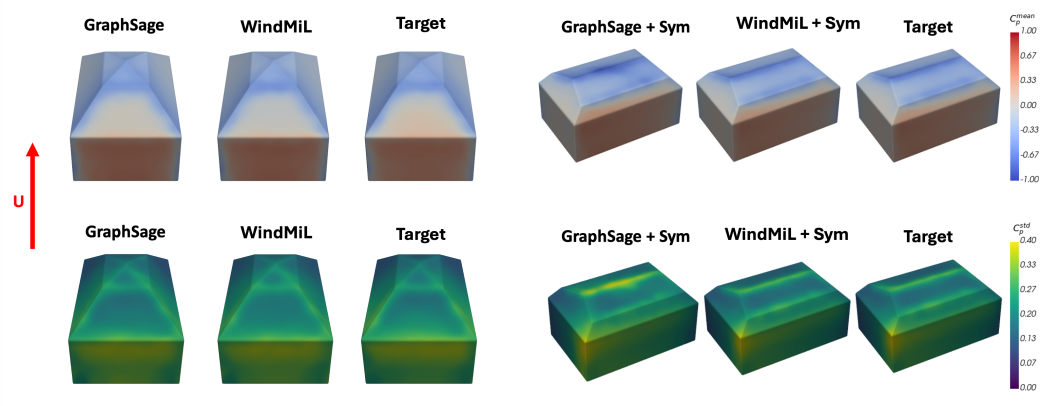

Figure 9: Contour plots of mean and std C_p for the random split. Columns show GraphSAGE, WindMiL, and LES ground truth; rows correspond to mean C_p and std C_p.

For the random test split shown in Fig. 9, both graph-based models capture the main spatial patterns of the pressure coefficients. They reproduce high-pressure regions on windward surfaces and low-pressure regions on leeward and roof surfaces, with smooth gradients that qualitatively agree with the LES reference. For mean C_p, both models provide close visual agreement with the ground truth, consistent with the quantitative results in Table 1. For std C_p, both models identify regions of high variability near roof edges and building corners, where flow separation and vortex shedding are expected to produce larger pressure fluctuations.

When reflected geometries are included in the test set, the qualitative behavior follows the trends reported in the main text. GraphSAGE can exhibit visible discrepancies on reflected configurations, particularly near geometrically and aerodynamically sensitive roof regions. In contrast, WindMiL remains visually consistent with the LES contours under reflection, supporting the quantitative finding that architectural equivariance enforces exact symmetry consistency while preserving spatial accuracy.