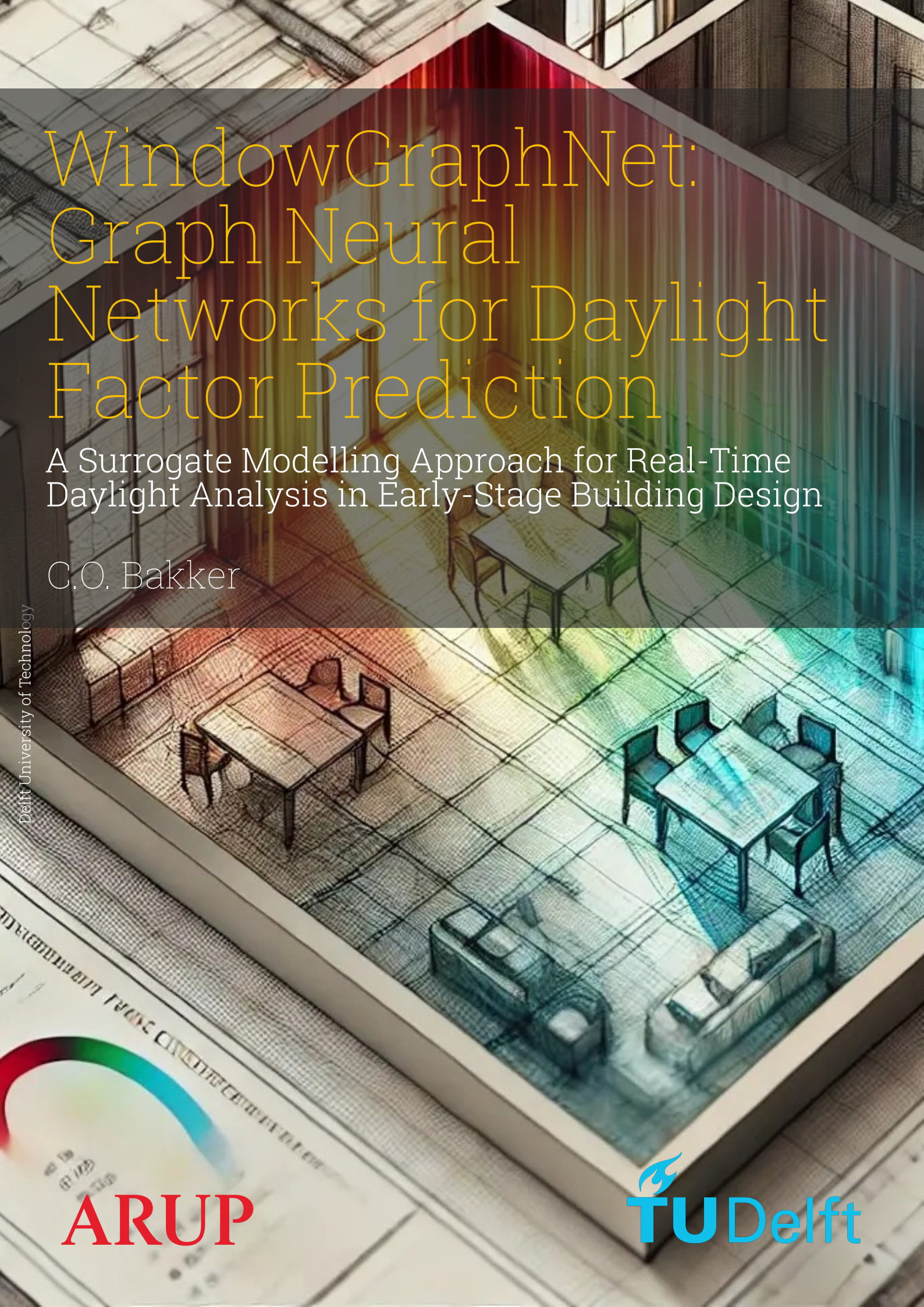


WindowGraphNet: Graph Neural Networks for Daylight Factor Prediction

A Surrogate Modelling Approach for Real-Time Daylight Analysis in Early-Stage Building Design

C.O. Bakker

Delft University of Technology

ARUP

TU Delft

WindowGraphNet: Graph Neural Networks for Daylight Factor Prediction

A Surrogate Modelling Approach for Real-Time Daylight Analysis in Early-Stage Building Design

by

C.O. Bakker

Student number:	4934288	
Project duration:	February 18, 2025 – October 18, 2025	
Thesis committee:	Assoc. Prof. Dr. Ir. F. P. van der Meer,	TU Delft, supervisor
	Asst. Prof. Dr. Ir. I. B. C. M. Rocha,	TU Delft, second supervisor
	Assoc. Prof. Dr. M. Turrin,	TU Delft, comittee member
	Ir. S. Lut,	Arup, company supervisor

Cover:	OpenAI. (2025). DALL-E (Version 3) [Artificial intelligence system]. https://openai.com/index/dall-e-3/ (modified)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

ARUP

TU Delft

“You can see what a ray is if you observe the sun’s light entering a dark room through a narrow opening. It extends in a straight line and impacts […] through the air on the other side: it settles there and does not slip or fall.”

— Marcus Aurelius, *Meditations* 8.57

”All models are wrong, but some are useful.”

- George E.P. Box

Preface

This thesis concludes a nine-month journey undertaken as part of the Master's programme in Civil Engineering at TU Delft, track Structural Engineering. Coming from a background in structural engineering, this project took me well outside that familiar domain. Exploring daylight modelling through graph-based machine learning methods was both challenging and deeply rewarding, and working on a topic that sits at the intersection of architecture, computation, and machine learning laid the foundation for a direction I hope to continue pursuing in my future work.

I would like to express my sincere gratitude to my supervisors at TU Delft, Frans van der Meer and Iuri Rocha, for their guidance, support, and many insightful discussions. Their openness, analytical clarity, and encouragement were essential throughout the research process. I am equally grateful to Michela Turrin for serving on my committee.

My thanks also go to Simon Lut and the Digital Services team at Arup for providing the case and for welcoming me into such an inspiring and energetic environment. My time at Arup was a highlight of this project; I thoroughly enjoyed working with the team and learned a great deal from their openness, expertise, and collaborative spirit. I am deeply grateful for the trust and enthusiasm I received throughout my internship, and I look forward to continuing this journey in the future.

Lastly, I would like to thank my father and sister for carefully proofreading countless pages and versions of this thesis. Their patience, support, and sharp eye for detail were indispensable throughout the writing process. I am also grateful to Isabel for her endless support and willingness to listen to my attempts to explain my thesis, even when the explanations made far more sense to me than to her.

Christiaan Bakker
Amsterdam, November 2025

Abstract

Assessing daylight in architecture increasingly depends on Radiance-based simulations that, although highly accurate, are too computationally demanding for rapid iteration during early-stage design. As layouts shift, openings are repositioned, or geometries become more complex, the turnaround time of physically based simulation limits the designer's ability to explore alternatives. Surrogate models offer a promising solution, yet most existing approaches rely on artificial neural networks (ANNs) trained on tightly controlled parametric datasets and evaluated only within their training distribution. As a result, their reliability for real, highly variable design geometries remains uncertain.

This thesis investigates graph neural networks (GNNs) as a new surrogate modelling paradigm for daylight prediction. Unlike vector-based ANNs, GNNs represent spatial relationships explicitly, allowing the model to "reason" about how light propagates through space rather than inferring patterns solely from coordinate inputs. To the best of the author's knowledge, this is the first formulation of daylight factor (DF) prediction as a graph-learning task, representing rooms as heterogeneous graphs in which window and sensor nodes exchange geometric and photometric information such as distance, direction, and relative orientation.

A deliberately restricted training setup, limited to simple square rooms, is used to test whether GNNs can generalise beyond the conditions they are trained on. The models are then evaluated on a wide range of unseen geometries, including elongated, asymmetric, rotated, and self-occluding layouts, alongside ANN baselines from the daylight literature. The proposed model, *WindowGraphNet*, consistently preserves the spatial structure of DF distributions across these unseen cases, whereas coordinate-anchored ANNs often fail when room form or orientation deviates from their training domain. The GNN's main limitation occurs in deeply recessed or fully occluded areas, where long light paths fall beyond the model's message-passing range, leading to systematic overestimation of low-illumination regions.

These findings demonstrate that generalisation in surrogate daylight modelling is governed primarily by the representation: embedding relational and geometric structure enables robustness that conventional ANNs cannot achieve through data alone. *WindowGraphNet* therefore establishes a new methodological foundation for surrogate daylight tools, with an extensible graph structure that can be expanded to incorporate surface nodes, external obstructions, or climate-based inputs. With further training on diverse datasets, such graph-based surrogates have the potential to deliver fast, geometry-aware daylight feedback directly within architectural design environments, supporting more informed and iterative decision-making in early-stage design.

Contents

List of Figures

List of Tables

1

Introduction

In architectural design, daylight operates simultaneously as a material and a constraint: it shapes spatial quality while influencing energy performance. How light enters and moves through a space determines its atmosphere, depth, and legibility; it reveals form, articulates surfaces, and provides orientation. The geometry and materiality of the built environment mediate the balance between direct sunlight and diffuse skylight, shaping the resulting light distribution within interiors [1]. As people spend an increasing proportion of their lives indoors, the character of this light has become a critical aspect of environmental design, linking occupants to natural rhythms that artificial lighting cannot replicate.

Physiologically, daylight is a key determinant of human health, well-being, and comfort. Natural light regulates circadian rhythms, mood, cognitive performance, and sleep quality [2]. Because the circadian system is highly sensitive to the intensity and spectral composition of daylight, insufficient exposure can lead to disruption, fatigue, and mood disorders [3]. Daylight also enhances visual perception and spatial awareness, reduces eye strain, and supports productivity and learning outcomes in work and educational environments [4].

Given its strong influence on spatial experience and human well-being, daylight must be deliberately integrated into design. This requires understanding the different forms of daylight and their contributions indoors: direct sunlight provides high-intensity illumination but may cause glare and overheating, diffuse skylight produces more uniform distribution, and reflected light from external or internal surfaces further shapes perceived brightness [3, 5]. Architectural daylighting strategies balance these components to optimise visual comfort, spatial quality, and energy performance. As a result, contemporary regulations increasingly include quantitative requirements, for example EN 17037 evaluates the fraction of the working plane that receives sufficient daylight under standard sky conditions, to guide architects in meeting minimum daylight quality.

Designers have long relied on experience, intuition, and simplified rules of thumb to estimate daylight performance during the early stages of design. Such methods offer rapid feedback but quickly break down for complex or unconventional geometries, where light propagation cannot be easily inferred by eye or captured through approximate formulas. When a design includes deep plan spaces, irregular layouts, or multiple openings, as illustrated in Figure 1.1, a physically based simulation becomes necessary to assess daylight availability with confidence. Importantly, such configurations are not exceptions but typical of early-stage practice, where designers rapidly iterate on massing, opening placement, and interior layout. These variations produce a wide range of spatial conditions that differ substantially from one design iteration to the next. Any practical daylight assessment method must therefore remain robust to forms and arrangements that fall outside narrowly defined geometric templates.

The most widely used simulation frameworks, such as Radiance and its Grasshopper-based interface Honeybee, employ ray tracing to model the transport of light through space with high physical fidelity [8]. Figure 1.2 shows a typical workflow: the designer defines a room geometry and internal sensor grid in Grasshopper; Honeybee converts these inputs into a Radiance scene and executes the simulation under a standardised sky condition. The simulation outputs an illuminance value at each sensor point,

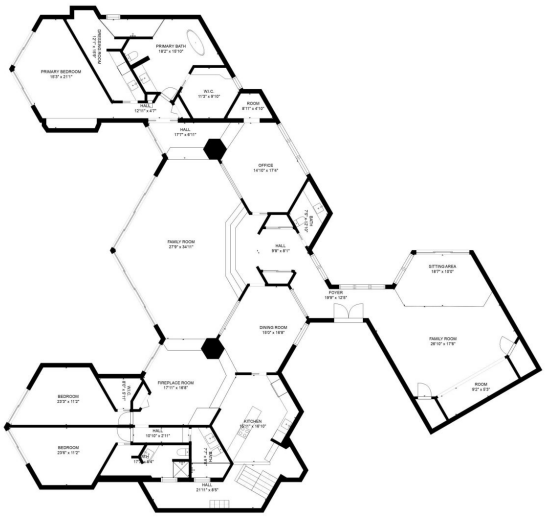


Figure 1.1: Example of a highly irregular floor plan illustrating the types of complex geometries that challenge rule-of-thumb daylight estimation and require physically based simulation. Adapted from CubiCasa [6, 7].

from which the daylight factor (DF) is computed as the ratio of indoor to outdoor illuminance. Although the DF is a scalar ratio, in practice it is evaluated across a dense grid of sensor locations, producing a spatial field commonly referred to as the DF grid (Figure 1.3). While such simulations provide accurate and reproducible results, their computational cost increases rapidly with scene complexity and sensor resolution. Even a single Radiance run may take several seconds to minutes, making it impractical to evaluate hundreds or thousands of design alternatives in workflows where geometry changes continuously [9].

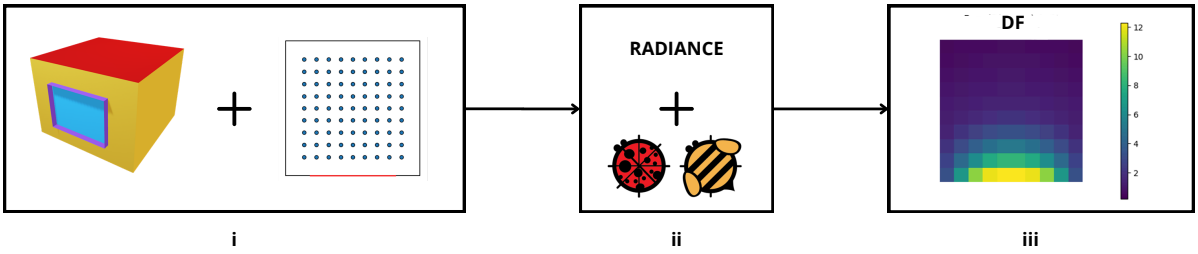


Figure 1.2: Overview of the DF simulation workflow. (i) A room geometry with an internal grid of sensor points defines the simulation setup. (ii) This input is processed by Radiance through the Honeybee interface, where backward ray tracing computes illuminance under a standardised overcast sky. (iii) The output is a grid of DF values, shown as a spatial distribution across the room, which provides the ground truth data for surrogate modelling.

This time constraint creates a methodological bottleneck: the designer must choose between speed and precision, either rely on intuitive but unreliable heuristics, or wait for accurate but slow simulations. As a result, daylight performance feedback is often deferred to later project stages, when major design decisions have already been made and optimisation potential is limited [9].

Surrogate modelling provides a way to bridge this gap. A surrogate model learns a mapping from input parameters to simulation outputs, offering a fast approximation of a more expensive computational process [10]. Such models have been successfully applied in adjacent domains such as energy and thermal-comfort prediction, where rapid evaluation is more valuable than exact physical accuracy [11, 12]. In daylight analysis, a surrogate can learn from Radiance-generated data to predict metrics such as the DF directly from geometric and material parameters, enabling near-instantaneous feedback for new design variants.

Machine learning (ML) provides a powerful framework for constructing surrogate models because of its ability to approximate complex, non-linear relationships directly from data [11, 13]. Among these approaches, artificial neural networks (ANNs) have been widely employed to predict building performance

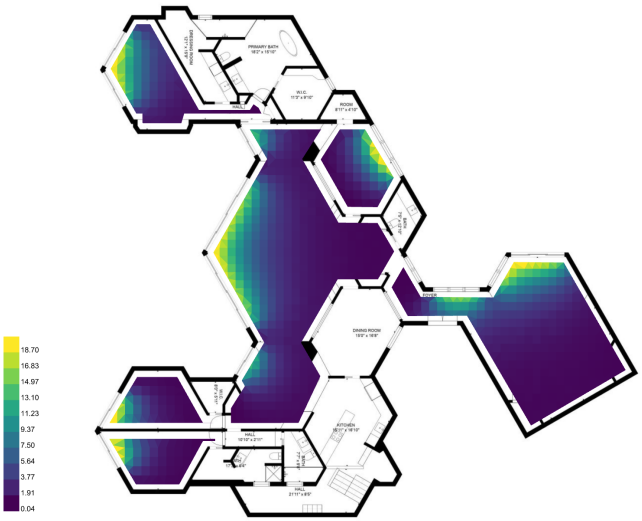


Figure 1.3: DF distribution superimposed on a highly irregular floor plan, illustrating how geometric complexity, multiple openings, and deep spatial configurations lead to strong spatial gradients in daylight availability. Adapted from CubiCasa [6, 7].

metrics from geometric and material parameters. By learning the mapping between inputs and outputs without relying on explicit physical equations, ANNs can capture interactions that are difficult to model analytically. However, conventional ANN architectures treat inputs as independent feature vectors, limiting their ability to represent the relational structure of geometry that governs spatial phenomena in built environments.

These limitations are reflected in current surrogate daylight models. Most ANN-based approaches are trained and tested on simplified, parametrically generated rooms and are evaluated only within the same distribution on which they were trained. They are rarely seldom evaluated on geometries that differ in form, scale, or configuration. Yet designers routinely produce layouts that fall outside these controlled parametric spaces. As a result, the reliability and applicability of existing machine-learning-based daylight predictors for real, highly variable architectural geometries remain largely unknown.

Graph neural networks (GNNs) extend ML to data represented as graphs, where relationships between entities are as important as the entities themselves [14, 15]. Unlike conventional architectures such as ANNs, which assume a fixed input structure (e.g., vectors), GNNs operate on arbitrary topologies and propagate information along the graph's edges. Through iterative message passing, they learn representations that encode both local neighbourhood interactions and global structural context.

In this framework, nodes and edges can represent elements such as sensor points, windows, and their geometric connections, allowing GNNs to capture spatial and relational dependencies that are otherwise lost in vector-based models. In the context of daylight prediction, such relational awareness offers a pathway toward more generalisable and physically informed surrogate models capable of predicting daylight distributions across diverse geometries.

Research Gap and Novelty

Machine learning has become a key strategy for accelerating building performance prediction by approximating complex simulation processes through data-driven models. Within daylight research, most existing studies employ ANNs trained on simulated data to predict daylight metrics from geometric and material parameters [16]. This paradigm has demonstrated that surrogate models can effectively reproduce the results of physically based simulations while reducing computational cost.

Three studies in particular, by Han et al. [17], Le-Thanh et al. [18], and Dieguez et al. [19], form the primary methodological benchmarks for this thesis. Collectively, these works illustrate a clear trajectory in feature design: from purely geometric encodings to the inclusion of directional, distance and photometric based features. While these models achieve high accuracy within their respective training domains, their evaluation has been consistently limited to geometries drawn from the same parametric distribu-

tions as the training data. None of these studies systematically examined the models' robustness to unseen typologies or geometric transformations such as rotation, scaling, or spatial rearrangement, factors critical to real-world design variability. In practice, architectural geometries rarely conform to the narrow distributions of a training dataset; even small variations in form, orientation, or window placement can fundamentally alter the daylight response of a space.

More broadly, existing data, driven daylight models, including ANNs and other machine learning formulations, remain primarily based on fixed or implicit spatial representations, such as vectorised features or regular grids. These approaches can capture spatial correlations only indirectly and do not encode the explicit relational dependencies that govern light transfer between architectural elements. A relational modelling approach is therefore needed to capture the geometric dependencies that underlie daylight propagation. However, no published work has applied GNNs to the prediction of daylight. This absence delineates a clear research gap and establishes the novelty of the present study, which introduces a graph-based surrogate modelling framework for daylight prediction and evaluates its generalisation against ANN benchmarks.

Research Aim and Research Questions

The aim of this research is therefore twofold. First, it seeks to develop a graph-based surrogate model for predicting the daylight factor distribution in early-stage building design, focusing on the formulation of geometric and relational features, the selection of graph operators, and the configuration of model architecture. Second, it aims to systematically evaluate the generalisation of this model to unseen geometries by comparing its performance with established ANN-based benchmark models under controlled distribution shifts.

To address this aim, the research is guided by the following questions:

i *Model formulation:* How can daylight factor prediction be represented as a graph learning problem, including the definition of nodes, edges, and feature representations suitable for GNNs?

ii *Architectural exploration:* How do graph operators, model depth, and architectural configuration influence the predictive performance of GNN-based daylight surrogates?

iii *Generalisation analysis:* To what extent can GNN-based surrogate models generalise to unseen room typologies and geometric transformations compared with ANN-based benchmarks?

Scope of the Study

The scope of this thesis is defined by its focus on the development and evaluation of graph-based surrogate models for predicting the DF distribution in early-stage building design. The work is situated within the domain of data-driven daylight modelling and examines how geometric and relational representations can be used to approximate simulation-based results efficiently. All analyses are based on synthetic data generated through parametric modelling in Grasshopper and Radiance simulations.

Conceptual Focus: Why DF and not CBDM

Daylight metrics can broadly be categorised into overcast-based and climate-based measures. Overcast-based metrics, such as the DF and the Vertical Sky Component (VSC), are evaluated under a standardised overcast sky and exclude direct sunlight. In contrast, Climate-Based Daylight Metrics (CBDM), including Daylight Autonomy (DA), Spatial Daylight Autonomy (sDA), and Useful Daylight Illuminance (UDI), incorporate hourly climatic variation and direct solar contributions [20]. While CBDM metrics are required by LEED and WELL [21, 22], DF remains supported by EN 17037 and BREEAM [23–25] and continues to play a central role in early-stage daylight assessment.

The DF is adopted as the target metric because it isolates the geometric and optical determinants of daylight performance under a standardised overcast sky. This makes it particularly suitable for investigating how spatial configuration and window placement govern diffuse light distribution, independent of climatic or temporal variation. Second, by excluding climatic and temporal variability, DF enables a controlled evaluation of feature representations and operator design. Third, its continued use in European daylight standards ensures practical relevance and establishes a clear foundation for future extensions toward climate-based metrics.

Methodological Boundaries

The study is limited to single-room configurations generated through parametric variation of spatial dimensions. Training is conducted exclusively on square-room geometries, with windows placed in the centre of the wall facing the South, to ensure a controlled learning domain, while rectangular and L-shaped configurations are reserved for evaluating generalisation to unseen typologies. All Radiance simulation parameters, including material reflectance, glazing transmission, and sky conditions, are held constant across the dataset so that only geometric variation drives differences in daylight performance. Two modelling paradigms are investigated: ANNs, following established daylight surrogate models in the literature [17–19], and GNNs, which introduce an explicit representation of geometric and relational structure. Model evaluation focuses on predictive accuracy and generalisation to unseen room typologies, whereas geometric transformations such as rotation and scaling are used during feature analysis and operator testing to assess invariance and robustness.

Integration into interactive design tools and multi-room, façade-scale or obstruction aware models lies beyond the present scope. Instead, the work establishes a controlled experimental framework to rigorously examine how graph-based formulations, feature representations, and operator configurations affect model performance and generalisation in daylight prediction.

Thesis Structure

The remainder of this thesis is organised as follows.

- Chapter 2 provides the theoretical and disciplinary background, introducing daylight simulation principles, ML strategies, and graph-based learning concepts.

  Chapter 3 reviews previous work on surrogate models for daylight prediction and identifies the studies that serve as benchmarks for this research. It additionally examines feature representation in GNNs, surveys existing message-passing operators, and discusses design considerations for GNN operator construction.
- Chapter 4 outlines the overall research framework, encompassing dataset generation, feature formulation, model configuration, and the evaluation procedures for both the GNN benchmarking and the final generalisation tests.
- Chapter 5 presents the results and discussion of feature ablation, operator selection, and architectural optimisation, leading to the identification of the final GNN configuration.
- Chapter 6 presents the comprehensive evaluation of all models on the *Final Test Dataset*, analysing how each architecture generalises across progressively more complex and unseen geometries, from rectangular to asymmetric and self-occluding configurations.
- Chapter 7 summarises the main contributions, discusses implications for design workflows, and outlines directions for future research.

2

Background

The background chapter provides the conceptual and disciplinary context for the research. It begins with an overview of daylight simulation in the context of early-stage design decision making, highlighting its architectural relevance and the challenges associated with current simulation-based approaches. Subsequently, different families of machine learning models are outlined, emphasizing their respective inductive biases and suitability for surrogate modelling tasks. Finally, GNNs are introduced conceptually, explaining their working principles and distinct advantages for capturing spatial and relational patterns in architectural data.

2.1. Daylight Simulation and Early-Stage Design

To develop a graph-based model for predicting the DF it is first necessary to understand how DF is defined, computed, and why its simulation poses challenges for data-driven modelling. DF expresses the relationship between the indoor illuminance at a given point and the simultaneous unobstructed outdoor illuminance under a standard overcast sky (Equation 2.1) [23]. By isolating purely geometric and material influences, without climatic or temporal variability, it provides a clear physical signal that captures how light is distributed through space.

$$DF = \frac{E_{\text{in}}}{E_{\text{out}}} \times 100\%. \tag{2.1}$$

The indoor illuminance E_{in} can be decomposed into the sum of three components, as shown in Figure 2.1: the *sky component (SC)*, representing light reaching a point directly from the visible sky; the *externally reflected component (ERC)*, accounting for light reflected off exterior surfaces before entering; and the *internally reflected component (IRC)*, capturing multiple reflections within the room:

$$E_{\text{in}} = E_{SC} + E_{ERC} + E_{IRC}. \tag{2.2}$$

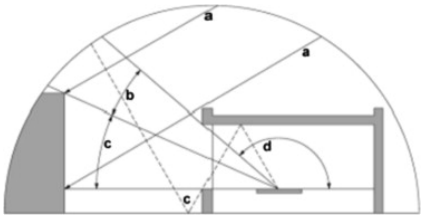


Figure 2.1: Daylight entering a room. a: direct sunlight, b: SC, c: ERC, d: IRC [3].

Regulatory frameworks such as EN 17037 define compliance based on the proportion of the reference plane meeting a target DF. A space typically satisfies the requirement when at least 50% of the reference

plane meets or exceeds the prescribed DF value (e.g., 2%, 4%, 6% for minimum, medium, or high daylight provision).

Accurately verifying such compliance relies on Radiance-based simulations, which remain computationally demanding despite their physical rigour. A single-room Radiance DF simulation typically requires several seconds to a few minutes, depending on model complexity, grid density, and the number of ambient bounces [26, 27]. When repeated across hundreds or thousands of design variants, as is typical in optimisation workflows, the cumulative runtime becomes a bottleneck; motivating the development of faster surrogate approaches based on machine learning.

2.2. From Simulation to Learning Frameworks

The computational cost of simulating DF across large geometric variations motivates the use of data-driven surrogates. Once a sufficient dataset of simulated cases is available, the mapping between geometry and DF can be learned rather than re-simulated. However, the suitability of a learning method depends on how effectively it can represent the spatial and relational structure underlying the daylight distribution. This distinction gives rise to several families of machine learning models, ranging from conventional vector-based approaches such as ANNs, to architectures explicitly designed to encode spatial or topological relations, such as convolutional and GNNs.

2.2.1. Families of Machine Learning Models

A wide range of ML models have been explored for surrogate modelling, each providing a different way of mapping inputs to outputs [28]. These approaches can be grouped according to whether they operate on unstructured feature vectors or exploit spatial relationships between elements.

Non-spatial models. Non-spatial models operate on vectorised inputs and do not represent geometric or topological relations explicitly. They include four main categories commonly used for surrogate modelling:

1. *Kernel-based methods.* Support Vector Regression (SVR) and Gaussian Process Regression (GPR) use kernel functions to quantify similarity between samples. The kernel defines a correlation structure in the input space, allowing the model to capture smooth and continuous relationships between variables [29]. GPR further provides probabilistic predictions, yielding both mean and uncertainty estimates for each output.
2. *Tree-based ensembles.* Decision tree ensembles, such as Random Forests and Gradient Boosting Machines, represent complex relationships by recursively dividing the input space into regions with homogeneous responses [30, 31]. Predictions are obtained by averaging or combining the outcomes of many individual trees, providing a flexible, piecewise representation of nonlinear functions.
3. *Artificial Neural Networks (ANNs).* ANNs consist of multiple layers of linear transformations followed by nonlinear activation functions. By composing these operations, ANNs can approximate a wide range of continuous mappings between inputs and outputs [32]. Their layered structure enables hierarchical feature extraction, making them well suited for high-dimensional surrogate modelling tasks.

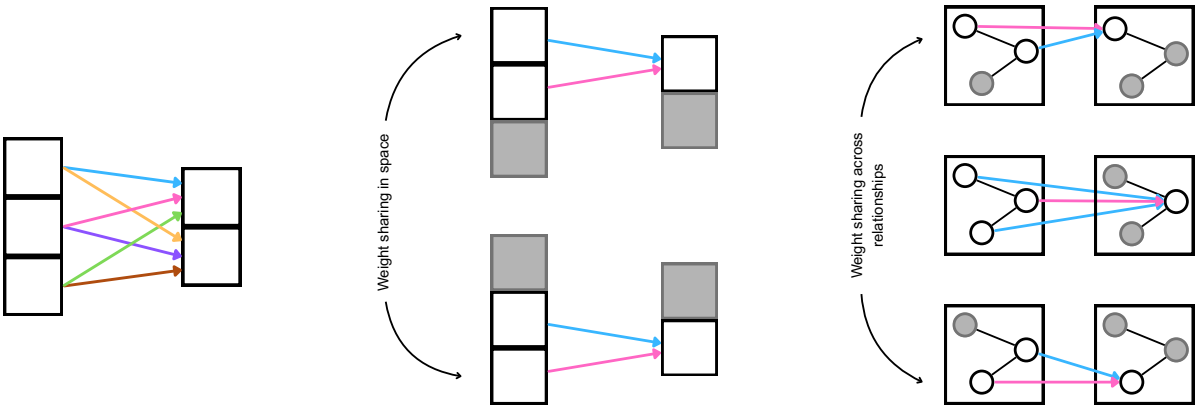
Grid-based models. Convolutional Neural Networks (CNNs) are designed for data defined on regular Euclidean grids, such as images or volumetric fields [33, 34]. They apply shared filters across local neighbourhoods, enabling the model to detect spatial patterns that repeat across the domain. Through convolution and pooling operations, CNNs capture local correlations and progressively build more abstract spatial representations.

2.2.2. Inductive Bias and the Transition to Relational Models

The notion of inductive bias is central to understanding the trade-offs between the model families. Inductive bias refers to the set of assumptions that a learning algorithm makes in order to generalise from finite training data to unseen inputs [35]. Models with explicit inductive biases, such as Gaussian Processes whose kernels encode smoothness or periodicity assumptions, can generalise efficiently when

these assumptions align with the data-generating process, but may underperform when they are violated. In contrast, models with a weak inductive bias, such as fully connected ANNs, are highly flexible and capable of approximating arbitrary functions [32], but require large datasets and careful regularisation to avoid overfitting. This lack of structural assumptions is reflected visually in Figure 2.2a, where every input–output connection is assigned an independent weight.

CNNs illustrate how incorporating the right inductive bias can dramatically improve efficiency and performance: by embedding locality, weight sharing, and translation invariance, they achieve state-of-the-art results in domains where these assumptions hold [36]. Figure 2.2bb illustrates this spatial weight sharing, where the same convolutional kernel is applied across different locations of the input grid. However, their reliance on Euclidean grid structures renders them unsuitable for domains where data are relational, irregular, or topologically complex.



(a) Fully connected **(b)** Convolutional (CNNs) **(c)** Graph convolutional (GNNs)

Figure 2.2: Illustration of inductive biases in neural network architectures. (a) Fully connected layers learn independent weights for every connection, with no sharing. (b) Convolutional layers in Euclidean domains (CNNs) share weights in space, enforcing translation invariance. (c) Convolutional layers in non-Euclidean domains (GNNs) share weights across relationships, enforcing permutation invariance. Both CNNs and GNNs are convolutional in nature, but their inductive biases differ according to the underlying data structure. Panels (a) and (b) are adapted from [37]

But many problems in engineering and science are inherently relational rather than grid-based. Even when they arise in Euclidean space, their dependencies do not follow a uniform lattice structure. Daylight simulation in buildings provides a clear example: the illuminance at a sensor point depends on its spatial relations to windows, walls, and obstructions, but these relationships vary from point to point and cannot be captured by a fixed convolutional kernel. In other words, while the geometry is Euclidean, the interaction topology is not; each sensor interacts with a unique subset of elements at varying distances and orientations. Such structure is therefore more naturally expressed as a graph, where nodes correspond to entities (e.g., sensors or windows) and edges encode geometric relationships such as distance, orientation, or visibility.

GNNs provide a principled framework for learning from such relational data. Rather than assuming a regular spatial grid as in CNNs, they aggregate information over arbitrary neighbourhoods defined by the graph structure, introducing a relational inductive bias that emphasises connectivity rather than spatial regularity [14, 37]. As shown in Figure 2.2cc, message-passing layers extend the principle of convolution to arbitrary graph neighbourhoods by sharing weights across edges rather than spatial positions.

The distinction between ANNs and GNNs thus captures the two extremes of this spectrum. ANNs, with minimal structural assumptions, serve as flexible yet unstructured function approximators. They provide a suitable baseline for assessing how much performance can be achieved without any encoded relational information. In contrast, GNNs introduce a relational inductive bias by defining learning directly over a graph representation of the system, allowing spatial and topological dependencies to be learned explicitly rather than inferred implicitly through feature design. Comparing these two families therefore isolates the effect of incorporating relational structure into the learning process, a central question that underpins the design and evaluation framework developed in the following chapters

The next section introduces the conceptual foundations of GNNs, discussing how graphs are represented, how message passing enables relational learning, and how the Graph Convolutional Network (GCN) serves as a minimal example of this framework. Finally, it outlines key limitations of message-passing GNNs, which motivate the diversity of operators reviewed in this thesis.

2.3. Graph Neural Networks

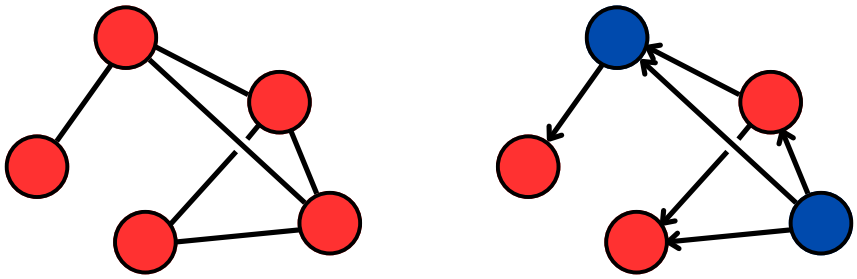
GNNs extend machine learning to data that are most naturally represented as graphs rather than vectors or grids. The family dates back at least to Scarselli et al. (2009), with the Graph Convolutional Network (GCN) of Kipf and Welling (2017) popularising modern message-passing formulations [38, 39]. At their core, GNNs operate through iterative message-passing schemes, where nodes exchange information with their neighbours along edges, aggregate these messages, and update their internal representations. This mechanism allows GNNs to capture both local dependencies and global structure, making them particularly suited to problems where the performance of one element depends on its relationships with others.

In the following subsections, the conceptual foundations of GNNs are introduced. First, the representation of graphs is formalised in terms of nodes, edges, and features. Next, the message-passing framework that underlies most modern GNN architectures is presented. This is followed by a concrete example based on the GCN operator, illustrating how information is aggregated and propagated within a single layer. Finally, the expressive power and known limitations of message-passing GNNs are discussed.

2.3.1. Graph Representation

A graph is a mathematical structure that encodes a set of entities and the relations between them. Formally, a graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of N nodes (or vertices), and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges [40, 41]. An edge $e_{ij} = (v_i, v_j) \in \mathcal{E}$ connects two nodes v_i and v_j. For an undirected graph, e_{ij} represents a bidirectional relation between v_i and v_j, whereas in a directed graph e_{ij} denotes an edge directed from v_i to v_j.

Graphs can be classified as weighted or unweighted[1]. In an unweighted graph, edges simply indicate the presence or absence of a connection, whereas in a weighted graph each edge is associated with a scalar value. More generally, edges may carry feature vectors $\mathbf{e}_{ij} \in \mathbb{R}^c$, where c denotes the dimension of the edge feature space. Similarly, nodes v_i can be enriched with feature representations $\mathbf{x}_i \in \mathbb{R}^d$, with d the dimension of the node feature space [37, 42].



(a) Homogeneous, undirected graph. All nodes belong to the same type (red) and edges are undirected (bidirectional).

(b) Heterogeneous, directed graph. Two node types (red and blue) are connected by directed edges (arrows).

Figure 2.3: Examples of basic graph types. (a) Homogeneous undirected graph with identical node types and symmetric edges. (b) Heterogeneous directed graph with multiple node types and asymmetric relations.

The presence of edges between nodes can be represented by an adjacency matrix $A \in \mathbb{R}^{N \times N}$, where $A_{ij} \neq 0$ indicates an edge from node v_i to v_j. The degree of a node v_i is the number of neighbours it has, and can be stored in a diagonal degree matrix D with entries D_{ii} equal to the degree of node v_i.

In addition to being directed or undirected, graphs can be further classified according to other structural properties. Homogeneous graphs consist of a single node and edge type, whereas heterogeneous

[1] In weighted graphs, edges carry numerical values such as distances, angles, or obstruction factors.

graphs include multiple node or edge types, each potentially with distinct feature spaces [43].

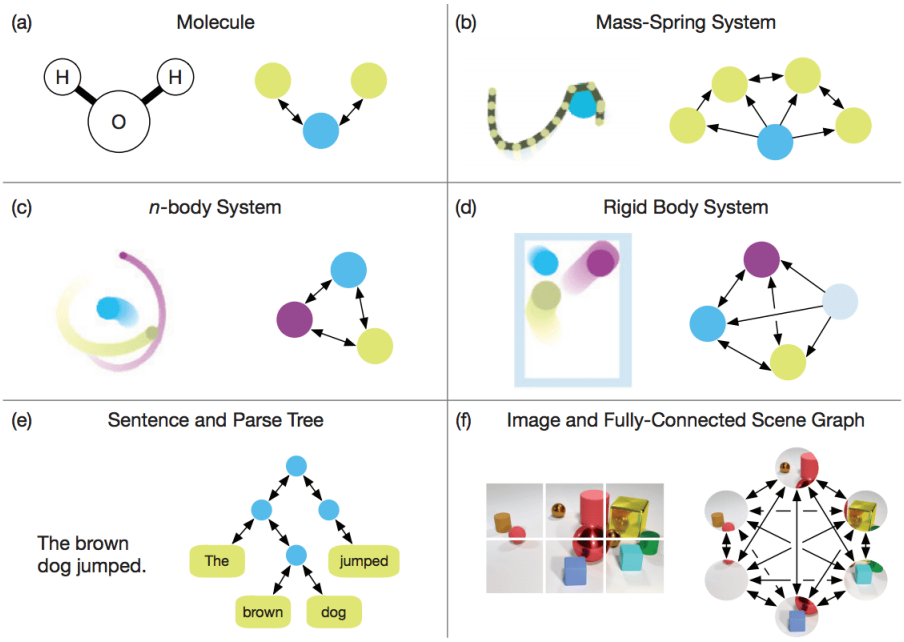


Figure 2.4: Examples of graph representations across domains, including molecules, physical systems, natural language, and images [37].

Beyond these structural categories, graphs also exhibit properties that are particularly important in the machine learning context. A fundamental property of graphs is their lack of a canonical ordering of nodes. This means that the same graph can be represented by different adjacency matrices depending on how nodes are indexed[2]. To handle this, GNNs are designed to be permutation invariant or equivariant[3] with respect to node indexing [40, 43]. Invariance is required when the output is global (e.g. a single performance score for the whole room), while equivariance is required for node-level tasks such as predicting DF at each sensor, where outputs must track the permutation of the input nodes. This ensures that predictions depend only on the actual graph structure and features, not on arbitrary indexing choices in memory. Stability to graph perturbations, such as small changes in connectivity or feature noise, is another critical property that ensures robust transferability across domains [43].

These considerations illustrate more broadly why graphs are powerful: they provide a flexible abstraction that can encode diverse relational systems while respecting properties such as heterogeneity, directionality, permutation equivariance, and stability to perturbations. This flexibility explains their adoption as a unifying representation across many domains, from molecules and physical systems to natural language and images (Figure 2.4) [37].

Beyond the diversity of domains, graphs also differ in how they are constructed. In some cases, the relational structure is explicitly given by the data (e.g., molecular bonds, social ties). In others, it must be induced from raw observations according to a design rule (e.g., k-nearest neighbours, spatial proximity, visibility, or learned affinities) [14, 37]. These are endpoints of a continuum, and many applications combine both.

2.3.2. Problem Settings in Graph Learning

Given such graph representations, GNNs can be applied to a variety of learning tasks, depending on whether predictions are made about nodes, edges, or entire graphs, as shown in Figure 2.5. This

[2]For example, a square room with four sensor nodes can be labelled {1,2,3,4} in clockwise order or {4,3,2,1} in counter-clockwise order. Both representations correspond to the same underlying geometry, but produce different adjacency matrices. A GNN must therefore yield consistent predictions regardless of this relabelling.

[3]Permutation invariance means that reordering nodes does not change the output: $f(P\mathbf{X}) = f(\mathbf{X})$. Permutation equivariance means that the output is permuted in the same way as the input: $f(P\mathbf{X}) = Pf(\mathbf{X})$, where P is any permutation matrix.

taxonomy is widely used across domains and helps situate the DF prediction task within the broader landscape of graph learning.

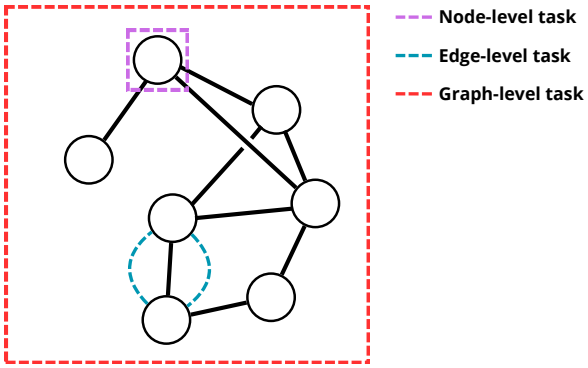


Figure 2.5: Problem settings in graph learning. Node-level tasks predict values for individual nodes (purple box), such as DF values at sensor points in this thesis. Edge-level tasks predict properties of connections (blue circle), such as whether a relationship exists or its weight. Graph-level tasks assign a label or property to the entire graph (red box), such as a global performance score.

Node-level tasks: Predictions are made for individual nodes, given graph connectivity and features. These can be formulated as classification (e.g., predicting the type of an atom in a molecule) or regression (e.g., predicting a continuous property such as atomic energy contributions). Benchmark datasets include *QM9*, where molecular properties are associated with atoms [44], and *ogbn-proteins*, which labels amino acids with biological functions [45]. The task in this thesis, predicting DF values at sensor locations, is a node-level regression problem, as each sensor node receives a continuous daylight prediction.

Edge-level tasks: Predictions concern the properties of edges. In classification form, this often corresponds to link prediction, such as inferring whether a connection exists between two users in a social network. In regression form, edge weights can be predicted, for example the strength of a molecular bond or the distance between two entities. For daylight modelling, edges are conceptually relevant because they encode light transport relations between windows and sensors. However, the aim of this thesis is not to predict the properties of these relations directly, but to use them as features that support node-level predictions of daylight outcomes. Edge-level prediction therefore lies outside the scope of the present work.

Graph-level tasks: Predictions assign a label or property to an entire graph, such as molecular toxicity, image class, or average material strength. Popular benchmarks include *ZINC*, a molecular property prediction dataset [46], and graph classification datasets such as *MNIST superpixels*, where entire digit graphs are labelled by class [47]. In the daylighting context, a graph-level formulation could be used to predict a single aggregate metric, such as the mean DF across a room, but this would discard the spatial richness of DF distributions that is crucial for informing early-stage design decisions. Although not the same task, insights from these benchmark datasets are still instructive.

This taxonomy illustrates the range of problems to which GNNs can be applied and clarifies why node-level regression is the most appropriate choice for this thesis. Unlike graph-level prediction, it preserves spatial detail at the sensor level, and unlike edge-level prediction, it aligns directly with the modelling objective: estimating daylight outcomes at sensor nodes rather than properties of the relations between them. Nevertheless, the broader literature on edge- and graph-level tasks remains valuable, as the associated benchmark datasets and operator evaluations provide insights that can inform model selection in this work.

2.3.3. Encoder-Decoder Framework

Understanding the levels at which predictions can be made raises the question of how GNNs actually perform such tasks. A widely adopted perspective for describing GNNs is the encoder–decoder framework [39, 48, 49]. Within this view, the encoder is responsible for transforming the raw graph structure and its associated features into informative latent representations, while the decoder interprets

these representations to produce task-specific outputs (see Figure 2.6). Between these two stages, a sequence of message-passing layers iteratively propagates and aggregates information across the graph, enabling each node's embedding to incorporate contextual information from its neighbourhood. This separation reflects a common principle in deep learning: the encoder learns general-purpose embeddings that capture domain structure, whereas the decoder applies a lightweight, task-dependent mapping [37].

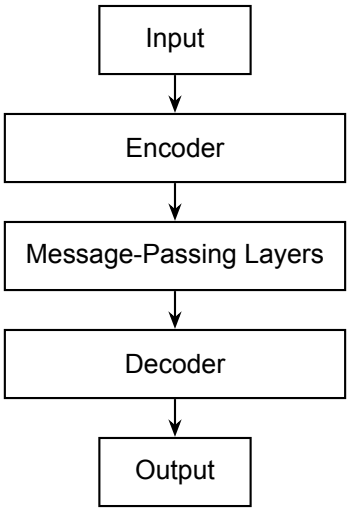


Figure 2.6: Schematic of GNN architecture showing encoder, message-passing, and decoder stages leading to the final output.

Formally, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node features $\mathbf{h}_i^{(0)}$ and optional edge features $\mathbf{e}_{ij}$, the encoder maps each node $v_i \in \mathcal{V}$ to a latent representation $\mathbf{h}_i^{(1)}$ in the feature space $\mathbb{R}^h$:

$$\mathbf{h}_i^{(1)} = \phi_{\text{enc}}\left(\mathbf{h}_i^{(0)}\right) \in \mathbb{R}^n. \tag{2.3}$$

Here, $\mathbf{h}_i^{(0)}$ denotes the raw input feature vector, and $\mathbf{h}_i^{(1)}$ the encoded node representation produced by the encoder. The embedding dimension n corresponds to the latent width of the node features, and is an architectural hyperparameter independent of the number of nodes $|\mathcal{V}|$. It may be either smaller or larger than the input feature dimension d, depending on predictive performance and computational budget [14, 37, 39, 48].

After the encoder, a sequence of message-passing layers updates node embeddings by exchanging learned signals across adjacent nodes. Formally, this intermediate operation can be expressed in simplified form as

$$\mathbf{h}_i^{(l+1)} = \text{MP}^{(l)}\left(\mathbf{h}_i^{(l)}, \{\mathbf{h}_j^{(l)}, \mathbf{e}_{ji} : j \in \mathcal{N}(i)\}\right), \tag{2.4}$$

where $\mathbf{h}_i^{(l)}$ and $\mathbf{h}_j^{(l)}$ denote the node embeddings at layer l, and $\text{MP}^{(l)}$ is a learnable transformation that aggregates neighbourhood information. This generic form summarises the message-passing process that bridges the encoder and decoder stages and will be formalised in detail in the next subsection.

The decoder then operates on these embeddings to produce predictions. Its form depends on the problem setting: in node classification, a linear transformation followed by a softmax produces class probabilities [39]; in link prediction, similarity functions between embeddings (e.g. dot products) estimate the likelihood of an edge [50]; and in graph-level tasks, pooled embeddings summarize the entire graph for classification or regression [51]. For node-level regression tasks, the decoder maps each final node embedding $\mathbf{h}_i^{(L)}$ to a continuous output $\hat{y}_i$ through a function ϕ_{dec}:

$$\hat{y}_i = \phi_{\text{dec}}\left(\mathbf{h}_i^{(L)}\right) \in \mathbb{R}^{d_y}. \tag{2.5}$$

Here, $\mathbf{h}_i^{(L)} \in \mathbb{R}^n$ denotes the latent representation of node i after the last message-passing layer, ϕ_{dec} is a task-specific readout function (often a single linear layer or a shallow MLP), and d_y is the output dimension, which equals 1 in scalar regression tasks such as daylight-factor prediction at sensor nodes.

This modular design highlights that most of a GNN's expressive capacity resides in the encoder, in particular its message-passing component, while the decoder is typically a lightweight, task-specific readout. Consequently, the encoder-decoder perspective offers a unifying abstraction that spans diverse architectures and problem settings in graph representation learning. The next subsection makes this mechanism explicit by formalising the message-passing framework through which node representations are iteratively updated from their neighbourhoods.

2.3.4. Message Passing Framework

As introduced in the previous section, the message-passing stage forms the core computational mechanism of a GNN, bridging the encoder and decoder by iteratively updating node representations through neighbourhood interactions. These layers enable information to propagate across the graph, allowing each node's embedding to become context-aware and reflect both its own features and those of its neighbours. At each layer, a node exchanges learnable messages with adjacent nodes, aggregates the incoming information through a permutation-invariant operator, and applies an update function to refine its state. This process can be expressed formally as

$$h_i^{(l+1)} = \mathrm{UPDATE}^{(l)}\Big(h_i^{(l)},\ \mathrm{AGGREGATE}^{(l)}\Big\{\mathrm{MESSAGE}^{(l)}\Big(h_i^{(l)}, h_j^{(l)}, \mathbf{e}_{ji}\Big) : j \in \mathcal{N}(i)\Big\}\Big), \tag{2.6}$$

where $h_i^{(l+1)}$ denotes the representation of node i after $l+1$ steps of message passing, $\mathcal{N}(i)$ is the neighbourhood of i, and $\mathbf{e}_{ji}$ represents the features of the edge between nodes j and i [52]. The procedure consists of three conceptual components:

- *Message,* computes information sent along edge from neighbours $j \in \mathcal{N}(i)$.
- *Aggregate,* combines all incoming messages into a single representation.
- *Update,* merges the aggregated neighbourhood information with the current node state to obtain the new embedding.

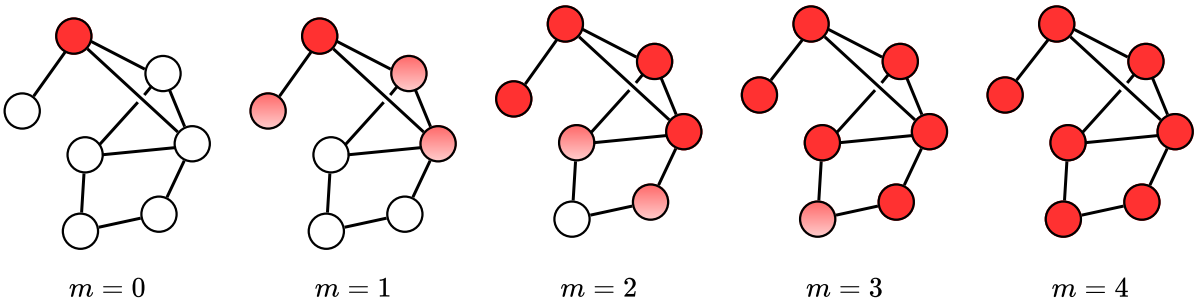


Figure 2.7: Example of message passing in a GNN. At each iteration m, nodes aggregate information from their neighbours and update their embeddings. The red colour indicates nodes that have already received the message, while the red shading denotes nodes updated in the current step. After $m = 4$, the message has propagated through the entire graph.

After m layers, the representation of a node captures information from its m hop neighbourhood [37]. Figure 2.7 illustrates this process: starting from an initial feature in the top node, messages propagate outward in successive steps until the entire graph has been updated.

In practice, the neighbourhood aggregation function AGGREGATE can take different forms. Throughout this thesis, aggregation is denoted generically by the operator $\bigoplus$, since different aggregation rules are tested and the notation should not be tied to a specific choice. The most common variants are illustrated in Figure 2.8 and can be summarized as follows:

- *Sum aggregation.*

$$\bigoplus_{j \in \mathcal{N}(i)} \equiv \sum_{j \in \mathcal{N}(i)} m_{ij}, \tag{2.7}$$

  which preserves the magnitude of neighbourhood information and is permutation invariant.

- *Mean aggregation.*

$$\bigoplus_{j\in\mathcal{N}(i)} \equiv \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} m_{ij}, \tag{2.8}$$

  which normalises by neighbourhood size and prevents embeddings from scaling by degree.

- *Max aggregation.*

$$\bigoplus_{j\in\mathcal{N}(i)} \equiv \max_{j\in\mathcal{N}(i)} m_{ij}, \tag{2.9}$$

  which highlights the strongest signal among neighbours but discards the rest.

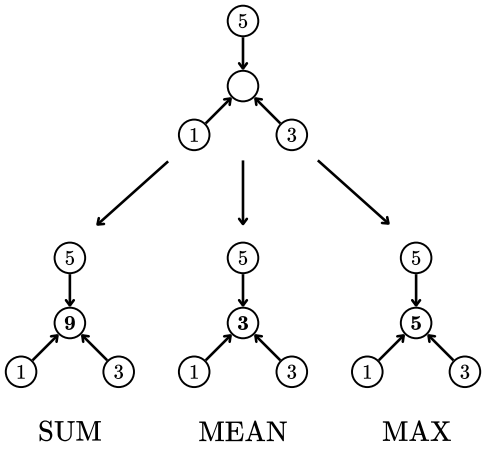


Figure 2.8: Common aggregation functions in GNNs. A central node i aggregates information from its neighbours j_1, j_2, j_3 with values $\{5, 3, 1\}$. Depending on the choice of aggregation, neighbourhood information is summarised as 9(sum), 3 (mean), or 5 (max). This intermediate result of the aggregation function is then combined with the node's own state in the update function to produce the final embedding. For this illustrative example, node features of node i and the edge features are not taken into account.

The sum, mean, and max functions provide the most basic ways of aggregating information from a node's neighbours. Building upon these, more sophisticated mechanisms such as attention or multi-aggregator schemes have been proposed.

2.3.5. Operators in GNNs

The design choices of *message*, *aggregation*, and *update* functions give rise to the diversity of GNN operators found in the literature. In this section, we focus on the Kipf–Welling Graph Convolutional Network (GCN) [39] as a minimal case. The GCN serves as a reference point to illustrate how these functions are instantiated in practice, providing the groundwork for the operator comparison presented later in this thesis.

The GCN represents the most widely used and conceptually simple instantiation of the message passing framework. In GCN, the update rule for all nodes in matrix form is

$$h_i^{(l+1)} = \sigma\left(\bigoplus_{j\in\mathcal{N}(i)\cup\{i\}} \frac{1}{\sqrt{\tilde{d}_i\tilde{d}_j}} h_j^{(l)} W^{(l)}\right), \tag{2.10}$$

where $h_i^{(l)} \in \mathbb{R}^d$ denotes the representation of node i at layer l, $W^{(l)}$ is a learnable weight matrix, and σ is a non-linear activation function. The aggregation $\bigoplus$ is instantiated as a weighted sum, with each neighbour j contributing its representation $h_j^{(l)}$ scaled by the symmetric normalisation term $1/\sqrt{\tilde{d}_i\tilde{d}_j}$. The neighbourhood is defined as $\mathcal{N}(i) \cup \{i\}$, which means that each node is connected to itself by an added self-loop. This self-loop ensures that a node's own features are retained in the aggregation, so that both self-information and neighbourhood information contribute to the update. Normalisation balances contributions from nodes of different degrees, preventing those with many neighbours from dominating the aggregation.

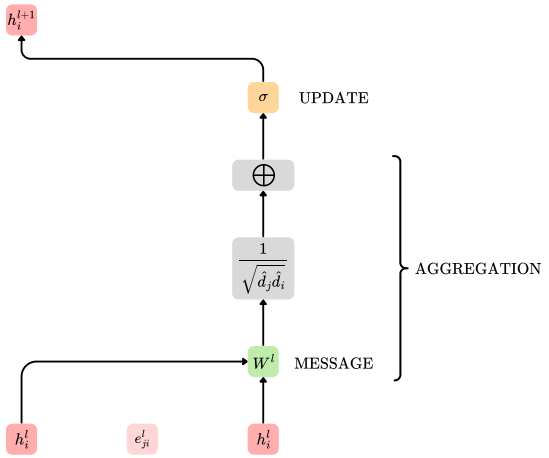


Figure 2.9: Graph Convolutional Network (GCN) as an instance of the message passing framework. Each neighbour j sends a message $h_j^{(l)} W^{(l)}$ (*Message*), which is combined by a degree-normalized sum (*Aggregation*). The aggregated representation is passed through a non-linearity σ to obtain the updated embedding $h_i^{(l+1)}$ (*Update*). Edge features e_{ji}^l are not used in this formulation.

The GCN formulation can be decomposed into the following message-passing components, which are illustrated for a single node in Figure 2.9:

- *Message.* Each neighbour j constructs a message $h_j^{(l)} W^{(l)}$, representing a linear transformation of its embedding.
- *Aggregation.* Messages are combined by a degree-normalised sum, indicated by $\bigoplus$ in the figure, where each contribution is scaled by $\frac{1}{\sqrt{\tilde{d}_i \tilde{d}_j}}$ and the neighbourhood includes the node itself ($\mathcal{N}(i) \cup \{i\}$). The inclusion of the self-loop ensures that a node's own state is retained alongside information from its neighbours.
- *Update.* The aggregated message is passed through a non-linearity σ to produce the updated representation $h_i^{(l+1)}$.

The GCN thus provides a minimal baseline against which later operators can be understood, and is presented here as a reference point for the message-passing framework, clarifying the roles of message, aggregation, and update functions.

2.3.6. Expressive Power and Limitations

Despite their flexibility, message-passing GNNs have well-understood theoretical and practical limitations. Recognising these constraints is essential, as they motivate new operator designs and guide architectural choices for specific application domains.

Message-passing GNNs have well-defined expressive limits that match those of the 1-dimensional Weisfeiler–Lehman (1-WL) test, a classical graph-refinement procedure used to tell apart non-isomorphic graphs[4] [51, 53]. Expressive power in this context refers to a model's ability to assign different representations to graphs that differ in structure. The 1-WL test iteratively updates node labels based on the multiset of neighbouring labels; message-passing GNNs perform an analogous operation by aggregating information from neighbouring nodes. Therefore, if the 1-WL test cannot distinguish between two non-isomorphic graphs, no standard message-passing GNN can separate them based solely on structural information [51, 53]. In practice, common aggregation functions (sum, mean, max) may map distinct but 1-WL-indistinguishable neighbourhoods to the same embedding. To extend this expressivity, researchers have proposed higher-order GNNs that emulate stronger WL variants [53, 54], or have incorporated additional structural or positional encodings to enrich the information available during message passing [55, 56].

[4] Two graphs are considered *isomorphic* if they are structurally identical under a relabelling of nodes and edges. In that case, they should be treated as equivalent.

While this theoretical bound is fundamental from a graph-isomorphism perspective, it is less restrictive in practical learning scenarios. Many real-world graphs include continuous node or edge attributes—such as spatial coordinates, distances, or physical quantities—that already differentiate structures the 1-WL test would treat as equivalent. Moreover, in graph-level or node-level regression tasks, the objective is to approximate a continuous mapping from graph inputs to target responses rather than to distinguish non-isomorphic graphs. In such contexts, the effective expressivity of message-passing networks depends more on the richness of available attributes and the capacity of the architecture than on the formal limits imposed by the WL hierarchy [14, 51, 54, 57]

Even when expressivity is not the main limitation, training deep GNNs faces another challenge known as oversmoothing. As layers are added, node embeddings become increasingly similar and eventually indistinguishable [58, 59]. This effect limits the useful depth of standard GNNs to only a few layers, which can hinder performance on tasks requiring long-range information. Common remedies include residual connections, normalisation, and identity mappings that slow this homogenisation of embeddings [39, 41].

A related limitation is oversquashing. Unlike oversmoothing, which makes embeddings too similar, oversquashing occurs when long-range information is compressed into fixed-size embeddings, creating a bottleneck that restricts information flow [60]. This effect is particularly pronounced in graphs with grid-like structures, where exponentially many paths are aggregated into a single vector. Proposed remedies include graph rewiring, positional encodings, and attention mechanisms that improve the flow of long-range information [60–62].

Taken together, these limitations show that message-passing GNNs, while powerful, are not universally expressive. Their ability to distinguish structural patterns is bounded by the 1-WL test, their effective depth is constrained by oversmoothing, and their capacity to capture long-range dependencies is reduced by oversquashing. These challenges have inspired new operator designs: GIN [51] to match 1-WL expressivity, residual and identity mappings [41] to mitigate oversmoothing, and attention or rewiring strategies [60, 63] to address oversquashing. Recognising these constraints provides context for the coming operator evaluation and informs the methodological choices made in this research.

Summary and Transition

The focus of this thesis is the surrogate modelling of DF, replacing costly simulation with efficient machine-learning prediction. Fully connected ANNs provide a natural starting point for such surrogates, but their inductive bias is minimal: they treat inputs as independent feature vectors and do not encode any geometric or relational structure. This motivates the consideration of architectures that introduce such structure explicitly. GNNs address this by operating directly on graph-structured data, where nodes and edges provide a natural way to encode geometric and relational structure within the model architecture.

GNNs work by assigning embeddings to nodes, updating them through message passing, and using the resulting representations for prediction. Their inductive biases ensure consistency under node relabelling and capture relational dependencies through neighbourhood aggregation. At the same time, they face well-known limitations: their expressivity is bounded by the 1-WL test, their depth by oversmoothing, and their long-range capacity by oversquashing.

In sum, GNNs offer a promising framework for modelling DF, but they are not a universal solution. The following chapter builds on this theoretical foundation by reviewing prior work on machine learning for daylight prediction and examining the role of GNNs in building engineering. It additionally examines feature representation strategies for GNNs, surveys commonly used message-passing operators, and discusses key design considerations relevant to GNN operator construction.

3

Literature Review

The previous chapter established the motivation for surrogate modelling of DF in early-stage design, highlighting the limitations of physics-based simulation and the promise of machine learning as a complementary approach. Building on this foundation, the present chapter reviews prior research that informs the methodological choices of this thesis. The discussion begins with surrogate models for daylight prediction, covering ANNs and other ML approaches, including feature formulations that will later serve as baselines. The focus then shifts to GNNs, by situating them within the broader building engineering domain. This serves to show how GNNs have already been applied to relational problems in design and engineering, thereby establishing their relevance and motivating their use for daylight modelling.

Following this overview, the chapter examines feature representation in GNNs, with particular attention to how node and edge attributes encode geometric relationships. This discussion outlines the role of edge features in message passing, their impact on model expressiveness, and the challenges of designing physically consistent representations for geometric learning tasks. Finally, the chapter reviews a selection of GNN operators. These operators are evaluated in terms of their message-passing formulations, computational complexity, and sensitivity to edge information, providing the foundation for the operator benchmarking and selection

3.1. Surrogate Models for Daylight Prediction

One recent review study illustrates the scope and trajectory of ML applications in daylight prediction. Liu et al. [16] conducted a systematic statistical analysis of 39 studies published between 2006 and 2022, categorising models according to task type (time-series, spatial, spatio-temporal) and algorithmic family. Their findings, summarised in Fig. 3.1, highlight a marked increase in ML use since 2014, with a clear dominance of ANN-based approaches, which account for the majority of applications. More traditional methods such as decision trees, random forests, gradient boosting, and support vector regression appear less frequently but remain relevant for specific prediction tasks. In recent years, convolutional and generative models (CNNs and GANs) have also started to appear, indicating a gradual shift toward architectures that can represent spatial patterns or distributional behaviour in daylight performance[1].

The review underlines the growing importance of surrogate models in architectural daylighting research. It also provides the context for this thesis: existing surrogate models are predominantly ANN-based and rely on hand-crafted geometric encodings, which motivates the exploration of graph neural networks as a more relational and geometry-aware alternative.

[1] To improve readability, models reported by Liu et al. [16] that use fully-connected multilayer perceptron architectures (ANN, FNN, FCNN, and BPNN) are grouped together as “ANN-based”, since these variants are typically treated interchangeably in the daylight surrogate modelling literature. The remaining approaches are reorganised into tree-based, SVM-based, deep structured (CNN/GAN), and a residual “Others” category.

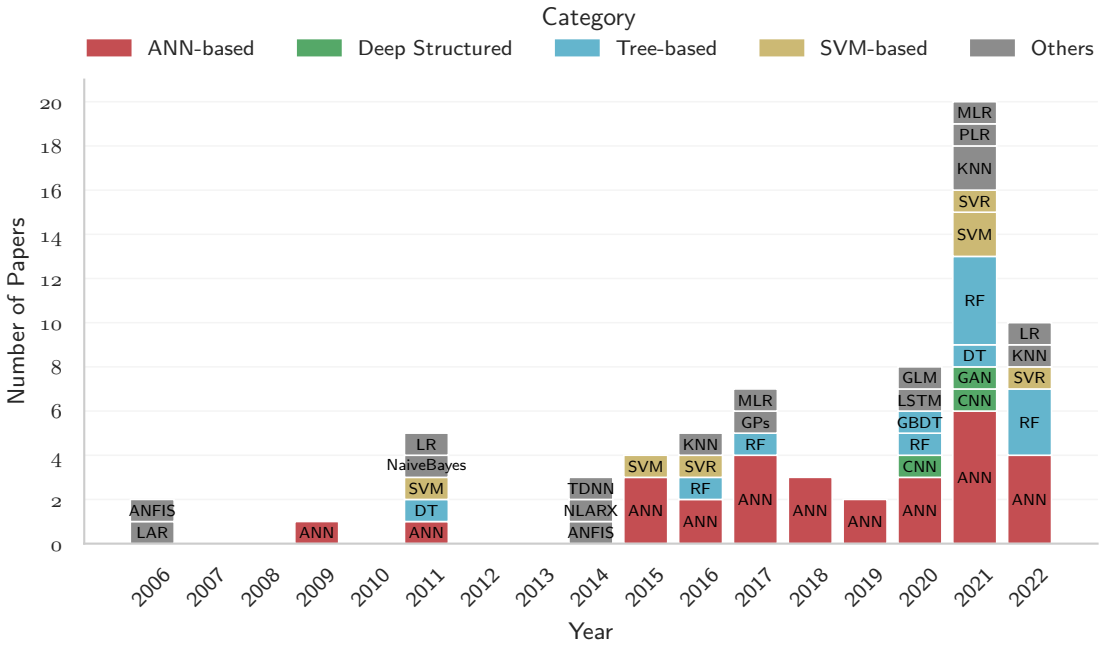


Figure 3.1: Distribution of machine learning surrogate models applied in daylighting research between 2006 and 2022, adapted from Liu et al. [16]. To improve clarity, the various ANN variants are consolidated into a single ANN-based category, and individual methods are grouped into five model families. Each block corresponds to one study and is annotated with the precise algorithm used. The timeline highlights both the rapid uptake of machine learning from 2014 onwards and the continued dominance of ANN-based approaches, with deep structured architectures (CNN/GAN) emerging only recently.

3.1.1. Artificial Neural Networks

ANNs have been widely applied to daylight metrics. Their success depends less on architectural novelty than on how geometric inputs are encoded into features. The main benchmarks for this thesis are drawn from three studies: Han et al., Le-Thanh et al., and Dieguez et al. [17–19]

Han et al. [17] trained a fully connected ANN on a procedurally generated dataset of 1610 rectangular office rooms with varying dimensions, window sizes, surface reflectances, and orientations. Daysim[2] was used to compute hourly illuminance at sensor grids, aggregated into DA and UDI. A key step was reformulating the problem at the point level: each sample combined a sensor position with local geometric descriptors. By normalising sensor coordinates within a room-local frame, the model learned spatial patterns transferable across room sizes and orientations. While effective within the sampled distribution, generalisation was not tested beyond the same parametric family, leaving invariance to unseen typologies, rotations, or scaling unverified.

Le-Thanh et al. [18] extended this approach by designing a more structured and physically grounded feature set. Using 400 synthetic building layouts generated from square modules with randomly placed windows, they simulated UDI with DIVA[3] for Ho Chi Minh City. Features were grouped into interpretable categories: distances to obstacles, distances between sensors and window corners, and relative angles to windows. This explicit encoding of geometric relations improved interpretability and predictive power. However, validation remained within generator-defined layouts , with no explicit tests of robustness to unseen design types.

Dieguez et al. [19] shifted attention to DF, a metric that has seen comparatively limited use in machine-learning studies on daylight prediction despite its regulatory relevance. Their dataset included approximately 15,000 synthetic rooms, both rectangular and L-shaped, embedded in Stockholm urban contexts with obstructions such as balconies and adjacent buildings. In the numerical track, DF was decomposed into window-by-window contributions, represented through features such as solid angle,

[2]Daysim is a daylight simulation engine that uses daylight coefficients derived from the Radiance ray-tracing framework to compute annual indoor illuminance and climate-based daylight metrics.

[3]DIVA is a daylight and energy simulation plugin for Rhinoceros and Grasshopper that provides an interface to Radiance and Daysim for computing indoor illuminance and climate-based daylight metrics.

relative orientation to the window normal, head height, and obstruction factors, which were used as inputs to neural network models. This strategy yielded accurate DF predictions and compliance assessments, but the generator still restricted geometries to orthogonal layouts, imposed an implicit size cap of about 12 m, and did not explicitly test on out-of-distribution layouts or uniform scaling.

Other ANN-based studies illustrate the diversity of encoding strategies, ranging from the integration of real-world sensor data and shading features [64], to image-like encodings of three-dimensional geometries [65], and large-scale climate-based datasets [66]. While these contributions broaden the methodological spectrum, the feature sets proposed by Han, Le-Thanh, and Dieguez remain the most relevant baselines for this thesis, both in their direct connection to DF and in their clear progression of inductive biases.

3.1.2. Other Machine Learning Approaches

Beyond fully connected ANNs, several other machine learning models have been explored for daylight prediction. Decision tree ensembles, particularly Random Forests (RF) and Gradient Boosting Machines (GBM), have also been employed in daylight prediction [19, 66, 67]. Their main strengths lie in robustness and interpretability, as they can provide clear rankings of parameter importance, such as the influence of window-to-wall ratio or glazing transmittance. However, tree-based models rely on piecewise constant approximations and tend to underperform compared to neural approaches when modelling high-dimensional geometric input spaces.

Among models incorporating spatial information, recent studies have applied CNN-based architectures, including ResNet variants, to predict daylight provision and view-related metrics from multimodal inputs that combine floor plan images with numerical descriptors [68, 69]. While such models can capture spatial structure effectively, they require large training datasets and remain constrained to regular grid representations, limiting their ability to generalise to irregular geometries [14].

A related development is the use of conditional generative adversarial networks (cGANs), which extend CNN-based architectures through a generator–discriminator framework that learns mappings between architectural encodings and daylight maps. Several studies translate image-like representations of room geometry into daylight outputs using pix2pix-style cGANs [70]. He et al. [69] apply pix2pix to general floor plans for multiple daylight metrics, demonstrating fast, visually faithful predictions. Dieguez et al. [19] likewise explore an image-based track in parallel to their numerical regression model, training a pix2pix network to generate DF distributions from binary masks augmented with window and obstruction channels. While effective, these approaches inherit typical cGAN limitations related to dataset coverage and training stability [71].

A different line of work involves physics-informed neural networks (PINNs), which embed governing equations, such as radiative transfer or diffusion, directly into the loss function to constrain predictions by physical laws [72]. This enhances physical consistency and interpretability but increases computational cost, and applications to daylight remain limited in scope [73].

Overall, while tree-based models, CNNs, cGANs, PINNs broaden the methodological landscape, they remain less established than ANN surrogates. Prior ANN-based studies demonstrate the feasibility of predicting daylight metrics with machine learning, but they also reveal two key limitations. The first concerns the use of flat vector encodings, which neglect the geometric and relational structure inherent in daylight transfer. The second, not limited to ANN models, is that model evaluation remained confined to data drawn from the same procedural distributions as the training and validation sets, without systematic assessment of robustness to new typologies or geometric transformations. Addressing this gap is a central aim of the present thesis. The feature sets of Han, Le-Thanh, and Dieguez therefore serve not only as historical milestones in daylight surrogate modelling, but also as benchmark encodings that will be reformulated and systematically evaluated in graph-based form. To date, however, no published work has applied graph neural networks to DF or CBDM prediction, underscoring both the novelty of the present study and its potential to extend surrogate modelling towards relational and geometry-aware learning.

3.2. Graph Neural Networks in the Building Engineering Domain

Since no prior studies have applied GNNs to daylight prediction, it is instructive to review their use in related building engineering. Existing work spans structural engineering, energy modelling, and generative design, providing precedents in prediction settings, feature representations, and methodological choices. These applications offer the closest reference points for positioning the present thesis, which focuses on node and edge features for modelling daylight at sensor nodes, where predictions depend both on intrinsic attributes and on relations to surrounding windows.

Across the wider building engineering domain, GNNs have been adopted for a diverse range of tasks, spanning scales from element-level operations within building information modelling (BIM) to city-scale predictions. In BIM and IFC [4] semantics, nodes represent building elements such as rooms, walls, or slabs, and edges encode adjacency or connectivity. Node features typically consist of geometric or categorical attributes, while edge features capture topological relationships. The task is node classification, enriching BIM models with functions or semantic labels [74, 75]. Similar node-level classification has been applied in construction quality assessment, where building elements are nodes with material or inspection features, and edges encode construction dependencies, allowing quality status to be inferred automatically [76].

In structural mechanics, GNNs have been employed as surrogates for finite element analysis (FEA), with mesh nodes as graph nodes and element connectivity as edges. Node features include spatial coordinates and material parameters, while edge features describe mesh adjacency. The prediction task is typically graph-level regression, approximating global system responses such as displacements or stresses [77–79]. Storm et al. [80] similarly represent integration points as nodes with features such as local strain, internal material variables, global strain inputs, and geometric distances to nearby voids, using relative coordinates as edge features to encode adjacency.

At the operational scale, spatio-temporal GNNs have been applied to building energy forecasting, where nodes represent zones or entire buildings, edges capture thermal or spatial adjacency, and features include occupancy, weather, or physical descriptors. The task is graph-level regression, forecasting multi-building or multi-zone energy loads [81, 82].

Other applications focus on geometry processing, particularly scan-to-BIM pipelines. Here, nodes correspond to superpoints in 3D point clouds, edges capture spatial proximity, and features describe local geometry such as coordinates or normals. The task is node classification or segmentation, enabling automated reconstruction of BIM elements [83, 84]. Recent reviews further synthesise these applications across domains including structural health monitoring, scheduling, and construction progress tracking, confirming that GNN adoption in building engineering is both diverse and rapidly growing.

Within design-related applications, two studies stand out as particularly relevant to this thesis. Lei et al. [85] proposed a GNN framework for predicting building characteristics at the urban scale. Each building is represented as a node with features such as type, height, and age, while edges encode spatial context derived from neighbouring objects. The prediction task is node regression and classification, where missing attributes are inferred from available geospatial data. This work provides a methodological precedent for daylight modelling, which likewise requires node-level prediction of performance values from geometric and contextual features.

Wu et al. [86] advanced the use of GNNs for surrogate modelling at the block scale. In their formulation, each building is a node with features such as floor plan type, height, window-to-wall ratio, projection lengths, and shape factors, while edges capture spatial relations including relative orientation, distance, and visibility percentage. The task is graph-level regression, predicting block-level performance metrics such as energy use, thermal comfort, and daylighting. Wu et al. demonstrated that graph attention networks outperform both ANNs and GCNs in this setting, while achieving significant speedups over physics-based simulations. Their explicit integration of node and edge features is especially relevant here, providing a clear precedent for the feature-centric approach of this thesis.

Other studies illustrate the breadth of emerging design-oriented applications, complementing the more directly relevant work of Lei and Wu. Park et al. [87] developed a floor plan recommendation sys-

[4] IFC (Industry Foundation Classes) is an open, vendor-neutral data schema developed by buildingSMART to enable interoperable exchange of building information models across software platforms.

tem, where rooms were represented as nodes, edges captured spatial adjacency, and node features encoded room types. The task was graph retrieval[5], supporting design exploration by recommending functional layouts. Barakathullah and Koh [88] applied heterogeneous GNNs to circulation modelling in shopping malls, where nodes represented shops and entrances and edges corresponded to corridors, predicting edge-level usage probabilities based on human–space interactions. These works extend the scope of GNNs beyond performance prediction to questions of spatial configuration and circulation, though their task types differ from daylight regression.

Taken together, these studies show that GNNs have been successfully applied across building engineering tasks ranging from semantic enrichment and construction quality to structural mechanics, energy forecasting, floor plan design, and urban-scale modelling. They also illustrate the variety of prediction settings that GNNs can address, including node classification, node regression, edge regression, and graph-level regression. However, despite this breadth, no work to date has applied GNNs to daylight prediction. This absence underscores the novelty of the present thesis, which adapts graph-based methods to the relational problem of daylight modelling, with a particular focus on node- and edge-level feature design. A structured overview of these prior applications is provided in Table 3.1, which groups existing studies by task type and highlights precedents most relevant to daylight modelling.

Table 3.1: Overview of GNN applications in the building engineering domain, grouped by task type and relevance to daylight modelling. Key precedents for this thesis are highlighted.

Domain / Study	Task type	Graph representation	Relevance to daylight
Landscape applications			
BIM semantics [74, 75]	i	Nodes: BIM elements; Edges: adjacency; Features: geometry, categories	Relational context helps, but categorical only
Construction quality [76]	i	Nodes: building elements; Edges: construction dependencies	Similar structure, but outputs categorical
Structural mechanics [77–80]	vi	Nodes: mesh nodes; Edges: connectivity; Features: coordinates, strain, material	Physics-based surrogates, local-to-global prediction
Energy forecasting [81, 82]	vi	Nodes: zones/buildings; Edges: thermal adjacency; Features: thermal, weather, occupancy	Relational forecasting, but aggregate
Scan-to-BIM [83, 84]	i	Nodes: point clusters; Edges: spatial proximity; Features: geometry descriptors	Geometry-to-graph encoding, not performance
Design-related applications			
Lei et al. [85]	i / **ii**	Nodes: buildings; Edges: context adjacency; Features: type, height, age	**Closest precedent: node-level prediction**
Wu et al. [86]	iv	Nodes: buildings; Edges: orientation, distance, visibility; Features: WWR, shape, height	**Closest precedent: surrogate with node/edge features**
Park et al. [87]	v	Nodes: rooms; Edges: adjacency; Features: room type	Shows layouts as graphs, retrieval not regression
Barakathullah & Koh [88]	iii	Nodes: shops/entrances; Edges: corridors; Features: function	Heterogeneous graphs, edge-level focus

Legend: i = Node classification, **ii = Node regression**, iii = Edge regression, iv = Graph regression, v = Graph retrieval.

Summary

The previous two sections has reviewed the development of surrogate models for daylight prediction, beginning with ANN-based approaches that established the feasibility of learning daylight metrics from geometric encodings, but also revealed their limited robustness to unseen configurations. Alternative

[5]Graph retrieval refers to the task of identifying graphs from a database that are most similar to a query graph. A GNN first embeds each graph into a latent representation, and similarity metrics (e.g., cosine similarity) are then used to rank or retrieve candidate graphs. In the context of floor plan recommendation [87], this means retrieving layouts whose spatial relationships between rooms best match the requirements of the query.

ML algorithms such as decision trees, CNNs, PINNs, and cGANs broaden the methodological landscape, yet ANNs remain the most widely adopted surrogates to date. The review then turned to GNNs, which have been successfully applied across building engineering domains ranging from BIM semantics and construction quality to structural mechanics, energy forecasting, and urban-scale modelling. These studies demonstrate the versatility of GNNs across node-, edge-, and graph-level prediction tasks, and highlight methodological precedents relevant to daylight modelling.

Developing a GNN-based surrogate model for daylight prediction requires constructing a graph representation of the problem, reformulating established ANN feature sets into node and edge attributes, and selecting appropriate message-passing operators. The next section therefore turns from the broader literature on GNN applications to the specific question of how daylight can be represented in graph form, focusing on the conceptual role of node and edge features and their influence on learning performance.

3.3. Feature Representation in GNNs

The review of GNN applications in building engineering reveals how graph-based learning can capture the relational structure of diverse physical and semantic systems. Across domains such as structural mechanics, BIM semantics, and energy modelling, nodes and edges are defined according to the entities and interactions most relevant to each task, be they joints connected by physical constraints, rooms linked by adjacency, or buildings related through energy exchange. These formulations demonstrate that the success of GNNs depends not only on the choice of operator but on how domain knowledge is translated into graph form.

For daylight modelling, this translation is non-trivial. Unlike the discrete or categorical relations in many engineering graphs, daylight propagation is continuous and geometry-dependent. Defining a graph suitable for this task therefore calls for identifying the appropriate entities (e.g., sensors and windows) and the relations between them that govern light transfer. It also demands careful selection of node and edge attributes that express geometric and photometric dependencies in a way that can be learned effectively by a message-passing network.

Consequently, the next part of this chapter shifts from reviewing prior GNN applications to investigating the conceptual foundations of constructing one for daylight prediction. It begins by examining how existing literature frames the role of node and edge features and the challenges associated with their design. The following sections review these considerations in turn, establishing a basis for later methodological choices.

3.3.1. Why Features Matter

But how much do features actually matter in GNNs? The broader literature makes it clear that they are often decisive. As Hamilton notes in his monograph on graph representation learning, the quality of node and edge attributes largely determines the ceiling of achievable performance, since operators cannot extract information that is not encoded in the input [15]. Building on this distinction, Bronstein et al. [49] frame it as a separation between *what* information is provided to the model, through the design of features, and *how* this information is processed, through the choice of operator. Xu et al. [51] further show that even maximally powerful operators are constrained by the informativeness of the features they process. More recently, Veličković [89] emphasises that, in practice, feature engineering and selection often outweigh operator choice in determining downstream accuracy.

Taken together, these insights highlight that feature design is not an auxiliary detail but a central determinant of GNN performance. This provides the rationale for beginning the present investigation with a focus on feature representation: before comparing operators, it is necessary to establish which encodings provide the most robust and transferable basis for daylight prediction.

3.3.2. Feature Sets from ANN-based Surrogates

As introduced in Section 3.1.1, prior ANN-based surrogates for daylight prediction largely differ in how they encode geometry and photometric relationships. From a feature design perspective, these studies can be grouped into three benchmark families that form the basis for this thesis:

- *Geometry-based descriptors (Han et al. [17])*: minimal encodings such as normalised sensor coordinates, providing scale- and orientation-consistent inputs without explicit reference to win-

dows.

- *Directional and distance descriptors (Le-Thanh et al. [18])*: structured relations between sensors and windows, embedding distances and angles that approximate photometric dependencies.
- *Obstruction-aware descriptors (Dieguez et al. [19])*: window-by-window contributions including solid angle, orientation, and obstruction factors, grounded directly in the physics of DF calculation.

Collectively, these feature families illustrate a progression from purely geometric descriptors, to relational encodings, to physics-informed formulations. Their shared limitation lies in representing interactions implicitly rather than explicitly, motivating the broader question of how geometric and photometric relations can be encoded as structured connections, an issue explored in the following discussion on edge features.

3.3.3. The Importance of Edge Features

Translating tabular feature sets into graph form makes node features relatively straightforward to define: intrinsic properties such as spatial coordinates or element dimensions naturally map onto node attributes. The greater challenge, and opportunity, lies in defining edge attributes, which determine how information flows between entities and thus govern the model's ability to capture relational dependencies.

The literature consistently underscores the importance of edge features. Gilmer et al. [90] demonstrated in molecular property prediction that message passing without edge conditioning fails to represent essential interactions, whereas incorporating bond types and distances substantially improved accuracy. Simonovsky and Komodakis [91] introduced edge-conditioned convolutions to show how edge attributes can modulate information flow adaptively, a principle later generalised in frameworks emphasising relational inductive biases [37]. More recent surveys similarly note that a GNN's representational capacity often depends less on the expressiveness of its operator than on whether edge-level relations are meaningfully encoded [89] .

Across domains, edge attributes have been used to express a wide range of continuous or categorical relations, geometric distances in physical systems [90], visibility between buildings [86], or adjacency weights in semantic building graphs [74]. These examples collectively demonstrate that incorporating edge information transforms message passing from a purely topological aggregation into a relation-aware process that can capture both the presence and the character of interactions.

Within the broader literature, this insight has motivated a variety of edge-feature formulations, from fixed scalar weights to learnable embeddings and directional vectors. Reviews consistently emphasise that such representations enhance expressivity, stability, and sample efficiency, particularly in tasks governed by spatial or physical relationships. At the same time, they introduce challenges in terms of feature normalisation, redundancy, and invariance, issues that have been the subject of growing theoretical attention [14, 60].

Taken together, these studies establish that edge features are not an optional refinement but a structural element in many high-performing GNN architectures. They enable networks to represent continuous relations more faithfully and to align message passing with the underlying geometry or physics of the domain. The conceptual and practical challenges of designing such attributes, robustness to transformation, sufficiency, and over-specification, are discussed in the following subsection.

3.3.4. Conceptual Challenges in Feature Design

Although graph representations extend the range of available encodings, they also raise new design challenges. Three considerations are particularly salient for daylight prediction, each of which can be linked to established limitations of GNNs.

First, robustness to geometric transformations. In geometric deep learning, invariance and equivariance to translation, rotation, and scaling are recognised as fundamental inductive biases [14, 54, 92]. Traditional lighting simulations naturally preserve these symmetries, but feature-based surrogates risk encoding absolute coordinates that break them. This challenge relates directly to the expressivity limits of the 1-WL test: if isomorphic graphs are distinguished only by arbitrary coordinate choices, generalisation is compromised. Recent advances, such as similarity-equivariant GNNs [93], show how

embedding symmetries improves data efficiency. In this thesis, rather than adopting an equivariant operator, robustness is probed empirically by systematically testing how different feature encodings behave under controlled transformations, using a fixed baseline operator.

Second, the balance between redundancy and sufficiency. Classical feature selection highlights that redundant descriptors increase dimensionality while insufficient ones limit predictive capacity [94]. In a GNN context, redundant features exacerbate oversquashing, as excessive information must be compressed into limited message-passing channels [60]. Conversely, insufficient features underutilise the available communication capacity of the graph. In this thesis, the trade-off is explored empirically by embedding feature families of varying scope into graphs and observing their performance and robustness.

Third, the risk of over-specification. Embedding physics-inspired priors too tightly can constrain the model to reproduce known formulas rather than discover more generalisable patterns. This is analogous to oversmoothing, where excessive bias or aggregation causes node representations to converge to indistinguishable embeddings [59]. In daylight surrogates, the open question is how much physical structure should be imposed through features versus learned directly from data. Rather than prescribing a fixed level of physics encoding, this thesis systematically tests feature sets ranging from minimal geometric descriptors to physics-grounded attributes.

These conceptual issues underscore that feature design is not a straightforward matter of including as many descriptors as possible. By linking them to expressivity (1-WL), communication limits (oversquashing), and representation collapse (oversmoothing), the challenges are anchored in the broader theory of GNN limitations. Building on these insights, the present thesis benchmarks ANN-inspired feature sets and their graph-based extensions to identify which representations provide the most reliable and transferable foundation for daylight prediction.

3.4. Operators in GNNs

Having motivated the importance of feature encodings first, the next step is to examine how different message-passing operators exploit them. This staged approach, features first and operators second, ensures that architectural choices are assessed on a consistent representational foundation.

This section reviews a selection of message-passing operators used in the GNN architecture, with attention to how they process both node and edge attributes. The discussion draws on insights from benchmark studies, operator complexity, and differences in how operators handle geometric and relational information.

3.5. Message-Passing Operator Design Considerations

GNN operators determine how information is exchanged along edges and thus how effectively a model can exploit the structure encoded in node and edge features. The literature highlights several recurrent considerations that influence suitability across spatial and physically grounded tasks:

- *Edge feature integration.* Operators differ in whether, and how, they incorporate continuous edge attributes into message computation, ranging from direct conditioning of weights to kernel-based interpolation and indirect concatenation [47, 90, 91, 95].
- *Depth stability.* Residual connections, normalisation, and tailored aggregators have been proposed to mitigate oversmoothing and gradient decay in deeper message-passing networks [41, 96].
- *Task alignment.* Some architectures are routinely applied to node-level regression, others to graph-level prediction or classification; performance can depend on how the operator's inductive biases match the target task.
- *Computational efficiency.* Reported scaling with latent dimension, edge dimension, and kernel resolution varies across families (e.g., edge-conditioned, kernel-based, attention-based), shaping the trade-off between expressivity and tractability [47, 90, 95].
- *Implementation maturity and reproducibility.* Comparative studies point to the value of operators with stable reference implementations in widely used frameworks and with established benchmark

results [55, 97].

Evidence for these considerations commonly derives from benchmark suites spanning molecular graphs (e.g., *QM9*, *ZINC*, *MD17*), spatial superpixel graphs (*MNIST superpixels*), and large-scale network datasets (e.g., *ogbn-proteins*, *ogbn-arxiv*, *ogbn-ppa*, *ogbn-molhiv*). Because training pipelines and budgets differ across studies, absolute numbers are not strictly comparable; nevertheless, the aggregated literature provides informative trends about operator behaviour under varying graph structures and the presence or absence of continuous edge attributes.

The following review surveys representative operator families, synthesises benchmark evidence, and contrasts their mechanisms for incorporating edge information. Any task-specific operator down-selection is deferred to the methodology.

3.5.1. Overview of Candidate Operators

The following operators are grouped by the extent to which they incorporate edge features, a dimension widely recognised as critical in tasks governed by geometric or physical relationships. The first group includes operators that natively support continuous edge attributes as part of their message computation. The second group covers operators that can incorporate edges indirectly or in restricted forms. The third group includes operators whose edge support is minimal or absent, serving primarily as baselines. A final group considers attention-based approaches.

Operators with Native Support for Continuous Edge Features

NNConv. The Neural Network Convolution (NNConv), also referred to as Edge-Conditioned Convolution (ECC), was independently proposed in closely timed works by Gilmer et al. [90] and Simonovsky & Komodakis [91]. Both formulations define a message-passing operator in which edge features dynamically parametrize convolutional weights via a neural network, typically an MLP. This makes NNConv well-suited for tasks involving continuous and physically grounded edge features. In practice, NNConv performs strongly on datasets like *QM9* and *ZINC* when edge information is included, particularly for molecular regression tasks.

CGConv. The Crystal Graph Convolution (CGConv), introduced by Xie and Grossman [98], was designed for property prediction in crystalline materials. It updates node features through a gated message-passing mechanism in which each neighbour's contribution is weighted by a sigmoid function of the edge features, enabling selective propagation of information based on geometric or physical relations. The operator explicitly supports continuous edge attributes, such as interatomic distances or bond descriptors, which are integrated directly into the message computation. Although CGConv was originally validated on materials science datasets and no results are reported on common benchmarks such as *QM9* or *ZINC*, its formulation is fully compatible with these datasets.

SplineConv. The Spline Convolution (SplineConv), introduced by Fey et al. [95], defines graph convolutional filters as continuous functions over edge pseudo-coordinates using B-spline basis functions. This approach allows locally supported, smooth kernels that are parameterized by a fixed set of learnable weights, enabling efficient and flexible geometric deep learning on graphs. The operator explicitly incorporates continuous edge attributes, such as Cartesian coordinates, polar angles, or other pseudo-coordinates, directly into the kernel computation, allowing for spatially aware message passing. While SplineConv has not been reported on benchmarks such as *QM9*, *ZINC*, or *OGB* node classification datasets, it has achieved strong results on spatially structured tasks such as *MNIST superpixels*, where geometry plays a central role.

GMMConv. The Gaussian Mixture Model Convolution (GMMConv), introduced by Monti et al. [47] as part of the MoNet framework, defines graph convolutional filters as mixtures of Gaussian kernels applied over edge pseudo-coordinates u_{ij} over each edge (i, j). These pseudo-coordinates, such as Euclidean distances or angular relations, allow the operator to explicitly incorporate continuous edge attributes into the message-passing process. Each kernel is parametrized by a learnable mean and covariance, enabling anisotropic and spatially localised filtering. Conceptually, GMMConv is closely related to SplineConv, with both operating on continuous pseudo-coordinates but differing in their choice of basis functions: Gaussian mixtures in GMMConv versus B-splines in SplineConv. While GMMConv

has not been specifically reported on benchmarks such as *QM9*, *ZINC*, or the *OGB* node classification datasets in its original form, it has demonstrated strong results on geometric learning tasks and remains competitive with other spatial operators.

Operators with partial or indirect edge use

ResGatedConv. The Residual Gated Graph Convolutional Operator [99] introduces edge-dependent gating and residual connections to improve message propagation in deep GNNs. Edge attributes are incorporated by concatenating them with the features of connected node pairs, allowing the gating mechanism to modulate information flow based on both node and edge characteristics. The residual pathway stabilises optimisation and enables the stacking of deeper layers without performance degradation. It performs competitively on *ZINC*, *MNIST*, *ogbl-ppa*, and *ogbl-molhiv*.

GENConv. GENeralized Graph Convolution (GENConv), also known as DeeperGCN, was introduced by Li et al. [41]. It combines learnable message aggregation (Softmax or PowerMean) with pre-activation residual blocks and message normalisation to stabilise very deep GNNs. Edge features are added directly to node features before ReLU activation, and the operator supports both fixed and dynamic aggregation control via learnable parameters. GENConv has demonstrated strong empirical performance on large-scale benchmarks such as *ogbn-proteins* and *ogbg-molhiv*.

PNAConv. The Principal Neighbourhood Aggregation operator, proposed by Corso et al. [96], combines multiple statistical aggregators (mean, max, min, standard deviation) with degree-scalers (identity, amplification, attenuation) to construct highly expressive message functions. This design increases theoretical discriminative power and empirical robustness in tasks that involve variable neighbourhood structures. Edge features are not required, but can be included through the message function. PNAConv has shown top-tier performance on graph regression and classification tasks such as *ZINC* and *ogbg-molhiv*.

Operators with limited edge support

GCN+. The Graph Convolutional Network (GCN) introduced by Kipf & Welling [39] aggregates strictly along adjacency edges but does not condition messages on continuous edge attributes. GCN+ extends this baseline with residual and identity mappings, dropout, and feed-forward networks, addressing oversmoothing and improving stability at depth [100]. In some formulations, GCN+ also supports explicit edge feature integration by adding edge attributes as weighted terms in the message function. Luo et al. [100] showed that this edge-augmented variant outperforms the version without edge features, confirming the benefit of even simple edge conditioning.

GeneralConv. GeneralConv represents a flexible design space for GNN architectures explored by You et al. through the GraphGym framework [97]. It spans design choices such as skip connections, number of layers, aggregation functions (sum, mean, max), and pre/post-processing MLPs. While edge features are not explicitly modelled in the standard formulation, GraphGym demonstrated how structured experimentation across such architectural components affects performance across benchmarks. The best designs typically include skip connections and sum aggregation, achieving strong results on *ogbg-molhiv* and node classification benchmarks. However, the lack of explicit edge conditioning may limit its applicability for edge-driven regression tasks.

Attention-based operators

GATv2Conv. The Graph Attention Network v2 (GATv2), introduced by Brody et al. [61], extends the original GAT architecture [101] by removing the static attention limitation, allowing attention coefficients to depend more flexibly on transformed source and target node features. This increases the model's expressiveness and improves performance on heterogeneous and high-variance graphs. GATv2 computes attention scores through a learnable mechanism, normalises them with a softmax function, and aggregates neighbour features as a weighted sum. Although it does not natively incorporate continuous edge attributes, these can be integrated into the attention mechanism through edge-conditioned transformations, enabling adaptation to tasks that rely on geometric relationships. While not as physically grounded as operators explicitly designed for edge features, GATv2 provides a strong, attention-based alternative.

Considered but not evaluated

Several other message-passing operators are frequently discussed in the literature but were not included in the comparative summaries below, either due to their computational cost or limited relevance to geometric edge conditioning. The original Graph Attention Network (GATConv) [101] was excluded in favour of its improved variant, GATv2, which offers greater expressiveness. TransformerConv [102] was omitted due to its higher computational cost relative to the dataset sizes considered and its weaker inductive bias for geometric edge features. The edge-aware GINEConv [103] was excluded because its advantages are most evident when large-scale pre-training is feasible. Finally, PDNConv [104] was not included due to its design focus on densely connected graphs with positional encodings, which do not align directly with the sparser, spatially structured graphs considered here.

Synthesis

Taken together, these operators span the main paradigms of edge conditioning: explicit, indirect, limited, and attention-based. This categorization clarifies the trade-offs between expressive power, inductive bias, and computational cost. In general, operators with explicit edge conditioning offer the most direct means of leveraging geometric and physically meaningful relations, while those with limited or indirect edge use often serve as useful baselines for assessing robustness. Attention-based approaches, in turn, provide an alternative mechanism for adaptively weighting neighbours, complementing more explicitly geometric methods

3.5.2. Comparative Assessment of Operators

The preceding subsections have introduced the candidate operators individually, outlining their design principles and theoretical properties. To position these architectures more concretely within the broader landscape of GNN design, a comparative assessment can be drawn along three complementary dimensions. First, empirical evidence from established benchmarks highlights which operators have demonstrated strong performance across molecular, computer vision, and large-scale graph datasets. Second, model complexity is often discussed in terms of parameter growth under different representational regimes, revealing trade-offs between expressivity and tractability. Finally, the extent and manner of edge feature integration are compared, since this aspect is particularly critical for tasks governed by geometric and physical relations. Together, these dimensions provide a framework for analysing the strengths and limitations of existing operator families.

Benchmark Analysis

Empirical performance across established benchmarks provides a complementary perspective on their practical capabilities. Table 3.2 summarises reported results for selected operators on representative datasets that vary in size, structure, and the presence of continuous edge attributes.

The datasets span a broad spectrum of domains, from citation networks (*Cora* [105]) and small molecular graphs (*QM9, ZINC* [46]) to spatial superpixel graphs (*MNIST Superpixels* [47]) and large-scale heterogeneous biological and social networks (*ogbn-proteins, ogbl-ppa, ogbg-molhiv* [45]). This variety enables a comparative view of operator performance under structurally diverse conditions. Datasets with continuous geometric edge features, such as *MNIST Superpixels* or molecular datasets, are structurally analogous to tasks where edges encode physically meaningful geometric relations, while those without continuous edge attributes test robustness when such information is absent. A detailed description of these benchmark datasets is provided in Appendix A.

It is important to note that the results in Table 3.2 are compiled from multiple independent studies. Consequently, surrounding model architectures, training pipelines, and hyperparameter configurations are not standardised across entries. For example, network depth, hidden dimensionality, optimiser choice, and regularisation vary between sources. The absolute performance values are therefore not directly comparable in a strict experimental sense; rather, they should be interpreted as indicative of each operator's potential when implemented within a well-tuned architecture. In addition, some results were obtained in settings optimised for classification or graph-level regression, which may not fully reflect behaviour in continuous node-level regression tasks.

Despite these caveats, several consistent trends emerge:

- *Edge-aware operators excel on geometric datasets:* NNConv achieves the highest accuracy on

MNIST superpixels, while spline-based (SplineConv) and Gaussian-kernel (GMMConv) methods remain competitive, confirming the value of continuous edge feature integration.

- *High-capacity aggregators dominate molecular regression:* PNAConv and GCN+ achieve state-of-the-art MAE on *ZINC*, highlighting the benefits of multi-aggregation and degree-scaling schemes for fine-grained continuous prediction. *Deep residual operators lead on large graphs:* GENConv consistently performs well on large-scale *OGB* datasets, benefiting from residual connections and message normalisation to enable stable deep architectures. ResGatedConv also reports competitive performance on several *OGB* benchmarks, though typically not at the level of GENConv.
- *Attention mechanisms remain competitive:* GATv2 achieves strong performance on *ogbn-proteins*, indicating potential for capturing long-range, heterogeneous interactions in spatial graphs, even when continuous edge features are not directly modelled.

Beyond absolute performance, two broader behavioural tendencies are evident in the literature. First, inductive bias, the architectural assumptions an operator embeds about spatial or relational dependencies, strongly influences how well it generalises across datasets with differing geometric structure [28, 51]. Operators with explicit geometric conditioning (e.g., NNConv, CGConv, SplineConv) show high accuracy on spatially structured data but reduced flexibility on non-geometric graphs. Second, several studies report pronounced hyperparameter sensitivity for kernel- and aggregator-based methods, where tuning parameters such as kernel resolution, number of aggregators, or degree scalers substantially affects convergence and stability [55, 96, 100]. These tendencies emphasise that reported benchmark rankings reflect not only intrinsic architectural capacity but also the degree of tuning and prior inductive assumptions embedded in each operator.

Across the reviewed benchmarks, these findings suggest that operators explicitly incorporating continuous edge features (e.g., NNConv, CGConv, SplineConv, GMMConv) tend to perform well in geometrically structured settings. At the same time, high-performing general-purpose operators (e.g., GENConv, PNAConv, GCN+) remain competitive across diverse datasets, indicating that strong architectural regularisation and aggregation schemes can partly compensate for limited edge-specific conditioning. ResGatedConv typically occupies an intermediate position between these groups, exhibiting moderate and consistently stable performance.

Table 3.2: Benchmark performance of selected GNN operators across multiple datasets. Best results per dataset are highlighted in **bold**. The † indicates that the operator did not utilize edge features.

Dataset	Metric	GENConv	GeneralConv	PNAConv	NNConv	CGConv	SplineConv	GMMConv / MoNet	GATv2Conv	GCN+	ResGatedConv
CORA	–	–	–	–	–	–	**89.48** [95]	81.69† [47]	83.50 † [106]	–	–
ZINC	MAE	–	–	0.188 [96]	–	–	–	0.397† [55]	–	**0.076** [100]	0.36 [96]
MNIST	Acc	–	–	97.69 [96]	**99.14** [91]	–	95.22 [95]	90.81† [55]	–	98.88 [100]	97.47 [96]
ogbn-proteins	ROC-AUC	0.860 [41]	–	–	–	–	–	–	0.77†[61]	–	–
ogbl-ppa	Acc	0.771 [41]	–	–	–	–	–	–	–	**0.808**	0.753 [100]
ogbl-molhiv	ROC-AUC	0.786 [41]	0.792† [107]	0.791† [96]	–	–	–	–	–	**0.802** [100]	0.769 [100]

Model Complexity Considerations

Beyond predictive performance, the computational complexity of a GNN operator is a central aspect of its practical usability. Complexity determines both the scalability of training and the model's effective capacity to generalise from limited data. In the broader literature, computational cost is typically characterised in terms of the number of trainable parameters or the asymptotic cost of message passing as a function of node feature dimension d, edge feature dimension e, and neighbourhood size [37, 90].

Operators that employ neural networks to generate edge-conditioned filters, such as NNConv and CGConv, exhibit quadratic or cubic scaling with respect to d, since each edge defines a distinct weight matrix [90, 91, 98]. Residual or normalised operators such as GENConv or GCN+ maintain more moderate growth of $\Theta(d^2 + ed)$, combining residual connections and message normalisation to preserve depth stability without excessive parameterisation [41, 100]. Kernel-based operators, including SplineConv and GMMConv, partially decouple complexity from d by defining convolutional filters over continuous pseudo-coordinates using spline or Gaussian kernels [47, 95]. Attention-based mechanisms such as GAT and GATv2 follow a similar $\Theta(d^2 + ed)$ scaling pattern but introduce additional constant factors due to key–query projections [61, 101].

Comparative studies have shown that increased model capacity can improve performance on complex datasets but often at the cost of longer training times and greater overfitting risk [55, 97]. Consequently,

operator selection in practice must balance expressivity, numerical stability, and computational feasibility, particularly in data-constrained regression settings where excessive model capacity may outweigh the benefits of architectural sophistication.

Edge Feature Handling Mechanisms

A key differentiator between message-passing operators lies in how real-valued edge attributes are incorporated into the aggregation process. In general, edge features encode relational or geometric dependencies between connected nodes, such as distances, angles, or interaction strengths, and their treatment strongly influences a model's ability to capture structured relationships [37, 90, 91].

Across the surveyed operators, four main strategies can be distinguished:

- **Direct addition:** Edge features are added to node messages before nonlinearity, as in GENConv or ResGatedGCN [41, 99]. This approach is parameter-efficient and stable at depth but may underexploit multi-dimensional edge descriptors.
- **Edge-conditioned weights:** Operators such as NNConv and CGConv generate edge-specific filters via a neural network applied to edge features [90, 91, 98]. This maximises expressivity and adaptivity but increases parameter counts and risk of overfitting.
- **Kernel interpolation:** SplineConv and GMMConv define continuous filters over pseudo-coordinates using B-splines or Gaussian mixtures [47, 95]. These approaches embed a smooth spatial bias and decouple complexity from node feature dimension, though performance can depend on kernel resolution and coordinate parametrisation.
- **Indirect integration:** Methods such as GATv2, PNAConv, or GCN+ incorporate edge information through auxiliary channels (e.g., concatenation with node features, pre-/post-aggregation MLPs, or attention scores) [61, 96, 100]. This design offers architectural flexibility but often reduces physical interpretability when edge attributes dominate the task.

This classification highlights a fundamental trade-off between simplicity, expressivity, and interpretability in graph operator design. Explicit edge conditioning and kernel-based formulations provide strong inductive biases for learning structured relations, whereas indirect or additive approaches prioritise architectural stability and computational efficiency. The choice among these strategies therefore depends on whether the target domain is primarily geometric or topological in nature.

Chapter Summary

This chapter has reviewed the theoretical and methodological background underpinning the development of a graph-based surrogate model for daylight prediction. It began with an overview of machine learning approaches for daylight modelling, where ANN-based surrogates established the feasibility of learning daylight metrics from geometric descriptors but remained limited in two respects: they encode spatial dependencies only implicitly, and they are almost exclusively evaluated within the same geometric distribution used for training. These gaps motivate the exploration of more relational model families capable of reasoning over unseen configurations.

The discussion then turned to GNNs within the building engineering domain, illustrating how graph-based learning has been applied across contexts such as structural mechanics, energy forecasting, and BIM semantics. These studies demonstrated that GNNs can capture both physical and semantic dependencies by structuring domain knowledge as nodes and edges, providing methodological precedents for representing daylight interactions between windows and sensors.

Building on this foundation, the chapter examined feature representation as a central determinant of GNN performance. The literature shows that predictive accuracy depends not only on the expressive power of message-passing operators but also on the informativeness and physical grounding of node and edge attributes. Three families of feature encodings from prior ANN-based surrogates, geometry-based, directional, and obstruction-aware, were identified as conceptual baselines for reformulation in a graph setting.

Finally, a survey of message-passing operators outlined how different operators incorporate edge information through mechanisms such as direct addition, edge-conditioned weighting, kernel interpolation, or attention-based aggregation. Comparative evidence from established benchmarks and theoretical

analyses clarified the trade-offs between expressivity, computational complexity, and edge-feature integration, leading to the identification of a focused set of candidate operators for further study.

Together, these four strands establish the conceptual basis for the research that follows. They show that the transition from ANNs to GNNs enables a more physically grounded and relational treatment of daylight prediction, while also providing a framework for testing geometric generalisation beyond the training distribution. The next chapter therefore translates these insights into a two-part methodological framework: first, the construction and optimisation of candidate GNN architectures, and second, their evaluation against unseen geometries using ANN surrogates as performance benchmarks.

4

Methodology

This chapter outlines the methodological framework used to develop and evaluate surrogate models for DF prediction. It first describes the simulation pipeline and dataset generation process, which produces consistent training, transformation, and Final Test Dataset following EN 17037 principles. The chapter then explains the construction of the benchmark ANN models and the graph-based WindowGraphNet, detailing how room geometries and sensor grids are represented as graphs. The subsequent sections define the training and evaluation protocol, including feature ablation, Bayesian Optimisation (BO), and operator selection. Finally, the performance of the selected model is evaluated on the Final Test Dataset to assess out-of-distribution generalisation. Together, these components establish a reproducible workflow linking physical simulation, data representation, and graph-based learning.

4.1. Dataset Creation

All datasets are generated with a shared simulation pipeline, but differ in their geometric parametrisation and methodological role. The simulation process, including all fixed parameters, is described first, followed by the dataset-specific structures.

4.1.1. Simulation Pipeline

All datasets are generated using a consistent simulation workflow, designed to follow the principles of EN 17037 [23] as closely as possible. The pipeline combines parametric modelling in Grasshopper, daylight simulation via the Honeybee (v1.8.0) plugin, and Radiance-based computation of DF.

Geometry setup. Room height is fixed at 3.0 m and wall thickness at 0.2 m. These values are chosen as representative of typical building practice, while keeping the parametrisation focused on width and window-to-wall ratio (WWR), which are among the most influential variables for daylight availability [108].

Surface properties. Standard Honeybee defaults are adopted for material properties, namely a reflectance of 0.8 for ceilings, 0.5 for walls, and 0.2 for floors, and a glazing visible transmittance of 0.6. These values lie within the recommended ranges specified in EN 17037 (0.7–0.9 for ceilings, 0.5–0.8 for walls, and 0.2–0.4 for floors) [23]. Surface reflectances do influence DFs, but their impact is comparatively modest: Simm and Coley [109] show that even a 0.1 change in wall reflectance alters the average DF by less than 0.06 in typical classrooms and offices, whereas variations in window size or room proportions produce much larger differences. Because the influence of reflectance is well understood and secondary to the geometric parameters under study, reflectances are held constant. This ensures that the analysis isolates the dominant drivers of DF while maintaining consistency with standard daylighting practice and enabling reproducibility.

Reference plane. EN 17037 specifies a reference plane at 0.85 m above floor level for DF calculations. In this study, a plane at 0.70 m is used. This value is also consistent with typical desk and table heights in residential and office contexts, and similar ranges (0.70–0.80 m) are commonly adopted in simulation studies [18, 110–112]. While this deviates slightly from the EN 17037 convention, it does not affect the

relative comparisons between geometries and models in this thesis, since all datasets are simulated consistently at the same plane. The deviation is documented explicitly to ensure reproducibility.

Band. In accordance with EN 17037, a band of 0.50 m is excluded from the perimeter of each room when generating the calculation grid. This exclusion removes zones that are strongly influenced by boundary effects, such as direct wall reflections or corner occlusions, which would otherwise bias the assessment of overall daylight availability in the space. From a methodological perspective, the use of a fixed 0.50 m band also ensures comparability between different room sizes and shapes, as the sensor placement is not dominated by local edge effects. All datasets in this thesis are generated using this exclusion rule, which is consistent with both the standard and typical practice in daylight simulation workflows.

Grid spacing. Within the remaining floor area, EN 17037 prescribes that the calculation grid should have a spacing of about 0.50 m, with sensors located at the centre of each cell. This rule is applied consistently in all datasets, resulting in a uniform and reproducible grid. The number of rows and columns followed directly from the room's length and width, ensuring that the grid adapted automatically to different geometries.

Sky condition and Radiance settings. All simulations used the CIE standard overcast sky, as required by EN 17037 for DF calculations. Radiance is executed with default Honeybee parameters, which are validated and widely used within the daylighting community. No custom overrides are applied to maintain reproducibility across datasets.

Outputs. For each simulation, the output consists of DF values computed at all sensor points within the room. The results are stored as an array indexed by sensor coordinates, providing a consistent spatial format that is maintained across all datasets and used for model training and evaluation.

Reproducibility. By explicitly documenting all fixed parameters, room height, wall thickness, surface reflectances, glazing transmittance, grid spacing and height, exclusion band, sky condition, and software defaults, the simulation pipeline can be exactly reproduced by other researchers. Although Honeybee defaults are used, these are verified to fall within the normative ranges of EN 17037, ensuring methodological consistency with the standard.

4.1.2. Dataset Structure and Purpose

While the simulation pipeline and calculation setup are consistent across all cases, the datasets generated from this pipeline differ in their geometric parametrisation and in their methodological role within the study. Each dataset supports a distinct stage of the research workflow: model training, feature and model selection, and final evaluation.

Three datasets are created in total, each designed to address one of these stages in a controlled and non-overlapping manner.

Training and Validation dataset. This dataset provides the foundation for model training and all subsequent analyses. It consists of square rooms with variable width and window-to-wall ratio (WWR), generated using the parametric pipeline described earlier and visualised in Figure 4.1. For all configurations, the window is consistently placed on the south-facing wall to ensure a uniform reference orientation across samples. All models, both ANN and GNN, are trained exclusively on this dataset. The validation subset is used throughout the thesis for early stopping, hyperparameter tuning, and performance monitoring. Because this dataset defines the statistical distribution on which models are fitted, it represents the in-distribution domain against which generalisation is later assessed.

Transformation Dataset. The *Transformation Dataset* introduces controlled geometric variations, including scaling and rotation of the base geometries, to probe robustness beyond the exact training distribution. As shown in Figure 4.1, it extends the base coverage using rotated and scaled versions of the same layouts. It is used jointly with the validation set to evaluate how feature formulations and model architectures respond to basic geometric transformations. This dataset therefore serves as a diagnostic testbed for both feature ablation and operator selection: it reveals whether specific feature encodings or message-passing operators retain accuracy under rotated or scaled variants of the same geometry, without introducing qualitatively new room types. Because it shares the same simulation

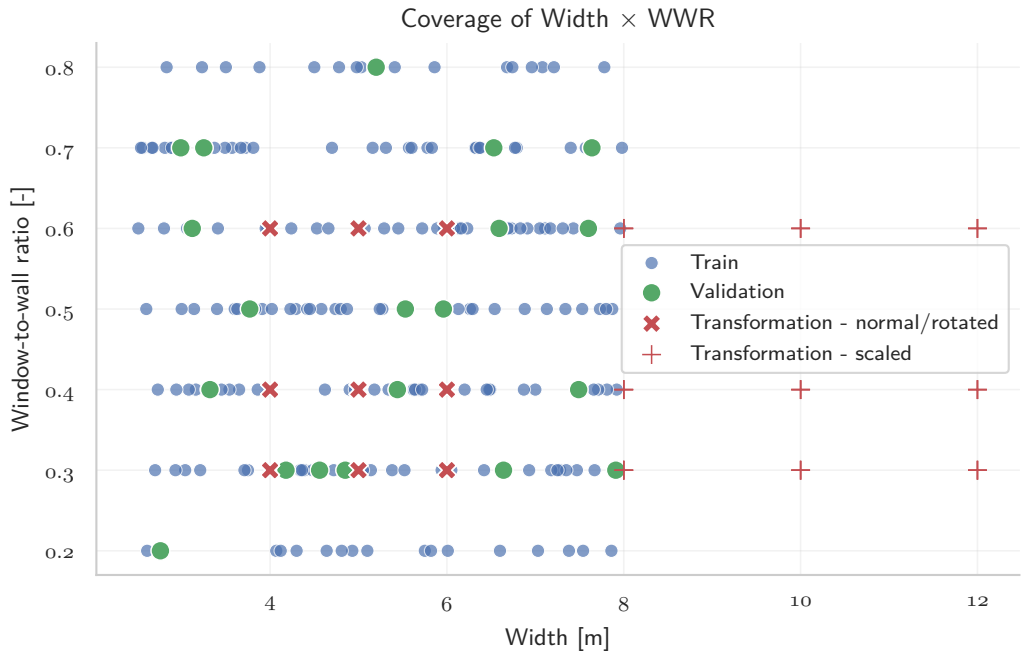


Figure 4.1: Scatter plot of sampled room configurations in the width–WWR space. Each marker represents one simulation case, with colours distinguishing the Training, Validation, and Transformation set. The Latin Hypercube Sampling ensures broad coverage of the design space, while the stratified split preserves consistency between Training and Validation.

parameters as the training data, performance differences observed here can be attributed purely to the models' internal representations rather than to shifts in physical conditions.

Final Test Dataset. The *Final Test Dataset* provides the ultimate evaluation stage, containing unseen geometries and boundary conditions that are never encountered during training. It includes rectangular rooms with offset windows and progressively more complex L-shaped floorplans with varying degrees of self-occlusion. This dataset is never used for training, tuning, or feature or model selection; it is reserved exclusively for comparing the final *WindowGraphNet* configuration, selected from the transformation-stage experiments, against benchmark ANN surrogates. Its purpose is to measure generalisation to novel spatial layouts and to determine whether the graph-based formulation offers a tangible improvement in predictive robustness over established ANN baselines.

Training and Validation Dataset

This dataset provides the foundation for model learning. It consists of 200 square rooms in which only width and WWR are varied, while all other parameters follow the simulation pipeline. The WWR is defined as the ratio of glazing area to total façade area,

$$WWR = \frac{A_{\text{glazing}}}{A_{\text{wall}}}. \tag{4.1}$$

Focusing on width and WWR is deliberate: these are among the dominant geometric drivers of DF, whereas variations in height or reflectance have a comparatively minor impact [108, 109]. Fixing the less influential parameters reduces dimensionality while keeping the analysis centred on the most relevant predictors.

The two variables are sampled using Latin Hypercube Sampling (LHS) across the ranges width $\in [2.5, 8.0]$,m and WWR $\in [0.2, 0.8]$, with a fixed random seed of 42 to ensure reproducibility. LHS provides space-filling coverage of the parameter space and avoids clustering along either axis. To maintain compatibility with the downstream pipeline, the generated values are lightly discretised (width rounded to two decimals, WWR to one decimal). The window is always placed centrally on the façade, so that only its size varied with WWR while its position remained fixed. In total, 200 distinct room configurations are obtained.

The dataset is then divided into 180 training and 20 validation cases. The validation set is used exclusively for early stopping and monitoring of training convergence, never for fitting. To ensure that both width and WWR are proportionally represented in the split, a stratified sampling strategy is applied: width is divided into five bins across its range, and WWR into four bins, with samples drawn proportionally from each stratum. Such stratification prevents imbalance between subsets and is a standard approach for constructing validation sets in ML [113]. The resulting distributions for width and WWR across the full, training, and validation subsets are shown in Figure 4.2. For visual clarity, the histogram bins in Figure 4.2 do not correspond to the stratification bins used for the training–validation split; the latter are defined independently to ensure proportional sampling across the joint width–WWR space.

The resulting dataset forms the common basis for model learning. Both the ANN benchmarks and the GNN models are trained exclusively on the training set, while the validation set supports early stopping, convergence monitoring, and Bayesian optimisation (BO) of model operators and hyperparameters. The same fixed training–validation split is maintained across all experiments to ensure comparability. Robustness to geometric variations is assessed using the *Transformation Dataset* during feature ablation and operator benchmarking, without incorporating these data into training. The final selected configuration is then evaluated on the *Final Test Dataset* to assess generalisation to unseen geometries and boundary conditions. All sampling and splitting operations are performed with fixed seeds, ensuring the dataset can be reproduced exactly or extended with additional samples if required.

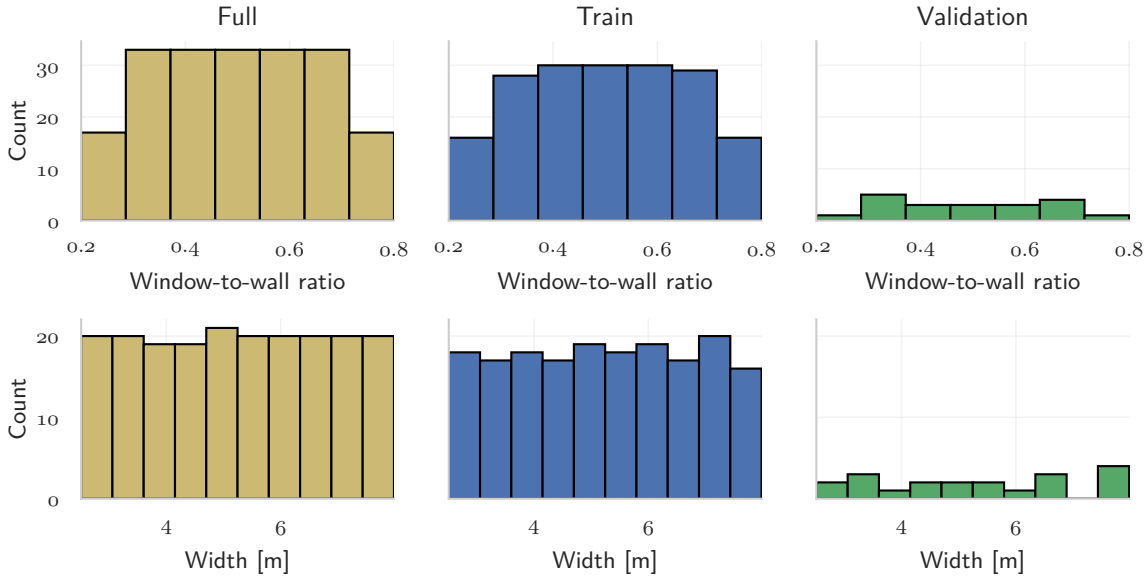


Figure 4.2: Distributions of window-to-wall ratio (WWR, top row) and room width (bottom row) for the full dataset (left), training subset (middle), and validation subset (right). The plotted histograms use visually uniform bins for clarity; these do not correspond to the stratification bins used to construct the training–validation split. Stratification is performed independently to preserve proportional representation of both variables across subsets, ensuring that training and validation data are drawn from comparable regions of the parameter space.

Transformation Dataset

This dataset introduces controlled geometric transformations to systematically evaluate the robustness of feature representations. The underlying parametrisation remains based on square geometries (equal width and depth), ensuring that observed effects arise solely from the applied transformations rather than shape differences. The *Normal* configuration corresponds to the original square-room geometries already represented in the training data and is included here solely as a baseline reference against which rotated and scaled variants can be compared. The composition of the dataset is summarised in Table 4.1. Invariance and equivariance to such geometric transformations are widely recognised as critical properties for geometric deep learning models [49, 51]. The dataset is therefore used exclusively for feature ablation and operator selection experiments, providing a controlled setting to assess whether features and message-passing operators encode relationships that generalise across transformations.

Figure 4.3 illustrates the four transformation scenarios: *Normal*, *Rotation* $90°$, *Rotation* $180°$, and

Table 4.1: Transformation dataset composition table

Transformation	Scaling factor	Rotation angle	No. of rooms	Description
Normal	$1\times$	$0°$	9	Base configurations
Rotation $90°$	$1\times$	$90°$	9	Window rotated $90°$
Rotation $180°$	$1\times$	$180°$	9	Window opposite wall
Scaled	$2\times$	$0°$	9	Double room size, same WWR

Scaled. For rotation, the room geometry (including the window) is rotated around the vertical axis by $90°$ or $180°$ while the sensor grid remains fixed in the global coordinate system. Consequently, the spatial sampling pattern of sensors is identical across rotation cases, but the aperture shifts to a different wall. Since physical properties and sensor–to-window distances are unchanged, the resulting DF fields are effectively invariant to rotation, as also reflected in the near-overlapping DF distributions in Figure 4.4 (right).

In contrast, the scaled case introduces a systematic change in both geometry and measurement resolution. The room width is multiplied by a factor of two while maintaining the same WWR, so glazing area scales proportionally with the façade. The sensor grid scales with the geometry: points that are approximately 0.5 m apart in the base case become 1.0 m apart under scaling. This preserves relative sampling density but doubles absolute sensor–sensor distances, reducing the window's subtended solid angle and attenuating illuminance. As shown in Figure 4.3 and in the DF distributions (right, Figure 4.4), consistent with the geometric attenuation of interior illuminance reported by Simm [109]. The left panel of Figure 4.4 confirms that the coordinate distributions are comparable across scenarios, except for the enlarged domain in the scaled case.

In total, 36 configurations are generated: 9 base or normal cases, 9 scaled cases (factor 2), and 18 rotated cases (90° and 180°). The compact and balanced design of this dataset is intentional. Rather than being incorporated into training, it functions purely as a diagnostic benchmark to evaluate how effectively features and message-passing operators capture geometric relationships under controlled transformations. Variations such as side windows or non-square floorplans are deliberately excluded at this stage, as they represent distinct typologies rather than transformations of a common geometry. Their assessment is deferred to the *Final Test Dataset*, which evaluates generalisation to previously unseen geometric families.

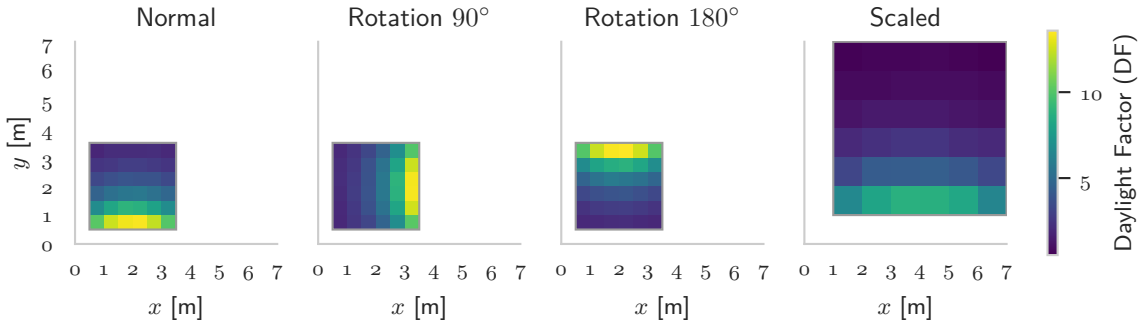


Figure 4.3: Example DF distributions for the four transformation scenarios: Normal, Rotation $90°$, Rotation $180°$, and Scaled ($\times 2$). All panels share the same spatial axes to highlight the geometric transformations applied during feature robustness testing.

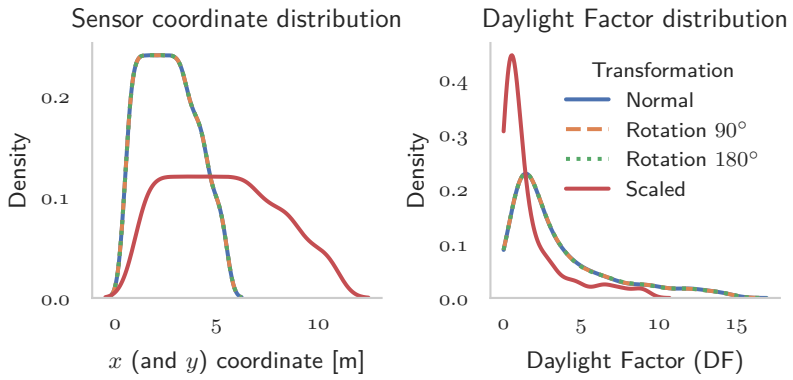


Figure 4.4: Distribution of sensor coordinates and DF values across the four geometric transformations. The left plot shows the spatial distribution of sensor coordinates (identical for x and y in square rooms), while the right plot illustrates the DF distribution. Scaling reduces DF values systematically due to increased sensor–window distance, whereas rotations mainly reposition the aperture relative to the fixed grid.

4.1.3. Final Test Dataset

To assess the generalisation ability of the surrogate models beyond the conditions seen during training, a dedicated *Final Test Dataset* is constructed. While the training data consists exclusively of square rooms with a centrally positioned window, this additional dataset introduces systematic variations in room geometry and window placement. The aim is not to optimise model performance on these new cases, but rather to evaluate the robustness of the learned representations when confronted with unseen and progressively more complex configurations.

To structure this evaluation, the dataset is organised into a series of tiers, each corresponding to a distinct type of out-of-distribution shift. The tiers are designed to increase in complexity: starting from simple geometric transformations of the training distribution, through rectangular extensions and window offsets, and culminating in L-shaped floor plans with varying levels of line-of-sight between sensors and the window, an overview is given in Table 4.2. This tiered setup makes it possible to pinpoint where and how the models begin to fail, and to compare their behaviour against benchmark ANNs under increasingly challenging conditions.

The following subsections provide an overview of each tier, summarising the geometric and positional variations introduced. Detailed generation procedures and parameter definitions are provided in Appendix C.1.

Table 4.2: Overview of the tiers in the *Final Test Dataset*.

Tier	Geometry / Window	Main Variation
0	Square, centred window	Control (training distribution)
1	Square, centred window	Rotations ($90°$, $180°$), scaling $\times 2$
2	Rectangular, centred window	Aspect ratios 2:1 + scaling $\times 5$
3	Rectangular, offset window	Lateral window displacement
4	L-shape, windows on long façades	Window on long façades (indices 3–4); partial self-occlusion
5	L-shape, windows on short façades	Window on short façades (indices 0–2, 5); deep self-occlusion and limited visibility

Façade indices follow the numbering in Fig. 4.6.

Tier 0: In-distribution baseline

The first tier serves as the in-distribution control set. Square rooms are generated with side lengths uniformly sampled between 2.5 m and 8.0 m and window-to-wall ratios (WWR) between 0.2 and 0.8, rounded to one decimal place to match the training distribution. These parameter ranges are identical to those used for model training, ensuring that Tier 0 represents the best-case scenario in which the test data are drawn from the same distribution as the training data. Each room has a single centrally

positioned window, resulting in 50 test cases that form the baseline against which all other tiers can be compared.

Tier 1: Geometric transformations
The second tier consists of geometric transformations applied to the Tier 0 base rooms. Specifically, each of the 50 square rooms is rotated by $90°$ and $180°$, and a uniformly scaled variant is generated with a factor of 2. The purpose of Tier 1 is twofold. First, it replicates the transformation tests already used for feature ablation and operator benchmarking, providing a direct link to earlier experiments. Second, it allows assessment of whether the models generalise consistently to rotated and scaled variants of otherwise identical rooms. In total, Tier 1 yields 150 transformed cases (three per base room).

Tier 2: Out-of-distribution rectangles and scaling
The second out-of-distribution tier introduces test cases that are not present during training or tuning. While the models are only exposed to square rooms, Tier 2 contains rectangular geometries and extreme scaling conditions. For each base room, two rectangular variants are created: a horizontal rectangle with doubled width and a vertical rectangle with doubled depth. Additionally, a uniformly scaled square variant is generated with a factor of 5, producing rooms with maximum side lengths up to 40 m. These three variants per base room result in 150 test cases. Tier 2 therefore probes the models under truly out-of-distribution conditions, evaluating their robustness to aspect ratios up to 2:1 and to very large room dimensions far beyond the training domain. Examples of the wide and tall rectangular variants are shown on the left in Figure 4.5, illustrating how aspect-ratio changes affect the distribution of daylight across the interior while keeping other parameters constant.

Tier 3: Rectangular offset windows
The third tier introduces lateral window displacement while maintaining a simple rectangular geometry. All rooms in this tier have similar aspect ratios to those in Tier 2, but the window is no longer centred on the façade. Instead, it is randomly offset to one side, resulting in asymmetric light distribution across the space. This configuration isolates the effect of window position from other geometric changes, allowing the models' sensitivity to lateral asymmetry to be assessed. Rooms are simple rectangles with a 2:1 aspect ratio, where the long side measures approximately 5–8 m and the short side 2.5–4 m. These dimensions are comparable to those used during training, meaning that this tier does not test scaling or extreme size variation. A total of 50 offset-window rooms are included. An example of the lateral window placement and resulting asymmetry in the daylight distribution is shown on the right in Figure 4.5.

Tier 4: L-shaped rooms with partial self-occlusion
The fourth tier introduces L-shaped geometries in which the window openings are located on the longer exterior façades of the plan. These placements, corresponding to façade indices 3 and 4 in Figure 4.6, result in broad façade exposure and relatively shallow recesses. Most of the interior remains in partial view of the window, with only a limited portion of the recessed area being indirectly lit. The resulting daylight distribution combines direct exposure near the façade with moderate attenuation into the recess, as reflected by the window-location density in Figure 4.7. All rooms share comparable overall dimensions with the rectangular tiers, with plan widths up to 8 m and recess depths of roughly half that length. Examples of Tier 4 configurations are shown on the left in Figure 4.8, illustrating how these geometries preserve wide façade visibility while introducing mild self-occlusion in the recessed zone. Tier 4 therefore examines whether the models can generalise to configurations that exhibit partial self-occlusion while maintaining extended façade exposure, testing their ability to capture smooth daylight gradients and gradual spatial decay of illuminance.

Tier 5: L-shaped rooms with deep self-occlusion
The fifth tier retains the same overall proportions and recess geometry as Tier 4 but repositions the windows onto the shorter façades of the L-shape. These placements, corresponding to façade indices 0, 1, 2, and 5 in Figure 4.6, are reflected in the distribution shown in Figure 4.7. Placing windows on the short façades reduces the illuminated surface area and produces strong directional dependence, as large parts of the recess lose direct line-of-sight to the opening. As can be seen in Figure 4.8, these configurations lead to much darker recess regions and lower overall DF values, consistent with the loss of direct sky exposure. Illuminance in these areas relies primarily on inter reflection from adjacent

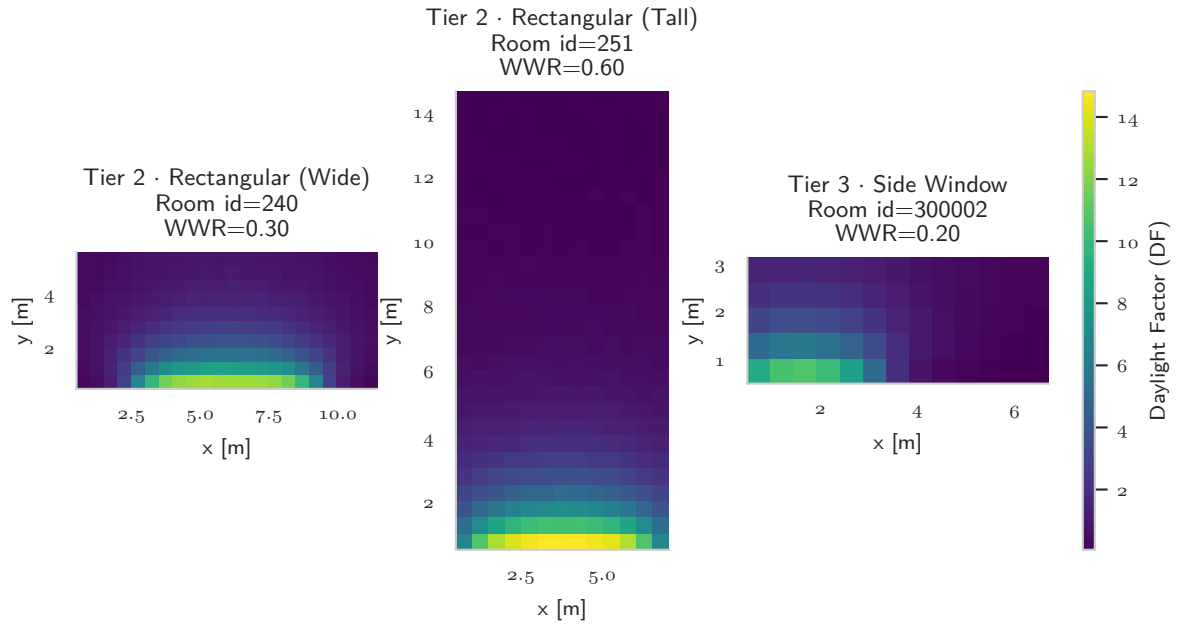


Figure 4.5: Representative DF distributions for rectangular and offset-window cases. The Tier 2 examples illustrate the *wide* (left) and *tall* (centre) rectangular variants, which introduce aspect-ratio variation without altering scale, while the Tier 3 example (right) shows the lateral window offset leading to asymmetric light distribution across the space. These cases highlight the controlled geometric shifts used to evaluate robustness to aspect ratio and window position.

walls and ceiling surfaces, resulting in steeper daylight gradients. As all geometric parameters are held constant with respect to Tier 4, this tier isolates the effect of deep self-occlusion and limited façade exposure. Tier 5 thus evaluates the models' ability to extrapolate DF patterns into partially or fully hidden zones, where illumination is dominated by indirect light rather than direct sky exposure.

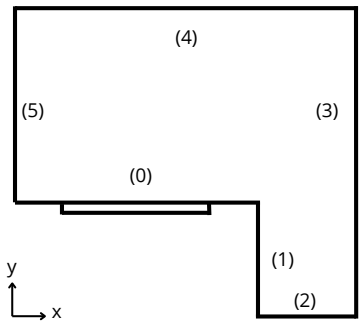


Figure 4.6: Schematic showing façade indexing for the L-shaped geometry. Window placements for Tier 4 occur on the longer façades (indices 3 and 4), while Tier 5 uses the remaining façades (indices 0, 1, 2, and 5).

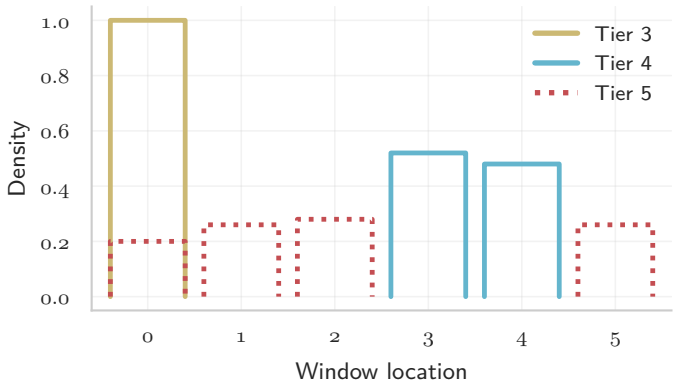


Figure 4.7: Empirical distribution of window-wall index g for Tiers 3–5, confirming façade assignments: $g{=}0$ for Tier 3 (long side of rectangle), $g{\in}\{3,4\}$ for Tier 4 (long façades), and $g{\in}\{0,1,2,5\}$ for Tier 5 (short façades).

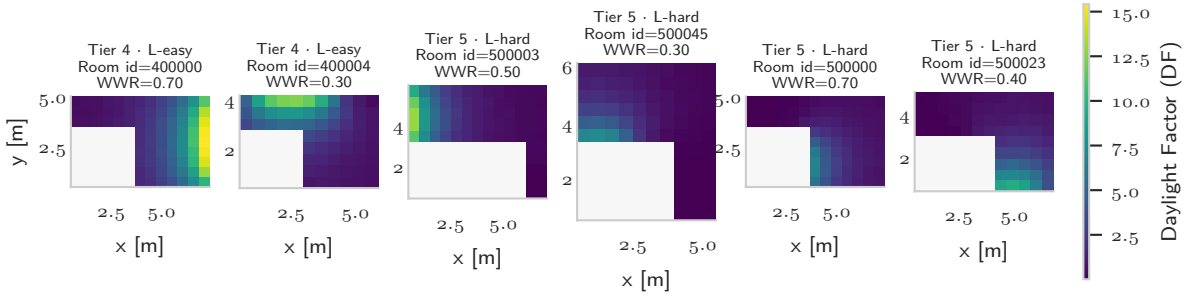


Figure 4.8: Example DF distributions for L-shaped configurations from Tiers 4 and 5. Tier 4 (left) shows windows placed on the long façades, resulting in broad exposure and partial self-occlusion, while Tier 5 (right) displays openings on the short façades, producing deeper occlusion and lower overall DF values. The variety of façade indices (0–5) corresponds to the window placements defined in Figure 4.6. Together, these examples illustrate how the *Final Test Dataset* spans a broad range of visibility conditions and occlusion severities.

Table 4.3: Summary of geometric characteristics and variants across the six tiers of the *Final Test Dataset*. Tiers 1 and 2 include multiple controlled variants, while Tiers 0 and 3–5 represent single configurations.

Tier	Variant	# Rooms	Width (m)	Depth (m)	WWR	Shape
0	–	50	2.5–8.0	2.5–8.0	0.2–0.8	Square
1	Rotation $90°$	50	same	same	same	Square (rot.)
	Rotation $180°$	50	same	same	same	Square (rot.)
	Scaling $\times 2$	50	same	same	same	Square (scale)
2	Rectangular (wide)	50	up to 16	2.5–8.0	0.2–0.8	Rectangular (2:1)
	Rectangular (tall)	50	2.5–8.0	up to 16	0.2–0.8	Rectangular (1:2)
	Scaling $\times 5$	50	up to 40	up to 40	0.2–0.8	Square (scaled)
3	–	50	5–8	3.5–4	0.1–0.4	Rectangular offset
4	–	50	4 – 8	4–8	0.1–0.7	L-shallow
5	–	50	4–8	4–8	0.1–0.7	L-deep

In Figure 4.9, the WWR panel shows that Tiers 0–2 follow the training range (0.2–0.8), whereas Tier 3 is shifted to lower values because the window is confined to one side of the façade, so the full wall width cannot be used. Tiers 4–5 are likewise lower on average: in the L-shape the openings are not centred and are constrained by shorter façade segments, which reduces the achievable glazing area and thus WWR.

The room-width panel indicates similar ranges for all tiers except Tier 2, which extends to much larger widths due to the $\times 5$ scaling variant. In the window-façade-width panel, Tier 5 is clearly lower because

windows are placed on the short façades of the L-shape, while the other tiers, including Tier 4 with long-façade placement, cluster around the common range.

Finally, the DF panel shows broadly comparable distributions for Tiers 0–4, but a marked shift toward lower values for Tier 5. This reflects the extensive self-occlusion in the recess when the opening is on a short façade, which reduces direct sky exposure and increases the proportion of indirectly lit sensors.

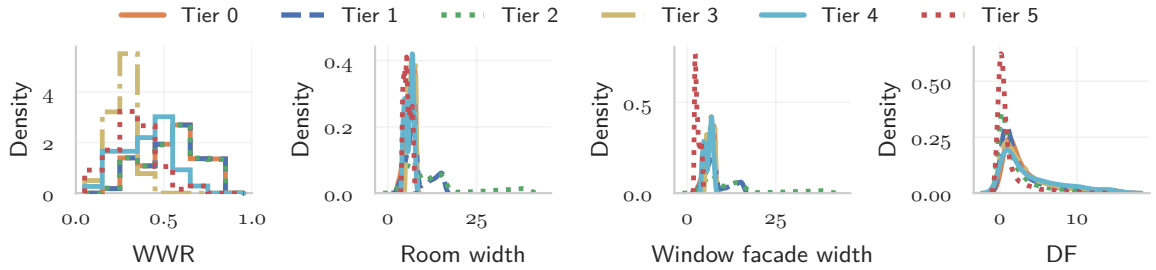


Figure 4.9: Comparison of key geometric and photometric distributions across the six tiers of the *Final Test Dataset*. The plots illustrate how parameters such as WWR, window wall width, window façade width, and DF distributions evolve from the in-distribution baseline (Tier 0) toward the most complex L-shaped configurations (Tier 5). Each tier introduces controlled geometric shifts while maintaining consistent material and simulation parameters.

4.2. Benchmark Models (ANNs)

ANNs trained on tabular feature vectors have been widely used in daylight surrogate modelling, with notable examples by Han et al. [17], Le-Thanh et al. [18], and Dieguez et al. [19]. In this thesis, three ANN baselines are defined, conceptually inspired by these prior studies but adapted to the present dataset and evaluation framework. The benchmarks serve two purposes: (i) to provide reference points for subsequent graph-based formulations, and (ii) to examine how progressively richer feature encodings affect predictive performance and robustness.

Raw baseline. Han et al. proposed a coordinate- and geometry-based encoding that achieves translation, rotation, and scaling invariance by normalising sensor coordinates and reorienting the coordinate basis. This invariance mechanism is not adopted here, as invariance properties are instead assessed explicitly at the feature level. A simplified baseline is therefore introduced using four scalar descriptors: sensor coordinates (x, y), room width, and WWR. These variables represent direct geometric quantities without further abstraction. The raw baseline establishes how far a model can progress without physics-informed or invariant features, functioning as a pedagogical lower bound for comparison.

Le-Thanh baseline. Le-Thanh et al. developed a feature encoding strategy designed to generalise across layouts by decoupling the input from the global coordinate frame. Their formulation combines (i) radial obstacle distances, (ii) Euclidean distances from sensors to window corners, and (iii) relative angular orientations. Together, these vectors capture obstruction, proximity, and directionality in a sensor-centric manner. This encoding is included as a second baseline, as it represents a computationally efficient abstraction of local geometry and provides a useful test of robustness under geometric transformations.

Dieguez baseline. Dieguez et al. proposed a comprehensive set of descriptors grounded in daylighting theory, explicitly linked to the three canonical components of the DF (direct, externally reflected, and internally reflected). These features include solid-angle projections, angular tilts, room-averaged metrics, and sensor-to-window relationships. The feature set is adopted largely verbatim, with the exception of frame ratio and external obstruction factors, which are not relevant to the single unobstructed façade considered in this study. In contrast to the raw and Le-Thanh baselines, this encoding is explicitly physically informed and therefore represents an upper bound on ANN performance. Both the original and a simplified network configuration are implemented to assess potential effects of overparameterisation. To examine the effect of architectural complexity, two network variants are implemented: the original Dieguez model, which uses a deeper structure with mixed sigmoid and ReLU activations; and a *Simple Dieguez* version, which applies the same feature set but uses the lightweight architecture from the *RAW* baseline, a two-layer ReLU network with 32 hidden units. This distinction isolates the influence of architectural design from the feature set itself, allowing a clearer comparison across baselines.

Taken together, the three ANN baselines span a spectrum of prior knowledge: from raw geometric inputs with minimal inductive bias, through sensor–window relational encodings that abstract local geometry, to physics-inspired descriptors with maximal inductive bias. All models operate directly on tabular feature vectors and predict DF at the sensor level. Their role within this thesis is to act as benchmarks against which the benefits of graph-based formulations can be assessed.

All benchmark models are implemented as fully connected multilayer perceptrons (MLPs), trained to predict the DF at each sensor position. Four configurations are considered: the *Raw* baseline (minimal geometric inputs), the *Le-Thanh* baseline (relational descriptors), and two *Dieguez* variants (the original five-layer architecture and a simplified two-layer version). For each feature family, the input layer dimension corresponds directly to the number of descriptors in the respective encoding. Hidden layers are arranged in a standard feedforward configuration with ReLU activations (except for *Dieguez*, chosen to balance expressiveness and training stability. The output layer consists of a single neuron with linear activation, providing a scalar DF estimate for each sensor. A full overview of the model architectures can be found in Table C.4. This setup reflects the convention established in prior ANN-based daylight surrogates, where predictive performance depends primarily on the informativeness of the input features rather than on architectural complexity. Exact hyperparameters, including layer widths, dropout rates, and optimisation settings, are reported in Appendix C.2.

4.3. Graph Construction

Having established the benchmark feature sets from the ANN experiments, these descriptors are now embedded into graph structures that enable physically meaningful message passing. The aim is to move beyond independent sensor predictions and instead represent the spatial and geometric relationships that govern daylight propagation. To this end, two graph variants are introduced. The homogeneous model represents the sensor grid alone, where edges capture local spatial coupling between neighbouring sensors. The heterogeneous model augments this with explicit window corner and centre nodes, allowing directional information to flow from apertures to sensors. Both variants share the same goals, data splits, and feature standardisation procedures as the ANN baselines; the only distinction lies in their graph construction.

4.3.1. Homogeneous WindowGraphNet

In the homogeneous formulation, the room is represented as a regular sensor grid in which each sensor corresponds to a node. Edges connect neighbouring sensors to capture local spatial interactions, enabling the model to represent the smooth variation of the DF field. This design builds on the idea that message passing between adjacent nodes can approximate the diffusive coupling inherent to daylight distribution, a behaviour often linked to Laplacian smoothing in graph convolutional networks [58].

Node features

Each sensor node is initialised with the feature descriptors introduced in the ANN baselines (Han et al. [17], Le-Thanh et al. [18], Dieguez et al. [19]), reformulated into node attributes. As the feature-ablation study will demonstrate, these descriptors vary across experiments, and thus the node feature input is not constant across all models. In addition to local sensor descriptors, global room-level scalars (length, width, window-to-wall ratio) are broadcast to all nodes rather than aggregated through a supernode. While virtual or supernodes are a common strategy to inject global context into graph-level tasks (e.g., GIN-virtual [103], Graphormer [114]), they concentrate all global information in a single hub and may exacerbate over-smoothing effects [115]. Broadcasting avoids this bottleneck and preserves the locality of information, which is more consistent with the sensor-level prediction task considered here.

Edge topology

Sensor nodes are connected according to an eight-connected (Moore) neighbourhood on the measurement grid, as shown in Figure 4.10. This ensures that each sensor exchanges information with its four axial and four diagonal neighbours, reducing the shortest-path distance across the grid. Compared to a four-connected (Von Neumann) scheme, the eight-connected topology allows diagonal relations to be represented in a single message-passing step rather than two, which better reflects the diffuse propagation of daylight. The approach is analogous to the nine-point stencil used in finite-difference

approximations of the Laplacian [116]. All edges are treated as undirected, meaning that for every (i, j) connection both (i, j) and (j, i) are included. This is consistent with the symmetric adjacency normalization adopted in standard message-passing GNNs such as GCN [39].

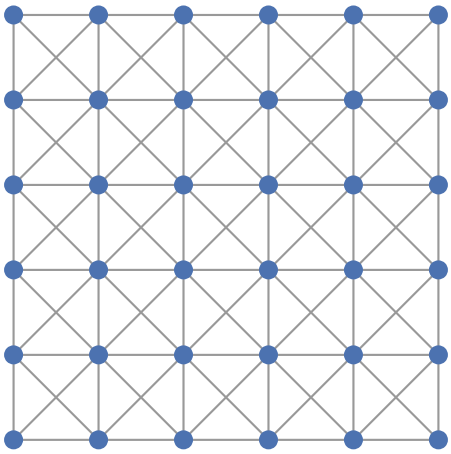


Figure 4.10: Eight-connected sensor grid used in the homogeneous graph construction. Each sensor corresponds to a node (blue dots) and is linked to its four axial and four diagonal neighbours. This connectivity reduces anisotropy and allows diagonal relations to be represented in a single message-passing step.

Edge attributes

Each edge is associated with a set of geometric attributes that describe the relation between source and target sensors. These features are derived directly from raw sensor coordinates and node-level descriptors, and include:

- Euclidean distance $\|d\|$, squared distance $\|d\|^2$, and normalised distance $\|d\|/\mathrm{diag}$, where $\mathrm{diag} = \sqrt{w^2 + l^2}$ denotes the room diagonal based on width w and depth l[1].
- Directional components, expressed as raw coordinate differences $(\Delta x, \Delta y)$ and as normalised direction vectors $(\Delta x, \Delta y)/\|d\|$.
- Differences in window-related sensor features: solid angle $(\Delta\mathrm{SA})$, distance to window $(\Delta\mathrm{DW})$, absolute distance difference $(|\Delta\mathrm{DW}|)$, and relative angular alignment $\cos(\Delta\phi)$, where ϕ denotes the angle of a sensor relative to the window normal.

All attributes are computed from unscaled, physically meaningful quantities taken directly from the simulation data. Only those terms defined as normalised by construction (e.g., $\|d\|/\mathrm{diag}$) are rescaled. The full set of formulas is listed in Appendix C.3.1.

4.3.2. Heterogeneous WindowGraphNet

The heterogeneous formulation extends the homogeneous graph by explicitly representing the window geometry in addition to the sensor grid. Three node types are introduced: four window corners, a single window centre, and the set of sensor nodes. Unlike the homogeneous model, sensor nodes are kept feature-light, with geometry instead encoded through edges. This mirrors the principle of multi-relational message passing in heterogeneous GNNs, where different edge types govern distinct interaction patterns [117]. By separating aperture nodes from receiver nodes, the model provides a more physically meaningful representation of how daylight enters through the window and propagates to the sensors.

Aperture node representation

Daylight enters a space through an aperture that acts as a single luminous source, but its contribution is shaped by the geometry of the window frame and its orientation in space. To capture this physical

[1] Unless otherwise specified, rooms are square such that $l = w$.

distinction, the heterogeneous graph represents the window using five aperture nodes: four corners encoding local frame geometry and one centre node capturing global aperture context. Corners provide directional cues associated with each boundary vertex, including occlusion patterns and variations in angular incidence that sensors experience across the room. The centre node, in turn, approximates the dominant optical behaviour of the aperture as a whole, acting as a compact representation of its position, scale, and effective normal direction. This decomposition mirrors analytical DF formulations, in which the window is treated as a single emitter while edge effects and visibility losses are governed by the frame geometry. By separating local boundary effects (corners) from global aperture context (centre), the heterogeneous formulation enables the model to learn how discontinuities at the window frame influence light distribution without overloading any single node with conflicting geometric roles. This modelling choice improves architectural fidelity while maintaining a lightweight graph structure suitable for fast message passing.

Node features

These aperture roles emerge entirely from graph relations, as no geometric information is encoded directly in node features. Instead, node features are kept intentionally minimal: a single binary indicator identifies window corners (1) versus non-corners (0). Centre and sensor nodes therefore share the same node attribute, but they remain distinguishable through their relational context. In particular, window nodes (centre and corners) send directed visibility- and orientation-based messages to sensors, whereas sensor nodes connect only to neighbouring sensors within the measurement grid. This avoids leaking semantic labels into node features and forces the model to infer functional distinctions from structure and geometry, ensuring that physically meaningful information is conveyed through the edge relations.

Edge topology

These relational distinctions are encoded explicitly in the graph structure. Four edge types are instantiated (Figure 4.11), each representing a different physical interaction:

- *Corner* $\rightarrow$ *centre* (dark red): encodes the span and orientation of the window by linking each corner to the aperture midpoint.
- *Centre* $\rightarrow$ *sensor* (red): propagates aggregate aperture information to all measurement points.
- *Corner* $\rightarrow$ *sensor* (red): conveys fine-grained directional information from each corner to the sensors.
- *Sensor* $\leftrightarrow$ *sensor* (pink): connects neighbouring sensors with bidirectional edges, enforcing local spatial smoothing as in the homogeneous model.

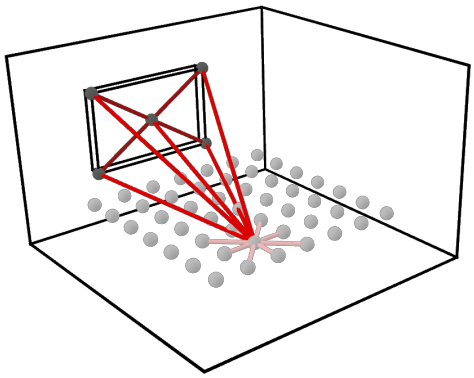


Figure 4.11: Heterogeneous graph construction with sensor nodes (grey) and aperture nodes (dark grey). Edges from corners to centre encode the size of the window (dark red). Edges from the aperture to the sensors encode directional daylight transport (red), while sensor–sensor edges maintain local spatial smoothing (pink).

Edge attributes

In contrast to the homogeneous formulation, where part of the geometric information is embedded as node features, the heterogeneous graph intentionally shifts this information into the edges. Each edge

is enriched with attributes derived from the relative positions of source and target nodes, ensuring that geometric and photometric relations are encoded in a physically meaningful way. This reflects the intuition that daylight transport from apertures to receivers is directional, and parallels geometry-aware message-passing models such as DimeNet [118], DimeNet++ [119], and equivariant architectures such as EGNN [120], which emphasise the role of relative geometry in capturing physical interactions. The edge features include:

- *Distance terms:* identical to those used in the homogeneous model, namely Euclidean distance $\|d\|$, squared distance $\|d\|^2$, and normalised distance relative to the room diagonal.
- *Scaled displacements:* component-wise displacements $(\Delta x, \Delta y, \Delta z)$ rescaled by the room dimensions (width, depth, height) to account for differences in geometry.
- *Directional cosines:* squared components of the normalised displacement vector $(\cos^2 \alpha_x, \cos^2 \alpha_y, \cos^2 \alpha_z)$.
- *Local-frame projections:* dot-products of the displacement vector with normal window n, global up vector u, and tangent vector $t = u \times n$, encoding orientation relative to the aperture.

The exact formulas are provided in the Appendix C.3.2.

4.3.3. Graph statistics

The two graph constructions lead to distinct graph characteristics. In the homogeneous case, each room produces a graph whose nodes correspond to sensors only, with edges following an eight-connected undirected grid. The number of nodes therefore equals the number of sensors per room, while the number of edges grows approximately as $16N$ for N sensors, with boundary effects reducing this slightly due to fewer neighbours along the room perimeter. [2]

In the heterogeneous graph, aperture–sensor relations are modelled as directed edges (corner→sensor, centre→sensor, corner→centre), whereas sensor–sensor edges are undirected via the duplicated eight-neighbour grid. This construction leads to an edge count that is higher than in the homogeneous case, consistent with the empirical averages reported in Table 4.5. One-way aperture edges are retained to reflect directional transport from the aperture to receivers and to avoid unnecessary parameter duplication.

Table 4.4 summarises these structural differences schematically, while Table 4.5 reports the realised averages across all graphs in the dataset. On average, heterogeneous graphs contain about five additional nodes but more than 400 additional edges compared to their homogeneous counterparts, highlighting the cost of explicitly representing window geometry. Extended statistics for all dataset splits, including minimum and maximum values, are provided in Appendix C.3.3.

Table 4.4: Average graph statistics for homogeneous and heterogeneous WindowGraphNet constructions.

Graph type	Avg. nodes	Avg. edges	Node types
Homogeneous	N_{sensors}	$\approx 16N$	Sensors only
Heterogeneous	$N_{\text{sensors}} + 5$	$\approx 16N + 5N + 4$	Sensors, 4 corners, 1 centre

Table 4.5: Average number of nodes and edges per graph across the full dataset, comparing homogeneous and heterogeneous constructions.

Graph type	Avg. nodes	Avg. edges
Homogeneous	82.2	1118.8
Heterogeneous	87.2	1533.9

In summary, the homogeneous construction emphasises sensor–sensor couplings to capture diffuse spatial smoothing, whereas the heterogeneous variant augments this representation with explicit aperture–sensor relations that encode directional geometry. Together, these two formulations provide complementary perspectives on daylight propagation. The full graph models, including all node and edge

[2] Since edges are stored in both directions, the edge index contains $2 \times 8N$ entries, corresponding to $16N$ directed edges for N sensors (before boundary corrections).

definitions as well as their masks, serve as the foundation for the subsequent feature-ablation and model-optimisation studies. Throughout this thesis, the graph structures themselves remain fixed; only the feature inputs and the neural architectures used to map graphs to outputs are varied.

4.4. Training and Evaluation Protocol

Before introducing the feature ablation study and model optimisation, the training and evaluation protocol is described in detail. As noted before, the edge features are derived from raw geometric quantities (x, y, window coordinates, width, length, and height). To ensure that all features contribute on a comparable scale, thereby improving convergence stability and avoiding dominance of variables with larger numerical ranges, both node and edge features are standardised. Standardisation is performed using z-score normalisation:

$$z = \frac{x - \mu}{\sigma}, \tag{4.2}$$

where μ and σ denote the mean and standard deviation computed exclusively on the training set. This procedure prevents data leakage by ensuring that no statistical information from the validation, feature ablation, model optimisation, or test sets influences the training process. The same μ and σ values determined from the training data are subsequently applied to all other sets. Target values (DF) are standardised where explicitly indicated, and inverse-transformed prior to evaluation.

Loss functions and evaluation metrics

All models are trained using mean squared error (*MSE*) as the loss function, consistent with prior ANN-based daylight surrogates [17–19]. MSE penalises large deviations quadratically, which is desirable in this context as significant errors at individual sensors can strongly affect room-level assessments.

For evaluation, three complementary metrics are computed: root mean squared error (*RMSE*), maximum absolute error (*MaxAbs*), and the structural similarity index (*SSIM*). *RMSE* is directly comparable to MSE but expressed in the same units as the DF, making it easier to interpret error magnitudes in physical terms. It is widely reported in daylight surrogate studies [18, 19] and remains sensitive to large deviations, thereby providing a stricter criterion than MSE alone.

MaxAbs highlights the worst-case prediction error across all sensors. This is particularly relevant for daylight analysis, where large local errors (for example in deep room zones) could compromise design decisions despite low average errors elsewhere. The inclusion of maximum-error metrics is consistent with best practice in performance-driven building modelling [121], where robustness against outliers is treated as a key reliability criterion.

Finally, *SSIM* measures the spatial similarity between predicted and ground-truth DF maps. This metric goes beyond pointwise error by assessing whether the model preserves the structural patterns of daylight distribution, an aspect that is critical for spatial design feedback.

Training regime

All models are trained using the Adam optimiser with a learning rate of 10^{-3} [122] and a dropout rate of 0.2 applied to hidden layers, matching robust defaults commonly used for regularisation in deep learning [123]. Unless otherwise specified, these hyperparameters are used consistently across all experiments. Models are trained for a maximum of 200 epochs, with early stopping applied if the validation MSE does not improve for 20 consecutive epochs. This regime prevents overfitting while ensuring that the final model corresponds to the epoch with the best validation performance.

Empirical studies across domains have shown that batch sizes in the range of 16–64, and particularly 32, frequently yield strong performance without requiring extensive hyperparameter tuning [124–126]. For this reason, a batch size of 32 has been widely adopted in deep learning practice and is used here for all experiments.

Optimisation settings and reproducibility

For BO experiments, features and hyperparameters are tuned using Optuna [127]. To balance reproducibility with adequate exploration of the hyperparameter search space, multiple random seeds are employed. Specifically, five independent Optuna studies are conducted for each optimisation task,

each initialised with a different global seed. Within a given study, the random seed for each trial is deterministically assigned as $1000 + t$, where t denotes the trial index. This scheme ensures that all five studies explore distinct regions of the search space while maintaining full reproducibility of individual trials.

Outside of BO experiments, global seeds are fixed to 42 for NumPy and PyTorch. All experiments are conducted with the dataset splits defined in Section 4.1.2.

Hardware

For standard training runs, models are executed on a local GPU workstation. For the BO experiments in feature selection and model optimisation, an NVIDIA A100 GPU is employed on Google Colab. This enabled parallel evaluation of candidate configurations at scale within the Optuna framework. Implementation relied on PyTorch and PyTorch Geometric for model training, Optuna for hyperparameter optimisation, and Radiance/Grasshopper for dataset generation. Further implementation details, including hardware specifications, are reported in Appendix C.5.

4.5. Feature Ablation

This section investigates which input descriptors most effectively predict DF while remaining robust under geometric transformations. To isolate the effect of feature representation from operator choice, a single edge-aware GNN backbone is fixed, and only the inclusion or exclusion of candidate node and edge attributes is varied through binary masking.

The analysis proceeds in two stages. First, information-theoretic screening is performed on the *Transformation Dataset*, informed by the behaviour of the three ANN benchmark feature sets. Before applying these criteria, four ANN surrogates are trained on the complete feature set to establish which descriptors carried predictive value: (i) the raw geometric inputs, (ii) the Le-Thanh feature set, (iii) the Dieguez feature set. Each model is evaluated on the held-out transformation data. Mutual information and redundancy scores are then computed on this *Transformation Dataset*, following an mRMR-like criterion to quantify descriptor relevance and redundancy.

Second, BO over feature masks is conducted to evaluate how different combinations of the remaining descriptors perform within graph architectures. Separate searches are carried out for homogeneous and heterogeneous graph formulations, treating node- and edge-level attributes on equal footing.

Performance is consistently evaluated on the *Transformation Dataset* using *RMSE*, *MaxAbs*, and *SSIM* . The outcome is a set of compact, high-utility feature masks that form the refined input spaces for the subsequent operator optimisation experiments.

4.5.1. Feature Preselection and Initial Screening

A systematic feature ablation study is conducted to identify the input descriptors that most effectively capture daylight behaviour. Table 4.6 summarises the complete set of node- and edge-level features considered. The analysis begins with a preliminary screening phase to identify descriptors with demonstrated predictive value. Four benchmark ANN surrogates are first evaluated on the *Transformation Dataset*, providing an empirical baseline of descriptor quality. The most informative features from this stage are then examined using information-theoretic criteria to quantify their relevance and redundancy. This two-step process ensures that the subsequent GNN experiments focus on physically meaningful and non-redundant features, reducing the dimensionality of the search space for BO.

Table 4.6: Complete set of candidate features considered for feature ablation.

Feature	Node/Edge	Homo/Hetero
x, y	Node (sensor coordinates)	Homo
Width	Node (global)	Homo
WWR	Node (global)	Homo
$x_{\text{thanh}}, d_{\text{thanh}}, w_{\text{thanh}}$	Node (sensor descriptor)	Homo
Solid Angle	Node (sensor descriptor)	Homo
Aspect Ratio	Node (global)	Homo
Solid Angle Tilt	Node (sensor descriptor)	Homo
Room Avg. Solid Angle	Node (sensor descriptor)	Homo
Distance to Window	Node (sensor descriptor)	Homo
Angle rel. to Window Normal	Node (sensor descriptor)	Homo
Window Head Height	Node (global)	Homo
Euclidean distance $\|d\|$	Edge	Homo / Hetero
Squared distance $\|d\|^2$	Edge	Homo / Hetero
Normalised distance $\|d\|/d_{\text{diag}}$	Edge	Homo / Hetero
Coordinate differences $(\Delta x, \Delta y)$	Edge	Homo
Normalised direction vectors $(\Delta x, \Delta y)/\|d\|$	Edge	Homo
Solid angle difference ΔSA	Edge	Homo
Distance-to-window difference ΔDW	Edge	Homo
Absolute Distance-to-window difference ΔDW	Edge	Homo
Relative angular alignment $\cos(\Delta\phi)$	Edge	Homo
Scaled 3D distance	Edge	Hetero
Scaled horizontal distance	Edge	Hetero
Scaled vertical separation	Edge	Hetero
Directional cosines $(\cos^2\theta_x, \cos^2\theta_y, \cos^2\theta_z)$	Edge	Hetero
Dot product with window normal $(d \cdot n)$	Edge	Hetero
Dot product with global up vector $(d \cdot u)$	Edge	Hetero
Dot product with tangent vector $(d \cdot t)$	Edge	Hetero

Benchmark evaluation of ANN surrogates. In the first stage, four benchmark ANN models are trained exclusively on the training set and evaluated on the held-out *Transformation Dataset*. This evaluation serves two complementary purposes: (i) to assess whether individual feature encodings generalise across geometric transformations, and (ii) to filter out descriptors or feature combinations that provide little predictive benefit or exhibit unstable behaviour. The resulting performance trends act purely as an empirical screening signal and do not involve any probabilistic modelling.

Mutual Information, Redundancy, and mRMR-like Scoring

In the second stage, feature relevance is assessed using information-theoretic measures computed directly from the *Transformation Dataset*. Mutual information (MI) and minimum-redundancy maximum-relevance (mRMR) principles are used to quantify each feature's unique and non-overlapping contribution to daylight prediction.

Mutual information (MI). The dependency between a feature f and the target DF y is measured by their mutual information:

$$I(f;y) = \sum_{f\in\mathcal{F}} \sum_{y\in\mathcal{Y}} p(f,y) \log \frac{p(f,y)}{p(f)\,p(y)}, \tag{4.3}$$

where $p(f,y)$ is the joint probability distribution of f and y, and $p(f)$ and $p(y)$ are their respective marginals. In practice, the joint distribution $p(f,y)$ is not modelled parametrically. Instead, mutual information is estimated non-parametrically using a k-nearest-neighbour approach [128], which supports continuous dependencies and avoids assumptions on distributional form.[3] A value of zero indicates independence, whereas higher values signify stronger statistical dependency.

Redundancy. To account for information overlap among features, redundancy is defined as the average mutual information between a given feature f and all other descriptors in the feature set:

$$R(f) = \frac{1}{|\mathcal{X}| - 1} \sum_{f'\neq f} I(f;f'). \tag{4.4}$$

Here, $\mathcal{X}$ denotes the complete set of input features. Thus, for each feature f, the redundancy score measures how much information it shares with the remaining descriptors. A high redundancy score

[3] ANN predictions are not used for MI estimation; the computation is performed directly on the raw feature–DF data.

indicates that the information carried by f is largely duplicated by other features, whereas a low score suggests that it provides unique information not captured elsewhere.

mRMR-like score. Relevance and redundancy are combined into a simplified mRMR-like criterion [129]:

$$\text{mRMR-like}(f) = I(f; y) - R(f). \tag{4.5}$$

A positive value indicates that f contributes unique predictive information about daylight behaviour, whereas a negative value suggests limited added value.

Together, the empirical benchmark evaluation and the information-theoretic scoring identify a concise and physically grounded set of descriptors for the subsequent BO of feature combinations within the graph-based models.

4.5.2. Bayesian Optimisation for Feature Masking

Following the initial screening phase, which establishes a reduced set of informative and non-redundant descriptors, the next step investigates how these features perform in combination within graph-based architectures. To this end, a series of BO searches is conducted to identify feature subsets that jointly maximise predictive accuracy and robustness across transformations.

Each feature subset is represented as a binary mask, where inclusion or exclusion of individual node and edge attributes defines a unique configuration. The search procedure systematically explores these binary spaces while keeping the model architecture and training setup fixed, ensuring that performance differences can be attributed solely to feature selection.

All experiments employ the ResGatedConv operator as a stable and edge-aware backbone. The following subsections describe the experimental strategy, architectural configuration, and optimisation setup in detail.

Experimental Strategy

To ensure that the BO isolates the effect of feature representation, the model architecture and learning operator are held constant throughout all runs. In graph neural networks, performance depends jointly on both the operator design and the chosen feature set; varying them simultaneously would make attribution ambiguous. A fixed, expressive, and edge-aware operator is therefore required to provide a fair and consistent test environment.

Among the available message-passing operators, the ResGatedConv [99] is selected as the backbone. Simpler alternatives such as the GCN introduced in Section 2.3.5 are excluded because they cannot incorporate edge features and have limited expressive capacity. Edge-conditioned variants such as ECC [91] or NNConv [90] are likewise avoided, as their parameter count scales with the dimensionality of the edge features, which would make feature ablations inherently unfair. Attention-based operators [63] are also omitted because neighbour competition through a softmax can obscure weak or noisy feature contributions, complicating attribution.

ResGatedConv offers a transparent and well-characterised alternative. It integrates multi-dimensional edge attributes through a sigmoid gating mechanism while maintaining a consistent parameterisation across all feature-mask configurations. Although the dimensionality of the input projections formally depends on both the node and edge feature widths $(d + e)$, the total number of trainable parameters remains constant in this study because the latent layer dimension is fixed and masked features are handled within that predefined width. This ensures that any observed performance differences arise from the informativeness of the features rather than variations in model capacity. Formally, the node update at layer l is defined as

$$\mathbf{h}_i^{(l+1)} = \mathbf{W}_{\text{skip}}^{(l)} \mathbf{h}_i^{(l)} + \bigoplus_{j \in \mathcal{N}(i)} \sigma\Big(\mathbf{W}_k^{(l)}[\mathbf{h}_i^{(l)} \,\|\, \mathbf{e}_{ij}^{(l)}] + \mathbf{W}_q^{(l)}[\mathbf{h}_j^{(l)} \,\|\, \mathbf{e}_{ij}^{(l)}]\Big) \odot \mathbf{W}_v^{(l)}[\mathbf{h}_j^{(l)} \,\|\, \mathbf{e}_{ij}^{(l)}], \tag{4.6}$$

where $\bigoplus_{j \in \mathcal{N}(i)}$ denotes the aggregation over neighbouring nodes, defined here as a simple summation:

$$\bigoplus_{j \in \mathcal{N}(i)} (\cdot) \equiv \sum_{j \in \mathcal{N}(i)} (\cdot)$$

Summation provides a permutation-invariant aggregation that treats all neighbours equally and ensures stable gradient propagation across varying neighbourhood sizes. It represents the most widely used aggregation scheme in message-passing networks, adopted in canonical formulations such as GCN [39] and GraphSAGE [48], and serves as a consistent baseline for comparative analysis. Here, σ denotes a sigmoid activation, $\|$ indicates concatenation, and $\odot$ the Hadamard product. A schematic of the operator can be found in Figure 4.12. The term $\mathbf{W}_{\text{skip}}^{(l)}\mathbf{h}_i^{(l)}$ represents the residual (skip) connection. While residual connections are valuable for stabilising deeper architectures, they can allow the model to rely excessively on self-features and thus obscure the effect of relational information. To ensure that predictive performance in this study reflects the contribution of node and edge features alone, the residual term is disabled during the BO runs. Nevertheless, residual connections remain an integral aspect of the ResGatedConv operator family and are employed in later model variants used for operator optimisation and evaluation.

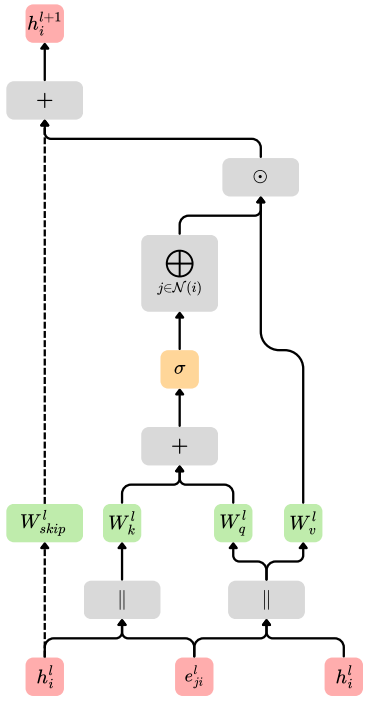


Figure 4.12: Message passing in the ResGatedConv layer. Node features from the source and target, together with edge attributes, are projected into key, query, and value terms. Their interaction produces a sigmoid gate that modulates the incoming message. The dashed skip connection indicates the optional residual term, which is disabled during ablation experiments to ensure that predictive performance depends fully on relational features.

This operator is chosen because it meets three key requirements for the present study:

1. *Edge sensitivity:* its channel-wise gating integrates edge attributes directly into the message computation without normalisation across neighbours;
2. *Residual flexibility:* skip connections can be selectively enabled or disabled to control the balance between self- and neighbour-dependence; and
3. *Parameter invariance:* its weight matrices map from $(d + e) \to d$, keeping the parameter count fixed even when edge dimensions vary (see Appendix B).

Together, these properties make ResGatedConv a balanced backbone for assessing the influence of node and edge feature selection. By fixing the operator, any observed variation in performance can be attributed directly to differences in the feature masks rather than architectural complexity.

Architectural configuration. Two ResGatedCo-based architectures are implemented, corresponding to the homogeneous and heterogeneous graph formulations introduced earlier.

- *Homogeneous architecture:* The homogeneous network comprised three ResGatedGraphConv

layers with 64 hidden units, followed by a linear readout. Residual connections are disabled during BO to ensure that all predictive information flows through the masked features and edges. This configuration provides a controlled balance between expressiveness and computational efficiency across multiple optimisation trials.

- *Heterogeneous architecture:* The heterogeneous WindowGraphNet variant instantiates separate ResGatedGraphConv modules for each relation type, corner→centre, corner→sensor, centre→sensor, and sensor↔sensor. Node embeddings for the three node types (corner, centre, sensor) are projected into a shared latent space, after which relation-specific convolutions are applied in a single message-passing step. A shallow MLP with 32 hidden units then maps the resulting sensor embeddings to DF predictions. Hidden dimensionality is fixed at 64 for all GNN layers.

In summary, this setup establishes a consistent and interpretable foundation for the BO study. Both architectures employ a dropout rate of 0.2 in their MLP or linear layers, while the ResGatedGraphConv operator itself does not include an internal dropout mechanism, and are trained using an initial learning rate of 10^{-3}.

Masking strategy. With the architectural setup fixed, the BO procedure is configured to explore the binary feature-mask space under controlled and repeatable conditions. Three full BO runs are carried out: two for homogeneous graphs and one for heterogeneous graphs.

- *Homogeneous stage 1:* A broad exploratory search over the full feature space. The purpose of this stage is not to identify the best mask, but to detect and prune consistently underperforming descriptors. Evaluation at this stage is performed at the level of individual features rather than masks.
- *Homogeneous stage 2:* A refined search over the reduced feature space, aimed at selecting robust feature masks. Both mask-level and feature-level statistics are retained for further analysis.
- *Heterogeneous stage:* A single BO run is sufficient due to the more compact feature space. The same edge feature mask is applied across all relation types, and the run is directly used to select candidate masks.

Having defined the masking stages, the subsequent optimisation procedure is designed to ensure consistent sampling, reproducibility, and comparability across all runs, as outlined below.

Optimisation strategy. All BO runs share the same setup to ensure comparability. Each optimisation is repeated across five independent studies with different random seeds $(40, 41, 42, 43, 44)$, to reduce variance and avoid artefacts from a single trajectory. Within each study, every trial is trained with a unique global seed affecting weight initialisation, data shuffling, and dropout masks. Each study comprises 200 trials, including 30 start-up trials drawn randomly to initialise the search space. The number of 200 trials per study is chosen as a balance between coverage of the high-dimensional mask space and computational feasibility, consistent with prior studies on discrete architecture search [130, 131].

The `TPESampler` is selected as acquisition strategy, as it provides a balance between exploration and exploitation and is particularly effective in high-dimensional, discrete search spaces such as binary feature masks [132]. Multivariate sampling is enabled to allow the sampler to capture dependencies between features, rather than treating each inclusion/exclusion decision independently [133].

Objectives. BO is conducted as a multi-objective optimisation with three metrics: *RMSE*, *MaxAbs*, and *SSIM*, all evaluated on validation and transformation sensor-level outputs. *RMSE* captures average predictive accuracy, *MaxAbs* penalises worst-case deviations, and SSIM reflects structural similarity of spatial daylight distributions. These objectives jointly balance overall accuracy, robustness, and fidelity of spatial patterns. These metrics are treated as simultaneous objectives within Optuna's multi-objective mode, producing a Pareto front of non-dominated feature masks rather than a single scalar optimum.

Evaluation of feature subsets

After completing the BO runs, the resulting set of 1000 evaluated trials (5 studies × 200 trials each) is subjected to post-optimisation analysis to identify robust and informative feature subsets. Because the optimisation problem involves multiple, and partly conflicting, objectives, these trials cannot be ranked by a single metric. Instead, a Pareto front is constructed to identify the subset of masks that represent optimal trade-offs across objectives. Subsequent analyses, including re-evaluation, frequency statistics, and contribution analysis, are then performed on this Pareto-optimal subset, providing a robust basis for feature selection.

Pareto front selection In multi-objective optimisation, no single criterion suffices to rank model configurations. A Pareto front selection strategy is therefore applied [134, 135]. A mask m is Pareto-optimal if there does not exist another mask m' that performs at least as well in all objectives and strictly better in at least one. Formally, the Pareto set $\mathcal{P}$ is defined as

$$\mathcal{P} = \left\{ m \in \mathcal{M} \;\middle|\; \nexists m' \in \mathcal{M} : \Big(f_j(m') \leq f_j(m) \;\; \forall j \in J, \quad f_k(m') < f_k(m) \;\; \text{for some } k \in J \Big) \right\}, \tag{4.7}$$

where $\mathcal{M}$ denotes the set of all evaluated masks, and f_j are the objective functions corresponding to the chosen performance metrics. The index set $\mathcal{J}$ enumerates these objectives, so that $j \in \mathcal{J}$ and $k \in \mathcal{J}$ refer to individual metrics (e.g. *RMSE, MAE, SSIM*). The resulting set $\mathcal{P}$ consists of the non-dominated solutions, i.e. those for which no strictly better alternative exists across all metrics.

The Pareto front captures the trade-off surface between objectives and reduces the set of candidates to those representing efficient compromises. All subsequent analyses are therefore conducted on $\mathcal{P}$.

In practice, Pareto front extraction is performed on the results of the Optuna studies, where each trial evaluates a different feature mask. Optuna explores the mask space using a stochastic sampler, and the number of trials that happen to fall near the trade-off surface varies naturally across studies. As a consequence, the size of the resulting Pareto set $\mathcal{P}$ is not fixed: some studies yield only a small number of non-dominated masks, whereas others produce larger fronts when several masks achieve mutually incomparable performance across the objectives.

This variability is expected. The Pareto set reflects the structure of the objective landscape induced by the masks themselves, rather than a predetermined selection budget. A small $\mathcal{P}$ indicates a clear separation between strong and weak masks, while a larger $\mathcal{P}$ arises when several masks offer different but non-dominated performance profiles. All subsequent feature analyses therefore operate on the study-specific Pareto sets returned by Optuna, each of which captures the efficient trade-offs present in that particular mask evaluation study.

Mask scoring and selection The Pareto set $\mathcal{P}$ can contain a large number of candidate masks, many of which differ only marginally in their feature composition or performance. In the first homogeneous phase, all Pareto-optimal masks are retained for re-evaluation to obtain a complete view of feature contributions. For the subsequent homogeneous and the heterogeneous phase, however, a more selective strategy is adopted.

Although Pareto front selection defines a set of non-dominated solutions, ranking within this set still requires a scalar criterion to facilitate re-evaluation and comparison across runs. To rank Pareto-optimal masks, each trial is assigned a composite score based on three metrics: *RMSE*, *MaxAbs*, and *SSIM*. Since these metrics are not directly comparable, they are first normalised to $[0, 1]$ using min–max scaling over all trials,

$$\mathrm{norm}(x_t) = \begin{cases} \dfrac{x_t - x_{\min}}{x_{\max} - x_{\min}}, & \text{if } x_{\max} > x_{\min}, \\ 0.5, & \text{otherwise}, \end{cases} \tag{4.8}$$

where x denotes either *RMSE* or *MaxAbs*. When all values coincide, a neutral score of 0.5 is assigned to prevent a degenerate metric from dominating the composite. Since higher *SSIM* indicates better structural agreement, this metric is inverted into a loss-like term,

$$L_{\mathrm{SSIM},t} = 1 - \mathrm{SSIM}_t, \tag{4.9}$$

so that lower values are preferable for all objectives.

The overall composite score is then defined as a weighted sum,

$$S_t = 0.55 \cdot \mathrm{norm}(\mathrm{RMSE}_t) + 0.30 \cdot \mathrm{norm}(\mathrm{MaxAbs}_t) + 0.15 \cdot L_{\mathrm{SSIM},t} \tag{4.10}$$

The weights reflect the relative importance of the evaluation criteria in the context of DF prediction. Accuracy, as measured by *RMSE*, is prioritised (55%) because the primary objective of the surrogate model is to reproduce simulation outputs with minimal average error across sensor points. Outlier sensitivity, captured by *MaxAbs*, is assigned a substantial but lower weight (30%), since extreme deviations at individual sensors can still affect design decisions and must therefore be penalised, but they are secondary to overall accuracy. Structural fidelity, expressed through *SSIM*, receives the remaining weight (15%), as it provides complementary information about the spatial consistency of predictions. Although important, SSIM is treated as an auxiliary criterion because it is less critical than global accuracy or robustness to local extremes. For completeness, an equal-weighted variant is also reported to verify that results are not strongly dependent on the specific choice of weights.

$$S^{\mathrm{even}} = \tfrac{1}{3}\,\mathrm{norm}(\mathrm{RMSE}_t) + \tfrac{1}{3}\,\mathrm{norm}(\mathrm{MaxAbs}_t) + \tfrac{1}{3}\,L_{\mathrm{SSIM},t}. \tag{4.11}$$

Within each group of trials corresponding to the same mask, the representative trial is defined as the one with the lowest composite score S_t. In the homogeneous phase 1, all Pareto-optimal masks are retained. In phase 2 and the heterogeneous phase, the nine masks with the lowest S_t values are shortlisted for re-evaluation. This approach ensures that not only individual features but also the overall quality of each mask is taken into account when deciding which configurations merit further analysis.

Re-evaluation of Pareto masks To assess the stability of Pareto-optimal masks and obtain unbiased performance estimates, the selected solutions (or a representative subset) are retrained. Each mask is re-evaluated under the same five random seeds $(40, 41, 42, 43, 44)$ used in the original optimisation runs. Reusing a consistent seed pool controls for stochastic factors such as model weight initialisation, data shuffling, and dropout, ensuring that performance differences arise from the feature masks themselves rather than favourable random effects. By repeating training under controlled random conditions, variance across seeds can be measured and reliable statistics (mean, standard deviation, worst-case performance) can be reported.

This re-evaluation filters out configurations that reached the Pareto front due to stochastic luck and highlights those whose performance is robust across repeated training. Only these re-evaluated solutions are carried forward into the frequency and contribution analyses.

Feature frequency analysis The frequency with which individual features appear in Pareto-optimal masks is quantified. For each feature f_i, the number of Pareto front masks in which the feature is active is counted and normalised by the total number of masks:

$$\mathrm{freq}(f_i) = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \mathbb{1}\{m_i = 1\}, \tag{4.12}$$

where $\mathcal{M}$ is the set of Pareto-optimal masks, $m \in \{0,1\}^d$ is a binary mask, and $\mathbb{1}\{\cdot\}$ is the indicator function. The resulting frequency lies in $[0, 1]$, representing the proportion of Pareto-optimal masks in which a feature is retained.

A high frequency indicates that a feature is consistently part of Pareto-optimal trade-offs, suggesting importance for robust performance. Conversely, a low frequency implies that the feature is rarely needed.

Feature contribution analysis The relative importance of features is further assessed through contribution analysis. The central idea is to compare the performance of models in which a given feature is included with those in which it is absent, based on the Pareto-optimal masks.

Let $\mathcal{M}$ denote the Pareto-optimal set and $m \in \{0,1\}^d$ a binary mask. For feature f_i, define

$$\mathcal{M}_i^+ = \{m \in \mathcal{M} \mid m_i = 1\}, \qquad \mathcal{M}_i^- = \{m \in \mathcal{M} \mid m_i = 0\}. \tag{4.13}$$

For evaluation metric Q, the means are

$$\bar{Q}_i^+ = \frac{1}{|\mathcal{M}_i^+|} \sum_{m \in \mathcal{M}_i^+} Q(m), \qquad \bar{Q}_i^- = \frac{1}{|\mathcal{M}_i^-|} \sum_{m \in \mathcal{M}_i^-} Q(m). \tag{4.14}$$

The contribution is defined as

$$\Delta Q_i = \begin{cases} \bar{Q}_i^- - \bar{Q}_i^+, & \text{for error metrics (\textit{RMSE}, \textit{MaxAbs})}, \\ \bar{Q}_i^+ - \bar{Q}_i^-, & \text{for similarity metrics (\textit{SSIM})}. \end{cases} \tag{4.15}$$

By this convention, $\Delta Q_i > 0$ indicates that including the feature improves performance.

The analysis can be applied jointly across all transformations (overall effect) or separately for each transformation (*Normal*, *Scaled* × 2, *Rotated* 90°, *Rotated* 180°), thereby capturing both global and transformation-specific trends. The frequency analysis, discussed earlier, provides complementary context for interpreting these contributions: whereas contribution quantifies the marginal effect of including or excluding a feature, frequency indicates how consistently that feature is retained in Pareto-optimal solutions. Considering both perspectives together enables a clearer distinction between features that merely improve performance and those that are also reliably selected across efficient configurations.

Mask ranking procedure To enable a systematic and reproducible comparison of candidate masks, a robustness-oriented ranking scheme is defined. Each mask is evaluated under the four considered geometric transformations, and aggregate statistics are computed to capture different aspects of performance:

- *Worst-case performance (minimax).* For each metric (*RMSE*, *MaxAbs*, *SSIM*), the most adverse value across the four transformations is retained. For SSIM, the minimum is reformulated as a loss $(1 - \mathrm{SSIM}_{\min})$, so that higher values consistently indicate poorer performance. This ensures that a mask cannot appear favourable if it fails catastrophically under a single transformation.
- *Tail risk (CVaR-2).* To penalise masks with repeated weak behaviour, the mean of the two largest *RMSE* values is computed. This measure is less sensitive than the strict minimax but still highlights unstable masks.
- *Average behaviour.* Mean *RMSE*, *MaxAbs*, and *SSIM* across transformations, as well as the standard deviation of *RMSE*, are recorded to capture central tendency and stability of predictions.

Each mask is then assigned a rank for these aggregate criteria, and the final *rank score* is defined as the mean of these ranks. In this construction, *RMSE* is deliberately given greater weight: it appears both as a worst-case statistic, a CVaR-2 measure, and an average. This emphasis reflects its role as the primary regression metric in both the training process (which minimises *MSE*) and in daylight modelling practice, where *RMSE* provides a direct and interpretable measure of predictive fidelity in DF units. *MaxAbs* and *SSIM* are included to ensure that extreme outliers and structural similarity are not neglected, but their influence is secondary to *RMSE*.

The resulting composite ranking therefore prioritises masks that are not only accurate on average but also robust across all transformations, while preventing selections that might appear strong in one metric but fail in others.

In summary, the feature ablation framework integrates information-theoretic scoring with BO, Pareto front selection, re-evaluation, and robustness-oriented ranking. This multi-stage evaluation provides a principled basis for identifying subsets of descriptors that are both informative and reliable across transformations. With the selected features established, the subsequent stage of the methodology focuses on optimising and evaluating GNN operators on these refined input spaces.

4.6. Architecture and Operator Design

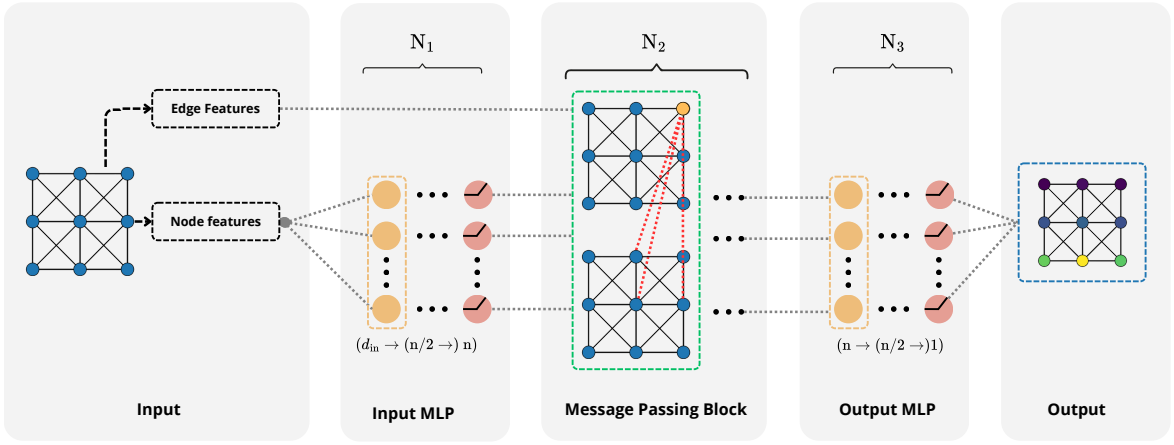
With the most informative node and edge descriptors established from the feature ablation study, the subsequent stage focuses on how these descriptors are embedded and propagated through graph

neural architectures. Following the ablation stage, the set of input features is fixed; no further modifications are introduced. The task now shifts from feature representation to architectural configuration and operator design.

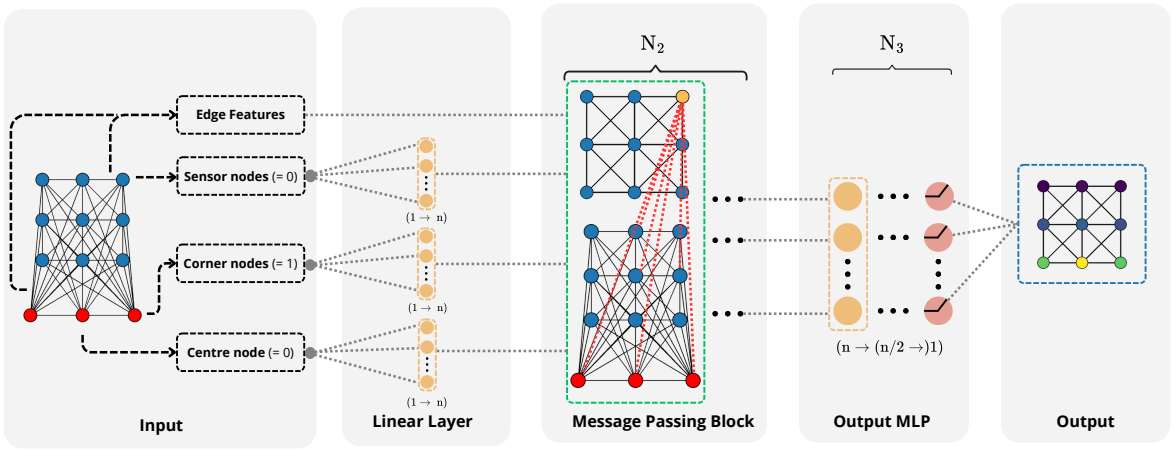
This section specifies a common architecture template shared across all models, narrows the operator set via a multi-criteria analysis, justifies the retained shortlist, and outlines how the shortlisted operators are optimised and compared. The sequence is as follows: model template and shared hyperparameters, operator scoring, operator selection and justification, BO setup, and scoring of the optimised models to nominate a single configuration for final testing.

4.6.1. Model template and shared hyperparameters

With the feature set fixed after ablation, the architecture is specified using a modular template comprising three stages: an input projection MLP (N_1), a stack of N_2 message-passing layers, and an output MLP head (N_3), as summarised in Figure 4.13. The latent width n defines the dimensionality of all hidden node representations: N_1 maps the raw input features into this n-dimensional latent space, each of the N_2 message-passing layers operates within this same latent dimension, and N_3 maps from the latent space to the final output dimension. This arrangement mirrors common GNN practice of (i) embedding raw features, (ii) propagating information across the graph, and (iii) mapping latent representations to task outputs [39, 48, 49, 90].



(a) Homogeneous graph architecture (sensor nodes only)



(b) Heterogeneous graph architecture (window & sensor nodes)

Notation: n — hidden dimension; N_1, N_2, N_3 — number of layers.

Figure 4.13: Homogeneous and heterogeneous model templates used in WindowGraphNet. Both variants follow the same three-block structure with an optional input MLP (N_1), a stack of N_2 operator layers, and an output MLP (N_3). The homogeneous variant applies a shared node encoder prior to message passing, while the heterogeneous variant employs type-specific encoders and omits the shared input MLP.

Input head. Consistent with the encoder–propagator–decoder structure introduced in Section 2.3, the input head specifies the encoder instantiation used in each architectural variant.

In the homogeneous setting, each node is a sensor with a shared descriptor vector. An input MLP with $N_1 \in \{1, 2, 3\}$ projects these descriptors into a common latent space before message passing. Such learned projections improve numerical conditioning and representational power by allowing feature rescaling and interaction in the embedding [51, 63, 136].

The internal layer widths follow the tapering configuration proposed by Wu et al. [86], who employed the same N_1–N_2–N_3 modular structure in graph-based building performance models. This configuration balances expressivity and parameter efficiency, aligning with general design guidelines for progressive channel reduction in deep networks [137, 138]. The first layer corresponds to the input feature dimension d_{in}, the intermediate layers (if present) use half the message-passing hidden size ($n/2$), and the final layer outputs dimension n, matching the operator's latent size. Concretely:

- $N_1 = 1$: $d_{\text{in}} \to n$,
- $N_1 = 2$: $d_{\text{in}} \to n/2 \to n$,
- $N_1 = 3$: $d_{\text{in}} \to n/2 \to n/2 \to n$.

This progressive expansion provides a smooth transition from raw geometric node descriptors to operator-compatible embeddings, while avoiding unnecessary parameter growth. The resulting latent representation aligns the diverse geometric descriptors prior to propagation, consistent with the encoder–propagator–decoder paradigm employed in graph regression networks [14, 90].

For the heterogeneous model, the design of the input head differs. All nodes share the same minimal descriptor: a single binary indicator distinguishing window corners (1) from all other nodes (0). To initialise the latent representations, each node type (sensor, corner, centre) is mapped to the latent dimension n using a type-specific linear embedding layer (Linear$(1 \to n)$), without any additional hidden depth or activation.

This contrasts with the homogeneous model, where sensors possess a richer set of geometric descriptors and are therefore processed by a multi-layer input MLP to enable feature rescaling and interaction before propagation. In the heterogeneous architecture, by contrast, window nodes provide contextual light–boundary information rather than rich attributes, and a lightweight linear projection is sufficient to align all node types to the required latent size.

Unlike the node descriptors, edge attributes are not passed through a shared edge-encoder MLP. The candidate operators incorporate edge information through mechanisms intrinsic to each design: NNConv learns an edge-conditioned linear map via an edge MLP, SplineConv applies continuous B-spline kernels to pseudo-coordinates, PNAConv modulates messages through degree-aware scaling, and attention-based layers (GATv2Conv, GCN+) rely on learned attention or normalisation weights rather than explicit edge transformations. Introducing a universal edge encoder would blur these operator-specific mechanisms and confound the interpretation of performance differences. Edge attributes are therefore provided in their native geometric form, ensuring that the search isolates and fairly evaluates the inherent geometry-handling capabilities of each operator.

Latent width. The latent dimension n controls the width of all hidden node representations and determines the capacity of the model to encode geometric and relational structure. Following common practice in GNN architecture design [39, 48, 51, 139], n is treated as a global channel size shared across the input head, message-passing layers, and output head. It is selected from $n \in \{32, 64, 96, 128\}$, a range widely used in GNNs to balance expressive power with computational efficiency. Smaller widths encourage compact representations and faster training, whereas larger widths increase modelling capacity at the cost of higher memory usage. Fixing a single latent width throughout the message-passing architecture ensures that all components operate in a consistent feature space, supporting stable message aggregation and residual connections across layers.

Message-passing depth. Both variants employ a stack of $N_2 \in \{1, \ldots, 5\}$ operator layers. Depth is limited to this range to capture local and mid-range interactions while avoiding over-smoothing and optimisation degradation that appear at larger depths [59, 90, 140].

To mitigate these effects and preserve gradient flow across stacked layers, each operator block supports an optional residual connection. Residual and skip connections have been shown to stabilise the training of deep GNNs by facilitating identity mapping and improving gradient propagation [58, 136, 138]. Their inclusion allows deeper architectures to maintain discriminative power without suffering from vanishing gradients or feature homogenisation. In the present framework, this functionality is implemented for all operators as an internal option that can be toggled on or off during optimisation. This ensures architectural consistency while allowing the residual pathway to be systematically examined in later benchmarking stages.

Centre-stream scheduling. The heterogeneous architecture exposes a scheduling flag for the centre stream. Centres are always initialised by a type-specific Linear$(1 \to n)$ projection and then updated once by a pre-loop corner$\to$centre convolution. After this pre-update, two regimes are possible. With the flag disabled, the centre embeddings remain fixed during the propagation loop: at each iteration the sensor updates combine messages from corner$\to$sensor, centre$\to$sensor, and sensor$\to$sensor edges, but the centre state is not further modified. With the flag enabled, the centre stream is included in each propagation step via a corner$\to$centre convolution (optionally followed by a small feed-forward block). A separate residual-connection flag controls whether the previous centre state is added back after each update, enabling iterative refinement; when disabled, repeated centre updates reduce to a constant transformation and have no further effect. As all operators in the search space either include the previous node state by design or can be wrapped with a residual connection when required, the dynamic centre-update mode has a tangible effect on the evolving centre representation whenever the residual flag is active.

The purpose of exposing this flag is to probe whether the daylight prediction task benefits from dynamic aperture context refinement or a static global context snapshot. Allowing iterative centre updates enables the centre embedding to evolve across propagation steps, increasing the expressive capacity of the model by allowing boundary influences to be accumulated rather than encoded only once. This is particularly relevant when the visibility of individual window corners varies across the geometry, as the centre can progressively adapt to which boundary regions predominantly contribute to illumination in different parts of the space, while keeping the centre fixed reduces computation and mitigates oversmoothing. The scheduling choice is therefore treated as an operator-agnostic hyperparameter in the heterogeneous search space.

Output head. Post-propagation, node embeddings are mapped to scalar DF predictions by an output MLP with $N_3 \in \{1, 2, 3, 4\}$ layers. Such heads are standard in regression-oriented GNNs, enabling flexible nonlinear mixing of latent channels prior to prediction [51, 90]. The internal layer configuration mirrors that of the input head but in reverse order, gradually contracting from the operator dimension n towards the scalar output, with intermediate layers sized at $n/2$. This symmetric design ensures balanced information flow between encoding and decoding stages and maintains architectural consistency across the model variants. For the heterogeneous model, the head is applied only to sensor nodes.

Shared hyperparameters and choices. The shared search space (Table 4.7) governs architectural and training stability:

(i) *dropout* and *weight decay* are included as complementary regularisation mechanisms. Dropout stochastically masks hidden activations to reduce overfitting [123] and is sampled in $[0, 0.5]$. Activation dropout is applied after each hidden layer in the input projection MLP (N_1), after every message-passing update in the propagation stack (N_2), and after hidden layers in the prediction head (N_3). Weight decay penalises excessive weight growth and encourages smoother function classes [141], and is sampled on a logarithmic scale to efficiently explore typical L2 magnitudes [142, 143]. Using both together is common practice in graph neural networks, including the original GCN architecture [39], and exposing them as independent hyperparameters enables the search process to determine the most effective pairing for each operator.

(ii) *learning rate* is also sampled on a logarithmic scale to ensure stable optimisation across operators [142, 143].

(iii) *normalisation* controls channel-wise stabilisation of the latent node embeddings. In this work, a global normalisation option is applied uniformly across operators within the MLP blocks used for node encoding (homogeneous) and prediction. The search space includes BatchNorm [144], LayerNorm [145], or disabling normalisation entirely, enabling the optimiser to determine whether normalisation contributes positively for a given operator configuration. Both BatchNorm and LayerNorm have been shown to improve optimisation stability in deep GNNs [14, 140].

(iv) *activations* govern non-linear transformation of node embeddings and are applied in the node-encoder MLP, and in the output-head MLP. The search space includes ReLU [146], GELU [147], and SELU [148], covering standard rectified, smooth probabilistic, and self-normalising behaviours commonly adopted in geometric deep learning [14]. Classical saturating nonlinearities such as sigmoid and tanh are excluded due to their bounded outputs, which cause vanishing gradients and slow convergence in deeper networks [137], and because theoretical studies show that ReLU-type activations preserve the maximal expressive power of message-passing architectures, whereas sigmoid and tanh reduce it [149].

Hidden dimension n is tuned per operator later; its ranges are reported with the operator-specific setup. The complete set of shared hyperparameters and their search ranges is summarised in Table 4.7. These shared parameters define the common architectural backbone for all model variants. With the overall structure specified, the following subsection focuses on the message-passing layers, where the candidate operators are compared and ranked prior to optimisation.

Table 4.7: Shared hyperparameters and search ranges used during BO.

Parameter	Search range / options	Description
N_1	$\{1, 2, 3\}$	Input MLP depth (homogeneous only).
N_2	$\{1, 2, 3, 4, 5\}$	Number of message-passing layers.
n	$\{32, 64, 96, 128\}$	Latent node embedding width.
N_3	$\{1, 2, 3, 4\}$	Output MLP depth.
Centre-stream scheduling	`{True, False}`	Update centre nodes at each message-passing step (heterogeneous only).
Dropout	$[0.0, 0.5]$ (step = 0.05)	Feature dropout probability.
Weight decay	$[10^{-6}, 10^{-2}]$ (log-uniform)	ℓ_2 regularisation factor.
Learning rate	$[10^{-4}, 10^{-2}]$ (log-uniform)	Adam learning rate.
Normalisation	`{batch, layer, none}`	Per-layer normalisation type.
Activation	`{ReLU, GELU, SELU}`	Nonlinear activation.

4.6.2. Comparative Assessment and Selection of Operators

Now that the overall model architecture and the shared hyperparameters have been defined, attention turns to the most critical component of the GNN framework, the message-passing operator. Continuing from the comparative review in Section 3.5.2, this section revisits the candidate operators and establishes the rationale for their selection in the WindowGraphNet benchmark. The discussion is structured in three analytical stages. First, empirical performance trends from established benchmarks are reviewed to contextualise how different operators behave across datasets of varying structure and scale. Second, the main strategies for incorporating edge features are examined, as they strongly determine the model's ability to capture geometric and physical relations. Third, a detailed model complexity analysis is conducted to quantify the computational implications of each operator under the homogeneous and heterogeneous regimes used in this study. Together, these three dimensions, empirical evidence, edge-feature handling, and computational complexity, form the basis for the operator ranking and final selection presented at the end of this section.

Empirical performance trends. Benchmark datasets such as *ZINC*, *MNIST superpixels*, and the *OGB* suite reveal consistent tendencies in operator behaviour. Edge-aware architectures including NNConv, CGConv, and kernel-based variants (SplineConv, GMMConv) consistently outperform simpler formulations on geometrically structured graphs, confirming the value of continuous edge conditioning for tasks governed by spatial relations. Conversely, residual and aggregation-based designs such as GENConv and PNAConv demonstrate strong scalability and depth stability across large, non-geometric datasets, highlighting their suitability for deeper networks and data with weaker edge semantics. These complementary strengths suggest that operators optimised for geometric precision and those optimised for architectural stability should be jointly represented in the subsequent evaluation.

Edge-feature integration strategies. A key differentiator between message-passing formulations lies in how real-valued edge attributes are integrated during aggregation. Four main mechanisms can be distinguished. Direct additive schemes (e.g., GENConv, ResGatedConv) treat edge features as auxiliary signals with minimal parameter cost. Edge-conditioned operators (NNConv, CGConv) generate filter weights from edge features through small MLPs, maximising expressivity at the cost of higher parameter counts. Kernel-based methods (SplineConv, GMMConv) generate filter weights by smoothly interpolating over geometric pseudo-coordinates using a fixed set of basis kernels, embedding inductive spatial priors while preventing parameter growth with the dimensionality of the edge attributes. Finally, indirect integration mechanisms (PNAConv, GCN+, GATv2) incorporate edge information through concatenation or attention, prioritising architectural flexibility over physical interpretability. Together, these categories delineate the design space from which a balanced subset of candidate operators will be drawn.

Model Complexity Analysis

Benchmark comparisons provide valuable empirical context but do not directly capture the computational trade-offs that arise when these operators are applied within a unified architecture. To quantify these trade-offs, the per-layer parameter complexity of each candidate operator is derived analytically and verified against the official PyTorch Geometric implementations. The resulting expressions, summarised in Table 4.8, report both the asymptotic scaling in $\Theta(\cdot)$ notation and the exact parameter count as a function of the latent node embedding width d, edge feature dimension e, and internal hyperparameters such as the layer expansion s, number of aggregators A, and kernel components C. Full derivations and validation examples are provided in Appendix B.

Notation. Standard GNN convention is followed and d denotes the latent node embedding width (i.e. the per-layer channel dimension). This corresponds directly to the parameter n used earlier to denote the optimised latent width selected through BO. For consistency with the broader literature, d is adopted throughout this complexity analysis, with $d \equiv n$ in the implementation.

It is important to highlight that the parameter complexity reported in Table 4.8 depends exclusively on the latent width d. Although the graph representations used in this work differ in how geometric information is allocated to nodes and edges (homogeneous versus heterogeneous configurations), all message-passing layers operate in a latent space of fixed width d and map $\mathbb{R}^d \to \mathbb{R}^d$. Minimal raw node descriptors in the heterogeneous case (e.g. $d_{\text{in}}{=}1$) are lifted to the latent width by a single learned input projection, after which the computational structure is identical to the homogeneous setting. Consequently, the per-layer parameter counts reported here apply equally to both representations.

Across operators, the theoretical parameter counts confirm that complexity is governed primarily by the latent width d. Residual and normalised architectures (e.g. GENConv, GeneralConv, ResGatedConv, GCN+) exhibit the expected $\Theta(d^2 + ed)$ scaling, with the quadratic term typically dominating at practical embedding widths. Operators with additional aggregation or kernel components (e.g. PNAConv, SplineConv, GMMConv) introduce proportional multiplicative factors. Edge-conditioned operators such as NNConv incur a higher cost of $\Theta(d^3 + ed)$ due to their learned edge-dependent transformations.

Overall, operator selection cannot rely solely on asymptotic scaling laws. In data-efficient surrogate modelling settings such as this one, where geometric cues are embedded directly into the graph structure, multiple operators achieve similar computational cost while offering distinct mechanisms for handling geometry and visibility. This motivates the inclusion of both residual general-purpose operators and geometry-specialised operators in the subsequent empirical evaluation.

Table 4.8: Asymptotic complexity $\Theta(\cdot)$ and per-layer parameter count for the considered GNN operators, expressed as a function of latent node embedding width d and edge feature dimension e. Counts refer to a single message-passing layer and are based on the official PyTorch Geometric implementations. Full derivations are provided in Appendix B.

Operator	$\Theta(\cdot)$	Parameter Count
ResGatedConv	$\Theta(d^2 + ed)$	$4d^2 + ed + 4d$
GENConv	$\Theta(d^2 + ed)$	$(ed + d) + d^2(2s + (L-2)s^2) + d((L-1)s + 1)$
GeneralConv	$\Theta(d^2 + ed)$	$(ed + d) + d^2(2s + (L-2)s^2) + d((L-1)s + 1)$
PNAConv	$\Theta(d^2 + ed)$	$d^2(2s + (L-2)s^2 + AS + 1) + d((L-1)s + e + 2)$
NNConv	$\Theta(d^3 + ed)$	$h_e(e + d^2 + 1) + 2d^2$
CGConv	$\Theta(d^2 + ed)$	$5d^2 + 2ed + 3d$
SplineConv	$\Theta(R^e \cdot d^2)$	$(R^e + 1)d^2 + d$
GMMConv	$\Theta(d^2 + Ce)$	$(C + 1)d^2 + 2Ce + d$
GATv2Conv	$\Theta(d^2 + ed)$	$2d^2 + ed + 2d$
GCN+	$\Theta(d^2 + ed)$	$(1 + 2s)d^2 + (s + 1)d + ed$

Notation and scope.

d = latent node embedding width (per-layer output channels)

e = edge feature width

s = MLP expansion factor inside an operator block

L = number of linear sub-layers inside the operator block (not network depth)

A = number of aggregators and S = number of degree scalers (PNA)

R = B -spline control points per pseudo-coordinate (SplineConv)

C = number of Gaussian components (GMMConv)

h_e = hidden width of the edge-conditioned MLP in NNConv; in this work h_e is chosen proportional to d (i.e. $h_e = \mathcal{O}(d)$), which yields the cubic dependence on d.

H = number of attention heads (GATv2)

For a fixed d, the parameter count of multi-head attention does not scale with H in PyG because heads are concatenated to a total width of d. FLOPs and activation memory are not reported here, as they additionally depend on average node degree and batch size.

Operator ranking

To consolidate the preceding analyses, the candidate message-passing operators are ranked using a qualitative decision matrix that integrates evidence from benchmark performance trends, edge-feature handling strategies, and theoretical complexity. This ranking serves as the final step in the operator-selection phase, identifying a concise and balanced subset for subsequent empirical evaluation. Ratings follow a three-level ordinal scale ('+' favourable, '±' mixed/neutral, '–' unfavourable) and aggregate across the following criteria:

(i) evidence of performance on tasks with continuous geometric edge features and on broader benchmarks;
(ii) parameter complexity;
(iii) inductive bias towards explicit geometric conditioning;
(iv) hyperparameter sensitivity;
(v) ease of implementation (data preparation and pitfalls);
(vi) PyTorch Geometric compatibility; and
(vii) interpretability.

All ratings are qualitative, desk-based assessments derived from published literature rather than from new training experiments. Inductive bias and hyperparameter sensitivity are treated as secondary criteria that complement empirical performance and computational complexity. Interpretability denotes

the extent to which an operator's learned parameters or aggregation weights can be meaningfully related to geometric or physical relations (for example, attention scores in GATv2 or spatial kernels in SplineConv). It is included here for completeness but does not serve as a primary factor in the present selection process.

Table 4.9: Qualitative decision matrix for candidate operators. Ratings: "+" favourable; "±" mixed or neutral; "–" unfavourable.

Operator	Performance	Complexity	Inductive Bias	Hyperparam Sensitivity	Ease of Impl.	PyG Compat.	Interpret-ability
GENConv (DeeperGCN)	+	+	+	+	+	+	±
GeneralConv	±	+	±	+	+	+	–
PNAConv	+	±	±	±	±	+	–
NNConv / ECC	+	–	+	+	+	+	±
CGConv	±	+	+	+	+	+	±
SplineConv	±	±	+	±	±	+	±
GMMConv (MoNet)	±	±	+	±	±	+	±
GATv2Conv	±	+	±	+	+	+	+
GCN+	+	+	±	±	±	±	–
ResGatedConv	±	+	+	+	+	+	–

Operators that have demonstrated strong performance on tasks with continuous edge attributes, such as NNConv, PNAConv, and GENConv, receive favourable ratings under *Performance*, acknowledging that reported benchmarks differ in scope and configuration. The *Complexity* reflects the relative parameter burden of each operator under the latent width d used in this work. While most architectures scale as $\Theta(d^2 + ed)$, NNConv introduces a cubic dependence on d due to its edge-conditioned $d \times d$ filters, making it noticeably heavier than the others in terms of parameter count. Accordingly, NNConv receives a lower complexity rating, whereas the remaining operators exhibit broadly comparable computational cost. Kernel-based methods, including SplineConv and GMMConv, obtain mixed ratings for *Hyperparameter Sensitivity* and *Ease of Implementation* because of their reliance on kernel resolution, and pseudo-coordinate normalisation. *Inductive Bias* is rated higher for operators with explicit geometric conditioning (NNConv, CGConv, SplineConv, GMMConv) than for those that incorporate edge information only indirectly (PNAConv, GeneralConv, GCN+, GATv2). *Interpretability* is strongest for attention mechanisms (GATv2) and for operators with explicit kernel or gating functions that can be inspected directly, whereas residual or ensemble-based designs are more opaque.

The ranking provides a principled basis for defining the operator subset to be implemented in the WindowGraphNet benchmark. Priority is given to architectures that combine geometric expressivity, depth stability, and practical ease of integration. To ensure diversity across message-passing paradigms while maintaining a tractable experimental scope, five operators are retained for implementation: GENConv, PNAConv, NNConv (or CGConv), GCN+, and SplineConv (or GMMConv). Together, these represent complementary inductive biases, from depth-stable generalists to explicitly geometric, edge-conditioned, and kernel-based formulations, providing a balanced foundation for controlled optimisation and comparative evaluation.

4.6.3. Selected Operators and Justification

The selected operators collectively span the main design philosophies identified in the comparative analysis. GENConv and PNAConv serve as depth-stable general-purpose baselines; NNConv and GCN+ offer contrasting strategies for incorporating edge features: NNConv uses an edge-conditioned filter generated by an MLP (parameter-intensive in the homogeneous regime, but tractable in our heterogeneous setting), whereas GCN+ integrates edges additively with a lightweight transform. SplineConv introduces a kernel-based geometric prior well suited to continuous spatial reasoning. This combination preserves diversity across architectural families while keeping the ensuing training and evaluation phases computationally manageable.

The following paragraphs summarise the motivations and potential limitations associated with each selected operator in the context of DF prediction. Together, they reflect complementary inductive biases spanning residual, aggregator-based, edge-conditioned, and kernel-interpolated message passing. Ta-

ble 4.10 provides a concise overview of their design characteristics and rationale for inclusion.

GENConv (DeeperGCN) is included as a depth-stable, residual-based generalist that directly integrates edge features and has demonstrated strong performance across large-scale benchmarks. Its message normalisation and pre-activation residual design enable deep architectures without degradation, making it well-suited for multi-hop dependencies inherent to DF prediction. While its inductive bias toward geometry is less explicit than kernel or edge-conditioned methods, it remains robust to hyperparameter choices and straightforward to implement in PyG.

PNAConv provides a complementary architecture through multi-aggregation and degree-scaling schemes that enhance expressivity across heterogeneous neighbourhood structures. Its strong performance on molecular and regression tasks such as ZINC demonstrates its suitability for capturing local variability, analogous to spatially diverse window–sensor configurations. Although more sensitive to hyperparameter tuning than GENConv, its distinct inductive bias justifies inclusion for architectural diversity.

NNConv (Edge-Conditioned Convolution) explicitly parameterises message functions via an MLP applied to edge attributes, directly aligning with the geometric relations governing DF (distances, angles, and solid angles). While parameter-intensive in the homogeneous regime due to its edge-dependent $d \times d$ filters, its cost scales linearly with edge dimension in the heterogeneous case used here. NNConv is therefore retained for its strong geometric grounding, with its edge-MLP width h_e later tuned to balance expressivity and efficiency.

GCN+ extends the classical GCN by introducing a learned linear edge transformation that enables additive edge integration. This mechanism maintains low complexity while improving sensitivity to geometric relations represented in edges. Empirical results on large-scale OGB datasets show that GCN+ matches or exceeds the performance of more complex models, offering a computationally efficient and stable baseline for DF prediction.

SplineConv introduces a continuous geometric prior through B-spline kernel interpolation over pseudo-coordinates, allowing messages to vary smoothly with spatial relations. This supports the modelling of light transport as a continuous geometric function. While performance depends on kernel resolution and coordinate normalisation, SplineConv offers a complementary inductive bias to parametric edge conditioning, enhancing diversity across the selected operators.

Table 4.10: Summary of selected operators and rationale for inclusion.

Operator	Edge Feature Handling	Depth Stability	Complexity Balance	PyG Compat.	Inductive Bias	Rationale
GENConv	Direct addition	High	Balanced	Yes	Residual generalist	Deep, stable, strong benchmark record
PNAConv	Aggregator + scalers	High	Slightly heavier (homog.)	Yes	Multi-aggregation	Captures varied graph structures
NNConv	Edge-conditioned weights	Moderate	Heavy (homog.), light (heterog.)	Yes	Strong geometric	Explicit modelling of continuous edge features
GCN+	Additive edge transform	Strong	Very light	Yes	General-purpose	Efficient, stable baseline with edge awareness
SplineConv	Kernel interpolation	Moderate	Light/moderate	Yes	Strong geometric	Smooth, spatially aware inductive bias

4.6.4. Operator-specific hyperparameters

Each message-passing operator introduces a distinct set of structural and functional parameters in addition to the shared hyperparameters defined in Section 4.6.1. These operator-specific hyperparameters govern how information is aggregated, normalised, and propagated across edges, and therefore determine both the model's inductive bias and its computational footprint. Their inclusion in the optimisation process ensures that each operator is evaluated under conditions that reflect its full representational potential rather than a restricted or arbitrarily fixed configuration. The following subsections summarise the hyperparameter search space defined for each selected operator, together with the rationale for including each parameter in the BO.

All operators are implemented in PyTorch Geometric (PyG) within a unified wrapper that standardises input–output interfaces and supports both homogeneous and heterogeneous (bipartite) message pass-

ing.[4] This wrapper ensures consistent handling of node and edge attributes across architectures and enables interchangeable use of operators within the same training pipeline. All operators except GCN+ use the native PyG implementations, while GCN+ is reimplemented from the authors' public repository to enable edge-aware message passing and integrated into the same interface [100]. The descriptions below summarise the operator-specific search dimensions rather than implementation details; the corresponding value ranges are listed in Appendix C.4.

GENConv (*DeeperGCN*) is optimised as a robust, depth-stable generalist with explicit edge integration and message normalisation. Its adaptive soft aggregation enables effective gradient propagation through deep stacks, and the learnable temperature and power terms regulate the sharpness of neighbour weighting.

Tuned parameters. The search space includes the aggregation function $\in \{\texttt{add}, \texttt{mean}, \texttt{max}, \texttt{softmax}, \texttt{softmax\_sg}, \texttt{power}\}$, softmax temperature $t \in [0.5, 2.0]$, and power exponent $p \in [0.5, 2.0]$, both optionally learnable (`learn_t`, `learn_p`). Message normalisation is toggled (`msg_norm`) with an optional learnable scale (`learn_msg_scale`). Architectural parameters comprise the internal MLP expansion factor (`expansion` $\in \{2\text{–}6\}$), the depth of stacked linear layers within each convolution block (`num_layers` $\in \{1, 2, 3\}$), and an optional bias term.

Constraints and rules. For `softmax_sg` aggregation, the temperature parameter is fixed (non-learnable).

PNAConv is configured as a degree-aware, high-capacity operator that aggregates messages through multiple statistical functions and scales them by the local degree distribution. This formulation allows it to capture distinct node connectivities and variable sensor densities, which are characteristic of the DF graphs.

Tuned parameters. The search space comprises combinations of aggregation functions (`aggregators` $\subseteq \{\text{mean}, \text{max}, \text{sum}\}$) and degree-scalers (`scalers` $\subseteq \{\text{identity}, \text{amplification}, \text{attenuation}, \text{linear}, \text{inverse linear}\}$). Architectural parameters include the number of towers (`towers` $\in \{1, 2, 3, 4\}$), the number of pre- and post-MLP layers (`pre_layers`, `post_layers` $\in \{1, 2\}$) and the division strategy of the input feature space (`divide_input` $\in \{\text{True}, \text{False}\}$).

Constraints and rules. The hidden feature dimension is required to be divisible by the number of towers, i.e. `hidden_size` $\bmod$ `towers` $= 0$, to satisfy the implementation constraint of tower partitioning. The degree histogram (`deg`) is pre-computed on the training set and passed as a fixed argument rather than optimised. In the homogeneous setup, a single histogram is used for all nodes, while in the heterogeneous configuration, degree statistics are computed separately for each edge type and provided as a dictionary `deg_by_etype[etype]`. These histograms ensure degree-aware normalisation across relations without adding extra learnable parameters.

NNConv (*Edge-Conditioned Convolution*) applies an edge-conditioned MLP to parameterise the message function, enabling the operator to learn continuous filters that depend directly on geometric edge attributes. This formulation aligns naturally with DF prediction, where distances, angles, and solid angles govern light transfer between nodes. The edge network is implemented as a small MLP that maps each edge's geometric attributes to an $n \times n$ filter matrix. When `edge_mlp_depth`=1, this reduces to a single linear projection from the edge features to the filter coefficients. For `edge_mlp_depth`=2, a hidden layer of width `edge_mlp_hidden` is inserted, introducing a controlled nonlinearity while keeping the generator lightweight. This shallow design provides sufficient expressive power to capture smooth geometric effects relevant for DF prediction without incurring the cost of deeper edge networks.

Tuned parameters. The edge network depth (`edge_mlp_depth` $\in \{1, 2\}$) and hidden width (`edge_mlp_hidden` $\in \{8, 16, 24, 32\}$) are varied to balance expressivity and cost. The aggregation scheme is selected from {`mean`, `add`}, and both `root_weight` and `bias` are toggled. The mid feed-forward block is disabled for NNConv (`use_mid_ffn=False`) to limit model depth and memory usage.

[4] All operators are wrapped in a unified `WindowGraphNet` interface that ensures consistent argument passing and compatibility with PyG's heterogeneous message-passing API.

Constraints and rules. Because NNConv generates a dense $n \times n$ weight matrix for every edge, its complexity scales quadratically with the hidden dimension. To maintain a comparable computational budget, the hidden size is restricted to $\texttt{n} \in \{32, 48, 64\}$, whereas other operators are allowed up to 128. Before training, each trial estimated memory consumption as

$$\texttt{est\_bytes} = E_{\max} \times n^2 \times 4,$$

where $E_{\max}$ denotes the largest edge count across the training graphs and four bytes represent the `float32` precision. Configurations with $\texttt{n} \geq 128$ or $\texttt{est\_bytes} > 3 \times 10^9$ are pruned pre-emptively to prevent GPU memory overflow.

GCN+ is applied only in the homogeneous configuration, as its formulation relies on a standard Laplacian-based propagation and lacks native support for bipartite message passing required by the heterogeneous setup. The operator extends the classical GCN with an additive edge-feature integration term and lightweight feed-forward enhancement, combining efficiency with improved expressivity for edge-conditioned learning.

Tuned parameters. The search space comprises the residual toggle ($\texttt{residual} \in \{\text{True}, \text{False}\}$), the operator's internal normalisation flag ($\texttt{use\_bn} \in \{\text{True}, \text{False}\}$), the feed-forward block toggle ($\texttt{use\_ffn} \in \{\text{True}, \text{False}\}$) with hidden multiplier ($\texttt{ffn\_hidden\_mult} \in \{2, 3, 4\}$), and dropout (`dropout`). The internal `use_bn` option does not conflict with the global normalisation setting, as it acts inside the convolution update rule whereas the global choice affects only the surrounding MLP layers, allowing both to be tuned independently without interference. Self-loops are explicitly disabled (`add_self_loops=False`) because, although classical GCN-based formulations include both self-loops and a residual pathway, the residual connection in GCN+ already preserves each node's own features; enabling self-loops in addition would therefore double-count self-information and provide no additional benefit.

Constraints and rules. Residual connections are treated as intrinsic to GCN+ and are therefore excluded from the global residual flag applied to other operators. The operator is evaluated only in the homogeneous experiments, where its Laplacian-based convolution and additive edge-conditioning are well defined.

SplineConv introduces continuous B-spline kernels defined over pseudo-coordinates derived from the edge geometry, providing a smooth geometric prior for message passing. This allows the operator to model light transport as a continuous function of spatial relationships while maintaining a compact parameterisation.

Tuned parameters. The pseudo-coordinate dimensionality is selected as $\texttt{d\_pseudo} \in \{2, 3\}$, with corresponding kernel size $\texttt{kernel\_size} \in \{3, \ldots, 6\}$ and spline degree $\texttt{spline\_degree} \in \{1, 2, 3\}$. The spline basis type (`is_open_spline`) is toggled between open and closed formulations. The aggregation function is chosen from {`mean`, `max`, `add`}, while `root_weight` and `bias` are optional. Two auxiliary feed-forward toggles are exposed: the mid-layer block after convolution (`use_mid_ffn`) and an intra-block feed-forward layer (`use_ffn`) with hidden multiplier $\texttt{ffn\_hidden\_mult} \in \{2, 3, 4\}$.

Constraints and rules. To prevent degenerate spline bases, trials are pruned when the kernel size is smaller than $\texttt{d\_pseudo} + 1$ or, for $\texttt{d\_pseudo} = 3$, exceeded 5. Pseudo-coordinates are generated from edge attributes via a fixed linear–sigmoid projector that maps all values to $[0, 1]$, ensuring consistent normalisation across relations and avoiding the need for dataset-specific scaling. Numerical safety margins ($\varepsilon > 0$) are applied to clamp extreme values. The operator is evaluated in both homogeneous and heterogeneous settings, using bipartite calls where supported by the PyG build.

With both shared and operator-specific parameters defined, the following subsection outlines the BO setup used to explore these configurations and determine the most effective architecture for each operator.

4.6.5. Bayesian Optimisation Setup

The hyperparameter search for the shortlisted operators is conducted through BO using Optuna [131]. The shared and operator-specific search spaces are identical to those defined in Section 4.5.2, where common parameters (*e.g.* hidden dimensions N_1–N_3, normalisation type, activation, dropout, learning

rate, weight decay, and residual connections) are searched jointly with operator-dependent parameters such as kernel size, temperature, or tower divisibility. All experiments are executed on GPU hardware as described in Section 4.5.2.

Optimisation protocol. For each operator, a dedicated Optuna study is created and optimised independently to prevent cross-operator parameter leakage. In contrast to the multi-objective feature ablation study, which allowed concurrent optimisation across multiple metrics and feature configurations, the operator tuning task involved parameter dependencies, *e.g.*, choosing `GENConv` activates only its corresponding temperature and aggregation parameters. As such, a single-objective formulation is required to ensure a valid search space. Each operator is therefore optimised using a weighted composite loss:

$$\mathcal{L}_{\text{BO}} = \text{RMSE}_{\text{val}} + 0.20\,\text{MaxAbs}_{\text{val}} + 5.0\,\big(1 - \text{SSIM}_{\text{val}}\big) \tag{4.16}$$

which down-weights sensitivity to local outliers (*MaxAbs*) and converts the structural similarity into a loss term. This scalar objective enables BO while preserving the trade-off structure of the downstream Pareto analysis.

Aside from the single-objective formulation described above, all training and optimisation settings are identical to those in Section 4.5.2. Each operator is optimised through five independent Optuna studies (random seeds $(40 - 44)$), each comprising 200 trials. The seeds affects only the 30 initial random samples that are used by the TPE sampler to initialise its surrogate model; the subsequent optimisation process is identical across studies.

Validation protocol and data separation. A strict separation between optimisation and evaluation data is maintained to prevent information leakage and to yield statistically valid performance estimates. Whereas the feature ablation phase used the *Transformation Dataset* to assess rotational and scaling invariance, the operator optimisation phase operated exclusively on the validation subset of the main dataset. This follows established good practice in surrogate modelling, where the validation set serves to guide model selection under the same distribution as training, while the *Transformation Dataset* is reserved for post-optimisation evaluation of operator generalisation. Including the *Transformation Dataset* during optimisation would introduce distributional overlap and risk overfitting to specific geometric transformations, thereby compromising the integrity of subsequent generalisation tests. Restricting BO to the validation data therefore ensures that hyperparameter tuning affects only model capacity and inductive bias within a controlled training–validation regime, leaving the *Transformation Dataset* as a genuinely unseen benchmark.

Study configuration and reproducibility. All studies are run with `multivariate=False` to avoid cross-parameter correlations within the TPE sampler, as such dependencies are invalid when different operators have non-overlapping parameter spaces. Full parameter ranges are listed in Appendix C.4.

Pareto-based selection of optimal configurations. The identification of well-balanced hyperparameter settings follows the same Pareto-based multi-objective selection principle used in the feature ablation study, but is here applied post hoc to the outcomes of the BO trials. Although the BO itself is performed with a single composite objective (Section 4.6.5), each trial recorded the full set of evaluation metrics, *RMSE*, *MaxAbs* and *SSIM*, all averaged over rooms to ensure that larger scenes with more sensor nodes do not dominate the results. This enables an equivalent multi-objective Pareto analysis to be carried out on the saved results, independent of the optimisation objective, and allows configurations that achieve balanced performance across all three criteria to be identified.

Unlike the feature ablation phase, where Optuna's internal best-trial tracking is used, the Pareto front is computed explicitly from the full set of evaluated trials using a custom implementation of fast non-dominated sorting (analogous to NSGA-II [150]). To enable this analysis, the objectives are cast as a mixed minimisation problem by treating $(\textit{RMSE}, \textit{MaxAbs}, 1 - \textit{SSIM})$ as the objective vector. A configuration i is said to dominate another configuration j if it performs no worse in any objective and strictly better in at least one:

$$i \prec j \quad \Leftrightarrow \quad \forall k:\ f_k(i) \le f_k(j) \quad \text{and} \quad \exists k:\ f_k(i) < f_k(j), \tag{4.17}$$

where $f_k(\cdot)$ denotes the value of objective k. The set of non-dominated configurations constitutes the Pareto front, representing all solutions for which no objective can be improved without degrading another. Because the number of non-dominated points can vary substantially across operators, a fixed subset of $K = 20$ Pareto-optimal configurations is retained for visualisation. Successive fronts are accumulated until this count is exceeded, after which the remaining configurations are selected from the last front according to their crowding distance to preserve diversity. To exclude pathological trade-offs (e.g., extremely low *MaxAbs* but unacceptably high *RMSE*), the search is limited to a feasible region defined by *RMSE* ≤ 0.15.

Deriving the selected configuration from the Pareto set. From the Pareto-efficient subset, two complementary summaries are reported. First, the elite reference corresponds to the single best trial within the Pareto set, defined as the configuration with the lowest validation *RMSE* averaged over rooms and seeds. Second, the selected configuration is derived by aggregating hyperparameters across all Pareto configurations to obtain a stable and representative choice. This aggregation is preferred in the optimisation phase because, unlike the feature ablation study, no additional retraining is performed before selecting the final operators. Selecting parameters that recur across near-optimal trials mitigates the risk of choosing an unstable outlier affected by stochastic variation and reflects consistent performance trends within the Pareto set. For discrete parameters, the most frequent value within the Pareto set is chosen; in case of ties, the option with the lowest median *RMSE* is retained. For continuous parameters, the marginal median is reported together with the interquartile range $[q_{25}, q_{75}]$ to indicate variability. A concrete configuration is then obtained by selecting the Pareto trial whose discrete parameters match the derived tuple and that minimises the validation *RMSE*, or, if multiple candidates exist, the one closest to the continuous medians. This ensures that the selected configuration represents a statistically stable, interpretable, and reproducible basis for downstream evaluation. The elite configuration is retained for reference in Appendix D.3.2, while the selected configuration is used for subsequent operator evaluation.

4.7. Final Model Evaluation and Analysis

Having obtained optimised configurations for each operator, the final phase evaluates their generalisation performance and data efficiency. This section first presents the operator comparison and selection procedure that determine the final WindowGraphNet architecture, followed by an empirical analysis of its learning behaviour and data efficiency through learning-curve experiments

4.7.1. Operator Evaluation and Final Model Selection

Following BO, the nine WindowGraphNet variants are retrained using their selected hyperparameter configurations under identical conditions to enable a controlled comparison of generalisation performance. Five homogeneous operators and four heterogeneous operators are evaluated, excluding GCN$^+$ from the latter due to incompatibility with the heterogeneous formulation. Each model is retrained across five random seeds $(40 - 44)$, using the same optimiser, scheduler, batch size, and early-stopping protocol described in Section 4.5.2. All other training settings, including data normalisation and batching strategy, are kept constant to ensure full comparability across operators.

Performance is evaluated on the *Transformation Dataset* to assess geometric generalisation beyond the training distribution. For each model, validation metrics are computed per transformation and aggregated across rooms to avoid bias toward larger scenes. The resulting test metrics are then summarised across transformations using the robustness-oriented ranking procedure introduced earlier. Specifically, worst-case, tail-risk (CVaR-2), and average-behaviour statistics are computed for each metric (*RMSE*, *MaxAbs*, *SSIM*) and combined into a composite rank score as defined in Section 4.5.2. This provides a unified measure of overall robustness rather than relying on a single best-performing configuration.

The operator achieving the lowest composite rank is selected as the final GNN configuration for subsequent analyses. This procedure yields a consistent and statistically grounded basis for identifying the most reliable operator across both homogeneous and heterogeneous regimes, while ensuring that the final choice reflects stable performance across random seeds and geometric transformations.

4.7.2. Learning-Curve Analysis

To evaluate the data efficiency of the final GNN architecture, a learning-curve experiment is conducted following the learning-curve sampling method proposed by Meek et al. [151]. In this framework, model performance is expressed as a function of the available training data, while the model configuration, optimisation schedule, and validation protocol remain fixed. The resulting curve quantifies how rapidly predictive accuracy improves as additional training samples are introduced, providing an empirical measure of the model's sample efficiency.

Experimental design. Following Meek *et al.* [151], the dataset is partitioned into progressively larger, nested subsets such that

$$D_1 \subset D_2 \subset \cdots \subset D_n. \tag{4.18}$$

This nesting ensures that performance differences arise solely from the inclusion of new samples rather than random composition effects. A fixed validation set of 20 rooms is kept constant for all repetitions, while the remaining rooms form the training pool. Within each repetition, the training pool is shuffled once using a pseudorandom seed, and prefix subsets of sizes $[1,\ 2,\ 4,\ 8,\ 16,\ 32,\ 64,\ 96,\ 128,\ 160,\ N_{\text{train}}]$ are extracted sequentially. For each subset size n, the minimum validation mean-squared error is recorded.

Repetition and random seeds. To account for stochasticity in optimisation and data partitioning, the entire procedure is repeated for five random seeds $(40 - 44)$. Each seed determines (i) the random split of validation rooms, (ii) the shuffle order defining the nested subsets, and (iii) the initialisation of all pseudorandom generators (`Python`, `NumPy`, and `PyTorch`). Fixing seeds ensures deterministic GPU execution and enables reproducible comparisons across runs. Averaging across seeds yields an unbiased estimate of the mean validation performance and its variability.

Power-law characterisation. Empirical studies of neural-network scaling behaviour [152, 153] and classical learning-curve analysis [151] show that generalisation error typically decreases with training-set size according to a power law:

$$L(n) \;=\; A\, n^{-\alpha} + B, \tag{4.19}$$

where A is a scaling coefficient, α is the data-efficiency exponent, and B represents the asymptotic performance plateau. After aggregating results across seeds, the mean validation loss for each subset size is fitted to Eq. (4.19). The plateau term B is estimated from the final three points of the curve, representing the near-saturation regime, while A and α are optimised by nonlinear least squares on the mid-range data ($4 \leq n \leq 128$). The fitted curve provides a compact characterisation of scaling behaviour, and the exponent α offers a quantitative measure of sample efficiency, larger values indicating faster performance gains with increasing training data.

4.8. Final Testing Methodology

Following the operator benchmarking phase (Section 4.6.3), the best-performing *WindowGraphNet* configuration is selected for comparative evaluation against the four ANN baselines. This stage examines whether a graph-based formulation provides measurable advantages over conventional feedforward architectures when tested under identical conditions.

Training Repetitions and Evaluation Procedure

All ANN baselines were already trained five times during the main benchmarking stage using seeds $(40$–$44)$, and these trained models are reused for the final evaluation. The selected *WindowGraphNet* is retrained using the same five seeds, ensuring that both model families are compared under identical random initialisations and optimisation conditions.

Performance is measured using *RMSE*, *MAE*, and *SSIM* computed over all sensor nodes, together with qualitative inspections of predicted–reference DF maps. For each model, the final score reported on every metric corresponds to the mean across the five repetitions, while the associated standard deviation quantifies sensitivity to the choice of random seed.

All models are evaluated on the *Final Test Dataset* (Section 4.1.3), which comprises six geometric tiers summarised in Table 4.2. Each tier isolates a specific transformation in room geometry or window

placement—including rotations, aspect-ratio changes, and progressively deeper L-shaped recesses—enabling model generalisation to be assessed across distinct families of spatial variation. To maintain interpretability, evaluation is performed per tier: models are tested on each geometric category separately, allowing differences in performance to be attributed directly to the underlying geometric and visibility conditions rather than masked by aggregated statistics.

Tiers 4 and 5 introduce L-shaped rooms with partial or deep self-occlusion, in which portions of the sensor grid are not directly visible from the window opening. This condition alters the effective input structure for both model families: for *WindowGraphNet*, the graph topology becomes more heterogeneous as sensor–window visibility edges are pruned, whereas for the ANNs, the same occlusion reduces the predictive information contained in the global geometric descriptors. Testing these tiers separately therefore provides insight into how well each model type extrapolates to scenarios with indirect lighting and limited visibility.

Feature Handling under Occlusion

When evaluating Tiers 4 and 5, a portion of the sensor grid lies in regions with no direct line of sight (LOS) to the window opening. This affects several input variables that depend on window visibility. To ensure physical consistency and comparability across model types, the following conventions are applied.

Le–Thanh-style features. In the original formulation by Le–Thanh *et al.*, each window is represented by a fixed pair of descriptors, a distance d_m and an angular parameter w_m, assigned to one of four predefined “4-square” slots. These descriptors are padded with zeros only when a window is *absent* from a given slot, ensuring a fixed input dimensionality across different room configurations. The zero therefore acts solely as a structural placeholder, not as an indicator of visibility.

Under occlusion, no changes are made to the window-based descriptors d_m and w_m. In the Le–Thanh method, visibility is conveyed exclusively through the ray-based vector x, where each ray x_i records the distance to the first intersected surface and takes the value 0 when the ray passes through a window opening. As a result, loss of line of sight is encoded via the pattern of non-zero ray lengths in x, while the geometric descriptors (d_m, w_m) remain valid and unmodified. This approach cleanly separates geometric information from visibility handling and is retained in the present work.

Dieguez-style features. In the numerical encoding proposed by Dieguez et al. [19], occlusion is not handled through explicit masking . Instead, all descriptors are computed directly from geometry, and regions without a view of the window are characterised by the natural collapse of the window projection on the spherical surface. In such cases the solid angle becomes extremely small or numerically zero, providing the primary indication of occlusion.

The remaining projection-based descriptors, the aspect ratio of the projection and the solid-angle tilt, are not modified by Dieguez et al.; however, both quantities become ill-defined when the projected window area degenerates. In this work these two descriptors are set to zero whenever the solid angle collapses, solely to avoid numerical instabilities in the Grasshopper implementation (e.g. division by zero or invalid geometry). This zero therefore denotes a degenerate projection rather than a physical occlusion mask. Other geometric descriptors, such as the sensor–to–window distance and the angle to the window normal, remain unchanged under occlusion.

Graph-based model. In the heterogeneous *WindowGraphNet*, occlusion is not represented through feature masking, as in the numerical baselines, but is embedded directly in the graph structure. Each sensor node is potentially connected to five window nodes (four corners and one centre), but an edge is created *only* when an unobstructed line of sight exists between the two. If the window corner or centre is occluded from a given sensor position, the corresponding window–sensor edge is simply absent from the graph. Thus, visibility is encoded structurally: the existence or non-existence of an edge replaces the need for an explicit visibility attribute.

This topological encoding has no analogue in either the Le–Thanh or Dieguez feature sets, both of which retain fixed input structures regardless of occlusion. In contrast, the heterogeneous formulation allows visibility to be expressed through the relational pattern of the graph itself: sensors located in

recessed or self-occluded regions naturally connect to fewer window nodes, while those with a full view maintain all five connections. As illustrated in Figure 4.14, occluded sensors simply possess fewer window–sensor edges, and this variation in connectivity enables the message-passing operator to distinguish between regions with direct, partial, or no line of sight.

Summary. Across the three feature families, occlusion is represented in fundamentally different ways. In the Le–Thanh encoding, window descriptors retain fixed values under occlusion and visibility is conveyed exclusively through the ray-based vector x. In the Dieguez feature set, no explicit visibility mechanism is used; instead, occlusion emerges implicitly through the geometric collapse of the window projection, most notably through the solid angle. In the heterogeneous *WindowGraphNet*, visibility is encoded structurally: window–sensor edges are created only when a direct line of sight exists, allowing occlusion to appear directly in the graph topology. Together, these approaches span scalar, geometric, and structural representations of visibility, providing a consistent basis for comparing model behaviour under self-occlusion. Crucially, no retraining is required when introducing occlusion: for the ANN baselines, occluded windows are handled naturally as their feature values reduce to zero, while for *WindowGraphNet* occlusion manifests as a topological change in the test graph, leaving the learned model weights unchanged.

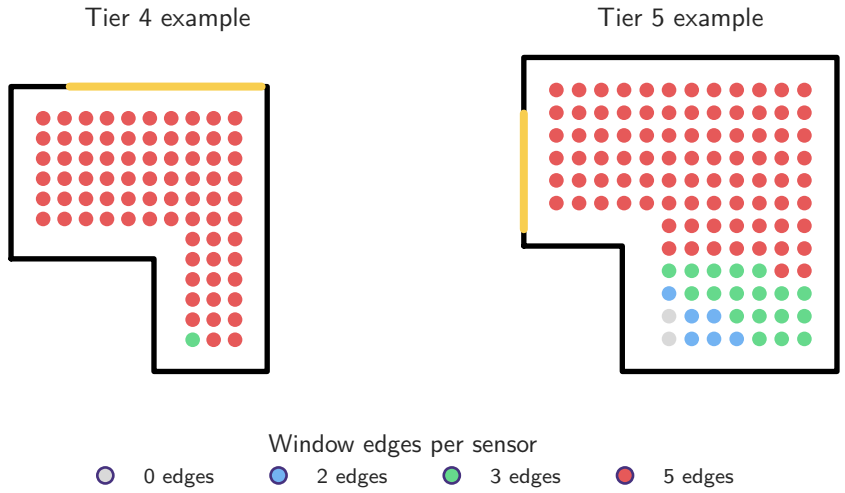


Figure 4.14: Sensor visibility categories based on the number of available window edges (0, 2, 3, 5) for two example geometries (Tier 4 and Tier 5). The black polyline shows the room boundary; the dark yellow segment marks the window.

4.9. Biases and Study Limitations

This subsection documents sources of bias arising from the dataset design, simulation assumptions, modelling choices, and evaluation protocol, and clarifies the scope of validity of the reported results.

Sampling and simulation biases.

(i) *Single sky condition.* All simulations use the CIE overcast sky mandated for DF, which removes weather/solar-geometry variability by design. While appropriate for DF, this restricts external validity to overcast conditions and precludes claims about metrics dependent on climate or sun position. Results should therefore be interpreted as overcast–condition surrogates rather than climate-general predictors.

(ii) *Fixed material palette.* Ceiling/wall/floor reflectances (0.8/0.5/0.2) and glazing $T_v = 0.6$ are fixed across all datasets. This isolates geometric effects but can bias magnitude calibration when real spaces deviate materially (e.g., darker walls). Because reflectance effects are deemed secondary and held constant to preserve focus and reproducibility, external validity is limited to similar ranges.

(iii) *Reference plane and grid policy.* A 0.70 m reference plane and a 0.50 m exclusion band with ≈0.50 m grid spacing are applied uniformly. The deviation from EN 17037's 0.85 m plane is consistent within the study and preserves relative comparisons, but absolute DF levels may differ from assessments

strictly following 0.85 m. Uniform banding and spacing also imply that near-boundary behaviour is intentionally under-sampled.

(iv) *Typology coverage.* Training/validation data comprise centrally glazed square rooms varying only in width and WWR; no multi-room or multi-window layouts, façade context, or external obstructions are modelled. Final tests introduce offset windows and L-shapes, but still within single-room, single-opening scenarios. This clean isolation of geometry can underrepresent complexities in practice (multiple apertures, adjacent shading, furnishings).

(v) *Transformation resolution effects.* In the "scale $\times 2$" and "scale $\times 5$" cases, sensor spacing scales with geometry, which preserves relative density but increases absolute sensor–window distances and coarsens sampling in large rooms. Part of the measured performance shift therefore couples physical attenuation with resolution change.

Dataset composition and distribution shift.
Training/validation use LHS over width $[2.5, 8.0]$ m and WWR $[0.2, 0.8]$, with a stratified split (180/20). Test tiers introduce systematic OOD shifts: rotations, scaling (to $\times 5$), rectangles (2:1), offset windows, and L-shapes with partial/deep self-occlusion. Some tiers (e.g., Tier 3–5) also shift the feasible WWR downward due to geometric constraints, which can confound "geometry vs. glazing area" effects and bias magnitude comparisons across tiers.

Model and training biases.
(i) *Architecture and capacity asymmetry.* Benchmarks include four MLPs (Raw, Le-Thanh, Dieguez, Simple-Dieguez) and graph models (homogeneous/heterogeneous WindowGraphNet). Although all share the same train/val splits and optimisation habits, differences in parameter counts and inductive biases can advantage certain regimes (e.g., feature-rich MLPs vs. relation-centric GNNs). Reported comparisons therefore reflect "model × representation" bundles rather than architecture alone.

(ii) *Hyperparameter search exposure.* Feature ablation and operator/model selections are conducted on the *Transformation Dataset* (with fixed seeds and repeated Optuna studies). This thorough use of a held-out diagnostic set mitigates noise but introduces a risk of *tuning to the transformations*, i.e., slight optimistic bias on the same transformation families relative to entirely novel conditions. The Final Test set, kept disjoint for selection, reduces but cannot fully eliminate this selection bias.

(iii) *Normalisation and target scaling.* z-score statistics for features are computed on training data and reused for all other sets, avoiding leakage; DF targets are standardised only where stated and inverse-transformed for evaluation. Any systematic magnitude bias in some tiers (e.g., in deeply occluded recesses) may interact with this scaling choice.

(iv) *Randomness and reproducibility.* Seeds are fixed globally (non-BO) and deterministically assigned (within BO). Nonetheless, GPU and library non-determinism can introduce small run-to-run variance; early stopping on a single validation split further couples final checkpoints to stochastic training trajectories.

(v) *Parameter tying vs. relation specificity.* In the *homogeneous* variant, a single set of weights is shared across all edges/nodes, enforcing a uniform relation model; in the *heterogeneous* variant, separate parameters are assigned per relation (e.g., window$\rightarrow$sensor vs. window$\rightarrow$window), allocating unequal capacity across relation types. These are structural capacity choices baked into the methodology rather than tuned outcomes.

(vi) *Graph size variability.* Because the sensor grid is part of the graph, changes in room size or grid density produce graphs with different node/edge counts. No batch- or loss-level normalisation is applied to compensate for graph-size differences, so per-batch compute and gradient magnitudes scale with the number of sensor nodes by design. This fixes the optimisation dynamics to be graph-size dependent within the study.

(vii) *Schema extensibility.* The current node/edge vocabulary does not include walls, or exterior context nodes. Extending to these factors requires schema changes (new node/edge types) rather than retraining alone, which constrains the study's scope *by construction*.

Evaluation protocol limitations.
(i) *Metric coverage.* Optimisation and reporting emphasise *MSE/RMSE*, *MaxAbs*, and *SSIM* at sensor level. These capture accuracy, worst case, and spatial fidelity, but do not directly evaluate task-level criteria (e.g., compliance areas per EN 17037), nor perceptual comfort indices. Model ranking may therefore differ if assessed on downstream KPIs.

(ii) *Single validation split.* A fixed 20-case validation set and compact transformation set are efficient and reproducible, but lack k-fold variance estimates. Confidence intervals reported across seeds do not reflect split uncertainty.

(iii) *Resolution coupling.* Spatial *SSIM* and error maps are computed on grids whose density varies with scale tiers, slightly complicating cross-tier comparability (see above).

Scope and external validity.
The surrogate models are intended for early-stage single-room design with unobstructed façades under overcast conditions, within material and geometric ranges similar to those simulated. Generalisation to multi-room layouts, complex fenestration systems, external shading/context, or climate-based metrics requires retraining or explicit domain adaptation. The Final Test tiers demonstrate robustness to several geometric shifts (rectangles, offsets, self-occlusion), but remain within the same simulation paradigm and should not be extrapolated to dissimilar tasks without caution.

Mitigations and transparency.
To reduce bias and support reproducibility: (i) all fixed simulation parameters and software versions are explicitly documented; (ii) data splits, seeds, and BO settings are fixed and reported; (iii) the tiered test design separates selection from final evaluation; and (iv) results are reported with complementary metrics (*RMSE/MaxAbs/SSIM*) and room-level visualisations to reveal spatial failure modes. These practices limit, but cannot eliminate, the above constraints.

5

Feature Ablation and Operator Optimisation

This chapter presents the design and optimisation of the WindorGraphNet model for DF prediction. It unites two complementary investigations: the identification of effective node and edge descriptors, and the selection of the most suitable message-passing operator. Together, these stages define the feature composition and model architecture that constitute the final GNN used throughout the remainder of this study.

The first part of the chapter focuses on feature ablation following the methodology outlined in section 4.5. An information-theoretic and optimisation-based framework is used to determine which input descriptors most accurately represent daylight behaviour while remaining robust under geometric transformations. By fixing the GNN operator and varying only the inclusion or exclusion of candidate features, the analysis isolates the contribution of feature representation from architectural effects. The resulting compact feature mask defines the model's input space for all subsequent experiments.

The second part concerns operator optimisation in accordance with the methodology described in section 4.6. A series of message-passing operators representing different architectural principles, residual, kernel-based, attention, and edge-conditioned, are trained and optimised under controlled conditions. Each operator undergoes Bayesian hyperparameter optimisation to identify the configuration that best balances predictive accuracy, and robustness. Quantitative comparisons across operators establish how differences in message aggregation and normalisation influence performance in DF prediction.

The combined outcomes of both investigations yield a single, fully specified GNN architecture. This configuration represents the definitive model design developed in this work and forms the basis for subsequent benchmarking against established ANN surrogate models from the literature.

5.1. Feature Ablation

The feature ablation phase investigates which combinations of geometric and physical descriptors most effectively characterise daylight behaviour within graph-based architectures. The objective is to identify concise and physically interpretable sets of node and edge attributes that maximise predictive accuracy and generalisation under geometric transformations.

The analysis proceeds in two main stages. First, benchmark ANN surrogates are trained to establish empirical evidence of feature relevance and stability across scaled and rotated configurations. Second, these insights are combined with information-theoretic scoring and BO over binary feature masks to evaluate feature subsets within both homogeneous and heterogeneous graph formulations. Each formulation is optimised independently to account for differences in relational structure and message-passing context, resulting in two refined feature configurations that serve as the respective inputs for the subsequent operator optimisation studies.

5.1.1. Benchmark surrogates: initial feature screening

The four benchmark ANN surrogates were trained and tested on their respective feature sets to obtain an initial signal of feature informativeness and transformation robustness. The complete training and validation curves, as well as the full numerical tables of results, are reported in section D.1; only the key outcomes are summarised here.

Figure 5.5 shows that the *Simple-Dieguez* feature set performs consistently well across all transformation scenarios, with the *Dieguez*-based descriptors providing strong generalisation. The *Raw* model (using x, y, width, and WWR) and the *Le-Thanh* feature set perform competitively in the normal and scaled cases but degrade sharply under rotation. This behaviour reflects fundamental differences in how each encoding represents spatial relationships between sensors and window openings, as visualised in Figures 5.1–5.4.

The degradation of the *Raw* and *Le-Thanh* networks under rotation arises from their representation strategy. The *Raw* set encodes only global coordinates (x, y), window-to-wall ratio, and width. When the geometry is rotated, the window position changes but the coordinates remain fixed in world space; consequently, the network receives no indication that the window has moved, and the learned mapping between input and output becomes inconsistent. This limitation is evident in Figures 5.2 and 5.3, where the predicted daylight fields remain aligned with the original orientation rather than adapting to the new window position. The *Le-Thanh* encoding was designed to improve generalisation by projecting local geometric information onto a series of angular rays. However, all training rooms in the present dataset contain windows exclusively on the southern façade. This means that for each sample, only a subset of the 40 directional inputs is active while the remainder are zero. After rotation, a different subset of rays becomes active, producing activation patterns that were never observed during training. As a result, the model fails to interpret these new input combinations, leading to a collapse in performance under rotation, as also visible in Figures 5.2 and 5.3.

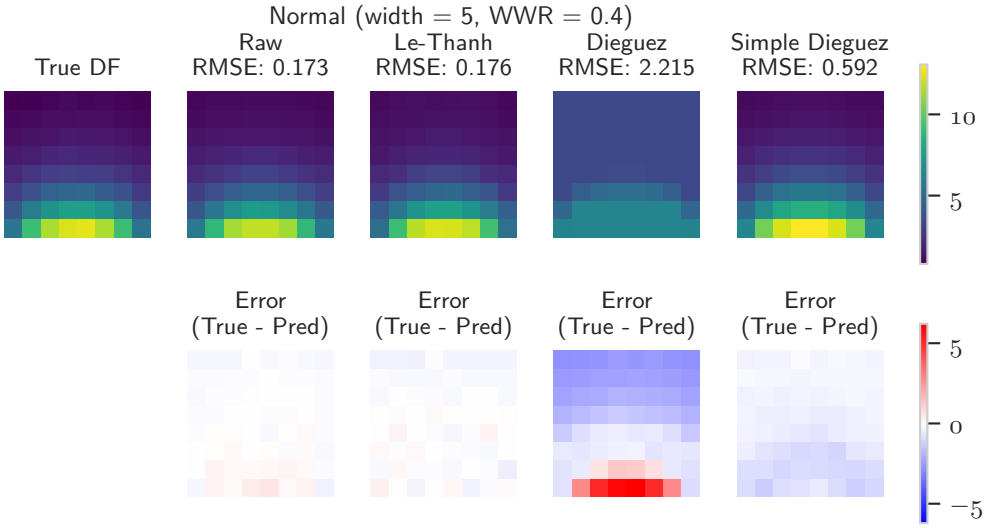


Figure 5.1: Comparison of DF predictions from four ANN surrogate models under the *Normal* configuration (width = 5 m, WWR = 0.4). The top row shows predicted DF distributions against the ground truth, while the bottom row visualises the corresponding spatial error maps (True–Pred). The results illustrate how the different feature encodings capture the spatial decay of daylight and the sharp luminance gradient near the window wall.

The *Dieguez* feature family incorporates explicit distance- and direction-based descriptors referenced to the window surface, which encode the spatial relationship between each sensor and the window directly. These features provide a physically meaningful signal that remains coherent under geometric transformations. The full *Dieguez* variant, however, suffers from oversmoothing effects caused by the sigmoid activations in its final layers, which induce saturation, hinder gradient flow, and lead to numerical instability during training. Because several layers operate in a narrow bottleneck regime (e.g. $16 \to 32 \to 16 \to 4 \to 2$), the network is particularly prone to collapsing its predictions toward a uniform DF field, failing to reproduce the façade–interior gradient.

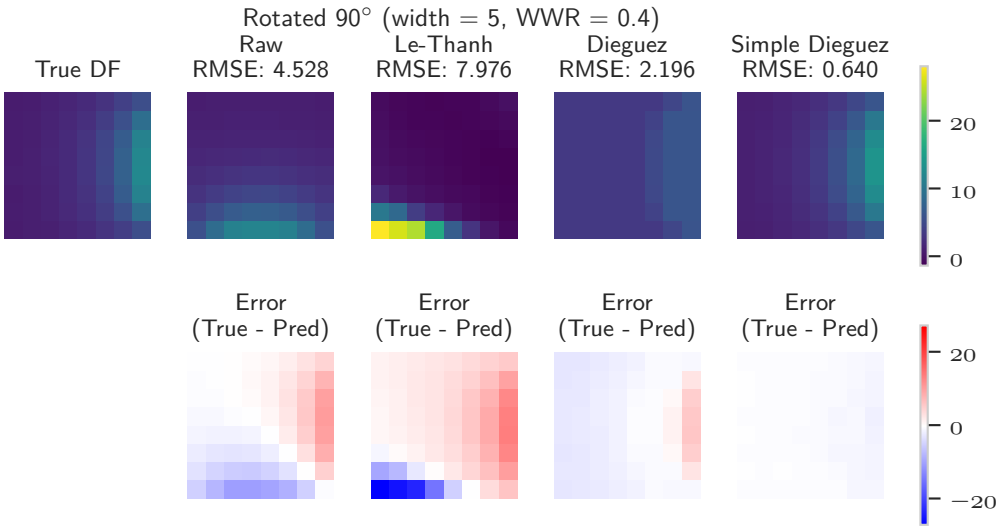


Figure 5.2: Comparison of ANN surrogate performance under a *90° rotation* of the room geometry (width = 5 m, WWR = 0.4). While the ground-truth DF field rotates accordingly, the Raw and Le-Thanh networks exhibit strong orientation bias, failing to adapt to the new window position. The Simple-Dieguez ANN maintains consistent spatial structure, indicating improved transformation robustness.

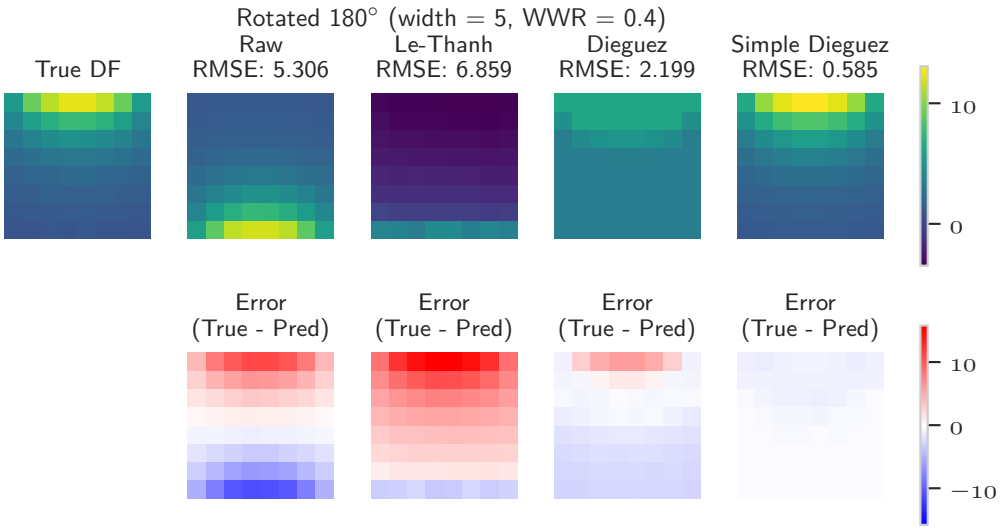


Figure 5.3: Performance of the ANN surrogates under a *180° rotation* of the room geometry. Models trained on explicit directional descriptors (e.g., Dieguez feature set) fail to reproduce the mirrored daylight pattern, while the simplified variant shows greater invariance. Error maps highlight the loss of spatial alignment and the presence of systematic over- and under-estimation across the field.

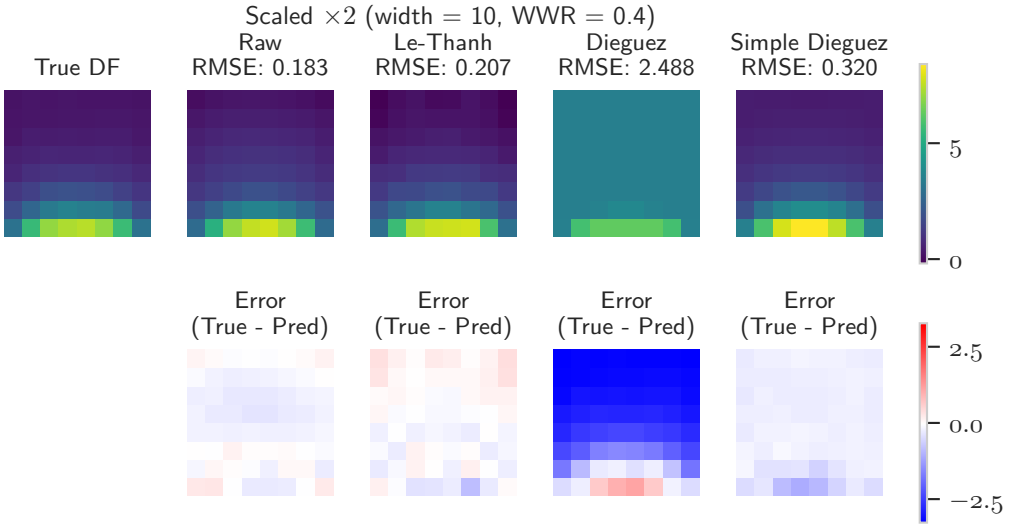


Figure 5.4: Comparison of ANN surrogate predictions under a *scaled* geometry (width = 10 (5 doubled), WWR = 0.4). The Raw and Le-Thanh ANNs maintain reasonable agreement with the true DF distribution, whereas the Dieguez model exhibits strong magnitude errors due to its dependence on absolute distances. The Simple-Dieguez variant remains stable, demonstrating improved scale consistency.

The simplified form, here referred to as the *Simple-Dieguez* model, improves stability by reducing the number of layers and replacing all sigmoid activations with ReLU, while keeping the original feature set unchanged. This architectural simplification preserves gradient flow and prevents activation saturation, allowing the physically informed descriptors to be exploited more effectively. The resulting predictions exhibit markedly improved stability and scale consistency across all geometric transformations, as illustrated in Figures 5.1 and 5.4.

Table 5.1: Mutual information (MI) and redundancy analysis for sensor coordinates and Le-Thanh encodings. Negative mRMR-like values indicate low relevance after accounting for redundancy.

Feature block	MI (all)	MI (mean)	Avg. redundancy	mRMR-like
x	0.28	0.16	0.34	-0.18
y	0.53	0.71	0.36	0.35
x_{thanh} (all)	0.28	0.26	0.31	-0.05
d_{thanh} (all)	0.46	0.22	0.38	0.03
w_{thanh} (all)	0.02	0.03	0.25	-0.09
Le-Thanh (all)	0.29	0.24	0.31	-0.04

The MI analysis in Table 5.1 supports these observations. The coordinate-based descriptors (x, y) and the majority of the *Le-Thanh* features exhibit both low mutual information and moderate redundancy, resulting in negative mRMR-like scores. These inputs carry little unique information beyond what is already captured by the Dieguez descriptors and do not encode rotationally invariant relationships. Although some d_{thanh} terms achieve higher MI when aggregated across all data, this effect is not consistent across transformations, yielding low averaged relevance.

Taken together, the quantitative results, visual inspection of the daylight fields, and the information-theoretic analysis point to a clear hierarchy of descriptor utility. Global or orientation-dependent inputs such as (x, y) and the *Le-Thanh* rays lack invariance and offer minimal unique information, while the physically grounded, window-referenced descriptors of the Simple-Dieguez set provide both accuracy and robustness under scaling and rotation. Consequently, the subsequent BO phase excludes (x, y) and all *Le-Thanh* features from the search space and concentrates on width, window-to-wall ratio, and the *Dieguez* descriptors. This refined feature basis defines the starting point for the systematic feature ablation experiments that follow, aimed at identifying the minimal yet sufficient set of geometric descriptors for reliable graph-based daylight prediction.

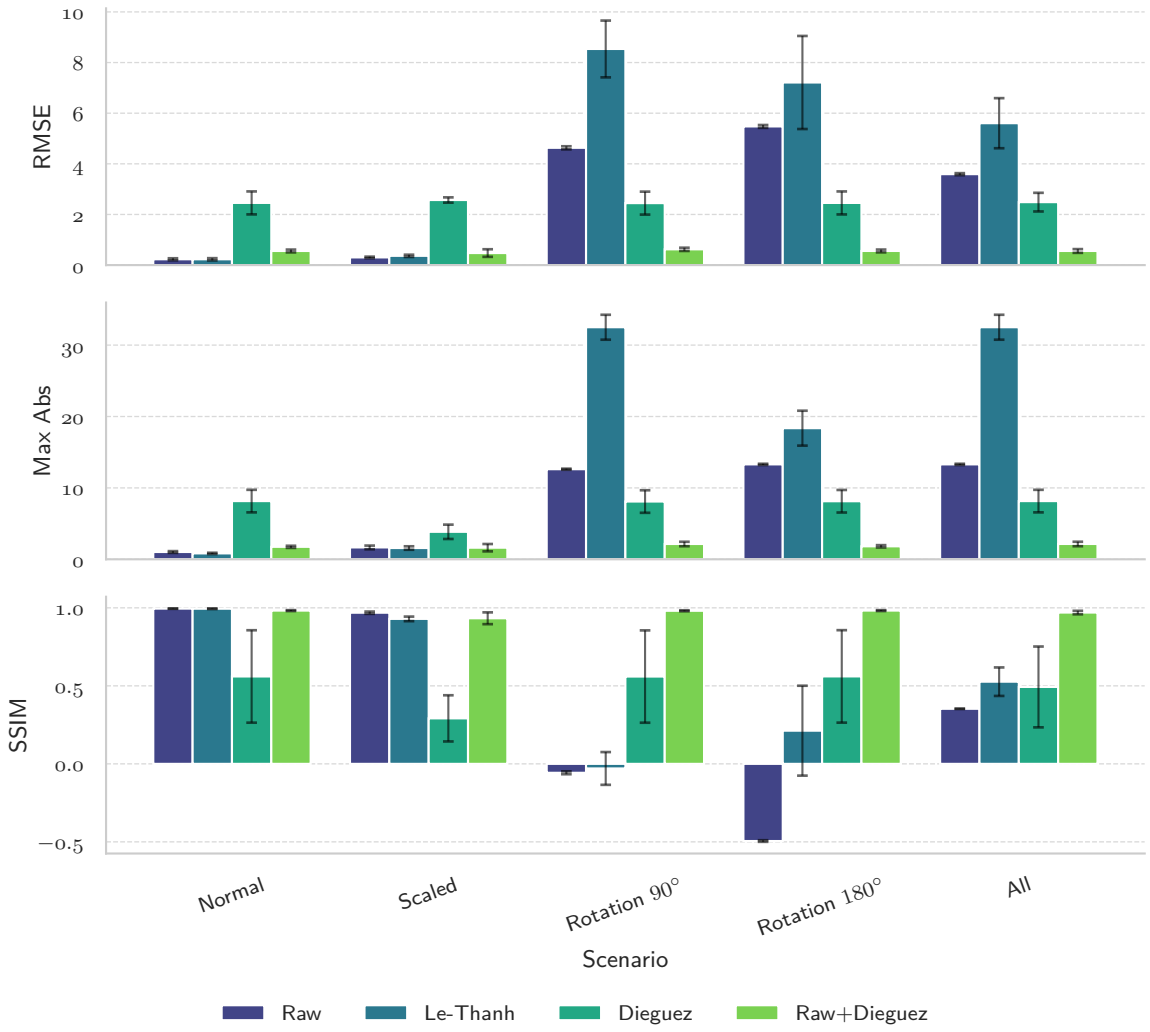


Figure 5.5: Benchmark feature set performance across transformations. Bars indicate mean over five random seeds; error bars show one standard deviation. Metrics are reported for RMSE, maximum absolute error (Max Abs), and structural similarity (SSIM).

5.1.2. Homogeneous graphs: Stage 1 coarse pruning

Building on the refined feature basis established in the benchmark screening, the next step applies the BO framework to systematically explore how different combinations of these descriptors affect predictive performance within graph-based architectures. In this first stage, the analysis focuses on the homogeneous formulation, using the physically grounded *Dieguez*-derived and global geometric features as the initial search space for the node attributes and the edge features as described in Section 4.3.2. The objective is to identify which subsets of these features jointly maximise accuracy and spatial fidelity while maintaining robustness under geometric transformations.

The first step of this process is to identify the set of feature masks that lie on the Pareto front. By jointly considering *RMSE*, *Max Abs* and *SSIM*, a total of 21 Pareto-optimal masks were selected (Fig. 5.6). From this point onward, all analyses are restricted to these Pareto-optimal configurations, which represent the most efficient trade-offs among accuracy, stability, and structural similarity. A complete overview of the selected masks is provided in Tab. D.2 in Appendix D.2. All results are retrained across five seeds, following the BO procedure outlined in Sec. 4.5.

The frequency of feature occurrence across the Pareto masks is shown in Fig. 5.7. Several features are never included in the best-performing masks and can therefore be safely discarded for the next phase: *Euclidean distance*, *squared distance*, *Δ position*, and *unit direction*. Their absence across all

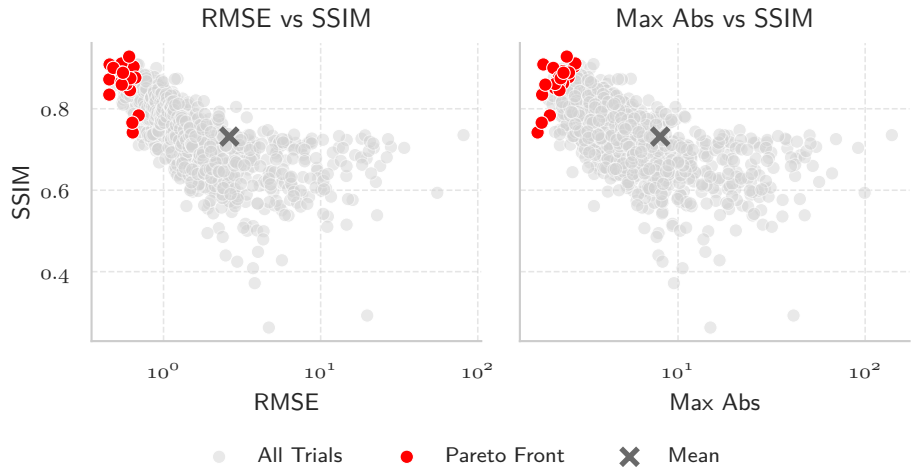


Figure 5.6: Multi-objective optimization results for Phase 1 (homogeneous feature ablation). The plots show the trade-off between predictive accuracy (RMSE and MaxAbs; log-scaled) and structural similarity (SSIM). Grey points correspond to all evaluated feature subsets, while red points highlight the Pareto-optimal fronts, representing feature combinations that achieve the best trade-off across objectives.

Pareto-optimal solutions indicates that they provide no unique contribution to performance. In addition, features such as *aspect ratio*, *angle relative to window normal*, *horizontal distance to window*, *width*, and *average solid angle* appear only infrequently. To determine whether these should be retained, their direct contributions to performance were further examined.

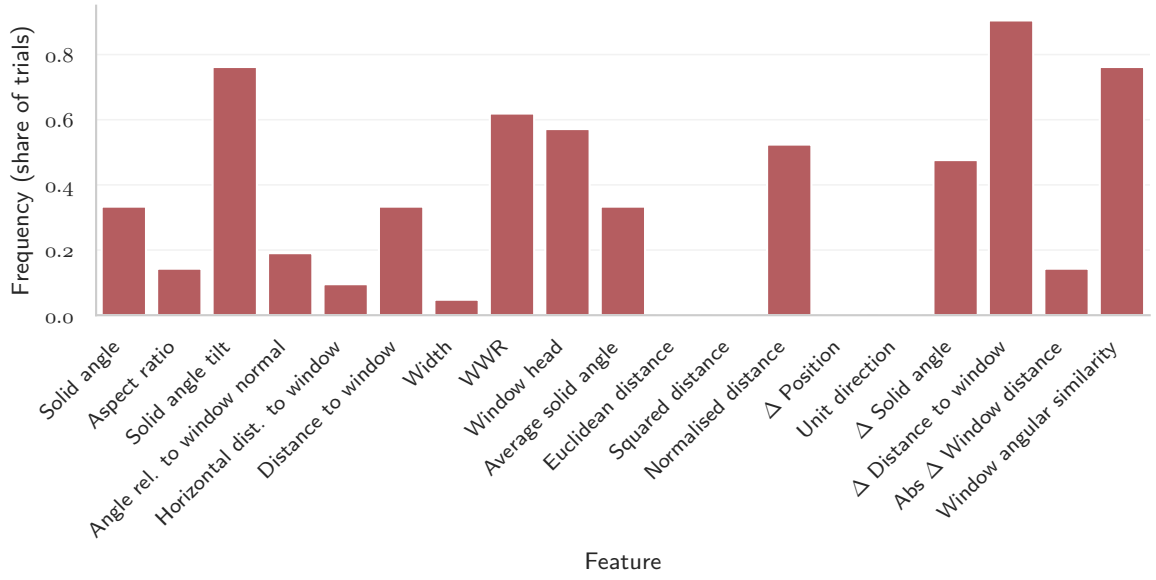


Figure 5.7: Phase 1 feature selection frequency in the Pareto front for the homogeneous graph model. Bars indicate the share of trials in which each feature was included in a Pareto-optimal configuration.

Figure 5.8 summarises the effect of each feature on the evaluation metrics. Three features, *aspect ratio*, *horizontal distance to window*, and *average solid angle*, consistently degrade performance across all three metrics, which explains their rare selection in the Pareto masks. These features are therefore excluded from Phase 2. Other features with mixed or weak contributions were retained for further testing in Stage 2, as their role could not be ruled out solely on this basis.

Finally, redundancy analyses were conducted to further guide feature selection. Pairwise synergy values (Fig. D.7) and Cramér's V redundancy scores (Fig. D.6) reveal clear patterns of detrimental overlap. *Aspect ratio* shows strong negative synergy with *window head*, reinforcing the decision to exclude it. The same applies to *average solid angle*, which exhibits negative synergy with both *normalised dis-*

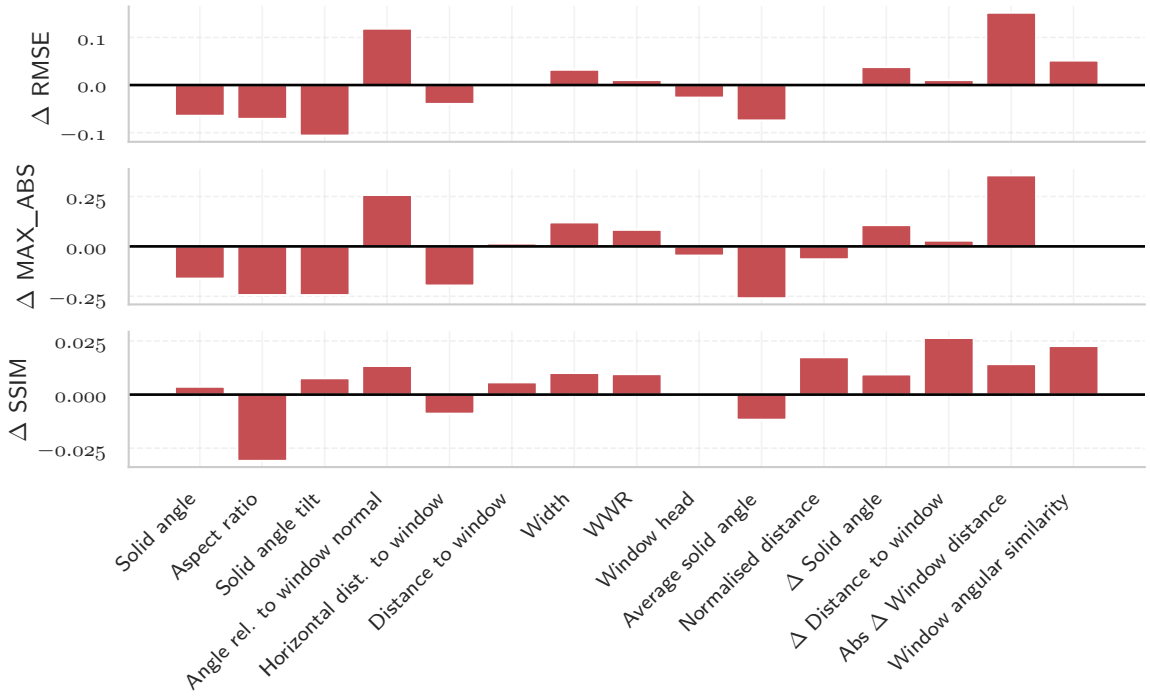


Figure 5.8: Phase 1 feature ablation results for the homogeneous graph model. Bars indicate the contribution of each feature to RMSE, Max Abs, and SSIM (*positive values indicate improvement*).

tance and *angular similarity*, further confirming that its exclusion based on contribution analysis was a well-founded choice. *Solid angle* exhibits negative synergy with *normalised distance*, the only distance-based edge feature consistently preserved in Pareto masks; to avoid redundancy and maintain the more informative descriptor, *solid angle* is excluded. Finally, *horizontal distance to window* negatively affects multiple metrics and shows strong detrimental synergy with *solid angle tilt*. While this interaction does not conclusively indicate which feature drives the redundancy, the consistently poor contribution of *horizontal distance* provides sufficient grounds to exclude it, while retaining *solid angle tilt* for further evaluation. The selected synergy values summarised in Tab. 5.2 accentuate these decisions, highlighting the consistent negative interactions of *aspect ratio*, *average solid angle*, and *horizontal distance to window*.

Table 5.2: Selected synergy values for feature redundancy analysis. Negative values indicate detrimental or redundant combinations.

Feature 1	Feature 2	ΔRMSE	ΔMax Abs	ΔSSIM
Horizontal distance	Solid angle tilt	-0.246	-0.455	-0.032
Average solid angle	Normalised distance	-0.177	-0.199	-0.012
Aspect ratio	Window head	-0.152	-0.351	–
Solid angle	Normalised distance	-0.110	-0.090	+0.007
Average solid angle	Angular similarity	-0.106	-0.131	+0.005

Conclusion. Stage 1 results in the exclusion of *Euclidean distance*, *squared distance*, *Δ position*, *unit direction*, *aspect ratio*, *horizontal distance to window*, *solid angle*, and *average solid angle*. The refined set of retained features is carried forward into Stage 2, where masks are systematically compared to identify the most effective combinations for the homogeneous model.

5.1.3. Homogeneous graphs: Stage 2 mask selection

BO was repeated on the reduced pool from Stage 1 to resolve interactions among the remaining descriptors and determine the final mask. The optimisation results are shown in Figure 5.9. When comparing to Figure 5.6 which shows the trials in Phase 1, it can be seen that for both *RMSE* and Max Abs the mean and outliers are better performing, while the outliers for *SSIM* are somewhat higher.

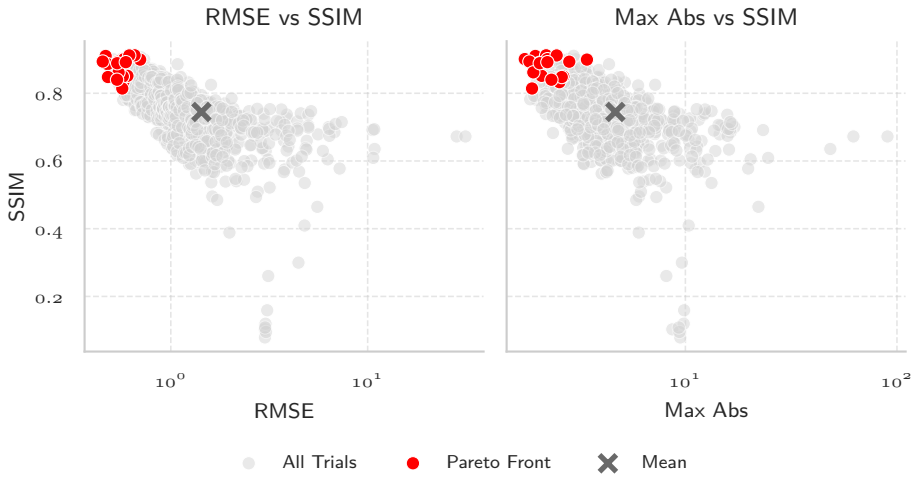


Figure 5.9: BO trials in metric space for Phase 2 homogeneous descriptors. Grey points denote all evaluated masks; red points indicate Pareto-optimal solutions. The dark grey cross marks the mean of all trials. Left: RMSE vs. SSIM. Right: MaxAbs vs. SSIM.

The summary statistics in Table 5.3 provide a clearer view of these differences. Across all trials, Phase 2 reduces the mean *RMSE* from 2.624 ± 4.684 to 1.431 ± 1.709 and the mean Max Abs from 8.027 ± 10.184 to 4.667 ± 4.500, while maintaining a comparable mean *SSIM* (0.732 ± 0.091 versus 0.745 ± 0.098). Considering only Pareto-optimal solutions, the changes are more modest: *RMSE* and Max Abs remain at a similarly low level, while *SSIM* shows a slight improvement. This suggests that the main benefit of Stage 2 lies in stabilising performance across the trial population and reinforcing the Pareto front already identified in Stage 1, by reducing the search space and producing similar solutions with less variability. From this point onward, the analysis therefore focuses exclusively on Pareto-optimal solutions, as these represent the most relevant configurations for final mask selection.

Table 5.3: Mean $\pm$ standard deviation of performance metrics for Phase 1 and Phase 2, comparing all trials with Pareto front solutions.

	Phase 1		Phase 2	
Metric	All	Pareto Front	All	Pareto Front
RMSE	2.624 $\pm$ 4.684	0.575 $\pm$ 0.069	1.431 $\pm$ 1.709	0.559 $\pm$ 0.061
Max Abs	8.027 $\pm$ 10.184	2.284 $\pm$ 0.295	4.667 $\pm$ 4.500	2.258 $\pm$ 0.404
SSIM	0.732 $\pm$ 0.091	0.863 $\pm$ 0.047	0.745 $\pm$ 0.098	0.878 $\pm$ 0.029

Feature-level analysis

Within the Pareto front, feature frequency and contribution analyses were performed to understand which descriptors drive performance. Despite the overall metrics being highly similar between Phase 1 and Phase 2, the underlying selection patterns reveal both continuity and subtle shifts. As illustrated in Figure 5.7 and Figure 5.10, several descriptors appear consistently across both stages: *Angle tilt*, WWR, *Window head height*, Δ *Distance to window*, and *Window angular similarity* are frequently chosen, while $|\Delta|$ *Window distance* is almost never selected. Differences emerge in the relative prominence of other features. In Phase 2 (Figure 5.10), the frequency of *Angle tilt* decreases, Δ *Solid angle* becomes more prominent, and *Normalised distance* is selected far less often. These results indicate that the core set of influential descriptors remains stable across optimisation stages, but the reduced pool in Phase 2 reshapes their relative importance, favouring physically grounded measures such as Δ *Solid angle* over more derived geometric ratios.

Across the Pareto-optimal configurations, no single descriptor exhibits a consistently positive effect across all evaluation metrics. Instead, the results indicate that performance improvements in Phase 2 arise from complementary interactions between multiple descriptors rather than from the isolated contribution of any individual feature. This confirms that the optimisation process has reached a regime

where the relative importance of single features diminishes in favour of collective synergy among the remaining inputs (see Fig. D.8 in Appendix D.2.1 for reference).

Given this interdependence, the analysis proceeds to the mask level, where entire feature combinations are compared directly. This shift from feature-wise to mask-wise evaluation enables a more holistic assessment of how the surviving descriptors interact within complete configurations, ultimately guiding the selection of the final homogeneous feature mask.

Figure 5.10: Selection frequency of homogeneous phase 2 features within the Pareto-optimal subset. Values indicate the proportion of masks in which a feature is active.

Mask-Level analysis

The 17 Pareto-optimal masks were ranked using the composite and evenly weighted scoring formulations defined in Equation 4.10 and Equation 4.11. In the composite score, *RMSE*, *MaxAbs*, and *SSIM* were assigned weighting factors of 0.55, 0.30, and 0.15, respectively, while the even score applied equal weighting to all three metrics. The resulting ranking, summarised in Table D.3, was used to identify the most promising configurations. From this ranked list, the top nine masks were selected for further evaluation. This number was not chosen arbitrarily: it ensures that if the lowest-ranked mask within the selection were to outperform higher-ranked candidates during retraining, several additional configurations would still be available to verify ranking stability. The corresponding feature compositions of the nine shortlisted masks are presented in Table 5.4.

The selection frequency of these nine masks (Figure 5.11) closely matches that of the complete Pareto set (Figure 5.10), confirming that the subset is representative of the broader optimisation landscape. The nine shortlisted masks were subsequently retrained across five random seeds to assess their robustness and determine which configuration yields the most consistent performance.

Table 5.4: Feature inclusion across the shortlisted homogeneous masks. A checkmark indicates that the feature is active in the corresponding mask, while a cross indicates exclusion.

Feature	Mask 0	Mask 1	Mask 2	Mask 3	Mask 4	Mask 5	Mask 6	Mask 7	Mask 8
Angle tilt	✓		✓	✓		✓	✓		✓
Angle to norm.	✓	✓			✓	✓	✓	✓	
Dist. to win.		✓	✓	✓	✓			✓	✓
WWR	✓	✓			✓	✓	✓	✓	✓
Win. head			✓	✓		✓		✓	
Norm. dist.	✓	✓	✓		✓	✓	✓		
Δ Solid angle		✓		✓		✓	✓	✓	✓
Δ Dist. to win.	✓		✓		✓	✓	✓		✓
|Δ| Win. dist.	✓	✓	✓	✓			✓		
Win. ang. sim.		✓		✓	✓	✓	✓		

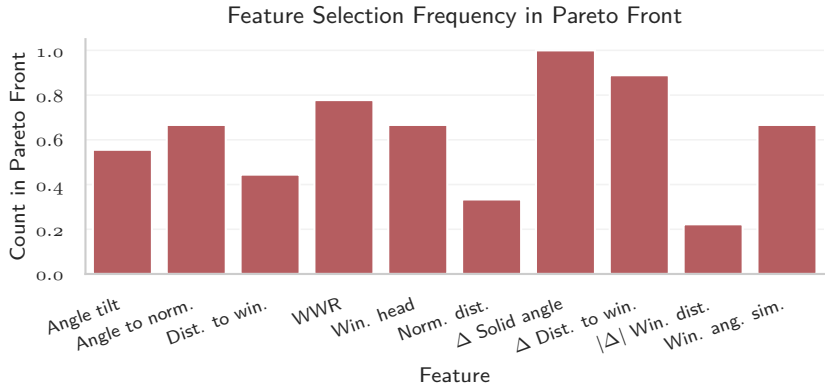


Figure 5.11: Selection frequency of homogeneous phase 2 features across the re-evaluated top nine masks. Frequencies are computed after retraining with five seeds, controlling for stochastic effects.

The mask-level comparison (Figure 5.12 and Table 5.5) demonstrates that the shortlisted configurations exhibit highly consistent behaviour across all geometric transformations, with only marginal variability in performance. The top-ranked masks (0, 1, 2, 5, and 8) all achieve comparably low *RMSE* and *MaxAbs* values while preserving strong structural similarity, indicating that the optimisation process has converged toward a family of equally robust solutions rather than a single dominant configuration. Among these, Mask 5 attains the best overall rank score and the lowest average *RMSE*, whereas Masks 0 and 1 display slightly higher stability across transformations. Mask 2 achieves strong performance in the normal and scaled cases but shows greater variance under rotation, while Mask 8 performs consistently but with a small accuracy trade-off. Together, these results confirm that the search space has entered a regime of diminishing returns, where alternative combinations of the same core descriptors yield near-equivalent outcomes.

The close clustering of these masks highlights the redundancy and complementarity within the refined feature pool: performance depends less on the inclusion of any single descriptor and more on balanced, physically coherent combinations. Mask 5 was therefore selected as the representative homogeneous configuration for subsequent experiments, not because it was uniquely superior, but because it offered a stable and interpretable balance between accuracy, robustness, and feature diversity. Its performance consolidates the homogeneous feature design, forming a reliable baseline for the heterogeneous formulation examined in the following section.

Table 5.5: Comparison of the nine homogeneous masks on robustness criteria. Minimax values report the worst case across transformations. CVaR-2 is the mean of the two largest *RMSE* values. Rank score is the composite metric defined in Section 4.5.

Mask ID	Worst RMSE	Worst MaxAbs	Worst SSIM loss	CVaR-2 RMSE	Avg. RMSE	Rank score
5	0.803	2.655	0.366	0.679	0.616	1.0
2	1.006	4.670	0.531	0.774	0.657	3.0
0	1.116	4.542	0.489	0.795	0.633	3.2
1	1.075	4.744	0.381	0.812	0.678	3.6
8	1.087	3.813	0.572	0.862	0.747	4.6
7	1.195	5.332	0.537	1.006	0.910	6.2
6	1.426	5.334	0.546	1.067	0.887	6.8
3	1.896	5.984	0.731	1.166	0.800	7.8
4	3.605	10.127	0.583	2.115	1.369	8.8

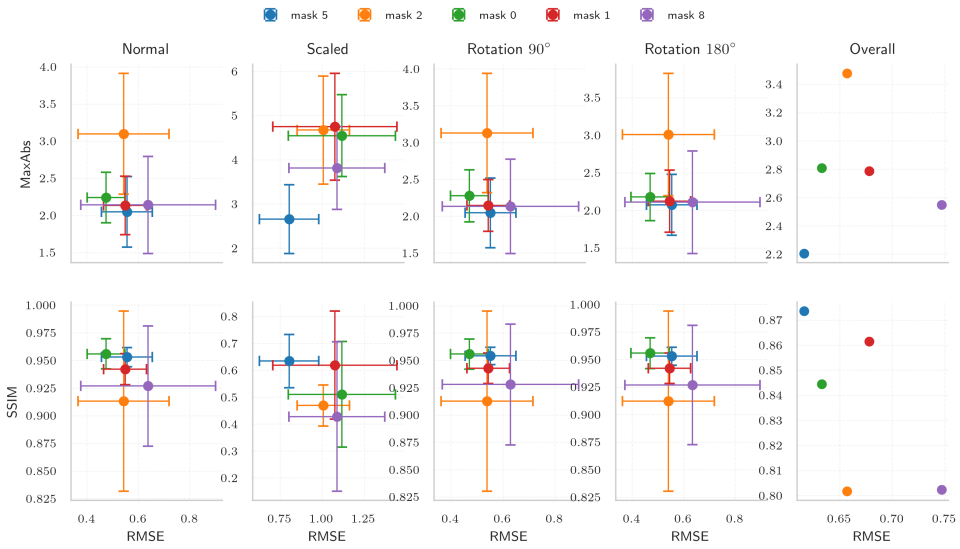


Figure 5.12: Performance of the five best performing homogeneous masks under each transformation scenario. Markers denote across-seed means; error bars show variance across seeds. The rightmost panels summarise overall aggregates used in the ranking procedure.

5.1.4. Selection of the heterogeneous feature mask

The selection of the heterogeneous feature mask followed a single-stage optimisation procedure tailored to the smaller and more uniform nature of the edge feature space. Given that the search space comprised only twelve descriptors, a single BO run was sufficient to explore all relevant combinations without requiring multi-phase pruning. The objective was to identify a compact yet expressive subset of edge features that jointly maximised predictive accuracy and robustness under geometric transformations, while maintaining interpretability and avoiding redundancy.

When compared with the homogeneous Phase 2 results (Figure 5.9), the Pareto front of the heterogeneous model (Figure 5.13) exhibits a substantially tighter distribution with lower variance and improved mean performance across all metrics (see Table 5.6). Both *RMSE* and *MaxAbs* values decrease considerably, while *SSIM* approaches unity for the Pareto-optimal configurations, indicating stronger structural fidelity. These results confirm that the heterogeneous formulation captures geometric relations more effectively and demonstrates higher data efficiency, making it a promising configuration for the subsequent operator benchmarking phase. The full set of Pareto-optimal configurations retained for comparison is reported in Table D.4.

Table 5.6: Mean $\pm$ standard deviation of performance metrics for the homogeneous Phase 2 and heterogeneous feature-ablation models, comparing all trials with Pareto front solutions.

	Homogeneous Phase 2		Heterogeneous	
Metric	All	Pareto Front	All	Pareto Front
RMSE	1.431 ± 1.709	0.559 ± 0.061	0.477 ± 0.294	0.186 ± 0.017
Max Abs	4.667 ± 4.500	2.258 ± 0.404	1.907 ± 0.818	0.944 ± 0.182
SSIM	0.745 ± 0.098	0.878 ± 0.029	0.927 ± 0.067	0.982 ± 0.004

Feature-level analysis

Figure 5.14 summarises the selection frequency of the twelve heterogeneous edge descriptors across all Pareto-optimal configurations. Two angular terms, $\cos^2\theta_x$ and $\cos^2\theta_y$, were never selected, indicating a lack of predictive relevance. Distance-based features such as *Euclidean distance* and *normalised distance* were retained in nearly all configurations, confirming their dominant role in predictive accuracy. *Squared distance* appeared regularly but did not improve performance, suggesting redundancy with the other distance encodings. *Scaled 3D distance* and *scaled vertical separation* contributed only under

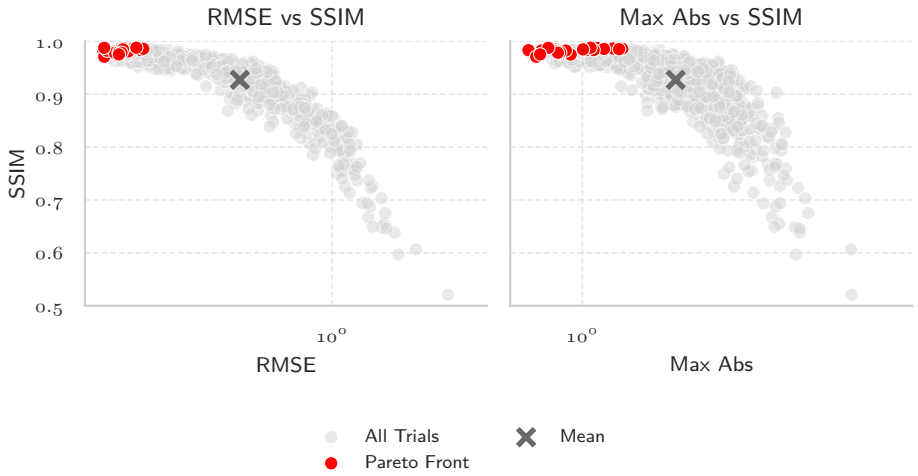


Figure 5.13: Comparison of mean ± standard deviation for homogeneous Phase 2 and heterogeneous configurations, showing substantial performance gains in both accuracy (RMSE, MaxAbs) and structural similarity (SSIM) for the heterogeneous formulation.

specific transformation conditions, while angular relations such as $\vec{d}\cdot\vec{t}$ and $\vec{d}\cdot\vec{n}$ provided complementary geometric cues. In contrast, $\vec{d}\cdot\vec{u}$ exhibited weaker and less consistent utility.

Overall, these findings indicate that the heterogeneous model derives most of its predictive strength from a compact subset of physically grounded edge descriptors, primarily distance measures and a limited number of angular relations, while scaled and squared variants contribute marginally. The corresponding contribution analysis, presented in Appendix D.2.2, corroborates these trends and provides a complementary perspective on the relative importance of individual features.

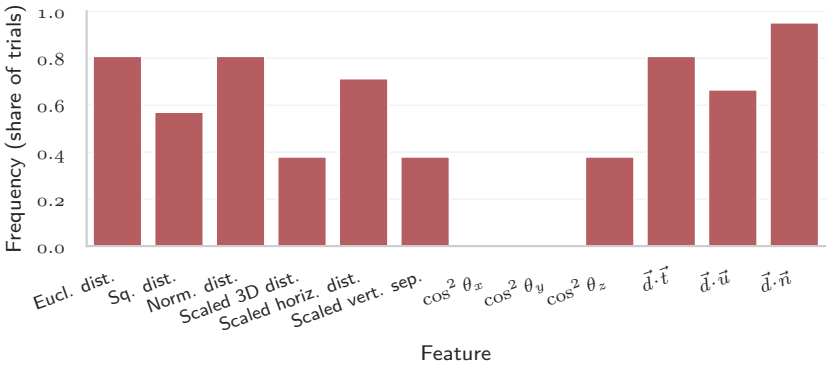


Figure 5.14: Selection frequency of heterogeneous edge features within the Pareto-optimal subset. Values indicate the proportion of masks in which a feature is active.

Mask-Level analysis

The nine shortlisted Pareto-optimal masks were re-evaluated across five random seeds to assess robustness and ensure stability under geometric transformations. This validation step provides a consistent basis for comparing configurations and identifying those that maintain reliable performance across normal, scaled, and rotated conditions. The feature composition of the selected masks is summarised in Table 5.7, with the corresponding selection frequencies shown in Figure 5.15. Several descriptors occur in all masks, namely *Euclidean distance*, *normalised distance*, $\cos^2\theta_z$, and $\vec{d}\cdot\vec{u}$. As these features are always included, their marginal effects cannot be isolated and they are excluded from the contribution analysis. In contrast, *scaled 3D distance* and *scaled vertical separation* appear infrequently, indicating limited relevance for robust configurations as mentioned before.

Following the re-evaluation of the shortlisted Pareto-optimal masks, a feature-level analysis was conducted to examine which edge descriptors contributed most consistently across the retrained config-

Table 5.7: Feature inclusion across the nine shortlisted heterogeneous masks. A checkmark indicates that the feature is active in the corresponding mask, while a cross indicates exclusion. The angular descriptors $\cos^2\theta_x$ and $\cos^2\theta_y$ are omitted, as they were not selected in any of the shortlisted configurations.

Feature	Mask 0	Mask 1	Mask 2	Mask 3	Mask 4	Mask 5	Mask 6	Mask 7	Mask 8
Eucl. dist.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sq. dist.	✓	✓	✓	✓	✓		✓		
Norm. dist.	✓	✓	✓	✓	✓	✓	✓		✓
Scaled 3D dist.	✓				✓				
Scaled horiz. dist.		✓	✓	✓	✓	✓	✓	✓	
Scaled vert. sep.	✓			✓					
$\cos^2\theta_z$	✓	✓	✓	✓	✓	✓	✓	✓	
$\vec{d}\cdot\vec{t}$	✓			✓		✓	✓	✓	✓
$\vec{d}\cdot\vec{u}$	✓	✓	✓	✓	✓	✓		✓	✓
$\vec{d}\cdot\vec{n}$	✓	✓		✓	✓	✓	✓	✓	✓

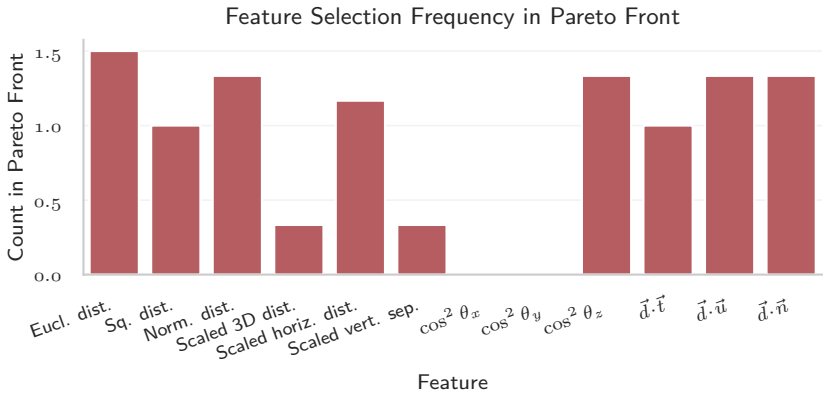


Figure 5.15: Selection frequency of heterogeneous edge features across the re-evaluated top nine masks. Frequencies are computed after retraining with five seeds, controlling for stochastic effects.

urations. Distance-based features, particularly *Euclidean* and *normalised distance*, were retained in nearly all masks, reaffirming their central role in encoding geometric relations between windows and sensors. *Squared distance*, by contrast, remained frequently included but exhibited mixed effects, suggesting partial redundancy with other distance metrics. Among the angular descriptors, $\vec{d}\cdot\vec{t}$ and $\vec{d}\cdot\vec{n}$ provided stable directional information, while scaled and vertical distance terms showed more variable, transformation-dependent behaviour.

A detailed per-transformation contribution analysis of these re-evaluated masks is presented in Appendix D.2.2. It demonstrates that distance-based relations dominate predictive performance, whereas scaled and angular descriptors contribute only under specific transformation scenarios. Since these trends do not alter the relative feature ranking or the final mask selection, the full plots are reported in the appendix for completeness.

Building on the feature-level analyses of the re-evaluated models, the nine shortlisted masks were next compared directly at the mask level. Robustness was assessed using the criteria defined in Section 4.5.

Table 5.8 further quantifies the robustness of each mask. Mask 5 exhibits the most consistent performance across all criteria, achieving the second lowest worst-case *RMSE* (0.518), the second-lowest CVaR-2 *RMSE* (0.346), and the best average *RMSE* (0.251). Its composite rank score of 2.0 reflects this stability, indicating strong and well-balanced behaviour rather than dominance in a single metric.

Figure 5.16 complements these tables by visualising mask-level performance across all transformations. Each panel shows *RMSE* versus *MaxAbs* or *SSIM*, with markers denoting across-seed means and error bars reflecting variance. The plots illustrate that Mask 5 not only attains strong average accuracy but also maintains stable performance across normal, scaled, and rotated configurations. Masks 2 and 4 show competitive average behaviour but are more variable under scaling and rotation, consistent with their weaker robustness scores.

Taken together, these results converge on Mask 5 as the most robust and reliable heterogeneous

Table 5.8: Comparison of the nine heterogeneous masks on robustness criteria. Minimax values report the worst case across transformations. CVaR-2 is the mean of the two largest RMSE values. Rank score is the composite metric defined in Section 4.5.

Mask ID	Worst RMSE	Worst MaxAbs	Worst SSIM loss	CVaR-2 RMSE	Avg. RMSE	Rank score
5	0.518	2.500	0.087	0.346	0.251	2.0
7	0.510	2.413	0.112	0.348	0.266	2.6
6	0.532	2.741	0.105	0.345	0.251	2.6
2	0.522	2.606	0.112	0.362	0.281	4.2
3	0.628	2.728	0.156	0.381	0.257	6.0
8	0.561	2.797	0.139	0.367	0.269	6.2
4	0.625	3.196	0.124	0.386	0.266	6.6
0	0.659	3.109	0.129	0.390	0.254	6.8
1	0.648	2.971	0.133	0.407	0.285	8.0

feature configuration. This mask combines a compact edge dimensionality (seven active features) with consistently strong minimax and CVaR-2 performance, making it the preferred choice for subsequent operator evaluation.

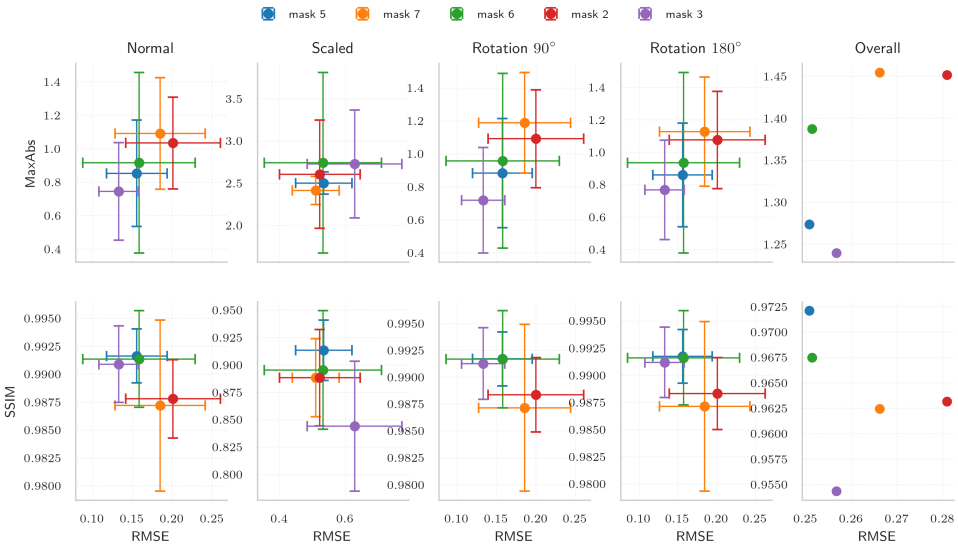


Figure 5.16: Performance of the nine shortlisted heterogeneous masks under each transformation scenario. Markers denote across-seed means; error bars show variance across seeds. The rightmost panels summarise overall aggregates used in the ranking procedure.

5.1.5. Summary and transition to architecture optimisation

The feature ablation phase established the final sets of node and edge descriptors used in the subsequent model architecture optimisation. Through a systematic combination of BO, Pareto analysis, and multi-metric evaluation, both the homogeneous and heterogeneous formulations converged on compact, physically interpretable representations that balance predictive accuracy, generalisation, and computational efficiency.

For the homogeneous formulation, the final node–edge feature configuration comprises: *Tilted angle*, *angle relative to window normal*, *window-to-wall ratio (WWR)*, *window head height*, *normalised distance*, Δ*solid angle*, Δ*distance to window*, and *window angular similarity*. This combination integrates geometric and directional cues that collectively describe the spatial relationship between sensors and openings, while maintaining transformation robustness through the inclusion of relative (difference-based) terms.

For the heterogeneous formulation, the final edge descriptor set includes: *Euclidean distance*, *normalised distance*, *scaled horizontal distance*, $\cos^2\theta_z$, and the three directional dot-products ($\vec{d}\cdot\vec{t}$, $\vec{d}\cdot\vec{u}$, $\vec{d}\cdot\vec{n}$). These features capture the core geometric dependencies between window and sensor nodes, combining distance-based and angular relationships in a manner that preserves physical interpretability and allows the model to generalise across scale and rotation.

A direct comparison between the final homogeneous and heterogeneous configurations, shown in Figure 5.17, reveals that the heterogeneous model consistently outperforms its homogeneous counterpart across all transformation scenarios. Equivalent plots for *MaxAbs* and *SSIM*, provided in Appendix D.2.3, exhibit the same trend, confirming that the observed improvement is consistent across all evaluation metrics. This performance gap raises an important question about the underlying cause. It may stem partly from architectural differences; specifically, the heterogeneous formulation employs a multi-layer MLP readout, whereas the homogeneous variant uses only a single linear output layer. However, the difference could also reflect a more fundamental distinction in how each model represents daylight transport. Whereas the homogeneous model primarily encodes local spatial gradients within the sensor grid, the heterogeneous formulation explicitly models directional energy transfer between windows and sensors, arguably aligning more closely with the underlying physics of daylight propagation. This distinction, however, cannot be conclusively attributed to architecture or feature structure alone at this stage; it will be revisited in the following phase once model optimisation results become available.

Together, these results complete the feature selection and ablation phase. The two configurations represent complementary strategies for encoding daylight-relevant geometry: the homogeneous model emphasises intra-domain spatial relations, while the heterogeneous model captures inter-domain directional dependencies. Both demonstrate stable performance under geometric transformations and provide an expressive yet parsimonious basis for subsequent learning.

The next stage of the study therefore shifts from feature representation to architecture optimisation, where the selected feature sets are embedded within candidate GNN operators and network configurations. This transition marks the move from input design to representational learning, enabling systematic benchmarking of message-passing operators and their capacity to exploit the geometric information encoded by the chosen features.

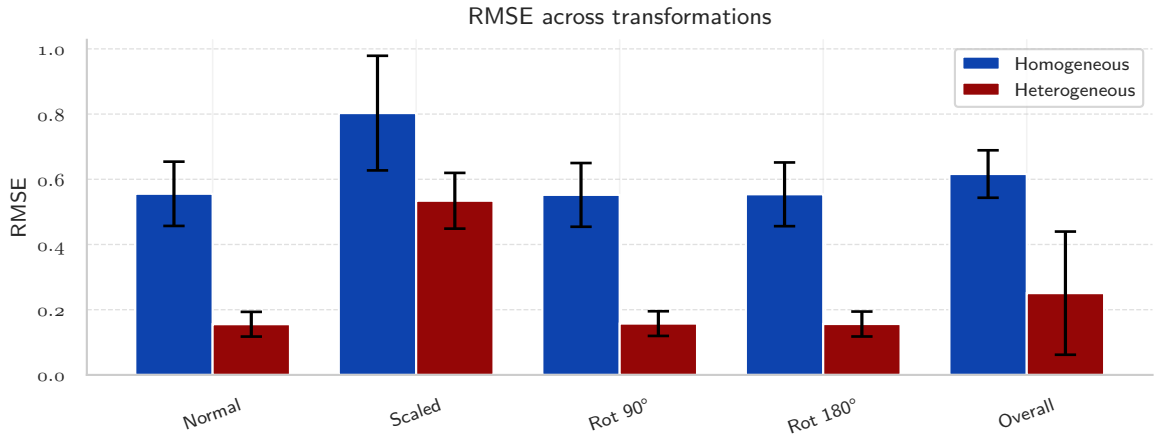


Figure 5.17: *RMSE* for the final homogeneous and heterogeneous feature configurations across transformations (Normal, Scaled, Rot 90°, Rot 180°, and Overall). Bars show mean values; error bars indicate one standard deviation across seeds.

5.2. Operator Optimisation

Following the feature ablation phase described in Section 4.6.3, a series of BO studies were conducted to identify performant and stable hyperparameter configurations for each operator. To ensure reproducible selection of final configurations, the top twenty trials per operator were selected using a self-defined Pareto ranking function that jointly minimised validation *RMSE* and model complexity.

Figure 5.18 shows the distribution of mean *RMSE* across hyperparameter choices for the heterogeneous GENConv operator. The red points indicate the Pareto-selected configurations, which cluster near the lower envelope of the *RMSE* distribution, confirming the consistency of the ranking criterion. Clear trends are observable across several parameters, such as a preference for `layer` normalisation, larger hidden sizes (96 or 128), and lower residual activation frequency (with `Residual=False` being slightly favoured).

As a second example, Figure 5.19 shows the corresponding violin plots for the homogeneous SplineConv

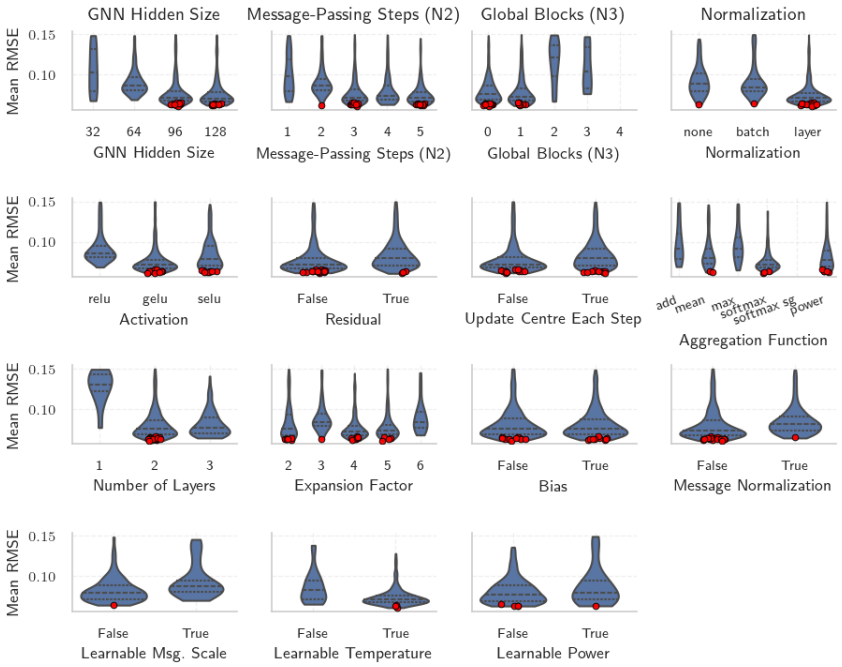


Figure 5.18: Distribution of mean *RMSE* across hyperparameter values for the heterogeneous GENConv operator. Red markers indicate Pareto-selected configurations. The clustering of red points near the minimum *RMSE* suggests that the Pareto ranking consistently identifies high-performing trials.

operator, which exhibited slightly broader variance across trials and less sensitivity to the choice of normalisation or activation. This illustrates the diverse response surfaces across operators and supports the inclusion of multiple operator families in the benchmark. All other violin plots are provided in Appendix D.3.1.

5.2.1. Global Hyperparameters

Table D.5 summarises the optimised shared hyperparameters obtained from the BO across all operators. Across both homogeneous and heterogeneous regimes, the search consistently converged towards configurations with negligible dropout, low weight decay ($< 10^{-5}$), and `gelu` activation functions. Layer normalisation was strongly preferred in heterogeneous models, while homogeneous configurations often performed equivalently without explicit normalisation.

Table 5.9: Optimised shared hyperparameters for homogeneous and heterogeneous GNN operators.

Graph type	Operator	n	N_1	N_2	N_3	Norm.	Act.	Res.	U.C.	Dropout	LR	Weight decay
Homo.	NNConv	48	2	5	2	layer	relu	True	–	0.0	4.23×10^{-3}	3.21×10^{-6}
Homo.	GENConv	128	2	5	1	layer	gelu	True	–	0.0	6.45×10^{-4}	5.24×10^{-6}
Homo.	SplineConv	96	3	5	2	none	gelu	True	–	0.0	5.60×10^{-4}	1.03×10^{-5}
Homo.	PNAConv	128	3	5	4	none	gelu	True	–	0.0	2.71×10^{-3}	1.54×10^{-6}
Homo.	GCN+	128	0	4	4	none	gelu	True	–	0.0	2.38×10^{-3}	1.68×10^{-6}
Hetero.	NNConv	48	–	1	2	layer	gelu	True	False	0.0	4.83×10^{-4}	3.2×10^{-6}
Hetero.	GENConv	96	–	3	1	layer	gelu	False	False	0.0	5.02×10^{-4}	3.0×10^{-6}
Hetero.	SplineConv	128	–	1	2	none	gelu	True	False	0.0	1.17×10^{-3}	1.58×10^{-6}
Hetero.	PNAConv	128	–	5	1	layer	gelu	True	True	0.25	1.36×10^{-3}	2.66×10^{-6}

The optimiser consistently pushed the models towards the upper limits of representational depth and width. The hidden dimension n reached the maximum of its range in almost all cases, and the message-passing depth N_2 attained its upper bound for every homogeneous operator (four–five layers) while varying more widely in the heterogeneous regime (one–three for most operators, five for PNAConv). This pattern aligns with the structural differences between graph types: homogeneous graphs lack explicit window–sensor separation and therefore require deeper propagation to integrate spatial dependencies. Heterogeneous graphs achieve comparable information exchange in fewer steps because

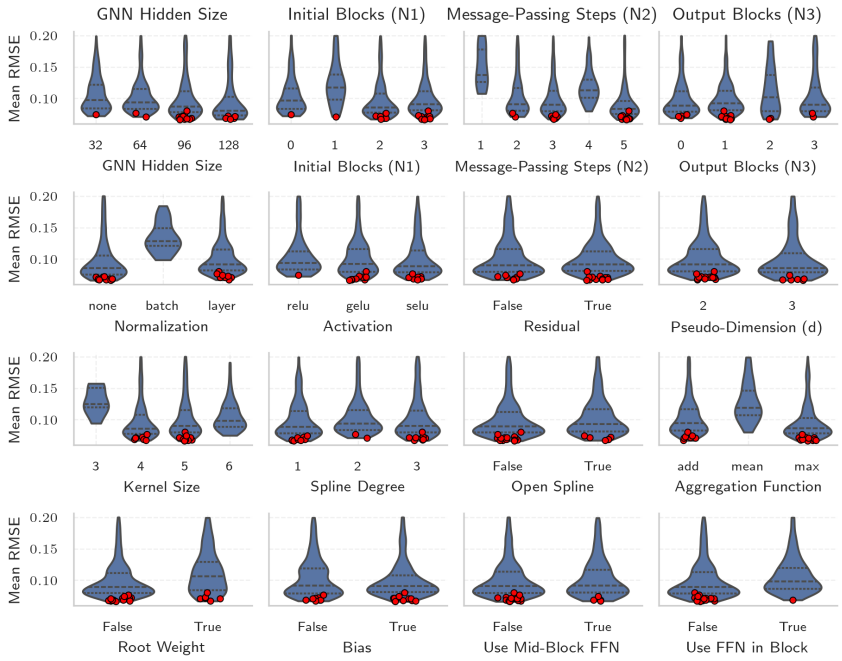


Figure 5.19: Distribution of mean *RMSE* across hyperparameter values for the homogeneous SplineConv operator. Red markers denote Pareto-selected configurations. The broader variance compared to GENConv reflects the higher sensitivity of SplineConv to edge-geometry parameterisation.

their bipartite topology provides direct cross-type links [59, 140]. Homogeneous models generally favour deeper output heads (up to four layers for PNAConv and GCN+), while heterogeneous models converge to shallower heads (one–two layers). This indicates that global nonlinear mixing is more beneficial when all nodes share a single feature space, whereas heterogeneous networks already encode type-specific structure that reduces the need for extensive post-propagation transformation.

GELU was preferred in nearly all optimised configurations, with homogeneous NNConv selecting ReLU. This agrees with prior findings that smoother activations such as GELU promote stable optimisation and gradient flow in deeper networks compared to hard-threshold ReLU [147, 154, 155]. Normalisation preferences separate clearly by regime: heterogeneous winners favour LayerNorm (except SplineConv, which performs well without it), while several homogeneous models omit normalisation entirely. This is consistent with evidence that LayerNorm improves stability when node embeddings originate from heterogeneous distributions [140, 145].

Residual pathways are enabled in nearly all cases (heterogeneous GENConv being the exception), confirming their stabilising effect on deep message passing [58, 138]. Dropout is optimised to zero everywhere except for heterogeneous PNAConv (0.25), suggesting that most architectures already achieve sufficient regularisation through architectural constraints and early stopping. Learning rates cluster tightly between 5×10^{-4} and 3×10^{-3} (with one outlier at 4.23×10^{-3}), paired with very small weight decay values ($\leq 10^{-5}$). This combination supports rapid yet stable convergence on the i.i.d. train/validation split, indicating that stronger explicit regularisation was unnecessary within the optimised capacity regime. Because the optimisation did not include transformation-based augmentation, the selected settings do not explicitly promote invariance to rotations, rescalings, or window repositioning. A larger generalisation gap is therefore expected on the transformed test sets, likely more pronounced for homogeneous configurations that rely on deeper message propagation and comparatively smaller, though not negligible, for heterogeneous configurations whose bipartite topology shortens information paths. In the subsequent results, per-transformation changes in *RMSE*, *MaxAbs*, and *SSIM* together with variance across seeds are reported to characterise this effect. For comparability and reproducibility, these optimised hyperparameters are retained for all subsequent retraining and evaluation.

5.2.2. Operator-Specific Hyperparameters

The operator-specific hyperparameters listed in Tables D.6 and D.7 reflect how each message-passing scheme adapts its internal configuration to the graph representation. While the shared parameters govern the overall network structure, these settings determine how messages are formed, weighted, and combined along edges, and therefore reveal the inductive biases of each operator. Clear and systematic differences emerge between the homogeneous and heterogeneous regimes, indicating that the optimiser consistently adapts internal complexity to the relational structure of the graph.

Table 5.10: Optimised operator-specific hyperparameters for heterogeneous GNN operators.

(a) NNConv

Param	Value
Edge-MLP depth	2
Edge-MLP size	16
Aggregation	mean
Root weight	True
Bias	False

(b) GENConv

Param	Value
Aggregation	`softmax_sg`
# Layers	2
Expansion	4
Bias	False
Msg norm	False

(c) SplineConv

Param	Value
d_{pseudo}	3
Kernel size	5
Degree	2
Open spline	False
Aggregation	add
Root weight	True
Bias	False
Mid-block FFN	False
FFN in block	True

(d) PNAConv

Param	Value
Aggregators	`mean_max`
Scalers	`identity_amp`
Towers	4
Pre-MLP layers	2
Post-MLP layers	2
Divide input	False

Table 5.11: Optimised operator-specific hyperparameters for homogeneous GNN operators.

(a) NNConv

Param	Value
Edge-MLP depth	2
Edge-MLP size	24
Aggregation	add
Root weight	False
Bias	False

(b) GENConv

Param	Value
Aggregation	`max`
# Layers	3
Expansion	6
Bias	True
Msg norm	False

(c) SplineConv

Param	Value
d_{pseudo}	2
Kernel size	5
Degree	1
Open spline	False
Aggregation	max
Root weight	False
Bias	True
Mid-block FFN	False
FFN in block	False

(d) PNAConv

Param	Value
Aggregators	`max`
Scalers	`identity att`
Towers	2
# Pre-MLP	1
# Post-MLP	2
Divide input	True

(e) GCNPlus

Param	Value
Use BN	False
Use FFN	True
Norm. ad.	True
Self-loops	False

Across operators, the heterogeneous configurations tend to employ smoother or averaging-style aggregation functions (`mean`, `add`, or `softmax_sg`), shallower internal MLPs, and reduced expansion factors. Bias terms are frequently disabled, and root weights are more often retained to stabilise updates in the presence of degree imbalances between node types. These tendencies suggest that when the bipartite structure already provides a clear directional flow between windows and sensors, less internal transformation depth and softer aggregation suffice to capture illumination relationships. By contrast, homogeneous configurations, where all nodes share a single type and connectivity must implicitly encode inter- and intra-type relations, favour more expressive or selective designs, including deeper internal blocks, harder aggregations (`max`), and higher expansion factors to compensate for the absence of explicit node-type separation.

NNConv. In the homogeneous setting, NNConv adopts an additive aggregation and disables the root weight, while the heterogeneous version uses a mean aggregation with the root weight enabled. The latter configuration smooths messages from window to sensor nodes and retains a self-term for stability, compensating for uneven edge densities in the bipartite graph. Both variants employ a shallow edge-MLP of depth 2, indicating that moderate expressivity is sufficient to parameterise the edge-conditioned filters governing geometric light transfer [90, 91].

GENConv. The heterogeneous configuration selects a softmax-based aggregation (`softmax_sg`) with two layers and an expansion factor of four, whereas the homogeneous counterpart employs a harder `max` aggregation with three layers and an expansion factor of six. In GENConv, these aggregation types operate over feature channels rather than neighbours: the softmax variants apply a smooth feature-wise normalisation that stabilises deep message passing, while the max variant performs a harder feature selection. The homogeneous model therefore relies on a stronger selection mechanism together with a larger internal expansion to propagate information through a deeper hierarchy [58].

SplineConv. Both variants converge to compact kernels of size 5 but differ in polynomial degree and aggregation behaviour. The heterogeneous model uses degree 2 splines with additive aggregation and an active root weight, promoting smooth spatial interpolation across window–sensor edges. The homogeneous model employs degree 1 splines and `max` aggregation without a root weight, favouring more contrastive feature extraction within the single node set. These patterns indicate that the bipartite setting benefits from slightly more flexible basis functions, whereas the homogeneous graphs rely on stronger nonlinear pooling to represent local luminance peaks [95].

PNAConv. The heterogeneous configuration activates both mean and max aggregators and employs amplification scalers with four towers, yielding a broad but balanced representation of varying connection strengths. The homogeneous model uses only the `max` aggregator with attenuation scalers and two towers, indicating a preference for more compact and selective message statistics when all nodes share similar degree distributions. The deeper pre- and post-MLPs in the heterogeneous setup further enhance its ability to combine information from diverse neighbourhoods [96].

GCN+. GCN+ is implemented only in the homogeneous regime and retains a feed-forward enhancement block while disabling batch normalisation and self-loops. Its use of normalised adjacency matrices indicates that internal normalisation and residual handling provide sufficient stability without external batch statistics, resulting in an efficient Laplacian-based baseline that remains competitive for edge-conditioned tasks.

Summary. Taken together, the optimised operator-specific hyperparameters show a consistent structural adaptation to graph heterogeneity. Heterogeneous models favour smoother aggregations, lower expansion, and fewer internal layers, leveraging the explicit window–sensor topology to simplify message propagation. Homogeneous models, by contrast, adopt deeper and more selective architectures to emulate the same relational structure implicitly through aggregation and internal transformation. These trends confirm that explicit graph heterogeneity reduces the need for strong inductive bias at the operator level, while homogeneous configurations compensate through increased architectural expressivity.

5.3. Testing and Selection

To evaluate the optimised architectures under consistent conditions, all nine GNN operators were retrained using five random seeds (40–44). The reported error bars in Figures 5.20 and 5.21 therefore represent one standard deviation across seeds, quantifying model stability under stochastic initialisation. The comparison integrates both homogeneous and heterogeneous message-passing regimes, using identical feature sets and training schedules to isolate architectural effects. Full numerical results for all operators and transformations are provided in Appendix D.4.

Overall comparison across operators. Figure 5.20 summarises transformation-level performance for all optimised operators across four test cases—*Normal*, *Scaled*, *Rot. 90°*, and *Rot. 180°*. Each subplot visualises mean and standard deviation in *RMSE* versus *MaxAbs* (top) and *SSIM* (bottom). Circles denote heterogeneous operators, while triangles represent homogeneous ones. A clear separation is observed: heterogeneous operators consistently cluster in the lower-left region, reflecting jointly lower *RMSE* and *MaxAbs* errors for the normal and rotation variants. Homogeneous operators, by contrast, exhibit higher errors and greater dispersion. In the scaled variant, this distinction largely disappears; both model types achieve comparable performance. Nevertheless, a marked degradation is evident across all models under scaling, with the drop being more pronounced for the heterogeneous configurations. This suggests that homogeneous models may retain slightly greater adaptability to geometric scaling.

Heterogeneous operator comparison. Figure 5.21 focuses on the four heterogeneous operators, providing a finer-grained view of mean *RMSE*, *MaxAbs*, and *SSIM* under *Normal*, *Rot. 90°*, and *Rot. 180°* transformations. All models perform nearly identically in these settings, with differences below 0.005 in *RMSE*, 0.02 in *MaxAbs*, and 5×10^{-5} in *SSIM*. SplineConv achieves the lowest values across all

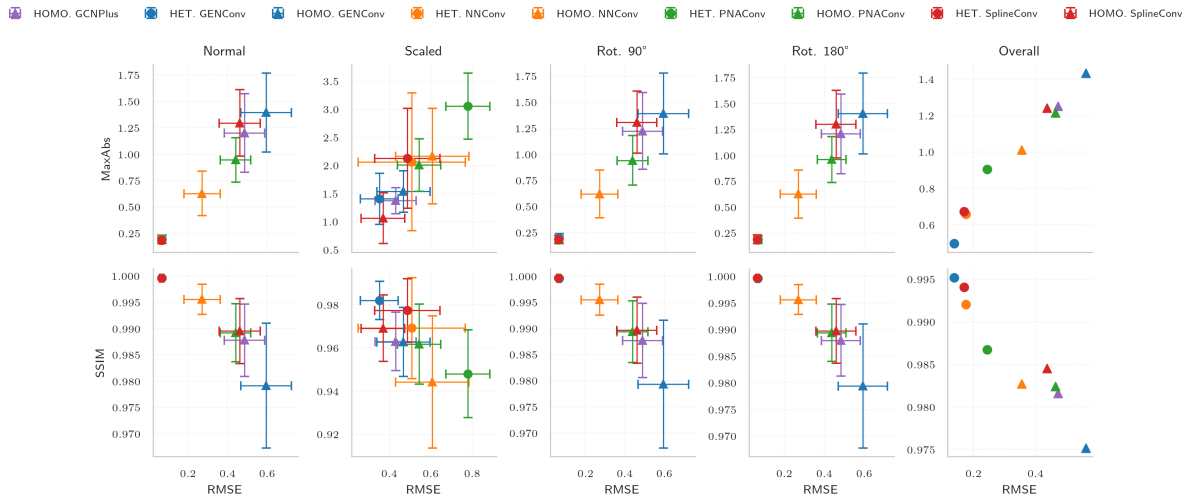


Figure 5.20: Summary of transformation-level performance for all optimised GNN operators. Each subplot shows the mean and standard deviation of RMSE versus MaxAbs (top row) and SSIM (bottom row) across four transformations: Normal, Scaled, Rot. 90°, and Rot. 180°. The final column aggregates overall averages across all transformations. Circles denote heterogeneous operators, while triangles represent homogeneous ones. The heterogeneous GENConv (blue circle) achieves the most consistent and lowest-error performance across all metrics and transformations, followed by SplineConv and NNConv heterogeneous.

metrics, while GENConv trails slightly, yet the magnitude of these differences is negligible. This behaviour aligns directly with the discussion in Section 3.3.1: once the heterogeneous graph provides the relevant geometric and relational structure, the operators converge to the same effective solution class, indicating that the feature representation, rather than the choice of message-passing scheme, determines the achievable accuracy. The only exception is scaling, where the operators no longer collapse as tightly. This is likely because the current feature set is only partially scale-normalised: several descriptors encode ratios or normalised distances, but Euclidean distance is still an absolute geometric magnitude. As a result, models must learn scale-dependent behaviour from data rather than from the features themselves, making operator-specific differences more visible under uniform scaling, especially since the DF field scales nonlinearly with geometry.

Quantitative robustness and final ranking. Table 5.12 consolidates worst-case and average results across transformations. The heterogeneous GENConv achieves the lowest average *RMSE* (0.138) and the best composite rank score (1.4), followed by both SplineConv (3.2 and 3.8) and heterogeneous NNConv (4.0). Homogeneous models exhibit both higher *RMSE* and greater variation, consistent with their need for deeper propagation to approximate window–sensor relations implicitly. Across all operators, the scaled transformation remains the hardest test. Here, the heterogeneous GENConv again achieves the lowest error (0.3507 ± 0.0907), with the homogeneous SplineConv as a close second (0.3682 ± 0.1046). This mirrors their overall ranking and reinforces that the operators performing best under scaling are also the strongest models in the general robustness comparison.

Table 5.12: Comparison of the nine optimised GNN operators on robustness criteria. Minimax values report the worst case across transformations. CVaR-2 is the mean of the two largest *RMSE* values. Rank score is the composite metric defined in Section 4.5.

Operator	Worst RMSE	Worst MaxAbs	Worst SSIM loss	CVaR-2 RMSE	Avg. RMSE	Rank score
HET. GENConv	0.351	1.408	0.018	0.209	0.138	1.4
HET. SplineConv	0.486	2.128	0.023	0.276	0.170	3.2
HOMO. SplineConv	0.461	1.309	0.031	0.460	0.436	3.8
HET. NNConv	0.507	2.065	0.031	0.287	0.175	4.0
HOMO. GCNPLUS	0.488	1.376	0.037	0.486	0.471	5.2
HOMO. PNAConv	0.542	2.010	0.038	0.491	0.464	6.6
HET. PNAConv	0.779	3.058	0.052	0.423	0.243	6.8
HOMO. GENConv	0.594	1.540	0.037	0.594	0.561	7.0
HOMO. NNConv	0.607	2.169	0.056	0.439	0.354	7.0

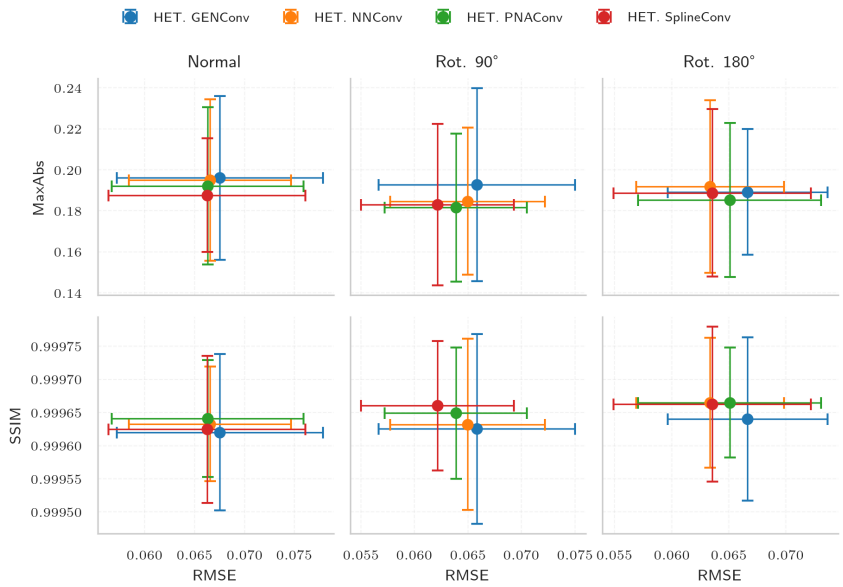


Figure 5.21: Comparison of the four heterogeneous GNN operators across the three evaluated transformations (Normal, Rot. 90°, and Rot. 180°). Points indicate mean values with horizontal and vertical bars representing the corresponding standard deviations in RMSE and MaxAbs (top) and SSIM (bottom). Among the operators, SplineConv consistently achieves the lowest RMSE, MaxAbs, and SSIM loss, whereas GENConv performs slightly worse on all three metrics. However, the absolute differences remain marginal, below 0.005 in RMSE, 0.02 in MaxAbs, and 5×10^{-5} in SSIM, indicating that all heterogeneous models behave nearly identically in terms of predictive accuracy. This suggests that, under identical feature representations, model performance is driven more by the shared input features than by the choice of operator architecture.

Discussion and model selection. Taken together, the figures and metrics demonstrate that heterogeneous models consistently outperform homogeneous ones in both accuracy and robustness, confirming the benefit of explicitly modelling window–sensor relations in the message-passing structure. Within the heterogeneous group, operator differences are minimal; the slight advantage of SplineConv is offset by GENConv's higher stability across seeds. The heterogeneous GENConv is therefore selected as the final operator, providing the best overall compromise between accuracy, robustness, and training consistency. This outcome reinforces the conclusion that explicit heterogeneity introduces the dominant inductive bias for daylight prediction, while further operator variation yields diminishing returns relative to the underlying feature representation. The selected heterogeneous GENConv configuration defines the final architecture of WindowGraphNet, which is evaluated in the next chapter against the ANN benchmarks on unseen geometries to assess generalisation beyond the training distribution.

5.4. Dataset Sufficiency Analysis

Before proceeding to the final model evaluation, the data-efficiency behaviour of *WindowGraphNet*, based on the heterogeneous GENConv architecture, was examined using the learning-curve framework described in Section 4.7.2. The resulting empirical curve, shown in Figure 5.22, expresses validation mean-squared error as a function of the number of available training rooms, averaged over five random seeds (40–44). The shaded region denotes one standard deviation across seeds.

Validation loss decreases by nearly three orders of magnitude as the training set expands from one to 32 rooms, then follows a clear power-law regime with exponent $\alpha = 1.26$, reflecting strong sample efficiency. Beyond approximately one hundred rooms, the curve stabilises around a plateau of 4×10^{-4} *MSE*, with negligible variance across seeds, confirming that the model has reached the data-saturated regime. This plateau indicates that the available dataset is sufficiently large for the GENConv-based surrogate to exploit its representational capacity fully; additional samples would yield marginal gains unless the feature set or architecture is further extended. Consequently, the present dataset is considered adequate for fair evaluation against ANN benchmarks and for assessing generalisation to unseen geometries in the subsequent chapter.

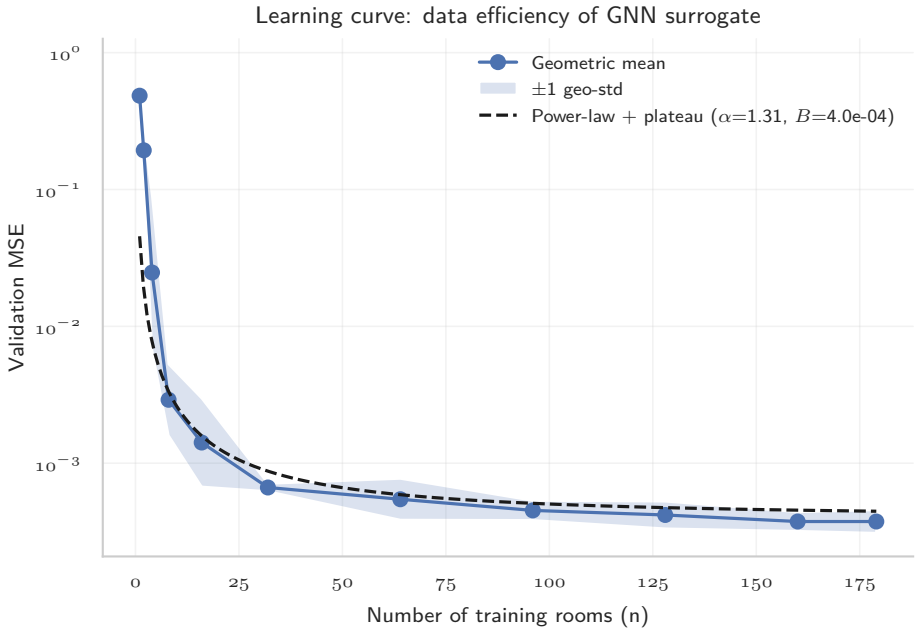


Figure 5.22: Learning curve of the final heterogeneous GENConv architecture showing validation mean-squared error as a function of the number of training rooms. Points indicate the mean across five seeds, with the shaded area representing one standard deviation. The dashed line shows a fitted power-law with plateau, $L(n) = An^{-\alpha} + B$, yielding $\alpha = 1.31$ and $B = 4.0 \times 10^{-4}$. The model exhibits rapid improvement at small sample sizes followed by a gradual saturation, indicating efficient learning and convergence toward its asymptotic error floor.

6

Final Evaluation on the Test Dataset

Having identified the best-performing configuration of WindowGraphNet through systematic feature ablation, operator benchmarking, and model optimisation in the previous chapter, the focus now shifts to its evaluation on unseen data. This chapter presents the conclusive test of model generalisation by comparing the selected WindowGraphNet architecture against benchmark ANN models on the final test dataset. The dataset is deliberately structured into tiers of increasing distributional shift, ranging from in-distribution squares to out-of-distribution rectangles and L-shaped geometries. By analysing predictive accuracy and robustness across these tiers, this evaluation provides a rigorous assessment of whether the proposed graph-based surrogate model offers tangible advantages over conventional ANN approaches when applied to design cases that differ from the training distribution.

6.1. Quantitative Results

The quantitative evaluation examines how predictive accuracy evolves across the six tiers of the Final Test Dataset. Performance is reported using the root-mean-square error (*RMSE*) computed over all sensor nodes and averaged across five random-seed repetitions for each model. This metric provides a direct measure of numerical deviation between predicted and reference DF values, allowing a consistent comparison between the proposed *WindowGraphNet* and the four ANN baselines. Results are presented in two groups corresponding to the lower-complexity tiers (0–2) and the higher-complexity tiers (3–5), which progressively increase the geometric and visibility challenges faced by the models.

6.1.1. Tiers 0–2: In-distribution and Rectangular Variants

Figure 6.1 and the detailed statistics in Table E.1 (Appendix E) summarise model performance across the first three tiers of the Final Test Dataset. These tiers represent progressively greater geometric deviation from the training distribution, ranging from the in-distribution control case (Tier 0) to rectangular and large-scale variants (Tier 2).

Tier 0 – Base (Square). For the base tier, *Raw ANN*, *Le-Thanh ANN*, and *WindowGraphNet* cluster tightly around 0.25–0.28 *RMSE* with negligible standard deviation, indicating stable convergence within this simple domain. In contrast, the *Dieguez ANN* exhibits substantially higher error ($\sim 2.51 \pm 0.5$), consistent with the optimisation instabilities discussed earlier: the combination of sigmoid activations and narrow bottleneck layers leads to oversmoothing and collapsed predictions even under training-like geometry. The *Simple-Dieguez ANN* occupies an intermediate range ($\approx 0.56 \pm 0.06$), reflecting the improved stability gained from removing sigmoids and reducing depth while retaining the physically grounded feature set. Variability across seeds remains minimal for all models (< 0.06), confirming consistent training behaviour aside from the known failure mode of the original *Dieguez* architecture.

Tier 1 – Geometric Transformations (Rotation / Scaling ×2). The rotational cases (90° and 180°) produce the strongest divergence between model families. *Raw ANN* and *Le-Thanh ANN* show dramatic error increases under rotation, rising to 4.68 ± 0.05 and 8.57 ± 1.08 ($90°$) and to 5.62 ± 0.05 and

7.50 ± 1.86 ($180°$), respectively. These values indicate that their learned representations do not transfer under spatial rotation. In contrast, *WindowGraphNet* maintains its Tier 0 performance (0.28 ± 0.06 for both rotations), and the *Simple-Dieguez ANN* remains stable in the range 0.56–0.62. The *Dieguez ANN* again mirrors its Tier 0 behaviour ($\approx 2.5 \pm 0.5$), reflecting the same architectural instability noted earlier.

For the scaled ×2 transformation, allmost all models show moderate increases relative to Tier 0. *WindowGraphNet* rises to 0.66 ± 0.13, while the *Raw ANN* and *Le-Thanh ANN* remain low at 0.35 ± 0.04 and 0.38 ± 0.04, respectively. The *Simple-Dieguez ANN* slightly to 0.65 ± 0.24, whereas the full *Dieguez ANN* decreases to 2.47 ± 0.15. Despite the larger scale change, no model experiences catastrophic degradation.

Seed-to-seed variability is markedly higher for the rotational tests— especially for the *Le-Thanh ANN* (e.g. 8.57 ± 1.08)—highlighting unstable generalisation under rotation. In contrast, *WindowGraphNet* shows negligible variance across all Tier 1 transformations.

Tier 2 – Rectangular and Large-Scale Variants. As the geometries depart further from the training domain, *RMSE* values span a broader range (≈ 0.4–4.9).

For the rectangular (wide) rooms, *Raw ANN* achieves the lowest error (0.62 ± 0.08), while the *Le-Thanh ANN* and *Dieguez ANN* rise to 3.22 ± 0.23 and 2.85 ± 0.55, respectively. *WindowGraphNet* attains 2.09 ± 0.12, with the *Simple-Dieguez ANN* remaining moderate at 1.52 ± 0.07.

For the rectangular (tall) configuration, the pattern is similar. *Raw ANN* again yields the lowest error (0.39 ± 0.06), followed by the *Simple-Dieguez ANN* (1.28 ± 0.09). *WindowGraphNet* produces 1.19 ± 0.13, outperforming both the *Le-Thanh ANN* (4.89 ± 0.31) and the *Dieguez ANN* (2.91 ± 0.22).

The scaled ×5 (square) case represents the strongest distribution shift within Tier 2. *WindowGraphNet* rises to 3.05 ± 1.79, approaching the highest ANN error (3.49 ± 0.48 for the *Dieguez ANN*). The remaining ANNs span 1.54 ± 0.40 (*Raw ANN*) to 2.33 ± 0.37 (*Le-Thanh ANN*), with the *Simple-Dieguez ANN* at 1.89 ± 1.02. Standard deviations increase substantially across all models, reflecting heightened sensitivity to extreme geometric scaling.

For the majority of architectures—*WindowGraphNet*, *Raw ANN*, *Dieguez ANN*, and *Simple-Dieguez ANN*—scaled ×5 is the most challenging Tier 2 configuration. The sole exception is the *Le-Thanh ANN*, which exhibits its largest degradation in the tall-rectangular variant rather than under uniform scaling.

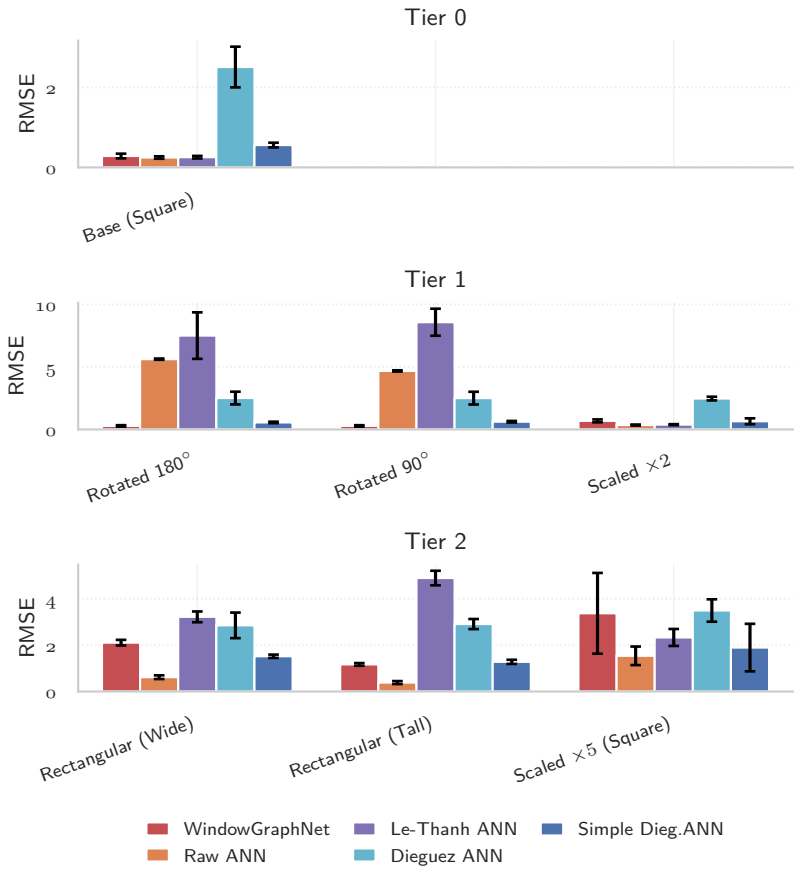


Figure 6.1: Root-mean-square error (*RMSE*) of *WindowGraphNet* and four ANN baselines across the first three tiers of the Final Test Dataset. Tier 0 represents in-distribution square rooms, while Tiers 1 and 2 include geometric transformations and out-of-distribution rectangular variants. Each bar shows the mean *RMSE* across five independent training runs; error bars indicate one standard deviation.

6.1.2. Tier 3: Offset-window Rectangles

Figure 6.2 and the quantitative results in Table E.2 (Appendix E) summarise model performance for Tier 3, which introduces rectangular rooms with laterally displaced windows. This configuration isolates the effect of window position asymmetry while maintaining comparable room dimensions and aspect ratios to those used in Tier 2.

Across all models, *RMSE* values range between approximately 1.9 and 3.0, indicating moderate numerical deviation under this form of geometric shift. The results exhibit narrower error dispersion than in the preceding tiers, with all models converging reliably across random seeds. *WindowGraphNet* and the *Simple Dieguez ANN* achieve the lowest mean errors (1.93 ± 0.07 and 1.99 ± 0.14, respectively), while the *Raw ANN* and *Le-Thanh ANN* show slightly higher values around 2.7–3.0. The *Dieguez ANN* performs intermediately at 2.41 ± 0.35 *RMSE*.

Maximum absolute errors follow a similar pattern, spanning from roughly 4.5 to 8.2 across models. SSIM values indicate moderate to high spatial correspondence (0.36–0.79), with the *Simple Dieguez ANN* achieving the highest structural similarity (0.79 ± 0.01). Overall, the Tier 3 results confirm stable model behaviour under window-position asymmetry, without the extreme error growth observed in the rotation or scaling transformations of Tier 1.

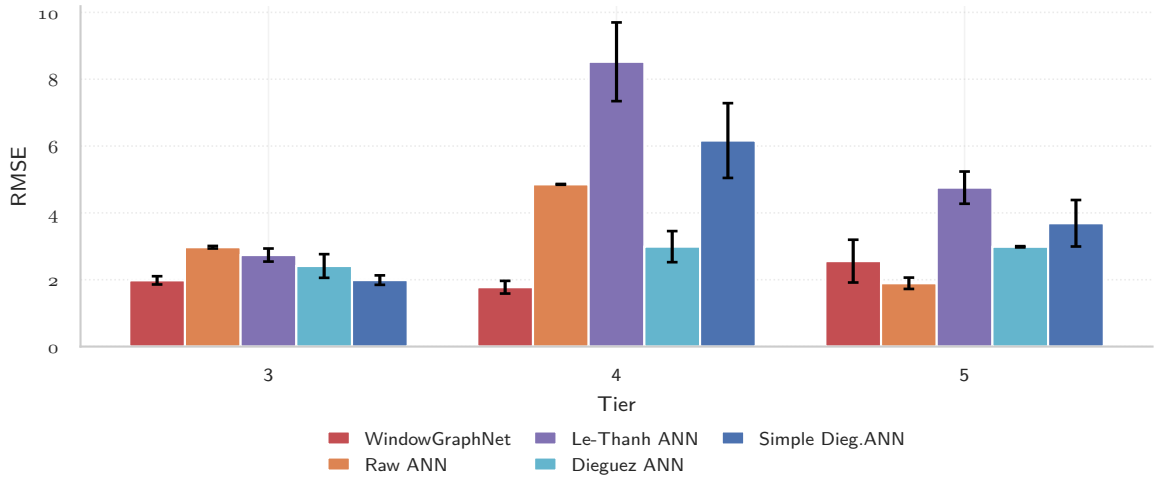


Figure 6.2: Root-mean-square error (*RMSE*) of *WindowGraphNet* and four ANN baselines for Tiers 3–5 of the Final Test Dataset. These tiers include rectangular rooms with offset windows (Tier 3) and L-shaped geometries with partial (Tier 4) and deep (Tier 5) self-occlusion. Bars show mean values across five repetitions; error bars represent one standard deviation.

6.1.3. Tiers 4–5: L-shaped Configurations with Occlusion

Figure 6.2 and the detailed results in Table E.2 (Appendix E) summarise model performance for the L-shaped geometries in Tiers 4 and 5. These tiers introduce partial and deep self-occlusion, respectively, representing the most challenging visibility conditions in the Final Test Dataset.

Tier 4 – Partial self-occlusion. In Tier 4, where windows are placed on the long façades, *RMSE* values range from approximately 1.8 to 7.5. *WindowGraphNet* attains the lowest error (1.78 ± 0.16), followed by the *Dieguez ANN* (2.99 ± 0.47). Higher *RMSE* values are observed for the *Raw ANN* (4.85 ± 0.01), *Simple Dieguez ANN* (6.16 ± 1.12), and *Le-Thanh ANN* (7.54 ± 1.28). This tier exhibits the widest overall error spread of all tested configurations, while seed variability remains moderate across models. SSIM values differ substantially between architectures, from high structural agreement (0.86 ± 0.01) for *WindowGraphNet* to low or negative similarity for the *Raw ANN* and *Le-Thanh ANN*.

Tier 5 – Deep self-occlusion. Tier 5, in which windows are placed on the short façades, produces a narrower error range of roughly 1.9–3.7 *RMSE* across models. *Raw ANN* achieves the lowest error (1.89 ± 0.17), closely followed by *WindowGraphNet* (2.63 ± 0.54) and *Dieguez ANN* (2.99 ± 0.02). The *Le-Thanh ANN* and *Simple Dieguez ANN* yield slightly higher values near 3.0–3.7. SSIM values for all models remain below 0.5, indicating reduced spatial agreement in these deeply occluded configurations.

Compared with Tier 4, the dispersion between models is smaller, and variability across seeds remains moderate for all runs.

Window-level analysis. Figure 6.3 and Table E.3 (Appendix E) present the same results grouped by window placement index (g). Indices $g{=}3$ and $g{=}4$ correspond to Tier 4 (windows on long façades), and $g{\in}\{0, 1, 2, 5\}$ correspond to Tier 5 (windows on short façades). Across façade indices, the lowest *RMSE* values occur at $g{=}2$ and $g{=}5$, while the highest appear at $g{=}3$ and $g{=}4$. *WindowGraphNet* maintains consistently low errors between 1.4 and 3.4 *RMSE*, whereas the *Le-Thanh ANN* and *Raw ANN* show larger variations across facades, reaching values above 5 RMSE for $g{=}3$ and $g{=}4$. The *Dieguez ANN* remains relatively stable (2.9–3.1 *RMSE*) across all façade positions, while the *Simple Dieguez ANN* exhibits higher overall variance (2.6–6.3 *RMSE*). Overall, façade index strongly influences performance in these L-shaped cases, with Tier 4 façades ($g{=}3, 4$) showing the highest overall errors and Tier 5 façades ($g{=}0, 1, 2, 5$) exhibiting smaller differences between models.

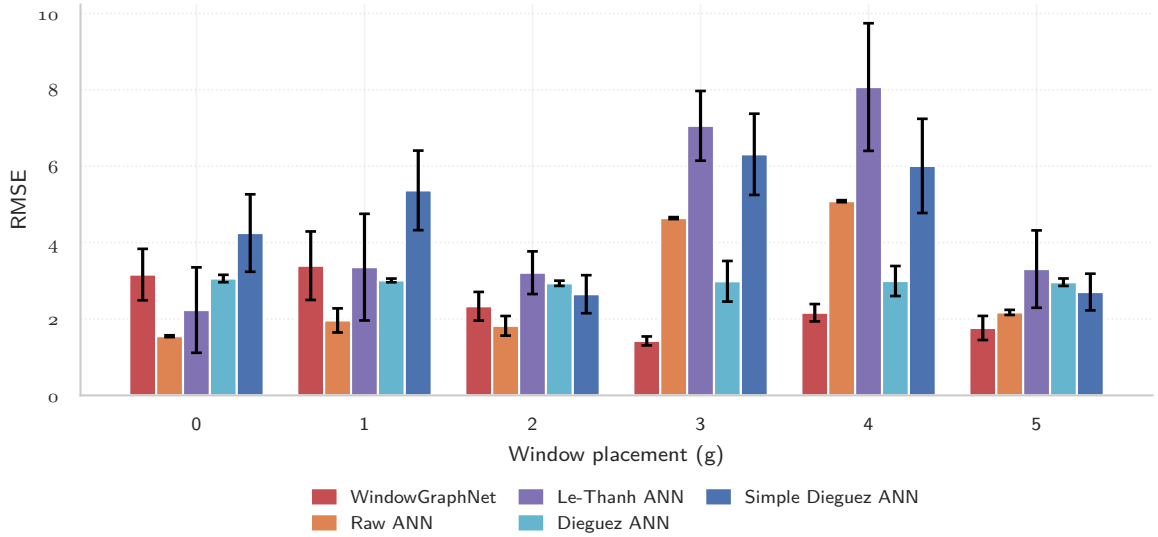


Figure 6.3: Root-mean-square error (*RMSE*) of *WindowGraphNet* and four ANN baselines across façade indices (g) in Tiers 4 and 5 of the Final Test Dataset. Indices $g{=}3, 4$ correspond to long façades (Tier 4, partial self-occlusion), and $g{\in}\{0, 1, 2, 5\}$ correspond to short façades (Tier 5, deep self-occlusion). Bars show mean *RMSE* across five independent training runs; error bars indicate one standard deviation.

6.1.4. Model Size and Inference Time

A comparison of the ANN baselines and *WindowGraphNet* reveals clear differences in model capacity, computational behaviour, and their implications for early-stage daylight prediction. Table E.4 summarises the parameter counts and architectural characteristics of all networks. Although all models were trained under identical optimisation settings, their parameter budgets differ by nearly two orders of magnitude.

Table 6.1: Model sizes and architectural characteristics of the ANN and GNN models.

Model	Architecture Type	Hidden Layers	Parameters
Raw ANN	Fully connected (shallow)	2	1,249
Le-Thanh ANN	Fully connected (medium)	3	5,971
Dieguez ANN	Deep alternating MLP	6	1,281
Simple Dieguez ANN	Reduced MLP architecture	2	1,249
WindowGraphNet (GENConv)	Heterogeneous GNN (shared GEN layers)	3 propagation steps	301,345

Capacity distribution and parameter sharing. All four ANNs are highly compact, containing fewer than 6 000 parameters. Their small size yields fast convergence and negligible memory usage, but

also limits their representational capacity. Among these models, the *Le-Thanh ANN* is the largest due to its wider intermediate layers.

The heterogeneous GNN contains approximately 301 000 parameters, almost entirely concentrated in the relation-specific GENConv modules. The input stems and output head together contribute fewer than 1 000 parameters, confirming that model complexity resides overwhelmingly in the message-passing layers. Importantly, all three propagation steps reuse the same GENConv weights: without this design choice, the parameter count would exceed 900 000. Thus, *WindowGraphNet* increases its effective depth while keeping the number of trainable parameters constant, achieving a balance between expressive power and regularisation.

All models use 32-bit floating-point precision. Under this representation, the ANNs require between 5–25 kB of storage, whereas the graph model occupies roughly 1.2 MB, still lightweight by contemporary deep learning standards.

Training behaviour. All models were trained on the same dataset of 180 training rooms and evaluated on 20 validation rooms per seed. Despite having the largest number of parameters, the graph network exhibits the shortest training time of all models. This behaviour follows directly from the granularity at which each architecture processes the training data:

- ANNs operate at the sensor level. Each sensor point is treated as an independent sample. With roughly 80 sensors per room, the 180 training rooms yield on the order of 1.4×10^4 samples. Even with minibatches of size 32, each epoch requires several hundred optimisation steps.
- The graph model operates at the room level. Each room is processed as a single graph containing all sensors and window nodes. With minibatches of 32 graphs, a full epoch consists of only 5 or 6 optimisation steps, even though each step is computationally heavier.

Thus, the graph model benefits from dramatically fewer optimisation steps per epoch. Combined with efficient GPU implementations of the convolutional kernels, this results in the lowest wall-clock training time (approximately 26 s), despite the model's substantially larger parameter count. By contrast, the Dieguez-type multilayer perceptron—despite having nearly two hundred times fewer parameters, requires roughly 373 s to train because it must iterate through tens of thousands of sensor samples.

Inference behaviour. During inference the situation reverses. Evaluation for the ANNs consists of a small number of dense matrix multiplications applied independently to each sensor sample. These operations are highly optimised on modern GPUs, yielding extremely low per-room latency: approximately 1.7 ms for the Raw ANN, 2.5 ms for the Dieguez variant, and about 5.5 ms for the Le-Thanh model.

The graph network, by contrast, must perform message passing, neighbourhood aggregation, and sparse tensor operations for each room. Although the absolute latency remains low (around 16 ms per room), it is an order of magnitude higher than for the ANNs. Thus, the GNN is the slowest surrogate during inference, but still operates in the millisecond range and remains substantially faster than any physically based simulation.

Comparison to Radiance. For context, a reference daylight-factor simulation was conducted in Grasshopper using Radiance on the full set of 500 test rooms in *Final Test Dataset*. This Radiance benchmark was executed once over the entire *Final Test Dataset*, and the reported statistics therefore reflect the distribution of per-room runtimes within that single full pass. The mean computation time per room was approximately 40.8 s, with a standard deviation of 2.9 s arising from geometric variability across rooms. This figure quantifies the baseline cost of physically based ray tracing under standard DF settings.

Table 6.2 summarises the training times, total test times, and per-room latencies for all surrogate models, alongside the Radiance baseline.

Relative to Radiance, even the slowest surrogate accelerates evaluation by several orders of magnitude. Using the average Radiance time of 40.8 s per room as a baseline, the speed-up factors are:

Table 6.2: Training time (180 training rooms, 20 validation rooms), total test time (500 rooms), and per-room inference latency for all surrogate models, averaged over five seeds, compared with the per-room Radiance runtime.

Model	Training time (s)		Test time (s, 500 rooms)		Per-room latency (s)	
	mean	std	mean	std	mean	std
Dieguez ANN	372.55	80.31	1.23	0.13	0.00247	0.00027
Simple Dieguez	77.87	23.28	1.20	0.28	0.00240	0.00056
Le-Thanh ANN	85.33	3.87	2.76	0.25	0.00552	0.00050
Raw ANN	74.39	13.61	0.87	0.11	0.00175	0.00022
WindowGraphNet	26.26	9.02	8.11	0.44	0.01623	0.00089
Radiance (Grasshopper)	-		-		40.85	2.90

- *Dieguez ANN*: $\approx 6.0 \times 10^5$,
- *Simple Dieguez ANN*: $\approx 5.9 \times 10^5$
- *Le-Thanh ANN*: $\approx 1.3 \times 10^4$,
- *Raw ANN*: $\approx 4.3 \times 10^5$,
- *WindowGraphNet*: $\approx 4.0 \times 10^4$.

Even the graph model, which is the slowest surrogate, provides more than 4 orders of magnitude acceleration over the underlying ray-tracing engine.

Implications for design workflows. These results show that architectural complexity does not imply prohibitive runtime. The graph model combines substantially larger capacity with the fastest training time and a per-room latency well below interactive thresholds. The ANNs offer extremely fast inference, but their coordinate-dependent inductive biases produce materially weaker generalisation on the Final Test Dataset. Radiance remains essential as a physically rigorous reference but is prohibitively slow for iterative design.

In practice, this establishes a clear trade-off: coordinate-based ANNs offer extreme speed but fragile generalisation, whereas the relational inductive bias of the graph network yields significantly more robust predictions at a computational cost that remains negligible in interactive applications.

These findings conclude the quantitative evaluation of the Final Test Dataset. The next section examines how these numerical differences manifest spatially through qualitative visual comparisons of predicted DF distributions.

6.2. Qualitative Evaluation and Interpretation

The quantitative evaluation across the six tiers highlights how each model's inductive bias governs its capacity to generalise beyond the training distribution. Before examining the qualitative outcomes, it is instructive to revisit these inductive biases in relation to the four ANN baselines and the proposed *WindowGraphNet*, since they determine the geometric and physical invariances each architecture can represent.

Coordinate dependence. The *Raw ANN* constitutes the most direct form of surrogate modelling, mapping absolute Cartesian coordinates and two global parameters (room width and window-to-wall ratio) directly to DF values. Lacking any relational or physical abstraction, the network learns purely positional correlations within the global frame. This strong coordinate dependence allows smooth interpolation within the training distribution but prevents recognising geometric equivalence between rotated, mirrored, or translated configurations. Nevertheless, the model retains an implicit scale awareness that enables surprisingly stable extrapolation to uniformly enlarged geometries, as reflected in its relatively low error for the scaled ×2 and even scaled ×5 variants. In practice, the network encodes spatial patterns as fixed image-like templates that rescale consistently but do not rotate or mirror correctly, explaining its sharp degradation under rotation yet robust performance under isotropic scaling.

Local geometric awareness. The *Le-Thanh ANN* introduces a structured representation in which each sensor's local surroundings are expressed through radial obstacle distances, Euclidean distances to window corners, and angular orientation features. These descriptors embed local shape and directional information, allowing the model to infer relative openness and orientation effects. However, because these vectors are computed in a global coordinate system, the encoding remains orientation-sensitive: a rotated room produces a completely different feature pattern. The resulting inductive bias is therefore one of partial geometric awareness: robust to small positional variation but not invariant to global rotation or reflection.

Physical feature grounding. The *Dieguez ANN* employs a set of physically inspired input features designed to approximate the three canonical components of the DF; direct, externally reflected, and internally reflected light. Quantities such as solid angle, aspect ratio, distance to the window, and the angle to the window normal describe photometric relationships rather than purely geometric ones. This physically grounded feature set encourages physically consistent behaviour but comes with a substantial parameter count and a deep, alternating-activation architecture. Given the limited training data available, this combination increases optimisation difficulty, often resulting in smooth yet biased predictions that generalise poorly to unseen geometries or window configurations.

Feature-Preserving Simplification. The *Simple Dieguez ANN* preserves the full set of physically grounded descriptors introduced by Dieguez *et al.*, but replaces the original deep architecture with a simplified, low-capacity multilayer perceptron (identical to the *Raw ANN* structure). This modification isolates the influence of model capacity from that of feature informativeness, allowing a clearer assessment of how much predictive power arises from the feature formulation itself. The simplified network converges more reliably and exhibits lower variance across random seeds, yet, being fully connected, it still treats each sensor independently and cannot exchange spatial information between neighbouring points. Its inductive bias therefore remains local, limiting its ability to capture room-scale light transport effects.

Relational representation. *WindowGraphNet* replaces manual feature vectors with a relational encoding of the scene as a bipartite graph of window and sensor nodes joined by edges that carry geometric–photometric descriptors. Each edge is parameterised by a compact, interpretable set: the *Euclidean distance* $\|\vec{d}\|$, its normalised and horizontally scaled variants, the squared cosine of the vertical incidence angle $\cos^2\theta_z$, and three directional dot products with the local façade frame $(\vec{d}\cdot\vec{t},\ \vec{d}\cdot\vec{u},\ \vec{d}\cdot\vec{n})$. Because these quantities are defined from relative vectors and a local frame, they are invariant to global translation and rotation while remaining directionally sensitive to the façade. Information is then propagated with three layers of *GENConv* (softmax aggregation, expansion factor 4, GELU), whose adaptive normalisation mitigates degree bias and stabilises training. This message-passing formulation couples local geometric cues (distance, orientation, visibility) with global context through iterative aggregation, naturally expressing distance attenuation and directional fall-off without anchoring to absolute coordinates. The resulting inductive bias is explicitly geometric and rotationally invariant, with approximate scale consistency over the training range, supporting the model's stable generalisation across rotations, aspect-ratio changes, and self-occluding layouts.

Together, these five inductive biases span a continuum from coordinate-dependent to relationally invariant representations. The following examples visualise the DF distributions predicted by each model for representative rooms across several tiers of the Final Test Dataset. Each figure compares the predicted DF maps and their corresponding residuals (True–Pred) against the Radiance ground truth, revealing where different inductive biases succeed or fail to capture the underlying illumination structure.

6.2.1. Manifestation of Inductive Biases Across Tiers

The behaviours observed in the quantitative results (Section 6.1) can now be interpreted in light of these inductive biases. Each model's feature representation dictates which geometric transformations it can treat as equivalent, and consequently, under which conditions it fails to generalise.

Transformation invariance (Tiers 0–1)

Figures 6.4–6.6 illustrate how the inductive biases of each model manifest spatially when the same room geometry is subjected to rotation. The base configuration (Tier 0, Figure 6.4) represents the training distribution, while the 90° and 180° rotations (Tier 1, Figure 6.5 and 6.6) provide controlled tests of geometric equivalence. Because the physical light transport problem is invariant under global rotation, any model that has learned the geometry of the scene rather than its absolute coordinates should reproduce the same DF field for all three orientations.

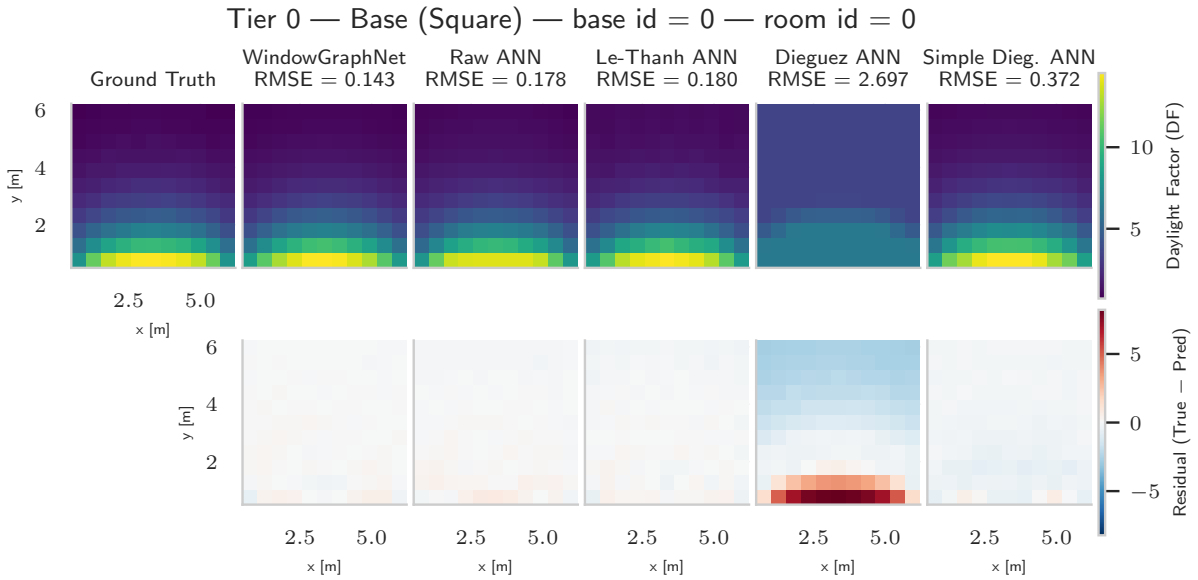


Figure 6.4: Qualitative comparison of DF distributions for the in-distribution base square case (Tier 0). The top row shows DF predictions from each model compared to the Radiance ground truth, while the bottom row presents the corresponding residuals (True–Pred). All models reproduce the façade gradient accurately, confirming consistent interpolation within the training domain.

In the in-distribution square case (Tier 0), all models except the original *Dieguez ANN* reproduce the façade gradient faithfully. *WindowGraphNet*, *Raw ANN*, and *Le-Thanh ANN* achieve near-perfect agreement with the ground truth, while the *Simple Dieguez ANN* remains slightly more diffuse but still accurate. The full *Dieguez ANN*, however, systematically underestimates DF magnitudes and produces a smoothed field with reduced contrast. This behaviour arises from its deep architecture combined with alternating sigmoid and ReLU activations, which compress high-intensity regions and expand low-intensity ones, effectively damping the spatial gradient. The simplified variant, using only ReLU activations, avoids this saturation and yields sharper, more physically consistent predictions.

Once the geometry is rotated by 90° (Figure 6.5), the differences in inductive bias become immediately apparent. The *Raw ANN* anchors its bright zone to the original axes: after rotation it produces a template-like field misaligned with the façade, yielding diagonal residuals. The *Le-Thanh ANN* exhibits the same anchoring, further distorted by its global angular descriptors. Both models therefore "anchor" the DF pattern to the coordinate axes rather than to the façade itself, demonstrating the inherent limitations of absolute coordinate encodings for spatial generalisation.

By contrast, the physically grounded models exhibit full rotational equivalence. The *Dieguez ANN* and *Simple Dieguez ANN* both reconstruct the rotated illumination pattern with essentially the same accuracy as in the base configuration, confirming that their scalar photometric features, solid angle, façade-normal angle, and distance, are invariant to global rotation. The *Simple Dieguez ANN* remains especially precise, slightly overestimating DF values only in the immediate vicinity of the window. Finally, *WindowGraphNet* reproduces the rotated distribution almost perfectly, with residuals close to zero across the entire domain. Because its edge features are defined in local façade coordinates and depend solely on relative vectors between window and sensor nodes, a rotation of the entire scene leaves these relationships unchanged, ensuring true geometric invariance.

The 180° rotation case (Figure 6.6) reinforces these observations. The coordinate-anchored models

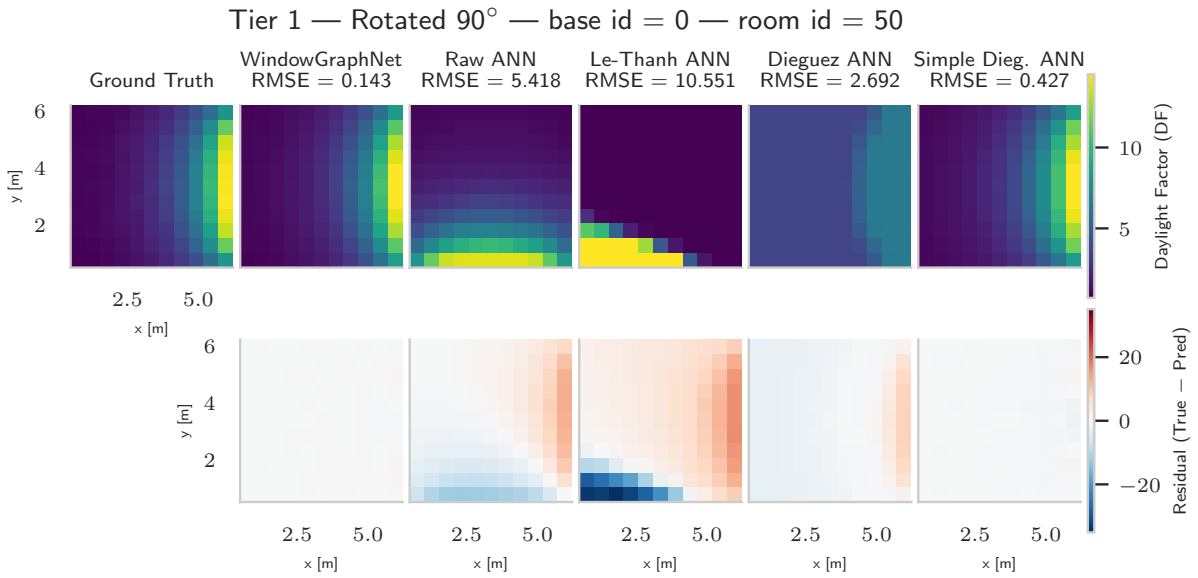


Figure 6.5: Comparison of DF predictions for the rotated configuration (Tier 1, 90° rotation). The *Raw ANN* and *Le-Thanh ANN* fail to recognise geometric equivalence between orientations, producing mirrored gradients and misplaced bright regions. In contrast, *WindowGraphNet* maintains rotational invariance, yielding near-identical DF fields to the base configuration.

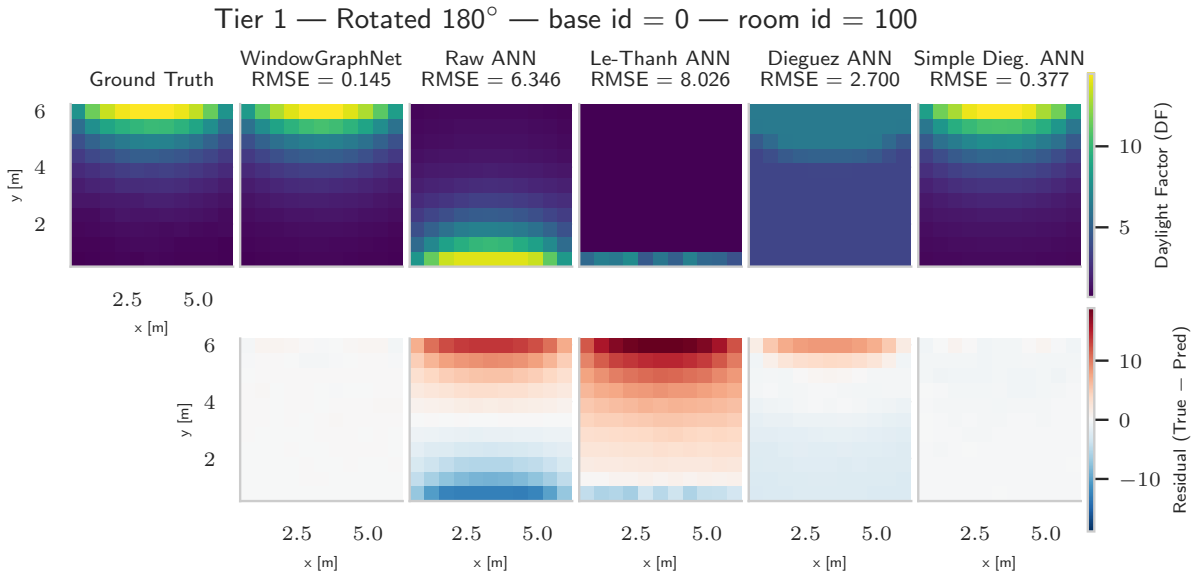


Figure 6.6: DF predictions for the 180° rotation of the base room (Tier 1). The coordinate-dependent models (*Raw ANN*, *Le-Thanh ANN*) invert the façade gradient, while the relational model *WindowGraphNet* preserves the spatial structure. This demonstrates full rotational invariance of the graph-based formulation.

again fix the bright region to the unrotated global axis, while the relational and physically grounded networks yield DF fields that are mirror images of the ground truth, matching both magnitude and structure. Together, these results confirm that only architectures based on relational or scalar physical representations, *WindowGraphNet* and the two *Dieguez* variants, achieve genuine rotational equivalence. This demonstrates that rotation invariance must be embedded directly in the feature representation itself.

Scaling cases (Tiers 1–2).
Scaling tests provide a complementary view to the rotation experiments by probing how each model responds to uniform enlargements of the geometry. Figures 6.7 and 6.9 show the predicted DF distributions for the scaled ×2 and ×5 variants of the base room, while Figure 6.8 provides a one-dimensional profile of DF along the room centreline. Because all geometric proportions and window-to-wall ratios remain identical, the physical illumination pattern should, in principle, preserve its shape and relative contrast. However, the grid spacing used for DF sampling increases proportionally with scale (from 0.5 m to 1 m and 2.5 m), which shifts the first sensor row further from the façade. As a result, the measured peak values drop from roughly 14 in the base case to 9 at ×2 and below 5 at ×5. The steeper measured gradient is largely explained by coarser sampling (first sensor row farther from the façade); residual discrepancies then reveal each model's scale handling.

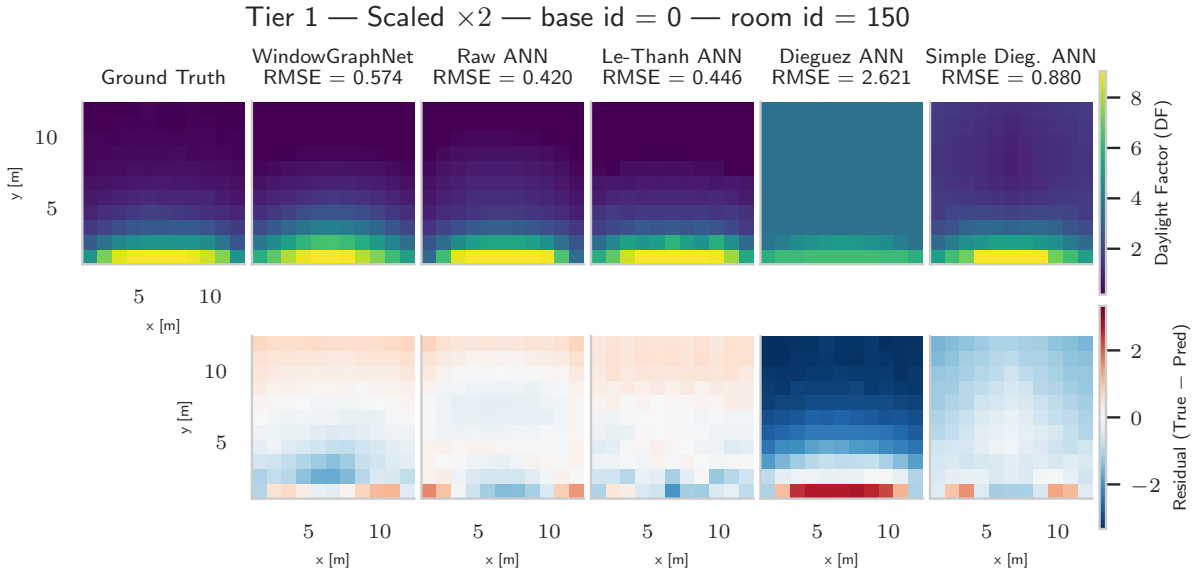


Figure 6.7: Comparison of DF distributions for the scaled ×2 configuration (Tier 1). All models qualitatively reproduce the overall light pattern, but gradient magnitude decreases faster due to doubled sensor spacing. *WindowGraphNet* retains physically consistent fall-off, confirming approximate scale invariance within the training range.

Among all models, the *Raw ANN* performs unexpectedly well under moderate scaling. At ×2 it reproduces the ground-truth profile almost perfectly, maintaining both gradient shape and absolute level. This occurs even though the network operates on absolute coordinates: because it also receives the room width W, the proportional increase in x and W preserves their effective ratio x/W, allowing the network to learn an implicit relative position within the plan. This yields emergent scale awareness up to moderate enlargements. At ×5, however, the model begins to fail physically, its predictions drop below zero beyond about 15 m from the façade, as seen in Figure 6.8. This negative tail reflects unconstrained extrapolation of the learned gradient rather than a geometric error. Without explicit non-negativity constraints or physically motivated asymptotic behaviour, the ANN extends its linear decay indefinitely, producing unphysical sub-zero DF values.

WindowGraphNet retains the correct geometric structure at both scales but systematically overestimates DF magnitudes. In the maps (Figures 6.7 6.9), the predicted fields exhibit a realistic façade-to-depth decay, yet the entire distribution is vertically offset above the ground truth, slightly at ×2 and markedly at ×5. The vertical profiles in Figure 6.8 confirm that the graph model barely differentiates between the two scaling factors: the ×2 and ×5 curves almost coincide, differing only by a reduced gradient at ×5. This behaviour originates from the feature formulation. Because the graph combines both

absolute and relative edge descriptors, the relative terms remain nearly invariant under scaling, while the absolute distances and node densities extend far beyond the range encountered during training. It is possible that this effect is further amplified by the use of normalised distance features, although this cannot be stated with certainty. As a result, the message-passing layers generate DF values appropriate for the Tier 0 scale even when the room expands fivefold. The outcome is geometrically correct but photometrically miscalibrated; accurate shape, incorrect level.

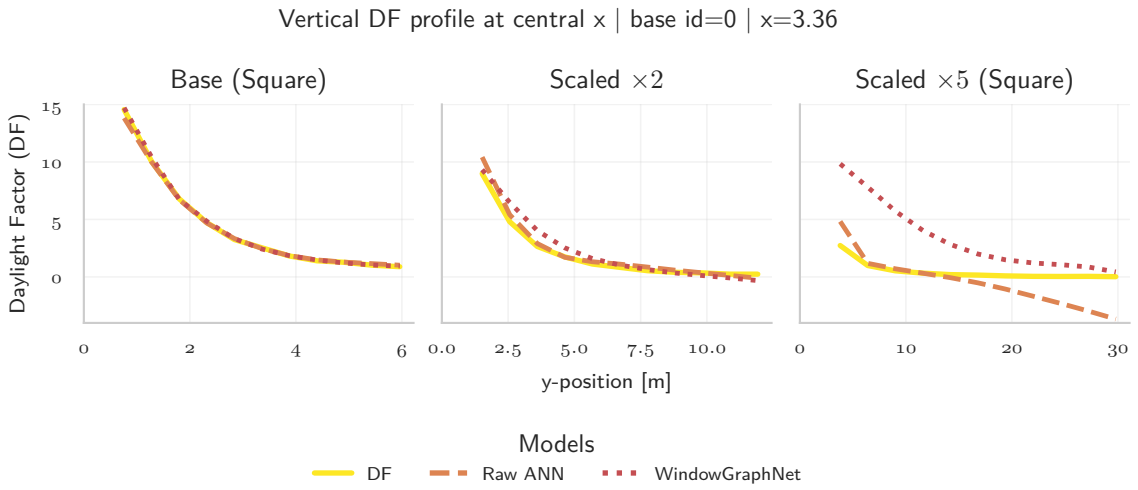


Figure 6.8: Vertical DF profiles at the central x-position for the base room ($\mathrm{base\_id} = 0$). Each panel shows a different geometric variant: the base square, a room scaled by $\times 2$, and one scaled by $\times 5$. Solid lines denote true DF, dashed lines the Raw ANN predictions, and dotted lines the *WindowGraphNet* predictions. The x-axis represents sensor positions measured from the window wall.

The physically grounded networks show contrasting behaviours. The full *Dieguez ANN* produces almost uniform, low-contrast maps, reflecting activation saturation and poor calibration when distance and solid-angle features exceed the ranges seen in training. The *Simple Dieguez ANN*, although more stable, develops local spikes near the upper corners in the ×5 case, artefacts of nonlinear feature interactions under extreme geometric proportions. The *Le-Thanh ANN* overestimates illumination near the window but severely underpredicts values deep in the room, as its radial and angular descriptors amplify decay with growing distance.

In summary, the scaling experiments reveal that all models retain geometric coherence but diverge in their treatment of absolute magnitude once distances and sensor densities exceed the training range. The next set of configurations extends this analysis to rectangular rooms, which can be viewed as directional, anisotropic counterparts of the uniform scaling tests. Tier 2 doubles one principal dimension while keeping the other constant, introducing elongated aspect ratios rather than uniform enlargement. Tier 3 then returns to training-scale dimensions but laterally displaces the window, creating rectangular rooms with side façades that isolate the effect of asymmetric window placement.

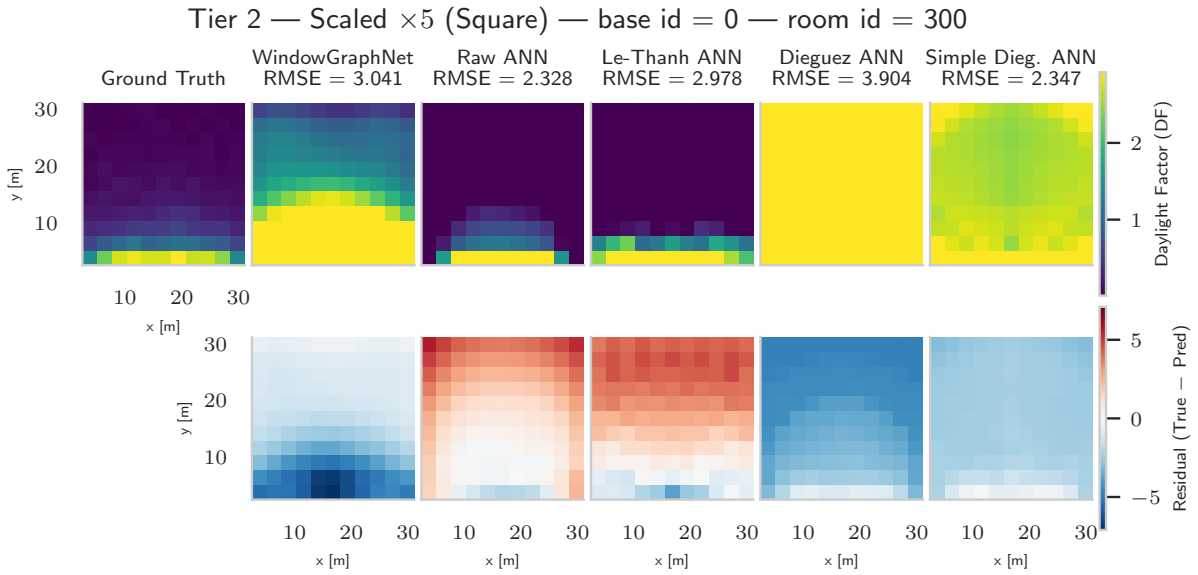


Figure 6.9: DF predictions for the extreme scale ×5 case (Tier 2). The increased spacing between sensors produces steep DF gradients that all models underpredict to varying degrees. Despite this, *WindowGraphNet* retains correct spatial structure, demonstrating partial but limited scale consistency.

Rectangular and offset-window cases (Tiers 2–3).
The rectangular and offset-window configurations test how the models generalise when geometric proportions and window placement deviate from the symmetric layouts seen in training. Figures 6.10 and 6.11 present the rectangular configurations, which serve as an intermediate step between the uniform scaling and the asymmetric window-placement cases. These variants test whether the models can maintain physically consistent predictions when only one principal dimension departs from the square training geometry. In both cases, the sensor-grid spacing returns to 0.5 m, eliminating the sampling artefacts observed in the scaling experiments, while the plan dimensions extend up to 16 m in width or depth, corresponding roughly to the $\times 2$ scaling range. The horizontally and vertically elongated plans shown in Figure 6.10 and Figure 6.11 therefore isolate the effects of anisotropic scaling, allowing the comparison of model performance under directionally biased yet still symmetric geometries.

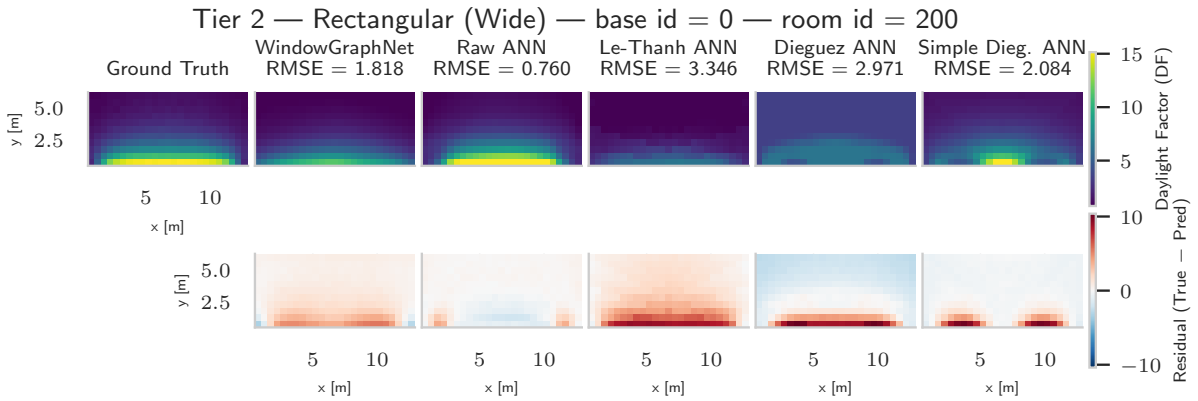


Figure 6.10: DF predictions for the horizontally extended rectangular case (Tier 2). The *Raw ANN* extrapolates well due to coordinate normalisation by room width, while *WindowGraphNet* maintains a consistent façade gradient across the enlarged span. Orientation-sensitive models such as *Le-Thanh ANN* show stronger lateral bias.

Here, the *Raw ANN* again performs well in tall rooms, depth is implicit in y, but in wide rooms it laterally stretches the learned template without re-normalising to aperture width, overestimating centrally and underestimating near the ends. Because depth information is implicitly encoded in the y-coordinate, the model continues to produce a physically plausible façade-to-depth decay with very low residuals in the tall configuration. In the wide variant, however, it begins to overestimate near the centre of

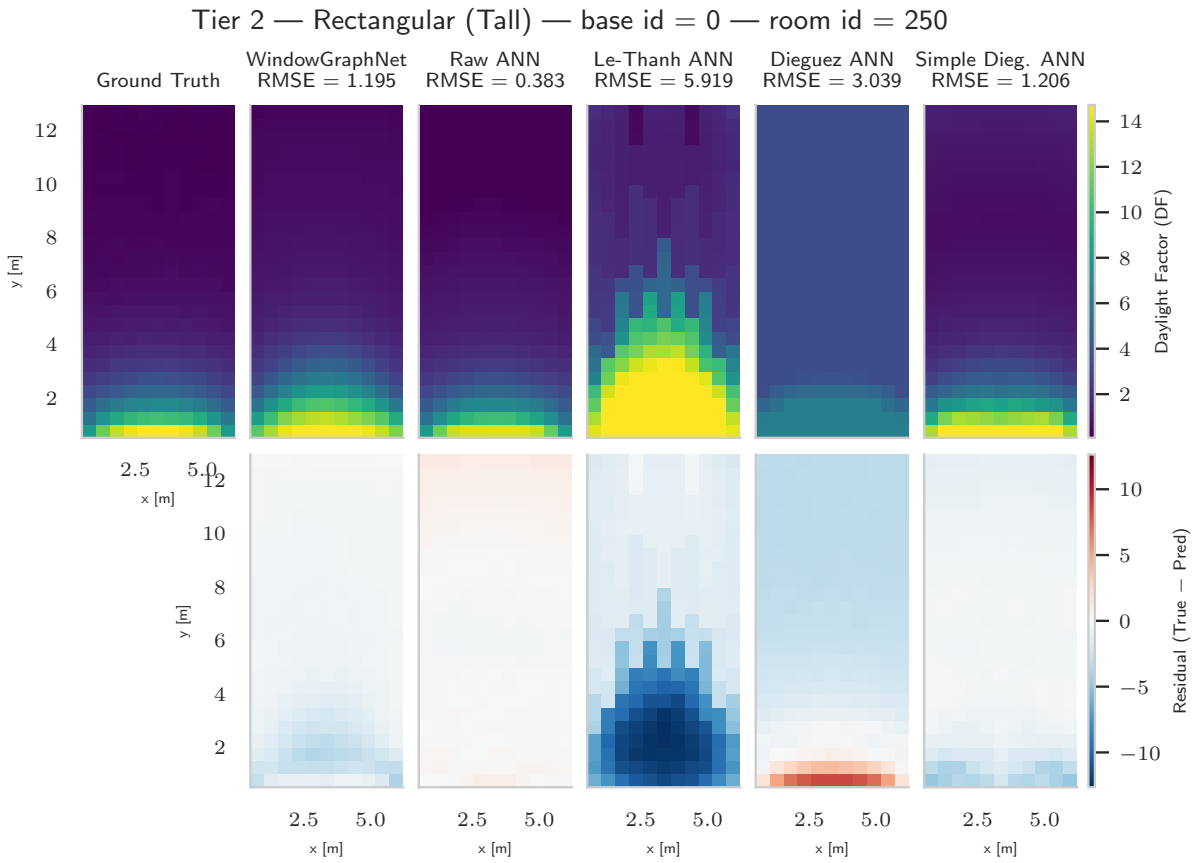


Figure 6.11: Qualitative comparison for the vertically extended rectangular case (Tier 2). *WindowGraphNet* reproduces depth-dependent attenuation more faithfully than the ANN baselines, which tend to over-brighten the upper regions of the room. This indicates robust generalisation to aspect-ratio changes that preserve façade alignment.

the façade and underestimate toward the edges, suggesting that the learned DF pattern is laterally stretched but not fully re-scaled to the extended aperture width. This behaviour reflects the model's partial ratio awareness: it can interpolate along depth effectively but lacks explicit cues to adjust the lateral extent of the illuminated region.

The *Le-Thanh ANN*, by contrast, deteriorates sharply under aspect-ratio changes. Although its descriptors are defined in a fixed global frame and thus invariant to rotation, they are not invariant to the relative scaling of the room axes. As the plan becomes elongated, the distances and angular relations between sensors and window corners deviate substantially from those seen during training, distorting the geometric symmetry assumed by the feature formulation. In wide rooms, the shorter distance to side walls combined with the extended window length produces inconsistent corner-angle patterns, while in tall rooms, the increased depth alters the ray-intersection geometry so severely that the model generates spurious bright zones and abrupt transitions. The breakdown therefore arises not from rotation but from the fact that its corner-based encoding implicitly assumes square or near-isotropic geometry.

The physically grounded *Dieguez ANN* maintains moderate stability but continues to exhibit smooth, low-contrast fields that under-represent spatial variation. The simplified variant performs noticeably better, though its behaviour differs between wide and tall cases. In wide rooms, it captures only the central portion of the window's contribution, effectively averaging the aperture into a single luminous patch and underestimating illumination near the window ends. This occurs because its global scalar features, solid angle, façade-normal angle, and average window distance, collapse lateral variation within the opening. In tall rooms, where the aperture dimensions and feature magnitudes align more closely with the training distribution, the predictions improve markedly, producing gradients comparable to those of *WindowGraphNet* with only a slight positive bias.

WindowGraphNet remains consistent across both configurations but mirrors the scale-dependent tendencies observed earlier. For wide rooms it underestimates near the façade and toward the lateral window edges, reflecting the increased separation between window nodes and the reduced influence of distant lateral messages. In tall rooms, by contrast, it slightly overestimates the overall DF magnitude while maintaining smooth depth-dependent decay. These complementary errors highlight that the graph model is more sensitive to the spatial distribution of window nodes along the façade than to the absolute depth of the room. Nevertheless, it preserves the correct illumination structure and shows minimal variability across repetitions, confirming stable convergence and a low overall bias.

Figures 6.12 and 6.13 illustrate the Tier 3 offset-window configurations, in which the window opening is shifted laterally along the façade while the overall room dimensions remain comparable to those in Tier 2. This setting isolates the effect of asymmetric window placement by introducing a strong lateral imbalance in daylight access while preserving the same façade orientation and boundary conditions.

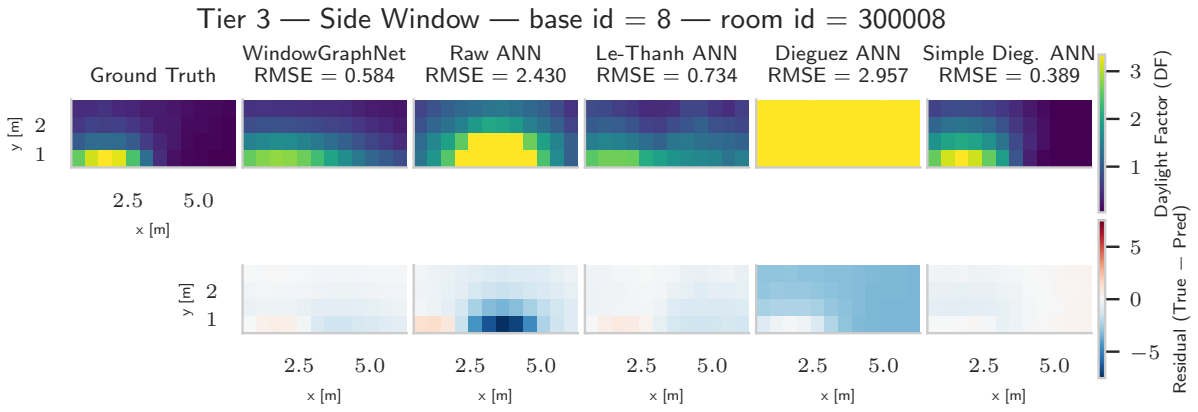


Figure 6.12: Comparison of DF fields for the laterally displaced window configuration (Tier 3, base id 8). All models capture the asymmetric façade exposure qualitatively, though *WindowGraphNet* and the physically grounded ANN variants maintain more accurate luminance decay across the interior.

The differences between models reflect their ability to re-anchor the predicted illumination field when the aperture no longer occupies the central façade position. The *Raw ANN* performs poorly under

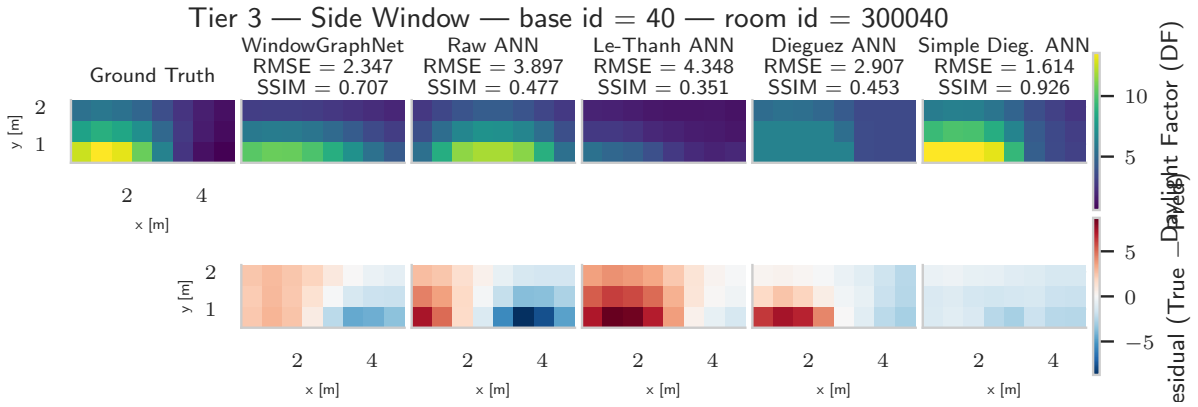


Figure 6.13: Comparison of DF fields for the laterally displaced window configuration (Tier 3, base id 40).

this displacement: its coordinate-dependent representation anchors the high-illuminance region to the centre of the plan, producing the same symmetric pattern as before. This failure is not apparent from its *RMSE*, which remains deceptively low because the overall magnitude of the DF field is small; the error metric does not penalise spatial misalignment strongly, allowing structurally incorrect predictions to appear numerically acceptable.

The *Le-Thanh ANN* localises the bright region more accurately but produces abrupt transitions and exaggerated contrast near the window. Its corner-based and ray-intersection features are defined in a global frame and lack lateral normalisation, so the feature distributions change nonlinearly when the window shifts. As a result, the model overemphasises the near-window zone while underestimating illumination across the far side, yielding blocky and discontinuous residual patterns.

The full *Dieguez ANN* again exhibits a nearly uniform, low-contrast field, reflecting activation saturation when solid-angle and distance features fall outside their calibrated range. In contrast, the simplified variant reproduces the asymmetric light distribution much more faithfully. Because its physically grounded features describe visibility through global scalars such as solid angle and façade-normal angle, they remain invariant to translation along the façade, allowing the model to reorient the bright zone without geometric reparameterisation. Residuals are small and spatially uniform, confirming the stability of the simplified formulation.

WindowGraphNet achieves similarly robust behaviour. The relational encoding of window–sensor geometry allows the network to reposition the high-DF region naturally: as the window nodes move laterally, the corresponding sensor–window edge relations simply update, preserving the learned propagation pattern. The resulting DF maps correctly capture the shifted façade gradient and its lateral decay, with only minor global overestimation near the aperture. Residuals are largely homogeneous, indicating that the remaining error stems from magnitude bias rather than structural misplacement.

Overall, the rectangular and offset-window configurations demonstrate that the models' ability to generalise depends on how geometric and photometric relationships are represented. Coordinate-based models such as the *Raw ANN* adapt well to simple geometric scaling but fail when symmetry is broken, while feature-engineered ANNs like *Le-Thanh* struggle once their isotropic assumptions are violated. Physically grounded and relational formulations, in contrast, remain stable across both aspect-ratio and window-displacement changes, with *Simple Dieguez ANN* and *WindowGraphNet* showing the lowest structural distortion and most consistent luminance decay patterns. These results suggest that representations encoding directional and visibility-based relationships generalise best to geometries that preserve unobstructed line-of-sight between sensors and window surfaces, even when proportions or placement shift.

The next analysis examines conditions that depart from this open-visibility regime. Tiers 4 and 5 introduce partial and deep self-occlusion through L-shaped geometries, testing how each model extrapolates when parts of the room are no longer directly visible from the window façade.

Obstructed and self-occluded geometries (Tiers 4–5).
The final set of configurations extends the analysis from purely geometric transformations to cases where direct visibility between window and sensor points is partially or completely obstructed. These L-shaped plans represent a fundamental shift in the illumination regime: rather than continuous geometric deformation, they introduce discontinuities in the light transport path itself. Tier 4 features windows placed on the long façades, creating recesses that are only partially exposed to the sky and therefore retain a direct line of sight for most sensors. Tier 5 moves the window to the short façades, producing deep self-occlusion in which an entire zone of the interior becomes detached from direct daylight access. Together, these two tiers test whether the models can generalise beyond the open-visibility assumption that underpinned all previous configurations, evaluating their ability to predict light propagation in spaces where part of the domain is physically unreachable from the aperture.

Tier 4 – Partial self-occlusion. Figures 6.14 and 6.15 illustrate the L-shaped configurations with windows placed on the long façades ($g = 3, 4$), producing shallow recesses that remain partly visible from the aperture. These layouts introduce a pronounced discontinuity in direct daylight access while still allowing some interreflected light to reach the recessed zone. They therefore test whether the models can represent the sharp DF drop across the bend without losing global coherence.

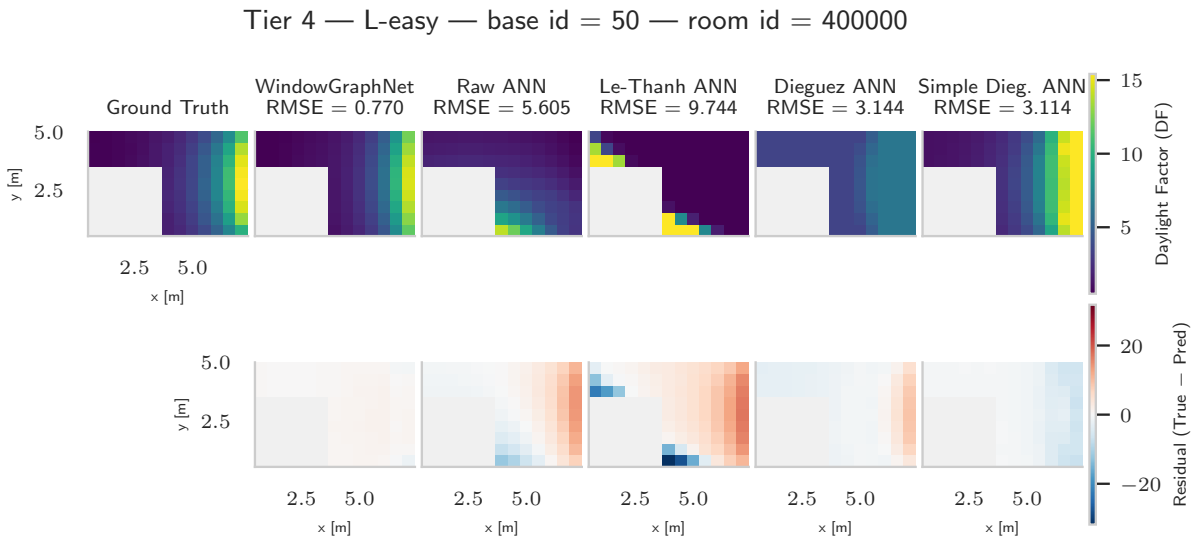


Figure 6.14: DF predictions for the L-shaped configuration with the window on the long façade (g=3). The recess remains partially visible from the aperture, creating a strong daylight gradient across the bend. *WindowGraphNet* accurately reproduces this transition, whereas the ANNs fail to capture the visibility break, producing either overly smooth (*Raw ANN*) or distorted (*Le-Thanh ANN*) illumination fields.

Across both façades, *WindowGraphNet* performs most robustly, reproducing the façade-aligned gradient and the rapid attenuation inside the recess with only minor overestimation near the transition edge. The residuals remain smooth and locally confined, indicating that the network successfully distinguishes between directly and indirectly lit regions through its relational encoding of window–sensor visibility. Its performance remains stable across both mirrored variants, confirming consistent handling of geometric symmetry and occlusion direction.

As neither the *Raw ANN* nor the *Le-Thanh ANN* is rotation invariant, both fail in a similar manner when the window is placed on the north or east façade, as observed in the rotation variants—the *Raw ANN* continues to form the same grid pattern from the centre, while the *Le-Thanh ANN* incorrectly positions the light source in the lower-left corner.

Both physically grounded variants (*Dieguez* and *Simple Dieguez* ANNs) display smoother, more plausible attenuation but with different limitations. The full *Dieguez ANN* consistently underpredicts the illuminated façade and over-smooths the entire field, leading to uniform low-contrast maps. The simplified variant retains higher contrast yet fails to capture the steep decay across the recess, slightly overestimating the exposed façade and underestimating the shadowed corner. These outcomes in-

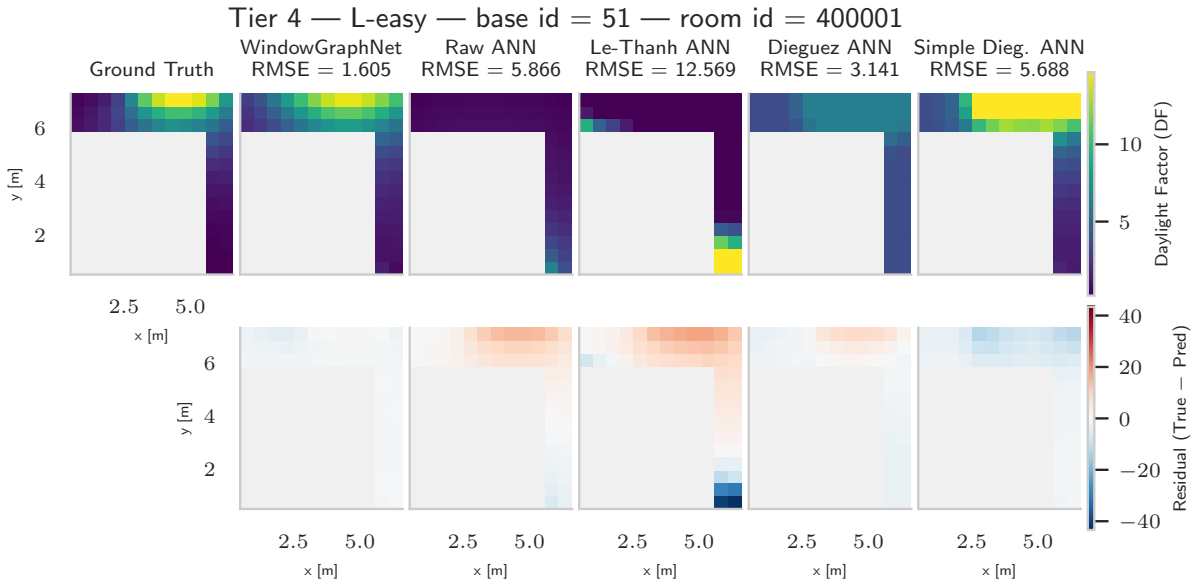


Figure 6.15: DF predictions for the mirrored L-shaped configuration with the window on the opposite long façade (g=4). Despite the reversed geometry, the trends mirror those of the previous case: *WindowGraphNet* retains stable performance, while coordinate-dependent and feature-engineered ANNs exhibit orientation-sensitive errors and poor shadow-region estimation.

dicate that while solid-angle and façade-angle descriptors improve physical plausibility, they remain insensitive to local self-occlusion geometry.

Overall, the Tier 4 results show that only *WindowGraphNet* consistently preserves the physical structure of the illumination field across partially occluded configurations. Its relational formulation inherently encodes visibility through edge connectivity, allowing it to separate directly and indirectly lit regions without explicit geometric cues.

Tier 5 – Deep self-occlusion. Figures 6.16–6.18 present the Tier 5 configurations, where the window is located on the short façade and the interior recess is fully detached from direct daylight access. Only a small portion of the room near the aperture receives direct light, while the remaining sensors depend entirely on interreflections. This tier therefore represents the most severe form of geometric occlusion in the dataset, testing each model's ability to extrapolate illumination into regions disconnected from the primary visibility graph.

The *Raw ANN* achieves good numerical performance, as indicated by low *RMSE* values that result more from averaging than from structural accuracy in some instances, but it fails to deliver physically meaningful predictions. As already noted, this discrepancy highlights the limitation of *RMSE* as a sole indicator of model quality and underscores the importance of inspecting the DF prediction plots directly: in cases such as Figure 6.16, the ANN produces an almost uniform field that yields a deceptively low error, whereas *WindowGraphNet*, despite a higher *RMSE*, more accurately reconstructs the façade–recess structure of the illumination field.

In Tier 5, the *Le-Thanh ANN* performs even less consistently. Only in the shallower configuration (base 51) does it retain a recognisable façade gradient, albeit with exaggerated brightness near the window. In all other cases, the model effectively breaks down: the feature geometry fails to capture the true façade orientation, and the network no longer recognises the location of the opening. In base 64 and particularly in base 86, the predictions collapse to a near-uniform field, with the model effectively "losing" the window entirely. This behaviour reinforces the same limitation observed in Tier 4, its globally defined angular and distance descriptors cannot represent visibility discontinuities or varying façade orientations. As a result, the *Le-Thanh ANN* remains strongly tied to the coordinate frame of the training data, preventing generalisation to rotated or self-occluding configurations.

The *Dieguez ANN* follows the same trend observed in earlier tiers, producing low-contrast, nearly uni-

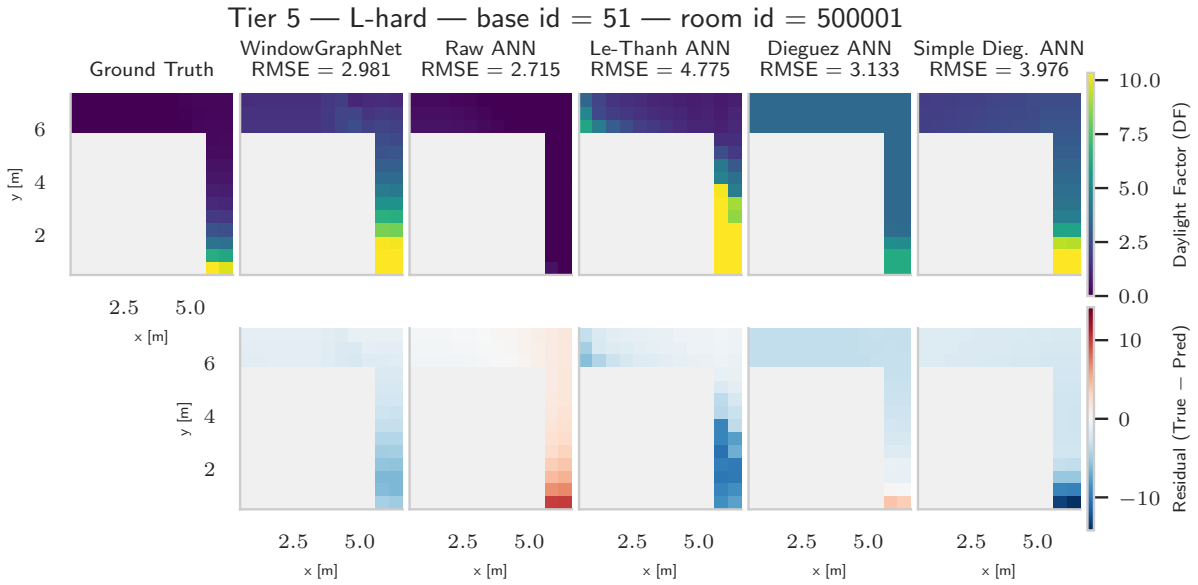


Figure 6.16: DF predictions for the L-shaped configuration with the window on the short façade (g=2). The recess is fully detached from direct daylight access. *WindowGraphNet* reproduces the spatial gradient correctly but slightly overestimates the DF magnitude within the recess, whereas all ANN baselines fail to capture the sharp visibility break.

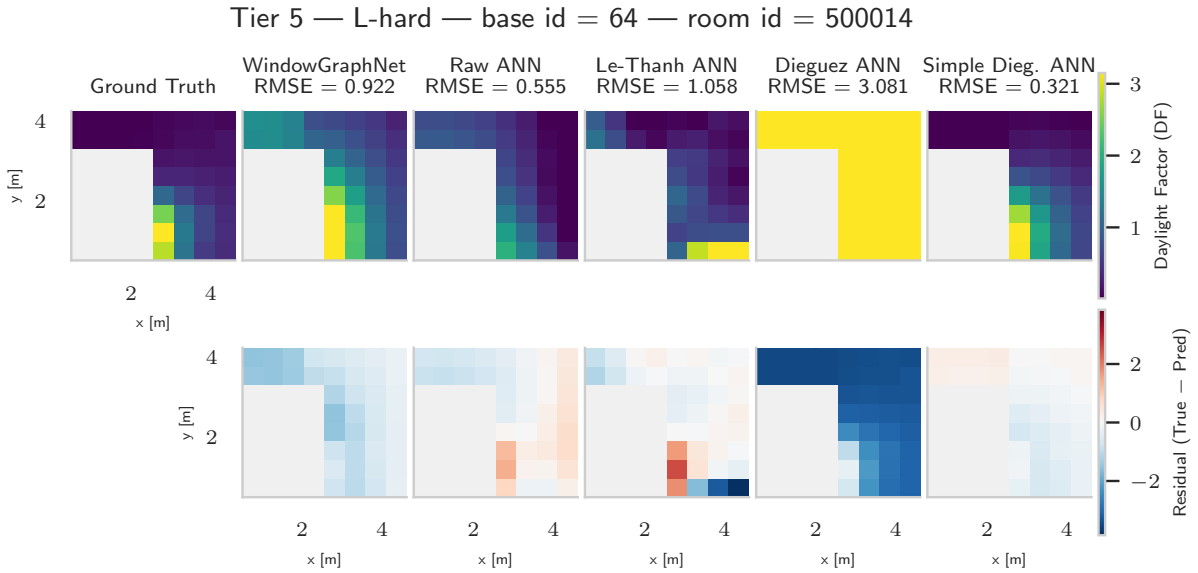


Figure 6.17: DF predictions for the L-shaped configuration with the window placed on the opposite short façade (g=1). The results reveal consistent overestimation of DF values in the recess for *WindowGraphNet* and diminished contrast for the ANNs, particularly the *Raw* and *Le-Thanh* models.

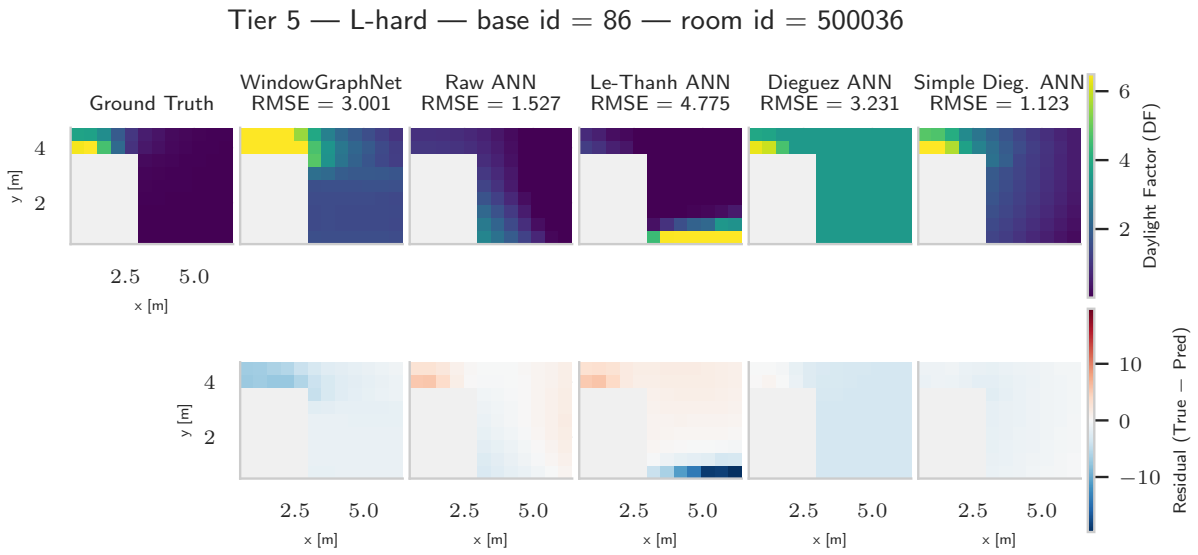


Figure 6.18: DF predictions for the most occluded configuration (Tier 5, $g{=}0$). Only a narrow zone near the aperture remains illuminated, while the recess receives negligible daylight. *WindowGraphNet* preserves spatial structure but overestimates magnitude across the interior; the *Simple Dieguez ANN* produces a more balanced but lower-contrast distribution.

form DF maps. This behaviour stems from activation saturation and from the dominance of global distance terms when solid-angle values approach zero. The model correctly captures the general dimness of the occluded region but loses all local variation.

The *Simple Dieguez ANN* performs surprisingly well given the absence of direct visibility. It successfully delineates the bright façade zone and the dark recess, albeit with reduced contrast and mild overestimation near the aperture. Because its features, such as average aspect ratio and distance to the window, remain well defined even when individual solid-angle terms vanish, it continues to encode a physically consistent gradient even surpassing the message-passing model in magnitude calibration in some cases. This can be contributed to the fact that, unlike *WindowGraphNet*, all sensors receive identical global information, which produces smoother transitions.

Finally, *WindowGraphNet* yields consistent predictions across all Tier 5 configurations. The bright–to–recess transition is correctly located and the spatial layout closely follows the ground truth, but a systematic magnitude bias persists, most clearly in Figures 6.17 and 6.18, with slight overestimation across the interior (blue residuals). In the shallower case (base 51; Figure 6.16) the façade–recess gradient remains physically accurate, whereas in deeper or lower-illuminance cases (bases 64, 86) the contrast between visible and occluded zones intensifies.

This behaviour is consistent with the model's relational encoding: as window–sensor connectivity diminishes, the information conveyed into the recess weakens, yielding physically plausible attenuation under reduced visibility. Sensor-to-sensor aggregation sustains a residual signal only over a limited propagation depth; beyond this effective range, distant recess sensors receive too little relational input to reconstruct the attenuated field, resulting in a gradual reduction of contrast rather than an abrupt loss of detail. Thus, the mechanism remains physically coherent at the level of local visibility: window edges regulate light ingress correctly, yet the finite communication range moderately limits the reproduction of the weak diffuse interreflections visible in the ground truth. The persistent overestimation near the façade is plausibly driven by local message accumulation around high-degree window nodes. Overall, *WindowGraphNet* captures visibility structure with high geometric fidelity, while its limited propagation depth constrains magnitude calibration in strongly self-occluded interiors.

Across all Tier 5 configurations (Figure 6.19), the recess sensors converge toward a nearly uniform value of approximately 1.5 DF. This level remains consistent across geometries and orientations and explains the systematic overestimation observed in the residuals: although illumination in the deepest recess should approach zero, the model stabilises at a low yet finite value. This behaviour arises because these sensors lie outside the training distribution; during training, every sensor maintained

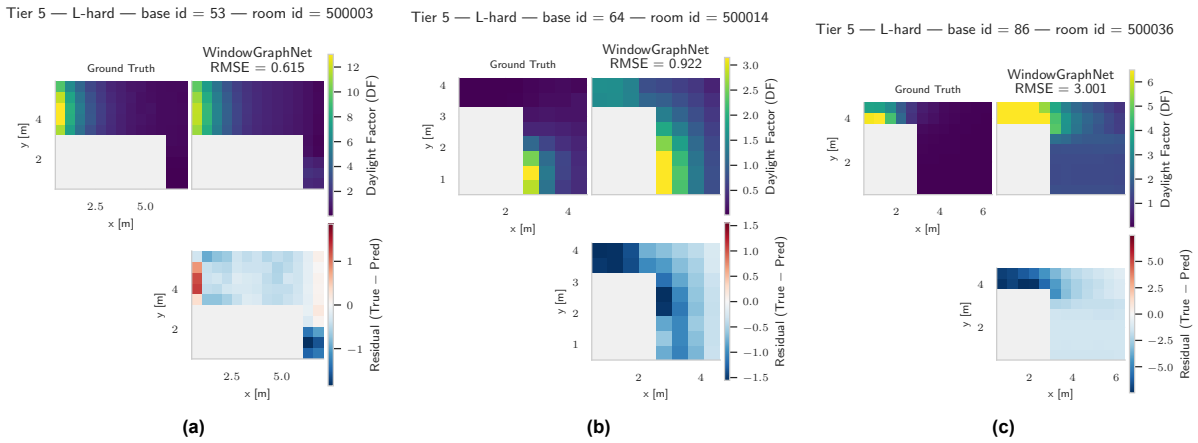


Figure 6.19: Tier 5 (L-hard) — *WindowGraphNet* predictions across three L-shaped rooms. The façade–recess transition is consistently captured, while occluded recess zones converge toward a sensor–sensor baseline and façade regions show mild overestimation.

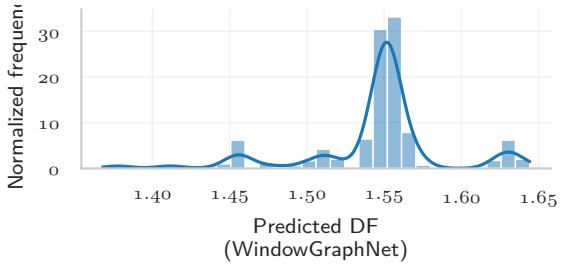


Figure 6.20: Distribution of predicted DF values from *WindowGraphNet* under full occlusion. The strong peak around 1.55 DF corresponds to the sensor–sensor baseline that emerges when no window edges are present, confirming a consistent window-independent plateau. Minor side peaks reflect subtle biases induced by the directional component n across different room configurations.

all connections to a window node. In fully occluded regions, therefore, the network lacks exposure to purely sensor–sensor relationships and cannot infer that illumination should vanish. With only three message-passing layers, signals from visible zones dissipate before reaching the recess, leaving those sensors fixed at an internal baseline rather than converging to darkness.

The aggregated distribution in Figure 6.20 corroborates this finding. To isolate the model's behaviour in the absence of any window information, the Tier 0 test rooms were re-evaluated after explicitly removing all window-to-sensor edges, effectively constructing a "no-window" variant of the dataset. The resulting distribution shows a distinct peak around 1.55 DF, reflecting the plateau predicted for occluded sensors and confirming that the model converges to a stable, window-independent baseline when no window connectivity is available. This plateau is not random noise but an emergent equilibrium determined by the finite communication range within the graph. Secondary peaks indicate minor orientation-dependent variations, suggesting that the network retains some directional tendency even without explicit visibility input.

The spatial patterns in the no-window study shown in Figure 6.21 clarify the origin of these variations. When all window edges are removed, the predicted fields across all room scales consistently fall between 1.35–1.65 DF, matching the plateau observed in Tier 5. This confirms that the baseline arises from the combination of limited message passing and out-of-distribution connectivity rather than from any architectural defect. Changing the surface-normal direction n modifies the spatial imprint of this baseline; while it does not create the baseline itself—which follows from the absence of window connectivity and finite receptive fields—it does impose a weak directional prior on the otherwise uniform illumination field.

Mechanistically, the sensor–sensor edge attributes include directional dot products $\vec{d} \cdot \vec{t}$, $\vec{d} \cdot \vec{u}$, and $\vec{d} \cdot \vec{n}$ (kept as channels 9–11), where $\vec{d}$ is the normalised, scaled sensor–sensor offset, $\vec{n}$ is the wall normal

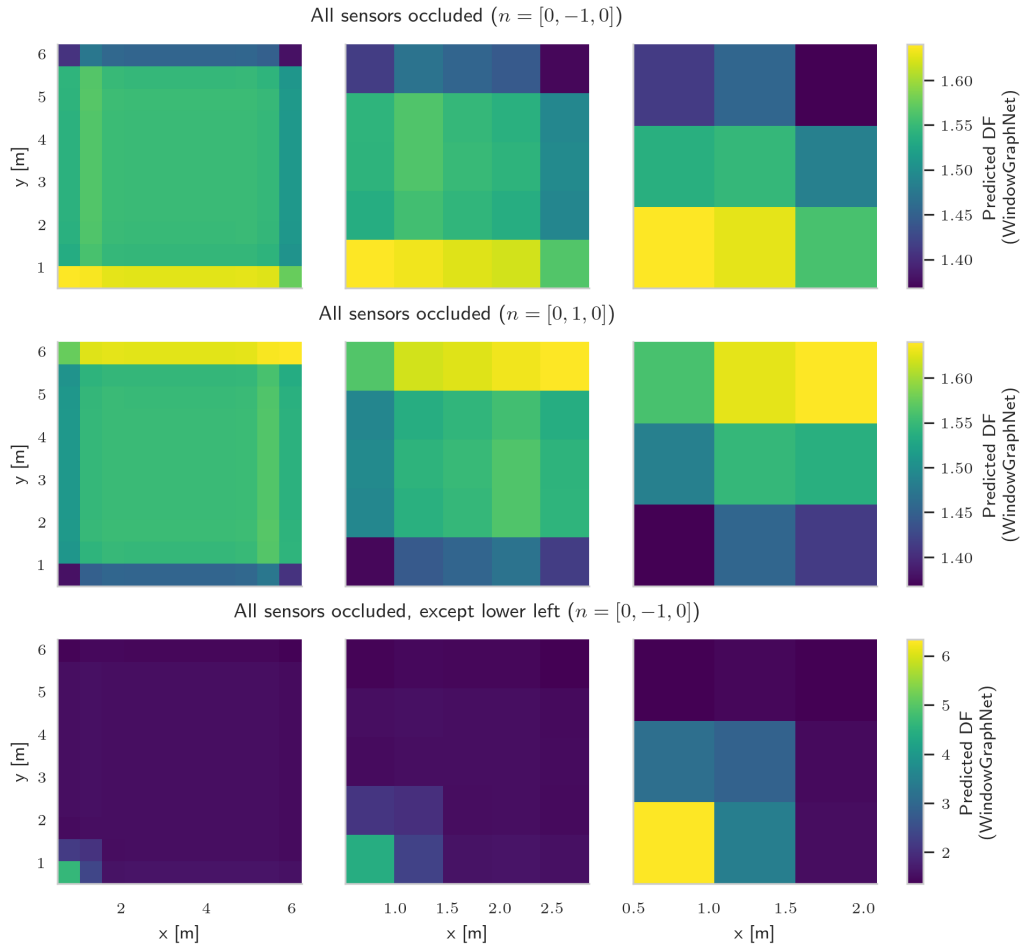


Figure 6.21: No-window study of *WindowGraphNet* predictions under forced occlusion. Each row shows the same three room scales with different visibility conditions. When all window edges are removed (top and middle rows), predictions settle into a nearly uniform sensor–sensor baseline (∼1.35–1.65 DF), with its level shifting according to the directional component n. When a single unoccluded sensor is retained (bottom row), local brightening is confined to a few neighbouring sensors, revealing the finite propagation depth of three message-passing layers.

associated with the window, $\vec{u} = (0, 0, 1)$ is the global vertical direction, and $\vec{t} = \vec{u} \times \vec{n}$ (all scaled and normalised). Even when all window edges are removed, the frame $(\vec{t}, \vec{n})$ remains defined via the window-wall metadata. The channels $\vec{d} \cdot \vec{t}$ and $\vec{d} \cdot \vec{n}$ therefore encode a preferred orientation relative to the notional window direction, whereas $\vec{d} \cdot \vec{u}$ only reflects vertical alignment.

During training, these window-referenced components correlate consistently with increased illumination toward the window wall; in the no-window setting the network thus retains a weak orientation bias, producing small gradients superimposed on the low plateau. In summary, the ∼1.5 DF plateau arises from limited message passing under out-of-distribution connectivity (graphs containing only sensor–sensor edges, unseen during training), while the residual directional pattern is a consequence of the retained $(\vec{d} \cdot \vec{t}, \vec{d} \cdot \vec{n})$ channels defined by $\vec{n}$.

In summary, the ∼1.5 DF plateau arises entirely from limited propagation depth under windowless connectivity. As shown in the last row of Figure 6.21, when only the lower-left sensor remains connected to the window, its influence travels only two to three hops through the sensor–sensor graph before saturating. Beyond this range, all sensors converge to the same baseline value of roughly 1.5 DF, which is the default prediction the network produces whenever no window edges are present during inference.

The subtle orientation visible in the windowless rooms is merely a secondary by-product of the remaining directional channels; it does not determine the plateau value nor the distance at which it appears,

but only adds a slight, anisotropic variation on top of an otherwise uniform field.

This propagation-limited behaviour explains the overestimation in the recessed regions of Tier 5: *WindowGraphNet* correctly identifies occlusion but cannot override the uniform baseline that emerges once explicit window information is no longer available.

Overall, Tier 5 produces smaller *RMSE* variation than Tier 4, as most models converge toward uniformly low DF levels. However, only *WindowGraphNet* and the *Simple Dieguez ANN* correctly reproduce the spatial structure of the illumination field. The former captures where light should reach, while the latter better preserves how much attenuation occurs. This contrast highlights the complementary nature of relational and physically grounded inductive biases: message passing effectively models visibility, whereas global photometric descriptors maintain continuity when the visibility graph becomes disconnected.

6.2.2. Summary

Generalisation in surrogate daylight prediction is governed by how spatial and photometric relationships are represented. The *Raw ANN* relies on absolute Cartesian coordinates and therefore encodes illumination as a fixed spatial template anchored to the global frame, enabling smooth in-distribution interpolation (and limited extrapolation) but failing under rotations or asymmetric layouts. The *Le-Thanh ANN* introduces distance- and angle-based features that provide partial geometric awareness, yet these descriptors are defined in a global coordinate system and remain orientation-sensitive, leading to similar breakdowns under rotated or mirrored configurations. Physically grounded ANNs (*Dieguez* variants) achieve true rotational invariance through scalar photometric descriptors but produce oversmooth fields and lose calibration when feature magnitudes exceed the training range.

In contrast, the proposed *WindowGraphNet* combines rotational invariance with preserved directional gradients, maintaining correct façade orientation and spatial structure across rotations, rectangular layouts, and occlusions. Its main limitation is a systematic magnitude bias under extreme scale changes and a low, visibility-consistent plateau in deeply occluded recesses where message propagation ceases.

Overall, relational and physically grounded encodings generalise most reliably across the six tiers, with *WindowGraphNet* providing the best balance between geometric invariance and spatial fidelity, and the *Simple Dieguez ANN* complementing it as a stable, physically grounded baseline for asymmetric or occluded configurations.

Taken together, these findings demonstrate that geometric generalisation in daylight prediction arises from the form of representation, whether coordinate-based, distance-based, physically grounded, or relational, rather than from architectural depth or parameter count. The comparative evaluation across all six tiers establishes a clear hierarchy of inductive biases: coordinate-based formulations interpolate but fail to transfer, distance-based encodings add partial geometric awareness yet remain orientation-sensitive, physically grounded models encode consistent but simplified behaviour, and relational encodings such as *WindowGraphNet* achieve the highest structural fidelity across transformations and occlusions. The concluding chapter reflects on these results in the broader context of surrogate modelling for architectural design, discussing their implications for model interpretability, integration into design workflows, and potential extensions.

7

Conclusion

Daylight performance is highly sensitive to the geometry of a space. As designers iterate through early-stage massing options, adjust window placement, or explore asymmetric and deeply recessed layouts, the resulting configurations can differ substantially from one moment to the next. Such variability cannot be captured reliably by simplified rules of thumb or intuitive judgement, making physically based simulation the only dependable method for accurate evaluation. Yet Radiance-based workflows remain computationally slow, creating a persistent bottleneck when many design alternatives must be assessed within short iteration cycles.

Surrogate models aim to overcome this limitation by providing rapid, simulation-free approximations of daylight outcomes. Their usefulness, however, depends on more than accuracy in idealised or highly controlled conditions. In practice, surrogate models must remain reliable across a wide range of geometric variations; variations that inevitably arise during early-stage design. Existing ANN-based daylight surrogates have typically been trained and validated on parametrically constrained datasets, and their performance outside these narrow domains has remained largely untested. The issue is therefore not that ANNs are known to perform poorly out of distribution, but that their generalisation ability has never been rigorously examined, despite being essential for real design use.

This thesis addressed this gap by reformulating daylight prediction as a relational learning problem and evaluating surrogate models under systematically increasing geometric deviation. By representing rooms as graphs and encoding geometric relations directly, the proposed GNN-based model was designed to capture how light propagates through space in a way that is sensitive to structure yet robust to global changes in form. The evaluation framework, spanning rotations, scaling, window offsets, and self-occluding L-shaped rooms, tested not only in-distribution accuracy but also whether the learned representations remained physically meaningful when confronted with geometries the model had never seen.

Main Findings and Contributions

In answer to RQ1 (model formulation)

How can DF prediction be represented as a graph learning problem, including the definition of nodes, edges, and feature representations suitable for GNNs? This thesis has shown that a graph-based formulation provides a natural and physically meaningful framework for modelling daylight. Each room was represented as a graph consisting of sensor nodes and window nodes, connected through edges that encode geometric and photometric relationships such as distances and angular orientations. The resulting model, *WindowGraphNet*, captures how light propagates through space by exchanging information between these connected elements. Because all features are expressed in relative, local terms, the formulation remains invariant to global translation and rotation, ensuring that identical daylight patterns are predicted regardless of room orientation. This graph representation therefore bridges the gap between geometric structure and light behaviour, establishing a robust foundation for learning-based daylight prediction.

In answer to RQ2 (architectural exploration)
How do graph operators, model depth, and architectural configuration influence the predictive performance of GNN-based daylight surrogates? A systematic operator benchmark and hyperparameter search revealed that performance depends more strongly on how relational and physically meaningful features are represented and processed than on the specific operator choice itself. A heterogeneous graph structure, distinguishing between window and sensor nodes, consistently improved results compared with homogeneous formulations, as it aligns the model architecture with the underlying feature semantics. The most effective configuration employed a three-layer GENConv with softmax aggregation, which provided stable optimisation and strong performance across architectures. Overall, these findings show that an appropriate combination of feature structure, relational encoding, and moderate network depth is more decisive for accuracy and robustness than model size or operator complexity alone.

In summary, RQ2 is addressed by showing that the alignment between architectural design and feature formulation is the primary determinant of high performance. While operator choice and depth matter, simpler models equipped with informative, well-structured features can often achieve comparable accuracy. This highlights the importance of prioritising feature design and relational structure over increasing architectural complexity.

In answer to RQ3 (generalisation analysis)
To what extent can GNN-based surrogate models generalise to unseen room typologies and geometric transformations compared with ANN-based benchmarks? Results from the Final Test Dataset show that *WindowGraphNet* generalised more reliably to unseen geometries and transformations than all ANN baselines, with the *Simple Dieguez ANN* a close second. Under rotations, coordinate-anchored ANNs degraded sharply (*Raw* and *Le-Thanh* ≈ 5–9 *RMSE*), whereas *WindowGraphNet* and *Simple Dieguez* remained below 0.7 *RMSE*. In the offset-window tier (Tier 3), *WindowGraphNet* achieved 1.93 ± 0.07 *RMSE* (vs. 1.99 ± 0.14 for *Simple Dieguez*); under partial self-occlusion (Tier 4) *WindowGraphNet* was best at 1.78 ± 0.16 *RMSE* with high structural fidelity (*SSIM* 0.86 ± 0.01), while *Le-Thanh* rose to 7.54 ± 1.28 *RMSE*.

For deep self-occlusion (Tier 5), *WindowGraphNet* delivered 2.63 ± 0.54 *RMSE* and preserved the façade–recess gradient far more convincingly than the ANN models. However, its predictions exhibited a consistent plateau around 1.5 DF in fully occluded recesses. This plateau arises because every training sensor retained at least one connection to a window; the model therefore never encountered purely occluded regions, and with limited message-passing depth it converges to a window-independent baseline rather than to darkness. Despite this magnitude bias in extreme cases, the spatial structure in *WindowGraphNet* predictions remained physically coherent, unlike the ANN baselines, which often collapsed to uniform or mis-shaped fields under severe occlusion.

By contrast, in extreme scaling (scale$\times 5$), *WindowGraphNet* overestimated DF magnitudes (3.05 ± 1.79 *RMSE*; ANNs 1.5–3.5), indicating residual sensitivity to absolute scale. Although the *Simple Dieguez ANN* matched *WindowGraphNet* on some aggregates, only *WindowGraphNet* consistently preserved correct spatial structure across all tiers—including asymmetric and L-shaped rooms with self-occlusion. These results support that relational inductive biases, rather than data augmentation alone, enable genuine geometric robustness; *WindowGraphNet* thus offers a more reliable basis for rapid daylight analysis across diverse design variants, with scale invariance and improved attenuation modelling remaining important targets for future refinement.

Contributions and Significance to Daylight Modeling
The outcomes of this research hold important methodological and practical implications for architectural daylight modelling. Methodologically, the work does not aim to produce the most accurate or exhaustive daylight surrogate model, but rather to establish a new foundation for how daylight can be learned from data. By formulating DF prediction as a graph-learning task based on explicit sensor–window relations, the thesis demonstrates that geometric structure and simplified photometric priors can be embedded directly into the model architecture. To our knowledge, this represents the first use of GNNs for DF prediction and the first explicit relational formulation of daylight propagation in a learning context. The resulting inductive biases within *WindowGraphNet*, including translational and rotational invariance and a physically grounded notion of locality, enabled the model to remain stable and meaningful when

tested outside its training distribution. Crucially, the systematic out-of-distribution evaluation conducted in this thesis shows that these relational biases directly influence geometric generalisation, revealing limitations in conventional ANN surrogates that would remain hidden under standard in-distribution testing.

Practically, the research provides an initial but essential step toward real-time, geometry-aware daylight tools for early-stage design. While *WindowGraphNet* is not positioned as a final or production-ready surrogate, it illustrates how graph-based formulations can support instantaneous daylight feedback across diverse spatial configurations. For deployment in architectural practice, the model would need to be trained and optimised on a broader and more representative range of geometries, but the underlying framework is inherently extensible and compatible with parametric modelling environments. With further development and expanded data coverage, this approach could offer a robust and efficient complement to simulation-based analysis, enabling rapid exploration of form and façade decisions in sustainable building design.

Future Research Directions

Building on the framework and insights developed in this thesis, several research directions emerge that can extend the capabilities and applicability of graph-based daylight prediction. The relational formulation and out-of-distribution evaluation introduced here define a foundation rather than a finished model, and the findings point directly to opportunities for deeper physical integration, broader geometric coverage, and larger-scale deployment. Future work may therefore focus on the following developments:

Extending to Multi-Window and Obstruction-Aware Configurations: The current graph formulation assumes a single dominant window normal n, which is reused in the directional components of sensor–sensor edges. This design is consistent with the present single-window setup, but becomes problematic in multi-window rooms, where different apertures can have distinct orientations. A single global n would then impose an incorrect directional prior on sensor–sensor interactions. Future work should therefore investigate alternative encodings for directional information in multi-window settings. Possible strategies include: (i) replacing the global n in sensor–sensor features by a weighted combination of window normals, for example based on distance or visibility, so that each sensor pair is influenced primarily by nearby apertures; (ii) removing n-dependent channels from sensor–sensor edges altogether, retaining them only on window–sensor edges, so that directionality is expressed explicitly with respect to actual apertures rather than implicitly through a global frame; or (iii) redefining the directional frame for sensor–sensor edges in terms of local wall or façade normals derived from geometry, decoupling these features from any particular window orientation. Any of these approaches would allow the model to handle multiple windows without inheriting the single-window bias that underlies the observed plateau behaviour.

Introducing explicit modelling of obstructions constitutes a related extension. In many real buildings, external elements such as adjacent façades, balconies, fins, and overhangs attenuate or occlude portions of the sky. These influences could be represented either through additional node types for obstruction surfaces, which block or modify messages travelling from window nodes to sensors, or through precomputed attenuation descriptors, such as sky-component reduction factors or view-factor percentages evaluated at window corners, and incorporated as window-level features. Both strategies would enable the graph to encode partial visibility and shadowing effects, moving the surrogate model closer to realistic façade and urban-context conditions.

Adapting to Grid Resolution and Irregular Discretisations: A complementary direction for future development concerns the discretisation of the sensor domain. The current model is trained on a fixed, uniform grid resolution, which contributes to the observed sensitivity under geometric scaling: as room dimensions increase, the physical spacing between neighbouring sensors grows, reducing message density and altering the effective receptive field of the network. Training on multiple grid resolutions would help stabilise this behaviour, enabling the surrogate to learn scale-consistent representations rather than inheriting a dependency on a specific sampling density.

A further extension involves supporting irregular sensor layouts. Practical daylight workflows frequently rely on non-uniform point distributions or mixed-element surface meshes, including discretisations containing both quadrilateral and triangular subdivisions. Such layouts depart from the regular Cartesian structure assumed in the present formulation. Extending *WindowGraphNet* to operate robustly on these

irregular discretisations would improve compatibility with real-world simulation pipelines and enhance generalisation across diverse spatial sampling schemes.

Extension of Graph Topology to Improve Attenuation Modelling: A key limitation observed in this study is the overestimation of DF in deeply recessed or fully occluded areas, where illumination should approach zero. This behaviour emerges because the model was trained only on geometries in which every sensor had at least one connection to a window; as a result, the network never encountered purely occluded configurations during training. Under such conditions, limited message-passing depth causes sensors in recesses to converge toward a window-independent baseline rather than learning the correct decay of diffuse light.

One natural avenue for improvement is to expand the training dataset to include recessed and occluded regions explicitly, together with a renewed hyperparameter search optimised for these cases. This would allow the model to learn attenuation patterns directly, without requiring changes to the graph structure.

A complementary direction is to enrich the graph topology itself. Introducing intermediate surface nodes, representing walls, corners, reveals, or façade elements, would create additional message-passing pathways through which information can propagate from exposed regions into deep recesses, approximating multi-hop attenuation more faithfully. Similar extensions can incorporate contextual obstructions by adding nodes for external elements such as adjacent buildings or overhangs, which intercept or attenuate messages before they reach the interior.

Together, these strategies highlight a core advantage of graph-based representations: the modelling framework can be improved either by broadening the training distribution or by extending the graph's physical expressiveness, without redesigning the underlying learning architecture.

Integration with Design Practice and Interactive Tools: Before deployment in design workflows, the limitations identified in this thesis must first be addressed. The restricted training distribution used here was intentional, designed to create a controlled setting for assessing geometric generalisation, but it does not reflect the full geometric variability encountered in practice. Expanding the dataset to include occluded regions, multiple window orientations, and a wider range of room typologies, together with renewed hyperparameter optimisation, would yield a more robust model suitable for practical use. With these enhancements, a GNN-based surrogate could be integrated into parametric modelling environments as a plugin or extension, providing real-time daylight feedback during early-stage design. Practitioner feedback would then be essential to refine usability and evaluate the model's impact in real architectural contexts.

Extension to Climate-Based Daylight Metrics: The current model focuses on DF under a static overcast sky. A natural next step is to extend the approach to climate-based metrics such as daylight autonomy or glare probability. Temporal sky conditions and solar positions could be incorporated into the graph, for example as global or time-dependent features, enabling the model to operate under realistic weather conditions. This addresses the limitation of static-sky analysis and enhances relevance for regulatory frameworks that require annual daylight evaluations.

Concluding Reflection

This thesis set out to examine whether a surrogate model can remain reliable when confronted with the geometric variability inherent to early-stage architectural design. By formulating daylight prediction as a graph-learning problem, and by developing *WindowGraphNet* as a heterogeneous representation of window–sensor interactions, the research demonstrates that relational structure provides a powerful inductive bias for generalising beyond a restricted training domain. The model consistently maintained the spatial character of daylight distributions across unseen layouts where conventional ANN surrogates, anchored to fixed coordinate inputs, frequently failed. In this sense, the study shows that robust generalisation arises not from model size or dataset breadth, but from the appropriateness of the representation itself.

At the same time, the work is deliberately positioned as a methodological foundation rather than a finished predictive tool. The limitations identified, particularly the model's reduced attenuation performance in deeply occluded regions and its sensitivity to extreme geometric scaling, highlight the need

for richer training distributions and potential extensions of the graph topology. Addressing these challenges opens the path toward surrogates that more closely approximate the full physical complexity of daylight propagation.

From a design perspective, these findings suggest a tangible opportunity for strengthening early-stage architectural workflows. Daylight assessments are often delayed until later project phases, when geometry has stabilised and formal alternatives have already narrowed. By providing fast, physically informed predictions directly from coarse geometric input, *WindowGraphNet* supports an analytic mode that can accompany sketch-level exploration rather than follow it. This lowers the threshold for engaging with daylight performance during concept development, enabling designers to test ideas while they are still fluid and to identify promising strategies before committing to detailed modelling. More broadly, such surrogate models can facilitate a clearer dialogue between architects and environmental consultants: rapid, interpretable feedback exposes performance trade-offs early, reducing late corrective iterations and helping ensure that design intent and environmental considerations evolve together rather than in sequence. Although *WindowGraphNet* itself is not yet a production-ready tool, its behaviour demonstrates the potential for integrating fast, relationally grounded daylight reasoning into the earliest stages of design—a direction that future, more mature surrogates may be able to realise in practice.

Taken together, the findings point toward a clear trajectory for integrating machine learning into architectural daylight analysis. By embedding geometric reasoning directly into the model structure, *WindowGraphNet* illustrates how fast, physically grounded daylight feedback can be brought into iterative design environments. Such tools have the potential to support more informed design exploration, enabling architects to balance form, space, and daylight quality with greater confidence. In doing so, this work contributes to the broader pursuit of creating healthier, more comfortable, and more responsive built environments.

References

[1] Nick Baker and Koen Steemers. *Daylight Design of Buildings: A Handbook for Architects and Engineers*. London: James James (Science Publishers) Ltd, 2002. isbn: 978-1902916210.

[2] Anna Wirz-Justice, Debra J. Skene, and Mirjam Münch. “The relevance of daylight for humans”. In: *Biochemical Pharmacology* 191 (2021), p. 114304. doi: 10.1016/j.bcp.2020.114304. url: https://doi.org/10.1016/j.bcp.2020.114304.

[3] Helmut F.O. Müeller. “Daylighting”. In: *Sustainability, Energy and Architecture*. Elsevier, 2014. Chap. 9, pp. 227–256. isbn: 9780123972699.

[4] Berta Garcia-Fernandez and Osama Omar. “Sustainable Performance in Public Buildings Supported by Daylighting Technology”. In: *Solar Energy* 257 (2023), pp. 341–356. doi: 10.1016/j.solener.2023.03.021. url: https://doi.org/10.1016/j.solener.2023.03.021.

[5] N. H. Wong et al. “A review of daylighting design in buildings: strategies, issues, and performance”. In: *Journal of Building Performance (or similar journal)* —.— (2017), pp. —. doi: -.

[6] CubiCasa. *The Golden Circle: The Why, How, and What of Floor Plan Apps*. Accessed: 12 November 2025. 2025. url: https://www.cubi.casa/the-golden-circle-why-how-what-of-floor-plan-apps/.

[7] CubiCasa. *CubiCasa Customer Story: Frank DiGiovanni*. Accessed: 12 November 2025. 2025. url: https://www.cubi.casa/cubicasa-customer-story-frank-digiovanni/.

[8] W. Merghani et al. “Integrating Radiance into grasshopper/honeybee for daylight and energy simulation”. In: *Proceedings of the Building Simulation Conference (or relevant conference)*. Use correct volume/issue/pages as per the paper. —for example, San Francisco, CA: IBPSA / relevant publisher, 2017.

[9] P. Solvang et al. “Surrogate modelling for daylight prediction in architectural design”. In: *Proceedings of the International Building Simulation Conference*. Include correct pages and volume. —: IBPSA / relevant publisher, 2020.

[10] Bianca Williams and Selen Cremaschi. “Novel Tool for Selecting Surrogate Modeling Techniques for Surface Approximation”. In: *Computer Aided Chemical Engineering*. Ed. by M. Türkay and M. Aydin. Vol. 50. 2021, pp. 451–456. doi: 10.1016/B978-0-323-88506-5.50071-1. url: https://par.nsf.gov/servlets/purl/10297247.

[11] Haiyan Zhao et al. “A review of surrogate models and their applications in engineering design”. In: *Journal of Mechanical Design* 134.4 (2012), p. 040801. doi: 10.1115/1.4006116. url: https://doi.org/10.1115/1.4006116.

[12] Shady Attia et al. “Assessing gaps and needs for integrating building performance optimization tools in net zero energy buildings design”. In: *Energy and Buildings* 60 (2013), pp. 110–124. doi: 10.1016/j.enbuild.2013.01.016. url: https://doi.org/10.1016/j.enbuild.2013.01.016.

[13] Saeed Razavi, Bahram A. Tolson, and Donald H. Burn. “Review of surrogate modeling in water resources”. In: *Water Resources Research* 48.7 (2012), W07401. doi: 10.1029/2011WR011527. url: https://doi.org/10.1029/2011WR011527.

[14] Michael M. Bronstein et al. “Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges”. In: *arXiv preprint arXiv:2104.13478* (2021). url: https://arxiv.org/abs/2104.13478.

[15] William L. Hamilton. “Graph Representation Learning”. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 14.3 (2020), pp. 1–159.

[16] Qiuping Liu et al. “A Review and Guide on Selecting and Optimizing Machine Learning Algorithms for Daylight Prediction”. In: *Building and Environment* 244 (2023), p. 110822. doi: 10.1016/j.buildenv.2023.110822. url: https://doi.org/10.1016/j.buildenv.2023.110822.

[17] Yunsong Han, Linhai Shen, and Cheng Sun. “Developing a parametric morphable annual daylight prediction model with improved generalization capability for the early stages of office building design”. In: *Building and Environment* 200 (2021). doi: 10.1016/j.buildenv.2021.107932.

[18] N. Le-Thanh et al. “Machine Learning-based real-time daylight analysis in buildings”. In: *Journal of Building Engineering* 52 (2022). doi: 10.1016/j.jobe.2022.104374.

[19] Dieguez et al. “Daylight Factor Prediction Using Machine Learning”. In: *Building and Environment* 18.2 (2025), pp. 145–160. doi: 10.1016/j.buildenv.2025.02.015. url: https://doi.org/10.1016/j.buildenv.2025.02.015.

[20] Christoph F. Reinhart, John Mardaljevic, and Zack Rogers. “Dynamic Daylight Performance Metrics for Sustainable Building Design”. In: *LEUKOS* 3.1 (2006), pp. 1–25. url: https://nrc-publications.canada.ca/eng/view/object/?id=f4b45bc0-978a-40d1-a6db-bcdff9752929.

[21] U.S. Green Building Council. *LEED v4 for Healthcare - Draft Credit EQc0: Daylight*. https://www.usgbc.org/credits/healthcare/v4-draft/eqc-0. Accessed: 2025-04-10. 2012. url: https://www.usgbc.org/credits/healthcare/v4-draft/eqc-0.

[22] International WELL Building Institute. *Daylight Modeling - WELL Building Standard v2*. https://standard.wellcertified.com/light/daylight-modeling. Accessed: 2025-04-10. 2020. url: https://standard.wellcertified.com/light/daylight-modeling.

[23] European Committee for Standardization (CEN). *NEN_EN_17037_2018+A1:2022 - Daylight in Buildings*. https://www.nen.nl/en/nen-en-17037-2018-en-262178. Dutch implementation of the European standard EN 17037, specifying methods and criteria for daylight in buildings. 2022.

[24] Dutch Green Building Council. *Daglichttoetreding - BREEAM-NL Nieuwbouw en Renovatie 2020 v1.0*. https://richtlijn.breeam.nl/credit/daglichttoetreding-10. Accessed: 2025-04-10. 2023. url: https://richtlijn.breeam.nl/credit/daglichttoetreding-10.

[25] J. Hřáska et al. “Revisiting EN 17037: Stricter daylight requirements and their implications for design”. In: *Lighting Research Technology (or relevant journal)* —.— (2024), pp. —. doi: -.

[26] Qinbo Li and Jeff Haberl. “Prediction of Annual Daylighting Performance Using Inverse Models”. In: *Sustainability* 15.15 (2023), p. 11938. doi: 10.3390/su151511938.

[27] John Mardaljevic. “Validation of a Lighting Simulation Program under Real Sky Conditions”. In: *Lighting Research Technology* 27.4 (1995), pp. 181–188. doi: 10.1177/14771535950270040701.

[28] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997. isbn: 978-0070428072.

[29] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. isbn: 978-0262182539. url: http://www.gaussianprocess.org/gpml/.

[30] Leo Breiman. “Random Forests”. In: *Machine Learning* 45.1 (2001), pp. 5–32. doi: 10.1023/A:1010933404324. url: https://doi.org/10.1023/A:1010933404324.

[31] Jerome H. Friedman. “Greedy Function Approximation: A Gradient Boosting Machine”. In: *Annals of Statistics* 29.5 (2001), pp. 1189–1232. doi: 10.1214/aos/1013203451. url: https://doi.org/10.1214/aos/1013203451.

[32] Kurt Hornik. “Multilayer feedforward networks are universal approximators”. In: *Neural Networks* 2.5 (1989), pp. 359–366. doi: 10.1016/0893-6080(89)90020-8. url: https://doi.org/10.1016/0893-6080(89)90020-8.

[33] Yann LeCun et al. “Gradient-Based Learning Applied to Document Recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. doi: 10.1109/5.726791. url: https://doi.org/10.1109/5.726791.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 25. 2012, pp. 1097–1105. url: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[35] Tom M. Mitchell. *The Need for Biases in Learning Generalizations*. Tech. rep. CBM-TR-117. New Brunswick, NJ, USA: Department of Computer Science, Laboratory for Computer Science Research, Rutgers University, 1980. url: https://apps.dtic.mil/sti/citations/ADA085561.

[36] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521.7553 (2015), pp. 436–444. doi: 10.1038/nature14539. url: https://doi.org/10.1038/nature14539.

[37] Peter W. Battaglia et al. "Relational inductive biases, deep learning, and graph networks". In: *arXiv preprint arXiv:1806.01261* (2018). url: https://arxiv.org/abs/1806.01261.

[38] Franco Scarselli et al. "The Graph Neural Network Model". In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. doi: 10.1109/TNN.2008.2005605. url: https://doi.org/10.1109/TNN.2008.2005605.

[39] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *arXiv preprint arXiv:1609.02907* (2017). doi: 10.48550/arXiv.1609.02907. url: https://arxiv.org/abs/1609.02907.

[40] Fernando Gama, Alejandro Ribeiro, and Joan Bruna. "Convolutional Neural Networks via Graphs". In: *ACM Computing Surveys* 53.4 (2020), pp. 1–36. doi: 10.1145/3398680. url: https://doi.org/10.1145/3398680.

[41] Guohao Li et al. "DeeperGCN: All You Need to Train Deeper GCNs". In: *International Conference on Machine Learning (ICML)*. 2021, pp. 6058–6069. url: https://arxiv.org/abs/2006.07739.

[42] Benjamin Sanchez-Lengeling et al. "A Gentle Introduction to Graph Neural Networks". In: *Distill* (2021). https://distill.pub/2021/gnn-intro. doi: 10.23915/distill.00033.

[43] Luis M. Ruiz, Fernando Gama, and Alejandro Ribeiro. "Graph Neural Networks: Architectures, Stability, and Transferability". In: *Proceedings of the IEEE* 109.5 (2021), pp. 660–682. doi: 10.1109/JPROC.2021.3069965. url: https://doi.org/10.1109/JPROC.2021.3069965.

[44] Raghunathan Ramakrishnan et al. "Quantum chemistry structures and properties of 134 kilo molecules". In: *Scientific Data* 1 (2014), p. 140022. doi: 10.1038/sdata.2014.22.

[45] Weihua Hu et al. "Open Graph Benchmark: Datasets for Machine Learning on Graphs". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020. url: https://arxiv.org/abs/2005.00687.

[46] John J. Irwin and Brian K. Shoichet. *ZINC - A Free Database of Commercially Available Compounds for Virtual Screening*. 2005. doi: 10.1021/ci049714+.

[47] Federico Monti et al. "Geometric Deep Learning on Graphs and Manifolds using Mixture Model CNNs". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5115–5124. url: https://arxiv.org/abs/1611.08402.

[48] William L. Hamilton, Rex Ying, and Jure Leskovec. "Inductive Representation Learning on Large Graphs". In: *arXiv preprint arXiv:1706.02216* (2018). doi: 10.48550/arXiv.1706.02216. url: https://arxiv.org/abs/1706.02216.

[49] Michael M. Bronstein et al. "Geometric Deep Learning: Going beyond Euclidean data". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42. doi: 10.1109/MSP.2017.2693418. url: https://doi.org/10.1109/MSP.2017.2693418.

[50] Muhan Zhang and Yixin Chen. "Link Prediction Based on Graph Neural Networks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2018. arXiv: 1802.09691. url: https://arxiv.org/abs/1802.09691.

[51] Keyulu Xu et al. "How Powerful are Graph Neural Networks?" In: *International Conference on Learning Representations (ICLR)*. 2019. arXiv: 1810.00826. url: https://arxiv.org/abs/1810.00826.

[52] Bharti Khemani et al. "A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions". In: *Journal of Big Data* 11.1 (2024), p. 18. doi: 10.1186/s40537-023-00876-4. url: https://doi.org/10.1186/s40537-023-00876-4.

[53] Christopher Morris et al. “Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 4602–4609. doi: 10.1609/aaai.v33i01.33014602.

[54] Haggai Maron et al. “Provably Powerful Graph Networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2019, pp. 2153–2164. url: https://papers.nips.cc/paper/2019/hash/54f0d77e11b2f1b65d2b3f2d6a8f8a0b-Abstract.html.

[55] Vijay Prakash Dwivedi et al. “Benchmarking Graph Neural Networks”. In: *International Conference on Learning Representations (ICLR)*. 2020. url: https://arxiv.org/abs/2003.00982.

[56] Jiaxuan You, Rex Ying, and Jure Leskovec. “Position-aware Graph Neural Networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2019. arXiv: 1906.04817. url: https://arxiv.org/abs/1906.04817.

[57] Balasubramaniam Srinivasan and Bruno Ribeiro. “On the Equivalence between Positional Node Embeddings and Structural Graph Representations”. In: *8th International Conference on Learning Representations (ICLR 2020)*. arXiv preprint arXiv:1910.00452 (2020). 2020. doi: 10.48550/arXiv.1910.00452. url: https://openreview.net/forum?id=SJxzFySKwH.

[58] Qimai Li, Zhichao Han, and Xiao-Ming Wu. “Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018. url: https://ojs.aaai.org/index.php/AAAI/article/view/11604.

[59] Kenta Oono and Taiji Suzuki. “Graph Neural Networks Exponentially Lose Expressive Power for Node Classification”. In: *International Conference on Learning Representations (ICLR)*. 2020. arXiv: 1905.10947. url: https://openreview.net/forum?id=S1ldO2EFPr.

[60] Uri Alon and Eran Yahav. “On the Bottleneck of Graph Neural Networks and Its Practical Implications”. In: *International Conference on Learning Representations (ICLR)*. 2021. arXiv: 2006.05205. url: https://openreview.net/forum?id=i80OPhOCVH2.

[61] Shaked Brody, Eran Yahav, and Yoav Goldberg. “How Attentive are Graph Attention Networks?” In: *International Conference on Learning Representations (ICLR)*. 2022. url: https://arxiv.org/abs/2105.14491.

[62] T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra. “A Survey on Oversmoothing in Graph Neural Networks”. In: *arXiv preprint arXiv:2303.10993* (2023). url: https://arxiv.org/abs/2303.10993.

[63] Petar Veličković et al. “Graph Attention Networks”. In: *arXiv preprint arXiv:1710.10903* (2018). doi: 10.48550/arXiv.1710.10903. url: https://arxiv.org/abs/1710.10903.

[64] Chuan-Hsuan Lin and Yaw-Shyan Tsay. “A metamodel based on intermediary features for daylight performance prediction of façade design”. In: *Building and Environment* 206 (2021). doi: 10.1016/j.buildenv.2021.108371.

[65] Wang et al. “Unfolding 3D Space into Binary Images for Daylight Simulation via Neural Network”. In: *Journal of Daylighting* 10 (2023), pp. 204–213. doi: 10.15627/jd.2023.17. url: https://doi.org/10.15627/jd.2023.17.

[66] J. Ngarambe et al. “Comparative performance of machine learning algorithms in the prediction of indoor daylight illuminances”. In: *Sustainability* 12 (2020). doi: 10.3390/su12114471.

[67] H. Nourkojouri et al. “Development of a hybrid machine-learning and optimization tool for performance-based solar shading design”. In: *Building and Environment* 206 (2021). doi: 10.1016/j.buildenv.2021.108371.

[68] Charlotte Jeline Kat et al. “Application of multimodal learning in daylight provision and view quality assessment of residential building layouts”. In: *International Journal of Architectural Computing* 22.4 (2024), pp. 605–627. doi: 10.1177/14780771241286614. eprint: https://doi.org/10.1177/14780771241286614. url: https://doi.org/10.1177/14780771241286614.

[69] He et al. “Predictive Models for Daylight Performance of General Floorplans Based on CNN and GAN”. In: *Journal of Architectural Science* 29.4 (2021), pp. 310–325. doi: 10.1016/j.jas.2021.04.007. url: https://doi.org/10.1016/j.jas.2021.04.007.

[70] Phillip Isola et al. “Image-to-Image Translation with Conditional Adversarial Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5967–5976.

[71] Martin Arjovsky, Soumith Chintala, and Léon Bottou. “Wasserstein Generative Adversarial Networks”. In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. PMLR, 2017, pp. 214–223. url: https://proceedings.mlr.press/v70/arjovsky17a.html.

[72] Labib et al. “Utilizing Physics-Informed Neural Networks to Advance Daylighting Simulations in Buildings”. In: *Journal of Sustainable Architecture* 22.1 (2025), pp. 50–65. doi: 10.1016/j.jsa.2025.01.008. url: https://doi.org/10.1016/j.jsa.2025.01.008.

[73] George E. Karniadakis et al. “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440. doi: 10.1038/s42254-021-00314-5.

[74] Zijian Wang, Rafael Sacks, and Timson Yeung. “Exploring graph neural networks for semantic enrichment: Room type classification”. In: *Automation in Construction* 134 (2022), p. 104039. doi: 10.1016/j.autcon.2021.104039.

[75] Adam Buruzs et al. “IFC BIM Model Enrichment with Space Function Information Using Graph Neural Networks”. In: *Energies* 15.8 (2022), p. 2937. doi: 10.3390/en15082937.

[76] Navid Kayhani, Brenda McCabe, and Bharath Sankaran. “Semantic-aware quality assessment of building elements using graph neural networks”. In: *Automation in Construction* 155 (2023), p. 105054. doi: 10.1016/j.autcon.2023.105054.

[77] Rini Jasmine Gladstone et al. “Mesh-based GNN surrogates for time-independent PDEs”. In: *Scientific Reports* 14.1 (2024), p. 3394. doi: 10.1038/s41598-024-53185-y.

[78] Yongcheng Li, Changsheng Wang, and Wenbin Hou. “A universal surrogate modeling method based on heterogeneous graph neural network for nonlinear analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 437 (2025), p. 117793. doi: 10.1016/j.cma.2025.117793.

[79] Eamon Jasper Whalen. “Enhancing surrogate models of engineering structures with graph-based and physics-informed learning”. PhD thesis. Cambridge, MA: Massachusetts Institute of Technology, 2021. doi: 10.48550/arXiv.2106.12345.

[80] J. Storm et al. “A microstructure-based graph neural network for full-field microscopic strain prediction”. In: *Computers Structures* 323 (2024), p. 107608. doi: 10.1016/j.compstruc.2024.107608. url: https://doi.org/10.1016/S0045-7825(24)00257-3.

[81] Yongzheng Liu et al. “BuildSTG: A Multi-building Energy Load Forecasting Method using Spatio-Temporal Graph Neural Network”. In: *arXiv preprint arXiv:2507.20838* (2025). url: https://arxiv.org/abs/2507.20838.

[82] Samuel Moveh et al. “Multi-Building Energy Forecasting Through Weather-Integrated Temporal Graph Neural Networks”. In: *Buildings* 15.5 (2025), p. 808. doi: 10.3390/buildings15050808.

[83] Dilong Li et al. “Graph Neural Networks in Point Clouds: A Survey”. In: *Remote Sensing* 16.14 (2024), p. 2518. doi: 10.3390/rs16142518.

[84] Xiaolei Chen et al. “GS-Net: Point cloud sampling with graph neural networks”. In: *Pattern Recognition* 170 (2026), p. 112054. doi: 10.1016/j.patcog.2025.112054.

[85] Binyu Lei et al. “Predicting building characteristics at urban scale using graph neural networks and street-level context”. In: *Computers, Environment and Urban Systems* 111 (2024), p. 102129. doi: 10.1016/j.compenvurbsys.2024.102129.

[86] Zhaoji Wu et al. “Developing surrogate models for the early-stage design of residential blocks using graph neural networks”. In: *Building Simulation* 18.3 (2025), pp. 679–698. doi: 10.1007/s12273-025-1237-7.

[87] Hyejin Park et al. “Floor plan recommendation system using graph neural network with spatial relationship dataset”. In: *Journal of Building Engineering* 71 (2023), p. 106378. doi: 10.1016/j.jobe.2023.106378.

[88] Malik M. Barakathullah and Immanuel Koh. “Prediction of Usage Probabilities of Shopping-Mall Corridors Using Heterogeneous Graph Neural Networks”. In: *arXiv preprint* (2025). arXiv: 2504.07645 [cs.LG].

[89] Petar Veličković. “Everything is Connected: Graph Neural Networks”. In: *Current Opinion in Structural Biology* 79 (2023), p. 102538. doi: 10.1016/j.sbi.2023.102538. url: https://arxiv.org/abs/2301.08210.

[90] Justin Gilmer et al. “Neural Message Passing for Quantum Chemistry”. In: *International Conference on Machine Learning (ICML)*. 2017, pp. 1263–1272. url: https://arxiv.org/abs/1704.01212.

[91] Martin Simonovsky and Nikos Komodakis. “Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3693–3702. doi: 10.1109/CVPR.2017.393. url: https://openaccess.thecvf.com/content_cvpr_2017/html/Simonovsky_Dynamic_Edge-Conditioned_Filters_CVPR_2017_paper.html.

[92] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. “E(n) Equivariant Graph Neural Networks”. In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. 2021, pp. 9323–9332. url: https://arxiv.org/abs/2102.09844.

[93] Fleur Hendriks et al. “Similarity equivariant graph neural networks for homogenization of metamaterials”. In: *Computer Methods in Applied Mechanics and Engineering* 439 (2025), p. 117867. doi: 10.1016/j.cma.2025.117867. url: https://doi.org/10.1016/j.cma.2025.117867.

[94] Isabelle Guyon and André Elisseeff. “An introduction to variable and feature selection”. In: *Journal of Machine Learning Research* 3 (2003), pp. 1157–1182. url: https://www.jmlr.org/papers/v3/guyon03a.html.

[95] Matthias Fey et al. “SplineCNN: Fast Geometric Deep Learning with Continuous B-Spline Kernels”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 869–877. url: https://arxiv.org/abs/1711.08920.

[96] Gabriele Corso et al. “Principal Neighbourhood Aggregation for Graph Nets”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020. url: https://arxiv.org/abs/2004.05718.

[97] Jiaxuan You, Rex Ying, and Jure Leskovec. “Design Space for Graph Neural Networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[98] Tian Xie and Jeffrey C Grossman. “Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties”. In: *Physical Review Letters* 120.14 (2018), p. 145301. doi: 10.1103/PhysRevLett.120.145301.

[99] Xavier Bresson and Thomas Laurent. “Residual Gated Graph ConvNets”. In: *arXiv preprint arXiv:1711.07553* (2018).

[100] Zhen Luo et al. “Can Classic GNNs Be Strong Baselines for Graph-level Tasks? Simple Architectures Meet Excellence”. In: *arXiv preprint arXiv:2501.02506* (2025). url: https://arxiv.org/abs/2501.02506.

[101] Petar Veličković et al. “Graph Attention Networks”. In: *International Conference on Learning Representations (ICLR)*. 2018. url: https://arxiv.org/abs/1710.10903.

[102] Meng Shi et al. “Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 1085–1093.

[103] Weihua Hu et al. “Strategies for pre-training graph neural networks”. In: *International Conference on Learning Representations (ICLR)*. 2020. url: https://arxiv.org/abs/1905.12265.

[104] Xin Li et al. “PDN: Position-aware Differentiable Neighborhood for Graph Neural Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[105] Andrew Kachites McCallum et al. “Automating the construction of internet portals with machine learning”. In: *Information Retrieval* 3.2 (2000), pp. 127–163.

[106] Michail Chatzianastasis, Giannis Nikolentzos, and Michalis Vazirgiannis. “Supervised Attention Using Homophily in Graph Neural Networks”. In: *arXiv preprint arXiv:2307.05217* (2023). url: https://arxiv.org/abs/2307.05217.

[107] Jiaxuan You, Rex Ying, and Jure Leskovec. “Design Space for Graph Neural Networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2021. url: https://arxiv.org/abs/2011.08843.

[108] Robert A. Mangkuto, F. Rohmah, and F.X. Soelami. “The Effects of Orientation, Window Size, and Lighting Control on Daylight and Energy Performance”. In: *Proceedings of Building Simulation 2019*. International Building Performance Simulation Association (IBPSA), 2019, p. 210677. url: https://publications.ibpsa.org/proceedings/bs/2019/papers/BS2019_210677.pdf.

[109] Stephen Simm and David Coley. “The relationship between wall reflectance and daylight factor in real rooms”. In: *Architectural Science Review* 54.4 (2011), pp. 329–334. doi: 10.1080/00038628.2011.613642. url: https://doi.org/10.1080/00038628.2011.613642.

[110] Luiz Bittencourt, Renato Cordeiro, and Virginia Melo. “Daylighting of Classrooms: Analysis of Designs and Simulation Results”. In: *Proceedings of the 7th International IBPSA Conference, Building Simulation 2001*. Rio de Janeiro, Brazil: International Building Performance Simulation Association (IBPSA), 2001, pp. 1349–1356. url: https://publications.ibpsa.org/proceedings/bs/2001/papers/bs2001_1349_1356.pdf.

[111] Ignacio Acosta, Miguel Á. Campano, and Jorge F. Molina. “Analysis of Daylighting and Energy Saving in Offices: Dynamic Daylight Metrics and Artificial Lighting Control Systems”. In: *Energies* 11.11 (2018), p. 3143. doi: 10.3390/en11113143. url: https://www.mdpi.com/1996-1073/11/11/3143.

[112] Fatma El Hussainy, Soad El-Sayed, and Hossam Ibrahim. “Performance-Based Simulation for Enhancing Daylight and Energy Efficiency in Educational Spaces”. In: *International Journal of Energy Engineering* 10.1 (2020), pp. 27–34. doi: 10.18576/ijee/100104. url: https://www.naturalspublishing.com/download.asp?ArtcID=22102.

[113] Lutz Prechelt. “Early Stopping—But When?” In: *Neural Networks: Tricks of the Trade*. Vol. 1524. LNCS. Springer, 1998, pp. 55–69.

[114] Chengxuan Ying et al. “Do Transformers Really Perform Bad for Graph Representation?” In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 34. 2021, pp. 28877–28888.

[115] Inyeop Hwang et al. “On the Analysis of Oversmoothing in Graph Neural Networks”. In: *Proceedings of the 39th International Conference on Machine Learning (ICML)*. PMLR, 2022, pp. 9377–9412.

[116] Randall J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. Society for Industrial and Applied Mathematics (SIAM), 2007. doi: 10.1137/1.9780898717839.

[117] Michael Schlichtkrull et al. “Modeling Relational Data with Graph Convolutional Networks”. In: *The Semantic Web – ISWC 2018*. Vol. 10843. Lecture Notes in Computer Science. Springer, 2018, pp. 593–607. doi: 10.1007/978-3-030-00668-6_38.

[118] Johannes Klicpera, Janek Groß, and Stephan Günnemann. “Directional Message Passing for Molecular Graphs”. In: *International Conference on Learning Representations (ICLR)*. 2020.

[119] Johannes Klicpera, Florian Becker, and Stephan Günnemann. “Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. 2020, pp. 14472–14483.

[120] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. “E(n) Equivariant Graph Neural Networks”. In: *International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 9323–9332.

[121] Vasileios Machairas, Aris Tsangrassoulis, and Kyriakoula Axarli. “Performance-driven architectural design and optimization techniques: A review”. In: *Solar Energy* 77.3 (2014), pp. 319–335. doi: 10.1016/j.solener.2014.01.023.

[122] Diederik P Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *ICLR* (2015).

[123] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research (JMLR)*. Vol. 15. 2014, pp. 1929–1958.

[124] Nitish Shirish Keskar et al. “On large-batch training for deep learning: Generalization gap and sharp minima”. In: *arXiv preprint arXiv:1609.04836* (2017).

[125] Dominic Masters and Carlo Luschi. “Revisiting small batch training for deep neural networks”. In: *arXiv preprint arXiv:1804.07612* (2018).

[126] Elad Hoffer, Itay Hubara, and Daniel Soudry. “Train longer, generalize better: Closing the generalization gap in large batch training of neural networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017.

[127] Takuya Akiba et al. “Optuna: A Next-generation Hyperparameter Optimization Framework”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.

[128] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. “Estimating mutual information”. In: *Physical Review E* 69.6 (2004), p. 066138.

[129] Hanchuan Peng, Fuhui Long, and Chris Ding. “Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (2005), pp. 1226–1238. doi: 10.1109/TPAMI.2005.159.

[130] James S. Bergstra et al. “Algorithms for hyper-parameter optimization”. In: *Advances in Neural Information Processing Systems*. Vol. 24. 2011, pp. 2546–2554.

[131] Takuya Akiba et al. “Optuna: A next-generation hyperparameter optimization framework”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. 2019, pp. 2623–2631.

[132] James Bergstra, Daniel Yamins, and David D. Cox. “Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures”. In: *Proceedings of the 30th International Conference on Machine Learning (ICML ’13)*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 1. Atlanta, Georgia, USA: PMLR, 2013, pp. 115–123. url: https://proceedings.mlr.press/v28/bergstra13.html.

[133] Shuhei Watanabe. “Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance”. In: *arXiv preprint arXiv:2304.11127* (2023). url: https://arxiv.org/abs/2304.11127.

[134] Kalyanmoy Deb et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197.

[135] Joshua Knowles and David Corne. “Pareto archived evolution strategy: A fast and elitist multiobjective evolutionary algorithm”. In: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197.

[136] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2016, pp. 3844–3852.

[137] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Deep Sparse Rectifier Neural Networks”. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 15. JMLR Workshop and Conference Proceedings. 2011, pp. 315–323. url: https://proceedings.mlr.press/v15/glorot11a.html.

[138] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[139] Weihua Hu et al. “Open Graph Benchmark: Datasets for Machine Learning on Graphs”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020. url: https://arxiv.org/abs/2005.00687.

[140] Ming Chen et al. “Measuring and Relieving the Over-smoothing Problem for Graph Neural Networks from the Topological View”. In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2020, pp. 3438–3445.

[141] Anders Krogh and John A Hertz. “A simple weight decay can improve generalization”. In: *Advances in neural information processing systems* 4 (1992), pp. 950–957.

[142] James Bergstra and Yoshua Bengio. “Random Search for Hyper-Parameter Optimization”. In: *Journal of Machine Learning Research* 13 (2012), pp. 281–305.

[143] James Bergstra, Daniel Yamins, and David D. Cox. “Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures”. In: *International Conference on Machine Learning (ICML)*. 2013, pp. 115–123.

[144] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *International Conference on Machine Learning (ICML)*. 2015, pp. 448–456.

[145] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. “Layer Normalization”. In: *arXiv preprint arXiv:1607.06450*. 2016. url: https://arxiv.org/abs/1607.06450.

[146] Vinod Nair and Geoffrey E. Hinton. “Rectified Linear Units Improve Restricted Boltzmann Machines”. In: *Proceedings of the 27th International Conference on Machine Learning (ICML)*. 2010, pp. 807–814.

[147] Dan Hendrycks and Kevin Gimpel. “Gaussian Error Linear Units (GELUs)”. In: *arXiv preprint arXiv:1606.08415* (2016). url: https://arxiv.org/abs/1606.08415.

[148] Günter Klambauer et al. “Self-Normalizing Neural Networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017, pp. 971–980.

[149] Sitao Luan et al. “On the Power of Graph Neural Networks and the Role of the Activation Function”. In: *Transactions on Machine Learning Research (TMLR)* (2022). url: https://arxiv.org/abs/1905.13138.

[150] Kalyanmoy Deb et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE Transactions on Evolutionary Computation*. Vol. 6. 2. 2002, pp. 182–197.

[151] Christopher Meek, Bo Thiesson, and David Heckerman. “The Learning-Curve Sampling Method Applied to Model-Based Clustering”. In: *Journal of Machine Learning Research* 2 (2002), pp. 397–418. url: https://www.jmlr.org/papers/volume2/meek02a/meek02a.pdf.

[152] Jared Hestness et al. “Deep Learning Scaling is Predictable, Empirically”. In: *arXiv preprint arXiv:1712.00409* (2017). url: https://arxiv.org/abs/1712.00409.

[153] Jared Kaplan et al. “Scaling Laws for Neural Language Models”. In: *arXiv preprint arXiv:2001.08361* (2020). url: https://arxiv.org/abs/2001.08361.

[154] Prajit Ramachandran, Barret Zoph, and Quoc V Le. “Searching for activation functions”. In: *arXiv preprint arXiv:1710.05941* (2017).

[155] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. 2019, pp. 4171–4186.

[156] Matthias Fey and Jan E. Lenssen. “Fast Graph Representation Learning with PyTorch Geometric”. In: *ICLR Workshop on Representation Learning on Graphs and Manifolds*. 2019. url: https://arxiv.org/abs/1903.02428.

[157] Gabriele Corso et al. “Principal Neighbourhood Aggregation for Graph Nets”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. 2020, pp. 13260–13271. url: https://proceedings.neurips.cc/paper/2020/hash/c5c3b0cfd182a20282e6e32d769c64b8-Abstract.html.

A

Benchmark Datasets for Operator Evaluation

To support the selection of suitable GNN operators for DF prediction, this appendix presents a summary of benchmark datasets commonly used in graph learning literature. While no existing dataset directly addresses DF prediction, several provide relevant analogies in terms of graph structure, edge feature types, and task formulation (e.g., node-level regression with geometric dependencies).

The datasets listed in Table A.1 have been selected based on their prevalence in the evaluation of edge-aware GNN operators and their structural resemblance to the DFprediction task. The table provides a high-level summary of each dataset's task type, feature composition, and relevance to the modelling of spatial and physical interactions characteristic of daylight simulations. A more detailed description of each dataset follows below.

Table A.1: Benchmark datasets and their structural similarity to DF prediction tasks.

Dataset	Task Type	Node Features	Edge Features	Similarity to DF Task
Cora [105]	Node classification	Binary word presence	Unweighted citation links (binary adjacency)	Canonical benchmark for semi-supervised node classification; while features are textual rather than geometric, the setup parallels predicting daylight at sensors from relational structure.
ZINC [46]	Graph regression	Atom types	Bond types (categorical)	Structured molecular geometry; widely used for evaluating edge-aware regressors.
MNIST Superpixels [47]	Graph classification	Pixel intensities	2D spatial adjacency	Validates spatial GNN kernels (e.g., SplineConv); 2D geometric graph structure.
OGBN-Proteins [45]	Node classification	Protein embeddings	Real-valued edge weights (8 types)	Node-level prediction using weighted interactions; analogous to predicting daylight at sensors from weighted window influences.
OGBL-PPA [45]	Link prediction	58-D one-hot species vector	Unweighted binary edges	Tests graph topology learning; less edge-feature detail but relevant for relational reasoning in DF graphs.
OGBG-MOLHIV [45]	Graph classification	Atom features (atomic number, chirality, etc.)	Bond features (type, stereochemistry, conjugation)	Continuous and categorical edge features; edge-aware molecular graphs parallel DF's geometric edge conditioning.

Cora

The Cora dataset is a citation network widely used as a benchmark for semi-supervised node classification [105]. It consists of 2,708 scientific publications represented as nodes, each described by a 1,433-dimensional bag-of-words feature vector indicating the presence or absence of specific terms. Edges denote citation links between papers and are treated as unweighted binary connections. The

classification task is to assign each paper to one of seven machine learning categories, such as neural networks, probabilistic methods, or reinforcement learning.

Cora remains one of the most canonical datasets for evaluating GNN architectures, especially in semi-supervised and transductive learning settings [39, 48]. While its features are textual rather than geometric, the dataset shares structural similarities with the DF task: predictions are made at the node level, and accuracy depends on propagating information across the relational graph. As such, Cora is included here as a baseline reference point for node-level classification benchmarks, complementing geometry-driven datasets by highlighting the role of relational structure independent of spatial features.

ZINC

ZINC is a molecular graph dataset used for graph-level property prediction, particularly focused on molecular solubility and chemical reactivity [46]. Each molecule is encoded as a graph with categorical node (atom) types and edge types representing chemical bonds. While these edge features are discrete, the task still relies on messagepassing mechanisms to capture local interactions.

ZINC is among the most widely used benchmarks for evaluating GNN architectures, including edge-aware models such as PNA, GINE, and NNConv [90, 96, 103]. Despite its discrete edge features, strong performance on ZINC is typically associated with models that effectively leverage edge-conditioned aggregation. This relevance translates to the DF task insofar as spatial and physical relationships must be embedded into the edge-processing pipeline of a GNN. As such, ZINC serves as a useful benchmark to compare the generalization and expressiveness of candidate GNN operators.

MNIST Superpixels

The MNIST Superpixels dataset is a graph-structured version of the original MNIST digit classification dataset [47]. In this representation, each image is transformed into a graph where nodes correspond to superpixels (grouped pixels with similar intensity), and edges are formed based on spatial proximity in the image plane. The node features encode local pixel intensities, while edge features often encode 2D geometric adjacency or relative coordinates.

Although MNIST Superpixels is a classification task rather than a regression task, it serves as a standard benchmark for evaluating spatially-aware GNN architectures, particularly those that rely on kernel-based edge functions (e.g., SplineConv, GMMConv) [47, 95]. Its use of 2D spatial connectivity and geometry-aligned edge features offers a conceptual bridge to indoor DF modeling, where edge features likewise reflect local spatial structure. As such, it is included for its utility in assessing the performance of convolutional operators in geometry-driven graph domains.

OGB Datasets

The Open Graph Benchmark (OGB) [45] provides a suite of large-scale, standardized graph datasets spanning multiple domains, including biology, chemistry, and social networks. These datasets are widely used to evaluate and compare Graph Neural Network architectures under consistent preprocessing, data splits, and evaluation metrics. They cover a variety of prediction tasks—node classification, link prediction, and graph-level property prediction—making them well-suited for assessing model generality and robustness.

For this study, three OGB datasets were selected that are particularly relevant to DF prediction. Each of them involves rich, real-valued edge attributes and requires the model to learn from graphs where edge features carry meaningful, domain-specific signals. This aligns closely with the current application, where continuous geometric relationships (e.g., distances, angles, solid angles) between windows and sensors are central to accurate prediction. Moreover, these datasets span different prediction settings—node-level, link prediction, and graph-level—which allows us to assess operator performance across task types.

- **ogbl-ppa** — A protein–protein association graph with nodes representing proteins from 58 species and edges indicating biologically relevant relationships such as interactions or co-expression. Each node has a 58-dimensional one-hot vector encoding its species. The task is link prediction: predicting new association edges from the training graph.
- **ogbn-proteins** — A weighted, undirected protein–protein association graph with nodes from 8 species and edges representing 8 biological relationship types, each weighted between 0 and 1.

The task is multi-label protein function classification (112 labels), evaluated by average ROC-AUC. This is analogous to predicting daylight intensity at sensor nodes based on spatially weighted influences from windows.

- **ogbg-molhiv** — A molecular property prediction benchmark where each graph represents a molecule. Nodes are atoms (9 features including atomic number, chirality, and other properties) and edges are bonds (3 features including type, stereochemistry, and conjugation). The task is to predict whether a molecule inhibits HIV replication, evaluated by ROC-AUC.

B

Model Complexity Derivation and Edge Integration

In this appendix, the detailed parameter-count derivations for each GNN operator considered in this study are presented. The goal is to obtain both the exact per-layer parameter counts and their asymptotic growth rates in $\Theta(\cdot)$ notation, focusing on the operator-only contribution to model complexity. By operator-only, it is meant that the shared input encoder and the final head are excluded, allowing for a fair comparison across architectures independent of dataset-specific input/output sizes.

B.1. Model complexity derivation

The derivations are performed under a unified notation: The hidden node-feature dimension, denoted by d, corresponds to the node feature vector as $\mathbf{h}_i \in \mathbb{R}^d$ (evaluated in a homogeneous setting with $d_{\text{in}} = d_{\text{out}} = d$), e is the edge-feature dimension; for each edge i, j, the associated feature vector is $\mathbf{e}_{ij} \in \mathbb{R}^e$. K is the number of layers, H for the number of attention heads (where applicable). For head concatenation, the per-head width is $d_{\text{head}} = d/H$. When an edge-MLP is present, its hidden width is denoted by h_e. R is the number of spline control points, and C is the number of Gaussian components for GMM-style convolutions. Unless otherwise stated, parameter counts include biases in the exact formulas; however, the asymptotic $\Theta(\cdot)$ expressions, terms of order $\mathcal{O}(d)$ are omitted when dominated by $\mathcal{O}(d^2)$ or $\mathcal{O}(ed)$.

Scope. Heterogeneous in/out channel adapters are not considered inside the convolution blocks for this comparison, as the goal is to isolate operator-intrinsic complexity. Per-relation formulas for heterogeneous settings are omitted for brevity.

For each operator, the:

1. Constituent learnable transformations are identified (e.g., linear projections for node and edge features, gating functions, kernel parameters)
2. Exact parameter count per layer are derived from the shape of these transformations.
3. Corresponding Θ-class complexity over K layers.

To ensure reproducibility and avoid discrepancies from implementation details, all derivations are cross-checked against the official PyTorch Geometric source code [156]. The results serve two purposes: first, to quantify differences in computational and storage demands among candidate operators; and second, to inform the trade-offs between predictive accuracy and model efficiency discussed in the main text.

B.1.1. ResGatedConv (Residual Gated Graph Convolution)

The ResGatedConv layer extends a standard message-passing architecture with (i) a learnable gating mechanism applied to edge-conditioned messages, and (ii) an additive residual connection from the

root node [99]. In the PyTorch Geometric implementation (`torch_geometric.nn.ResGatedGraphConv`), the gating coefficients are computed from key and query projections of the concatenated node and edge features, and are applied element-wise to a value projection before aggregation.

Learnable components (default configuration)

1. **Key projection**
$$\text{lin\_key} : \mathbb{R}^{d+e} \to \mathbb{R}^d$$
Parameters: $(d+e) \cdot d + d = d^2 + ed + d$
2. **Query projection**
$$\text{lin\_query} : \mathbb{R}^{d+e} \to \mathbb{R}^d$$
Parameters: same as key: $d^2 + ed + d$
3. **Value projection**
$$\text{lin\_value} : \mathbb{R}^{d+e} \to \mathbb{R}^d$$
Parameters: same as key: $d^2 + ed + d$
4. **Residual/skip connection** (if `root_weight=True`)
$$\text{lin\_skip} : \mathbb{R}^d \to \mathbb{R}^d$$
Parameters: d^2 (with `bias=False`)
5. **Output bias** (if `bias=True`)
$$\mathbf{b} \in \mathbb{R}^d$$
Parameters: d

Per-layer parameter count Assuming `root_weight=True` and `bias=True`:

$$\begin{aligned} N_{\text{layer}} &= 3(d^2 + ed + d) + d^2 + d \\ &= 3d^2 + 3ed + 3d + d^2 + d \\ &= 4d^2 + 3ed + 4d \end{aligned}$$

Complexity class
Bias terms scale as $\mathcal{O}(d)$ and are dominated by quadratic and bilinear terms. Hence:

$$N_{\text{layer}} \in \Theta(d^2 + ed)$$

Correspondence to message-passing formulation
Given node features $\mathbf{h}_i, \mathbf{h}_j \in \mathbb{R}^d$ and edge features $\mathbf{e}_{ij} \in \mathbb{R}^e$, the update is computed as:

$$\mathbf{h}_i^{(l+1)} = \mathbf{W}_{\text{skip}}^{(l)} \mathbf{h}_i^{(l)} + \bigoplus_{j \in \mathcal{N}(i)} \sigma\Big(\mathbf{W}_k^{(l)}[\mathbf{h}_i^{(l)} \,\|\, \mathbf{e}_{ij}^{(l)}] + \mathbf{W}_q^{(l)}[\mathbf{h}_j^{(l)} \,\|\, \mathbf{e}_{ij}^{(l)}]\Big) \odot \mathbf{W}_v^{(l)}[\mathbf{h}_j^{(l)} \,\|\, \mathbf{e}_{ij}^{(l)}]$$

where σ is a sigmoid nonlinearity and $\|$ denotes feature concatenation. This corresponds exactly to the three $(d+e) \to d$ projections and the optional residual connection.

B.1.2. GENConv (Generalized Graph Convolution)

The GENConv layer from DeeperGCN applies a generalized aggregation (e.g., softmax or power-mean) to edge-conditioned messages and then feeds the aggregated signal through a post-aggregation MLP [41]. In the PyTorch Geometric implementation (`torch_geometric.nn.conv.GENConv`), edge attributes are optionally projected to the hidden width and added to neighbor node features before aggregation; after aggregation (and residual addition), an MLP with configurable depth and expansion factor produces the output. Its message construction is:

$$\mathbf{h}_i^{(l+1)} \;=\; \mathrm{MLP}^{(l)}\left(\mathbf{h}_i^{(l)} \;+\; \bigoplus_{j \in \mathcal{N}(i)} \Big\{\mathrm{ReLU}\big(\mathbf{h}_j^{(l)} + \mathbf{e}_{ij}^{(l)}\big) + \varepsilon\Big\}\right).$$

Learnable components (default configuration)

1. **Edge projection (optional)**

$$\text{lin\_edge}: \mathbb{R}^{e} \to \mathbb{R}^{d}$$

Parameters: $e \cdot d + d \;=\; ed + d$. *Note.* If $e = d$, an identity may be used and this term contributes 0.

2. **Post-aggregation MLP**

$$\text{MLP}: \underbrace{[d \;\to\; ds \;\to\; \cdots \;\to\; ds \;\to\; d]}_{L \text{ linear layers, expansion } s}$$

Weights:

$$d{\cdot}(ds) \;+\; (L{-}2)\,(ds){\cdot}(ds) \;+\; (ds){\cdot}d \;=\; 2d^2s \;+\; (L{-}2)d^2s^2.$$

Biases:

$$(ds) \;+\; (L{-}2)(ds) \;+\; d \;=\; (L{-}1)ds \;+\; d.$$

Total MLP parameters:

$$d^2\big(2s + (L{-}2)s^2\big) \;+\; d\big((L{-}1)s + 1\big).$$

3. **Aggregator output adapter (optional)**

$$\text{lin\_aggr\_out}: \mathbb{R}^{d_{\text{agg}}} \to \mathbb{R}^{d}$$

Parameters: $d_{\text{agg}}d + d$.

Note. Instantiated only when a multi-aggregator changes channel count; absent for a single aggregator.

Per-layer parameter count (single aggregator). Let L be the MLP depth and s its expansion. With edge projection active ($e \neq d$) and ignoring optional adapters that do not instantiate under this setting,

$$N_{\text{layer}} = \underbrace{(ed + d)}_{\text{lin\_edge}} + \underbrace{\Big[d^2\big(2s + (L{-}2)s^2\big) + d\big((L{-}1)s + 1\big)\Big]}_{\text{MLP}} \quad \underbrace{\big[+\,(d_{\text{agg}}d + d)\big]}_{\text{if adapter present}}$$

A common choice $L{=}2$, $s{=}2$ with a single aggregator, yields:

$$N_{\text{layer}} \;=\; (ed + d) \;+\; \big(4d^2 + 3d\big) \;=\; 4d^2 \;+\; ed \;+\; 4d$$

Complexity class

Bias terms are $\mathcal{O}(d)$ and are dominated by $\mathcal{O}(d^2)$ and $\mathcal{O}(ed)$. Hence:

$$N_{\text{layer}} \in \Theta\big(d^2 + ed\big)\,.$$

Correspondence to message-passing formulation

The PyG implementation computes messages as $\mathrm{ReLU}(\mathbf{h}_j + \tilde{\mathbf{e}}_{ji}){+}\varepsilon$ where $\tilde{\mathbf{e}}_{ji} = \text{lin\_edge}(\mathbf{e}_{ji})$ if $e \neq d$; the aggregated result is residually combined with $\mathbf{h}_i$ and passed through the MLP. This corresponds exactly to the counted edge projection and post-aggregation MLP.

B.1.3. GeneralConv (General Graph Convolution)

The General Graph Convolution layer (`torch_geometric.nn.conv.GeneralConv`) unifies several GNN architectures by enabling flexible fusion (additive or multiplicative) of node and edge features, followed by a message MLP. It supports optional input/output adapters when node feature dimensions vary, but these are excluded here due to our assumption of a homogeneous setting with $d_{\text{in}} = d_{\text{out}} = d$.

Learnable components (homogeneous, default configuration)

1. **Edge projection (optional)**
$$\text{lin\_edge} : \mathbb{R}^e \to \mathbb{R}^d$$
Parameters: $ed + d$ Used when $e \neq d$ or explicit projection is enabled.
2. **Message MLP** After fusing source, destination, and edge features (typically by addition), the fused message is passed through an MLP:
$$\text{MLP} : [\, d \to ds \to \cdots \to ds \to d \,]$$
Parameters: Weights: $2d^2s + (L-2)d^2s^2$ Biases: $(L-1)ds + d$ Total:
$$d^2(2s + (L-2)s^2) + d((L-1)s + 1)$$
3. **Input adapters (omitted)** The default implementation includes `lin_src` and `lin_dst` when $d_{\text{in}} \neq d$, which we exclude here since the setting is homogeneous.

Per-layer parameter count With $L = 2$, $s = 2$, and edge projection enabled:

$$\begin{aligned} N_{\text{layer}} &= (ed + d) + d^2(2s + (L-2)s^2) + d((L-1)s + 1) \\ &= (ed + d) + (4d^2 + 3d) \\ &= 4d^2 + ed + 4d \end{aligned}$$

Complexity class
Bias terms scale as $\mathcal{O}(d)$ and are dominated by the quadratic and bilinear terms:

$$N_{\text{layer}} \in \Theta(d^2 + ed)$$

Correspondence to message-passing formulation
Given $\mathbf{h}_i, \mathbf{h}_j \in \mathbb{R}^d$ and edge feature $\mathbf{e}_{ij} \in \mathbb{R}^e$, the GeneralConv layer performs:

$$\mathbf{m}_{ij} = \text{MLP}(\mathbf{h}_j + \mathbf{e}_{ij}) \quad \text{or} \quad \text{MLP}(\mathbf{h}_j \odot \mathbf{e}_{ij})$$

depending on the fusion strategy. The fused message is aggregated over neighbors and added to the root node via skip connection. The learnable components used here are precisely the edge projection (if $e \neq d$) and the post-fusion MLP, matching the parameter count.

B.1.4. PNAConv (Principal Neighbourhood Aggregation)

The Principal Neighbourhood Aggregation layer (`torch_geometric.nn.conv.PNAConv`) enhances message expressivity by combining multiple aggregation functions and degree-scalers. As proposed by Corso et al. [157], PNA applies a shared message MLP to edge-conditioned node interactions, followed by aggregation over multiple statistical moments and degree-scalers, and concludes with a mixing MLP. Edge features are incorporated directly by concatenation into the message function.

$$\mathbf{h}_i^{(l+1)} = \text{MLP}_{\text{mix}} \left(\mathbf{h}_i^{(l)} \,\|\, \bigoplus_{\substack{a \in \mathcal{A} \\ s \in \mathcal{S}}} s \left(a \left\{ \text{MLP}_{\text{msg}}(\mathbf{h}_i^{(l)} \,\|\, \mathbf{e}_{ij}^{(l)} \,\|\, \mathbf{h}_j^{(l)}) \right\}_{j \in \mathcal{N}(i)} \right) \right)$$

Here, $\mathcal{A}$ is the set of A aggregators (e.g., mean, max, min, standard deviation), and $\mathcal{S}$ is the set of S degree-scalers (e.g., identity, amplification, attenuation). The operator outputs a representation in $\mathbb{R}^{ASd}$, which is mixed back to $\mathbb{R}^d$ via the final MLP.

Learnable components (default configuration)
Let A be the number of aggregators, S the number of scalers, L the depth of the MLP, and s the hidden expansion factor.

1. **Message MLP** (shared across all edges and aggregators)

$$\text{MLP}_{\text{msg}} : [\, d \to ds \to \cdots \to ds \to d \,]$$

Parameters: Weights: $2d^2s + (L-2)d^2s^2$ Biases: $(L-1)ds + d$ Total:

$$d^2(2s + (L-2)s^2) + d((L-1)s + 1)$$

2. **Post-aggregation mixing MLP**

$$\text{MLP}_{\text{mix}} : \mathbb{R}^{ASd+d} \to \mathbb{R}^d$$

Parameters: $(ASd + d) \cdot d + d = ASd^2 + d^2 + d$

3. **Edge encoder (optional)** If used, the edge encoder is a single linear layer:

$$\text{lin\_edge} : \mathbb{R}^e \to \mathbb{R}^d$$

Parameters: $ed + d$

Per-layer parameter count

$$N_{\text{layer}} = d^2 \left(2s + (L-2)s^2\right) + d\left((L-1)s + 1\right) + \left(ASd^2 + d^2 + d\right) + (ed + d)$$

Or simplified to:

$$N_{\text{layer}} = d^2 \left(2s + (L-2)s^2 + AS + 1\right) + d\left((L-1)s + e + 2\right)$$

Assuming $L = 2$, $s = 2$, and edge encoder is active:

$$\begin{aligned} N_{\text{layer}} &= (4d^2 + 3d) + (ASd^2 + d^2 + d) + (ed + d) \\ &= (5 + AS)d^2 + ed + 5d \end{aligned}$$

Complexity class
Bias terms are $\mathcal{O}(d)$ and are dominated by quadratic and bilinear terms. Hence:

$$N_{\text{layer}} \in \Theta\left((5 + AS)d^2 + ed\right)$$

Given that $A, S = \mathcal{O}(1)$ (i.e., constant number of aggregators and scalers), the complexity class simplifies to:

$$N_{\text{layer}} \in \Theta(d^2 + ed)$$

Correspondence to message-passing formulation
Given node features $\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)} \in \mathbb{R}^d$ and edge features $\mathbf{e}_{ij}^{(l)} \in \mathbb{R}^e$, each message is computed as:

$$\mathbf{m}_{ij} = \text{MLP}_{\text{msg}}(\mathbf{h}_i^{(l)} \,\|\, \mathbf{e}_{ij}^{(l)} \,\|\, \mathbf{h}_j^{(l)})$$

Messages are aggregated over neighbors using a composite of A aggregators and S degree-scalers, yielding a vector in $\mathbb{R}^{ASd}$. This is concatenated with the root node's features and passed through the mixing MLP to produce the final update. This formulation corresponds exactly to the PNA design proposed by Corso et al. [157], combining statistical aggregation with degree-sensitive modulation.

B.1.5. NNConv (Neural Network Convolution)

The Neural Network Convolution layer (`torch_geometric.nn.conv.NNConv`) computes edge-conditioned filters via a small MLP, which is used to transform neighbor features. Originally proposed by Gilmer et al. [90], NNConv allows the message function to adapt to edge attributes by generating a weight matrix per edge.

$$\mathbf{h}_i^{l+1} = \Theta\, \mathbf{h}_i^l \bigoplus_{j \in \mathcal{N}(i)} \mathbf{h}_j^l \cdot \mathsf{MLP}(\mathbf{e}_{ij}),$$

where Θ is a learnable linear transformation applied to the root node $\mathbf{h}_i^l$, corresponding to the root-weight matrix (typically $\Theta \in \mathbb{R}^{d \times d}$). Here $\bigoplus$ denotes a generic permutation-invariant aggregation operator; in standard message-passing GNNs this operator is instantiated as a summation.

Learnable components (default configuration)

The message for each edge $j \to i$ is computed as:

$$\mathbf{m}_{ij} = \mathbf{h}_j^l \cdot \mathsf{MLP}(\mathbf{e}_{ij}) \quad \text{with} \quad \mathsf{MLP}(\mathbf{e}_{ij}) \in \mathbb{R}^{d \times d}$$

1. **Edge MLP (produces $d \times d$ matrix)**

   $$\mathsf{MLP}_{\text{edge}} : \mathbb{R}^e \to \mathbb{R}^{d \times d}$$

   We implement this as a standard MLP:

   $$\mathsf{MLP}_{\text{edge}} : e \to h_e \to d^2$$

   Parameters: Weights: $eh_e + h_e d^2$ Biases: $h_e + d^2$ Total:

   $$eh_e + h_e d^2 + h_e + d^2$$

2. **Root transformation (optional)**

   $$\mathsf{lin\_root} : \mathbb{R}^d \to \mathbb{R}^d$$

   Parameters: d^2 (if `root_weight=True`, `bias=False`)

Per-layer parameter count With root weight enabled:

$$\begin{aligned} N_{\text{layer}} &= (eh_e + h_e d^2 + h_e + d^2) + d^2 \\ &= h_e(e + d^2 + 1) + 2d^2 \end{aligned}$$

Complexity class

Assuming $h_e = \mathcal{O}(d)$ (as in most applications), the dominant terms are:

$$N_{\text{layer}} \in \Theta(d^3 + ed)$$

Correspondence to message-passing formulation

Each edge $\mathbf{e}_{ij}$ is passed through a shared edge MLP to yield a $d \times d$ kernel $\mathbf{W}_{ij}$, which is then multiplied with $\mathbf{h}_j$. The messages are summed over neighbors and added to a root transformation of $\mathbf{h}_i$ (if enabled). This formulation directly explains the cubic term in the parameter count: a fully learnable filter for each edge feature embedding into a matrix of size $d \times d$.

B.1.6. CGConv (Crystal Graph Convolution)

The Crystal Graph Convolution (CGConv) operator was introduced by Xie and Grossman [98] for property prediction of atomistic crystals. It models neighbor interactions via a learned gating mechanism applied to the concatenated node and edge features. In `torch_geometric.nn.conv.CGConv`, this is implemented using two learnable linear projections—one for gating, and one for message transformation—followed by elementwise multiplication. The final message is aggregated across neighbors and optionally added to a learnable transformation of the root node.

$$\mathbf{h}_i^{(l+1)} = \mathbf{h}_i^{(l)} + \sum_{j \in \mathcal{N}(i)} \sigma\left(\mathbf{z}_{ij}^{(l)} \mathbf{W}_f^{(l)} + \mathbf{b}_f^{(l)}\right) \odot g\left(\mathbf{z}_{ij}^{(l)} \mathbf{W}_s^{(l)} + \mathbf{b}_s^{(l)}\right)$$

where:

- $\mathbf{z}_{ij}^{(l)} = \mathbf{h}_i^{(l)} \,\|\, \mathbf{h}_j^{(l)} \,\|\, \mathbf{e}_{ij}^{(l)}$ is the concatenated input,
- $\mathbf{W}_f^{(l)}$, $\mathbf{W}_s^{(l)} \in \mathbb{R}^{(2d+e)\times d}$ are learnable weight matrices,
- $\mathbf{b}_f^{(l)}$, $\mathbf{b}_s^{(l)} \in \mathbb{R}^d$ are bias terms,
- σ is the sigmoid function, g is a non-linear activation (e.g., ReLU),

Learnable components (default configuration)

Assuming homogeneous input with hidden dimension d and edge feature dimension e:

1. **Gating linear projection**
$$\mathbf{W}_f : \mathbb{R}^{2d+e} \to \mathbb{R}^d$$
Parameters: $(2d + e) \cdot d + d = 2d^2 + ed + d$
2. **Message linear projection**
$$\mathbf{W}_s : \mathbb{R}^{2d+e} \to \mathbb{R}^d$$
Parameters: same as above: $2d^2 + ed + d$
3. **Root node transformation (optional)**
$$\text{lin\_root} : \mathbb{R}^d \to \mathbb{R}^d$$
Parameters: d^2 (if `root_weight=True`, `bias=False`)
4. **Bias (optional)**
$$\mathbf{b} \in \mathbb{R}^d$$
Parameters: d

Per-layer parameter count Assuming `root_weight=True` and `bias=True`:

$$\begin{aligned} N_{\text{layer}} &= 2 \cdot (2d^2 + ed + d) + d^2 + d \\ &= 4d^2 + 2ed + 2d + d^2 + d \\ &= 5d^2 + 2ed + 3d \end{aligned}$$

Complexity class

The bias term is linear in d and negligible. Hence:

$$N_{\text{layer}} \in \Theta(d^2 + ed)$$

Correspondence to message-passing formulation

For each edge (i, j), CGConv first forms the concatenated input $\mathbf{z}_{ij} = \mathbf{h}_i \,\|\, \mathbf{h}_j \,\|\, \mathbf{e}_{ij} \in \mathbb{R}^{2d+e}$. This vector is projected in two parallel branches:

- One linear layer produces gating coefficients, passed through a sigmoid nonlinearity: $\sigma(\mathbf{z}_{ij}\mathbf{W}_f + \mathbf{b}_f)$
- The second projects the same input to a transformed message vector with ReLU activation: $g(\mathbf{z}_{ij}\mathbf{W}_s + \mathbf{b}_s)$

The resulting two vectors are combined via elementwise multiplication, and messages are summed across neighbors. A residual connection from the root node completes the update:

$$\mathbf{h}_i^{(l+1)} = \mathbf{h}_i^{(l)} + \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij} + \mathbf{b} \quad \text{with} \quad \mathbf{m}_{ij} = \sigma(\cdot) \odot g(\cdot)$$

This structure corresponds exactly to two dense $(2d + e) \to d$ projections, plus an optional root transformation and bias, as counted in the parameter analysis.

B.1.7. SplineConv (Spline-based Graph Convolution)

The Spline-based Graph Convolution operator (`torch_geometric.nn.conv.SplineConv`) was introduced by Fey and Lenssen [95] as a continuous, spatial message-passing kernel based on B-splines. Instead of using neural networks to parameterize the messages, SplineConv employs a B-spline tensor product basis to define a continuous kernel function, enabling efficient and spatially-aware graph convolutions.

$$\mathbf{h}_i^{(l+1)} = \bigoplus_{j \in \mathcal{N}(i)} \mathbf{K}(\mathbf{e}_{ij}) \cdot \mathbf{h}_j^{(l)}$$

where:

- for SplineConv, the default is mean aggregation, i.e. $\bigoplus_{\sum_{j \in \mathcal{N}(i)}} = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)}$
- $\mathbf{e}_{ij} \in [0, 1]^e$ are pseudo-coordinates assigned to edge $(j \to i)$,
- $K : [0, 1]^e \to \mathbb{R}^{d \times d}$ is a kernel function defined over a tensor product of e one-dimensional B-spline bases,
- $\mathcal{N}(i)$ is the set of neighbors of node i.

Learnable components

Let d be the feature dimension, e the number of edge dimensions (pseudo-coordinates), and R the number of B-spline control points per dimension.

1. **B-spline kernel tensor**

   $$\mathcal{W} \in \mathbb{R}^{R^e \times d \times d}$$

   Here, R denotes the number of control points (or knots) used per edge feature dimension in the tensor-product B-spline basis, and e is the dimensionality of the edge coordinate $\mathbf{e}_{ij} \in [0, 1]^e$. Since B-spline interpolation is performed independently along each axis, the full kernel space is discretized into R^e grid cells.

   Each of the R^e control points defines a trainable $d \times d$ linear transformation. During message passing, the kernel value $\mathbf{K}(\mathbf{e}_{ij})$ is computed as a convex combination of the neighbouring basis weights, determined by the pseudo-coordinate $\mathbf{e}_{ij}$.

2. **Root transformation (optional)**

   $$\text{lin\_root} : \mathbb{R}^d \to \mathbb{R}^d$$

   Parameters: d^2 (only if `root_weight=True`)

3. **Bias (optional)**

   $$\mathbf{b} \in \mathbb{R}^d$$

   Parameters: d

Per-layer parameter count

$$N_{\text{layer}} = R^e \cdot d^2 + d^2 + d = (R^e + 1)d^2 + d$$

Complexity class
Bias terms are negligible, and the dominant term arises from the spline kernel tensor:

$$N_{\text{layer}} \in \Theta(R^e \cdot d^2)$$

assuming $R^e \gg 1$.

Correspondence to message-passing formulation
Each message is computed as:

$$\mathbf{m}_{ij} = \mathbf{K}(\mathbf{u}_{ij}) \cdot \mathbf{h}_j$$

The kernel **K** is evaluated as a weighted sum of $d \times d$ matrices determined by local support B-spline functions:

$$\mathbf{K}(\mathbf{u}_{ij}) = \sum_{p \in \mathcal{P}(\mathbf{u}_{ij})} B_p(\mathbf{u}_{ij}) \cdot \mathcal{W}_p$$

where $\mathcal{P}(\mathbf{u}_{ij})$ denotes the $s = (m+1)^e$ active control points based on B-spline support. This allows sparse evaluation of the kernel and efficient GPU implementation.

The final output includes optional root transformation and bias addition:

$$\mathbf{h}_i^{(l+1)} = \text{lin\_root}(\mathbf{h}_i^{(l)}) + \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij} + \mathbf{b}$$

B.1.8. GMMConv (Gaussian Mixture Model Convolution)

The Gaussian Mixture Model Convolution (GMMConv) operator defines a spatially-aware message-passing mechanism over graphs by learning a continuous kernel function over edge pseudo-coordinates. It shares conceptual similarity with `SplineConv` in that both perform edge-conditioned filtering based on relative geometry, but differs in its parametrization: whereas `SplineConv` uses a fixed B-spline basis over a regular control grid, `GMMConv` models the kernel as a sum of C Gaussians with learned means and variances.

The node update rule is:

$$\mathbf{h}_i^{(l+1)} = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \sum_{k=1}^{C} \omega_k(\mathbf{u}_{ij}) \cdot \mathbf{W}_k \mathbf{h}_j^{(l)}$$

where:

- $\mathbf{u}_{ij} \in \mathbb{R}^e$ is the pseudo-coordinate vector associated with edge $(j \to i)$,
- $\omega_k(\mathbf{u})$ is the weight assigned to component k by the Gaussian kernel:

$$\omega_k(\mathbf{u}) = \exp\left(-\frac{1}{2}(\mathbf{u} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{u} - \boldsymbol{\mu}_k)\right)$$

- $\mathbf{W}_k \in \mathbb{R}^{d \times d}$ is the learnable linear transformation associated with component k.

Learnable components
Let d be the hidden feature dimension and e the pseudo-coordinate dimension:

1. **Gaussian kernel parameters**

$$\boldsymbol{\mu}_k \in \mathbb{R}^e, \quad \boldsymbol{\sigma}_k \in \mathbb{R}^e \quad \text{for } k = 1, \ldots, C$$

Parameters: $2Ce$

2. **Component-specific weight matrices**

$$\mathbf{W}_k \in \mathbb{R}^{d \times d} \quad \text{for } k = 1, \ldots, C$$

   Parameters: $C \cdot d^2$

3. **Root node transformation (optional)**

$$\text{lin\_root} : \mathbb{R}^d \to \mathbb{R}^d$$

   Parameters: d^2

4. **Bias (optional)**

$$\mathbf{b} \in \mathbb{R}^d$$

   Parameters: d

Per-layer parameter count

$$N_{\text{layer}} = C \cdot d^2 + 2Ce + d^2 + d = (C+1)d^2 + 2Ce + d$$

Complexity class
Assuming $C = \mathcal{O}(1)$ and $e = \mathcal{O}(d)$ (as is common in geometric graphs), we obtain:

$$N_{\text{layer}} \in \Theta(d^2 + ed)$$

Correspondence to message-passing formulation
Messages are computed as:

$$\mathbf{m}_{ij} = \sum_{k=1}^{C} \omega_k(\mathbf{u}_{ij}) \cdot \mathbf{W}_k \mathbf{h}_j$$

where $\omega_k(\cdot)$ is the Gaussian activation for pseudo-coordinate $\mathbf{u}_{ij}$ with respect to component k. The resulting messages are averaged over neighbors, added to a root transformation (if enabled), and passed through a bias. The spatial bias is encoded by the continuous, differentiable kernel formed via the Gaussian mixture model.

Unlike `SplineConv`, which relies on structured interpolation over a fixed grid of control points, `GMMConv` allows for a more flexible kernel shape by learning both the centers and widths of each Gaussian basis function. This flexibility comes at the cost of additional parameters, particularly in high-dimensional edge spaces.

B.1.9. GATv2Conv (Graph Attention v2 Convolution)

GATv2 (`torch_geometric.nn.conv.GATv2Conv`) improves on GAT by computing attention scores dynamically, based on the concatenated source and target node representations (optionally including edge features), followed by a shared linear transformation and LeakyReLU activation. This removes the static attention limitation of the original GAT [61].

The multi-head attention mechanism computes:

$$\mathbf{h}_i^{(l+1)} = \|_{h=1}^{H} \sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij}^{(h)} \, \mathbf{W}_t^{(h)} \mathbf{h}_j$$

where the attention coefficients are:

$$\alpha_{ij}^{(h)} = \frac{\exp\left(\mathbf{a}^{(h)\top} \,\text{LeakyReLU}\left(\mathbf{W}_s^{(h)} \mathbf{h}_i + \mathbf{W}_t^{(h)} \mathbf{h}_j + \mathbf{W}_e^{(h)} \mathbf{e}_{ij}\right)\right)}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp\left(\mathbf{a}^{(h)\top} \,\text{LeakyReLU}\left(\mathbf{W}_s^{(h)} \mathbf{h}_i + \mathbf{W}_t^{(h)} \mathbf{h}_k + \mathbf{W}_e^{(h)} \mathbf{e}_{ik}\right)\right)}$$

Notation

- $\mathbf{W}_s, \mathbf{W}_t \in \mathbb{R}^{d \times d}$: linear projections for source and target node features (optionally shared),
- $\mathbf{W}_e \in \mathbb{R}^{e \times d}$: projection of edge features,
- $\mathbf{a} \in \mathbb{R}^d$: learnable attention scoring vector,
- LeakyReLU$(\cdot)$: non-linear activation (typically with slope 0.2),
- α_{ij}: normalized attention weight via softmax over neighbors.

Note. Edge features are not part of the original GATv2 paper, but are supported in PyG's implementation and included in this study.

Learnable components
Let H be the number of attention heads and $d_{\text{head}} = d/H$ the feature dimension per head.

1. **Source and target projections**
$$W_s, W_t : \mathbb{R}^d \to \mathbb{R}^d$$
Each is implemented as a linear map without bias. Parameters: $2d^2$
2. **Attention vector**
$$\mathbf{a} \in \mathbb{R}^d$$
Parameters: d
3. **Edge projection (optional)**
$$W_e : \mathbb{R}^e \to \mathbb{R}^d$$
Parameters: ed
4. **Output bias**
$$\mathbf{b} \in \mathbb{R}^d$$
Parameters: d

Per-layer parameter count
$$N_{\text{layer}} = 2d^2 + ed + 2d$$

Complexity class
$$N_{\text{layer}} \in \Theta(d^2 + ed)$$

Correspondence to message-passing formulation
Node and edge features are first projected and summed inside the attention mechanism:

$$\alpha_{ij} = \text{softmax}\left(\mathbf{a}^\top \text{LeakyReLU}(\mathbf{W}_s \mathbf{h}_i + \mathbf{W}_t \mathbf{h}_j + \mathbf{W}_e \mathbf{e}_{ij})\right)$$

The resulting attention scores weight the target node features:

$$\mathbf{m}_{ij} = \alpha_{ij} \cdot \mathbf{W}_t \mathbf{h}_j$$

These are summed across neighbours, and the outputs from all heads are concatenated.

B.1.10. GCN+ (Enhanced Graph Convolution with MLP)

The GCN+ operator builds upon the classical Graph Convolutional Network (GCN) by incorporating architectural refinements that improve both stability and expressiveness in deeper networks. The full variant used in this study includes (i) edge features, (ii) residual connections, (iii) Batch Normalization, (iv) Dropout regularization, and (v) a two-layer MLP applied after aggregation. This architecture was proposed by Li et al. [**li2021unlocking**] to overcome limitations of oversmoothing and vanishing gradients in deeper message-passing networks.

The message-passing rule implemented in the PyTorch Geometric variant is:

$$\mathbf{h}_i^{(l+1)} = \text{MLP}\left(\text{Dropout}\left(\sigma\left(\text{BN}\left(\sum_{j\in\mathcal{N}(i)\cup\{i\}} \frac{1}{\sqrt{\hat{d}_j\hat{d}_i}}\left(\mathbf{h}_j^{(l)}\mathbf{W}^l + \mathbf{e}_{ij}\mathbf{W}_e^l\right)\right)\right)\right) + \mathbf{h}_i^{(l)}\right)$$

Here, $\hat{d}_i$ denotes the degree of node i with self-loops included. The term $\mathbf{e}_{ij}\mathbf{W}_e^l$ enables edge-aware message construction, while the residual addition ensures better gradient flow across layers. The result is passed through Batch Normalization (BN), a nonlinearity σ (e.g., ReLU), dropout, and a two-layer MLP for final transformation.

Learnable components (default configuration)

1. **Node feature transformation**

   $$\text{lin\_node} : \mathbb{R}^d \to \mathbb{R}^d$$

   Parameters: d^2

2. **Edge feature transformation**

   $$\text{lin\_edge} : \mathbb{R}^e \to \mathbb{R}^d$$

   Parameters: $e \cdot d$

3. **Post-aggregation MLP**

   $$\text{MLP} : \mathbb{R}^d \to \mathbb{R}^{ds} \to \mathbb{R}^d$$

   Weights: $d \cdot ds + ds \cdot d = 2dsd$
   Biases: $ds + d$
   Total: $2dsd + ds + d = s(2d^2 + d) + d$

Per-layer parameter count The total number of parameters per layer is:

$$\begin{aligned} N_{\text{layer}} &= \underbrace{d^2}_{\text{node transform}} + \underbrace{ed}_{\text{edge transform}} + \underbrace{s(2d^2 + d) + d}_{\text{MLP}} \\ &= (1 + 2s)d^2 + (s + 1)d + ed \end{aligned}$$

Complexity class
Assuming edge feature dimension $e = \mathcal{O}(1)$, the complexity becomes:

$$N_{\text{layer}} \in \Theta(d^2 + ed)$$

Correspondence to message-passing formulation
Given node features $\mathbf{h}_j^{(l)} \in \mathbb{R}^d$ and edge features $\mathbf{e}_{ij} \in \mathbb{R}^e$, each message is constructed as:

$$\mathbf{m}_{ij} = \mathbf{h}_j^{(l)}\mathbf{W}^l + \mathbf{e}_{ij}\mathbf{W}_e^l$$

The messages are summed using symmetric normalization and represent the aggregated neighborhood signal for node i. After aggregation, the result is passed through BatchNorm, a nonlinearity, and Dropout, before being combined with a skip connection from the input. Finally, a two-layer MLP refines the representation. This sequence introduces modern deep-learning practices into the GCN framework, resulting in deeper and more stable graph networks without requiring architectural overhauls.

C

Supplementary Methodological Details

C.1. Final Test Dataset

The *Final Test Dataset* is constructed to evaluate how well the selected models generalise beyond the training distribution. Whereas the training and validation sets consist exclusively of square rooms with a centrally placed window, the final test set introduces controlled geometric and transformational variations that probe the limits of the learned representations.

The dataset is organised into six tiers (Tiers $0 - 5$), arranged to form a progressive departure from the distribution seen during training. Tier 0 reeproduces base square geometries that lie fully within the training domain. Tier 1 introduces th same rigid transformations (rotations and uniform scaling) as the *Transformation Dataset*; these cases are not used for training but remain structurally consistent with the original square geometry. Beginning with Tier 2, the dataset introduces room shapes and window configurations that are entirely unseen during training: rectangles with altered aspect ratios (Tier 2), side-window rectangles (Tier 3), and L-shaped rooms with windows placed on outer or recessed façades (Tiers 4 and 5). The full dataset therefore provides a structured progression from in-distribution cases to increasingly complex out-of-distribution scenarios.

C.1.1. Tiers 0-2: Squares, Transforms, and Rectangles

Tiers 0–2 originate from a common pool of 50 base square rooms. Each configuration is defined by a width and a window-to-wall ratio (WWR) sampled using Latin Hypercube Sampling (LHS) in the space

$$\text{width} \in [2.5, 8.0]\ \text{m}, \qquad \text{WWR} \in [0.2, 0.8],$$

with a fixed seed (43). The samples are lightly discretised by rounding width to two decimals and WWR to one decimal.

Tier 0 (Base squares). Each LHS sample is instantiated as a square with depth $=$ width, unit scale (scale $= 1$), and zero rotation (rotation $= 0°$). These correspond exactly to the training distribution and serve as the in-distribution reference cases.

Tier 1 (Rotated and uniformly scaled squares). Tier 1 repeats the transformations used in the *Transformation Dataset*: a rotation of $90°$, a rotation of $180°$, and a uniformly scaled variant in which the entire room is enlarged by a factor of two. Apart from these rigid motions and global scaling, the underlying geometry remains identical to that of the base squares.

Tier 2 (Rectangles and large squares). Tier 2 introduces the first genuinely new geometries:

- **Horizontal rectangles**: the base square is stretched in the x-direction, doubling its width while keeping its depth unchanged;
- **Vertical rectangles**: the base square is stretched in the y-direction, doubling its depth while keeping its width unchanged;
- **Large squares**: a uniformly scaled variant in which the base square is enlarged by a factor of five.

These geometries depart from the training distribution through changes in plan aspect ratio or overall size while maintaining the same WWR.

Summary of parameters. The primary parameters defining Tiers 0–2 are summarised in Table C.1.

Table C.1: Parameters defining Tiers 0–2 of the *Final Test Dataset*.

Parameter	Description
`base_id`	Index of the originating LHS sample.
`tier`	Tier label (0–2).
`width`	Room width before scaling.
`depth`	Room depth before scaling.
`WWR`	Window-to-wall ratio.
`scale`	Uniform scale factor.
`rotation`	Plan rotation angle in degrees.

C.1.2. Tiers 3--5: Side-window Rectangles and L-shaped Rooms

Tiers 3–5 introduce entirely new plan geometries not present in training. All follow the parametric conventions shown in Figure C.1, summarised in Table C.2. Unlike Tiers 0–2, these rooms include non-centred windows, altered façade lengths, and—in the L-shaped cases—self-occluding recesses.

Parameter	Description
a	Room length in the y-direction.
b	Room width in the x-direction.
c	Horizontal notch offset (negative), defining the L-shape.
d	Vertical notch offset (negative), defining the L-shape.
e	Room height (set to 3 m).
f	Wall thickness (set to 0.2 m).
g	Index of the window-bearing façade (0–5).
h	Window centre position along façade g.
i	Window width.
j	Sill height.
k	Window height.
wall length	Usable façade length for façade g.
WWR	Window-to-wall ratio on façade g.

Table C.2: Parameters defining Tier 3–5 geometries. [1]

Parameter overview.

Tier 3 (Side-window rectangles). Tier 3 consists of rectangular rooms without indentation ($c = d = 0$). The plan is parameterised as $(a, b) = (s, 2s)$ to ensure an asymmetric layout, and the window is placed on the long façade. Its position and size are sampled along the available span, and WWR is computed relative to that façade alone.

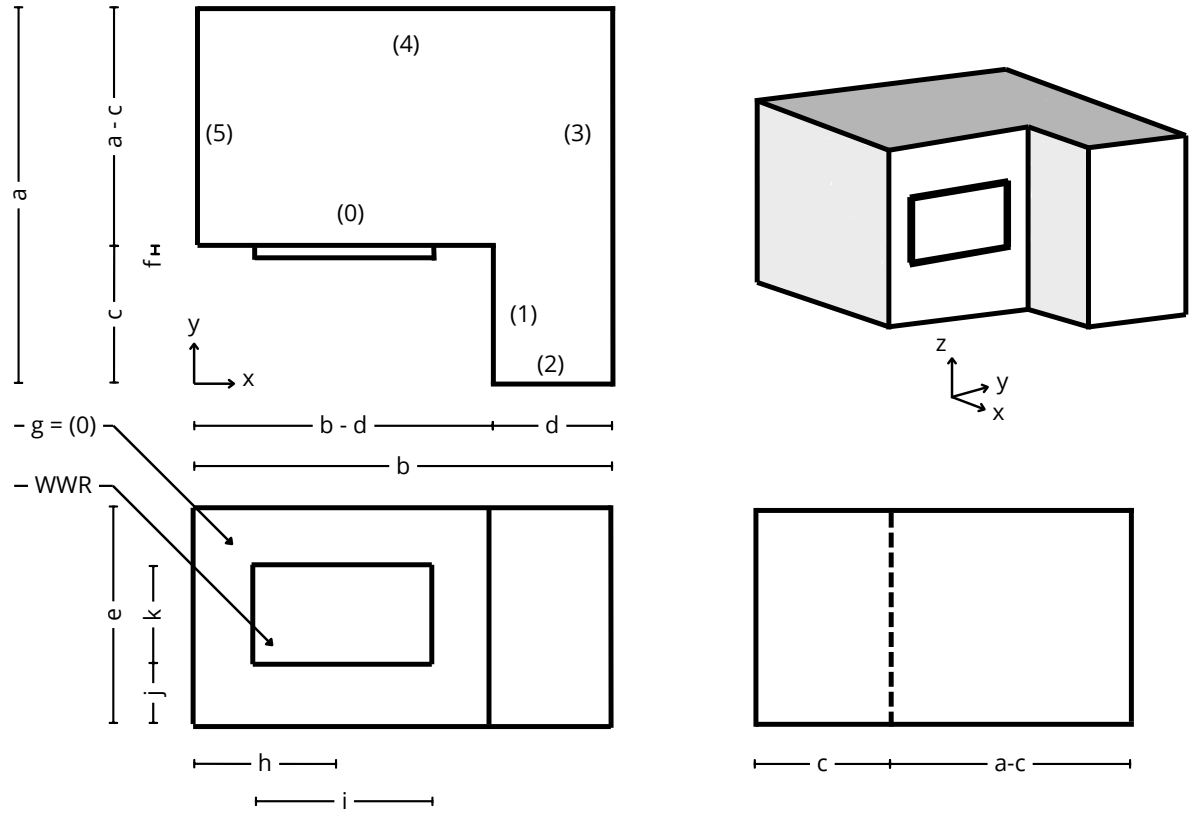


Figure C.1: Parametric construction of Tier 3 side-window rectangles and Tier 4–5 L-shaped rooms.

Tier 4 (L-shaped rooms with outer-wall windows). Tier 4 introduces L-shaped plans by applying negative offsets (c, d) that carve out a rectangular recess. Windows are placed on the *outer* long façades. The available façade length depends on (a, b, c, d), and all window parameters (h, i, j, k) are sampled accordingly.

Tier 5 (L-shaped rooms with recessed windows). Tier 5 uses the same L-shaped geometry as Tier 4 but positions the window on one of the *inner* or recessed faces of the plan. These façade segments are shorter and partially occluded, representing the most challenging and strongly out-of-distribution test cases.

C.2. Benchmark ANN Models

C.2.1. Raw baseline

The raw baseline represents the simplest form of ANN surrogate modelling, using only direct geometric descriptors as input features. Each sensor is described by its Cartesian coordinates (x, y) within the room, supplemented by two global parameters: room width and window-to-wall ratio (WWR). No abstraction, projection, or physically derived transformations are applied, and the four scalars are merely normalised prior to training. This minimal encoding functions as a pedagogical lower bound, allowing the contribution of more advanced feature engineering strategies to be assessed in subsequent models.

The surrogate model is implemented as a lightweight MLP, consistent with prior ANN-based daylight surrogates that employ compact, fully connected networks [17–19]. Given the low input dimensionality, a shallow architecture was adopted to prevent overparameterisation while retaining sufficient expressive power to capture spatial gradients in DF values. The network comprises two hidden layers of 32 neurons each, with ReLU activation functions and dropout regularisation ($p = 0.2$). The output layer is a single linear neuron producing the DF estimate for each sensor point. This configuration reflects established practice in surrogate modelling of building performance, where predictive capacity is dominated

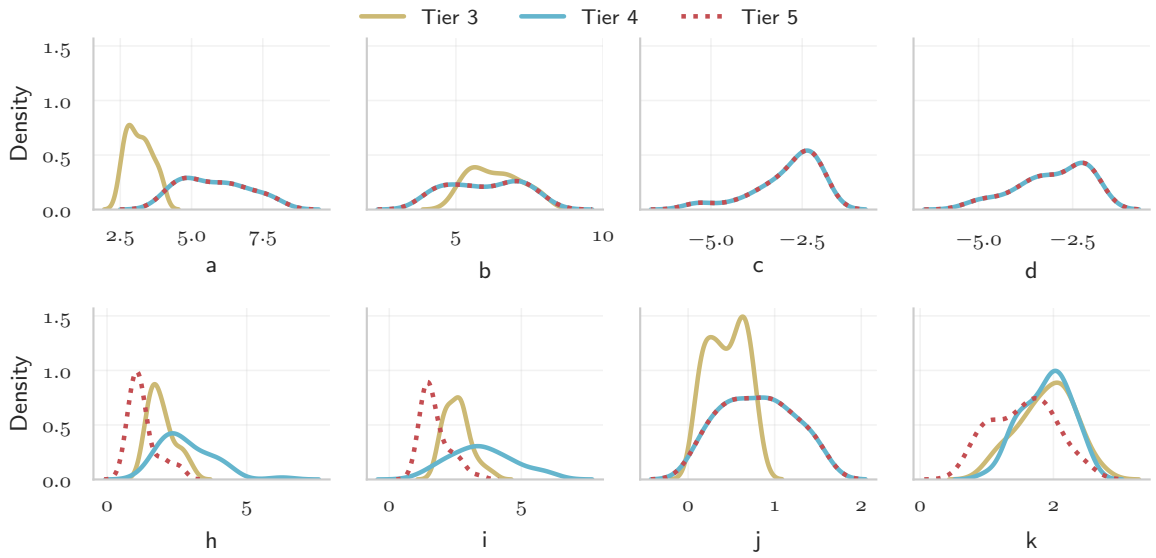


Figure C.2: Parameter distributions of (a, b, c, d, h, i, j, k) across Tiers 3–5.

by feature informativeness rather than architectural complexity.

C.2.2. Le-Thanh et al.

The surrogate model introduced by Le-Thanh et al. [18] employs a structured encoding strategy designed to generalise across arbitrary floor plan layouts. Instead of relying on absolute Cartesian coordinates, spatial relationships are abstracted into three sensor-centric feature families: (i) radial obstacle distances x, (ii) Euclidean distances d to window corners, and (iii) angular orientations w of these vectors relative to a global axis. This design preserves locality, directionality, and occlusion effects in a compact form, largely independent of the global coordinate frame.

Obstacle distance encoding (x).

From each sensor s_i, n rays are cast radially across the $360°$ plane. The length of the first intersection is recorded: if the ray intersects a wall, the distance is stored; if it intersects a window, the value is set to zero. This vector encodes local openness and obstruction around the sensor. Following the findings of Le-Thanh et al., $n = 40$ rays were selected, which was shown to yield the best balance between accuracy and efficiency (Table 7 in [18]).

Window corner distance encoding (d).

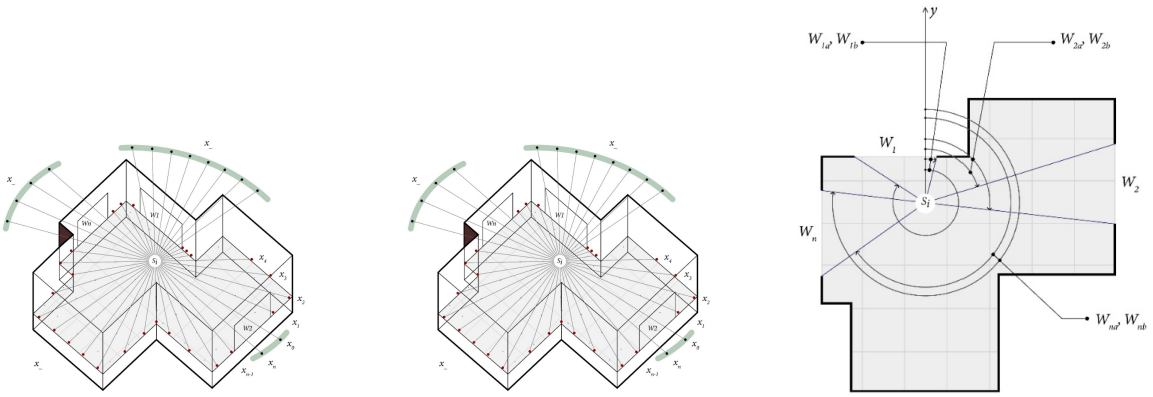
For each window W_m, the Euclidean distance between s_i and its four corners is computed. To ensure compatibility with multi-window layouts, the maximum number of windows was set to two, yielding eight distances per sensor. Although only single-window cases are present in the dataset, this choice preserves flexibility for future multi-window tests without introducing unused padded inputs during training.

Window orientation encoding (w).

To complement the magnitude-based d features, angular information is included by measuring the difference between the global $+y$ axis and the projection of the sensor–corner vector onto the floor plane. Two angular values are recorded per window, yielding four orientation features in the present setup.

Together, these descriptors yield a fixed-length input vector of $40 + 8 + 4 = 52$ features per sensor.

The feature vector was processed by a MLP with three hidden layers (60, 30, 30 neurons), each with ReLU activation and dropout regularisation ($p = 0.2$). This configuration corresponds to the best-performing architecture reported by Le-Thanh et al. (Table 5 in [18]), which was shown to outperform shallower or wider alternatives in terms of RMSE, MAE, and R^2. The output layer consists of a single linear neuron producing the DF prediction per sensor. Although the original study focused on UDI and



(a) Obstacle distance encoding. Rays are cast from each sensor across $360°$, with distances stored until the first intersection. Window intersections are assigned zero.

(b) Window-corner distance encoding. Euclidean distances between sensors and window corners capture local geometric proximity to openings.

(c) Window orientation encoding. Angles between sensor–corner vectors and the global $+y$ axis encode directional orientation of windows relative to each sensor.

Figure C.3: Feature encoding diagrams from Le-Thanh et al. (2022) for daylight prediction [18].

employed TensorFlow, the present implementation was reproduced in PyTorch and adapted to the DF prediction task. Detailed layer specifications and training settings are reported in Table C.4.

C.2.3. Dieguez et al.

Dieguez et al. [19] proposed a comprehensive encoding scheme for DF prediction, designed to reflect the three canonical components of the metric: direct skylight (DC), externally reflected light (ERC), and internally reflected light (IRC). Unlike coordinate-based encodings, this strategy introduces physically meaningful descriptors that approximate how light propagates through an interior. The features are grouped below according to their corresponding DF component.

Direct component (DC) features.

- *Solid Angle*: The portion of the sky visible through the window from a given sensor point. It is computed as the projected surface area of the window on a unit sphere centred at the sensor, expressed in steradians:

$$\Omega = \frac{A_{\cup}}{r^2} \tag{C.1}$$

  Here, $A_{\cup}$ denotes the union of the interior and exterior window apertures. As illustrated in Figure C.4, e corresponds to the exterior opening, f to the interior opening, and g to their union. By projecting the combined aperture, the calculation explicitly accounts for wall thickness, distinguishing the Dieguez formulation from earlier surrogate models, which typically represent the aperture as a zero-thickness plane. This enables the descriptor to capture partial occlusion effects from deep reveals and recessed windows, making it a more physically realistic measure of daylight availability.
- *Aspect Ratio*: The ratio of width to height of the projected union of the window shape on the sensor's spherical view. This indicates whether the visible aperture is horizontally or vertically elongated, influencing how light is distributed in the room.
- *Solid Angle Tilt*: The vertical tilt angle of the window projection's centroid relative to the horizon. This distinguishes whether upper or lower parts of the sky dome are more visible.
- *Frame Ratio* (excluded): Encodes the fraction of the window area blocked by framing elements. This was omitted here, since all façades in the dataset assume fully glazed openings with negligible framing.

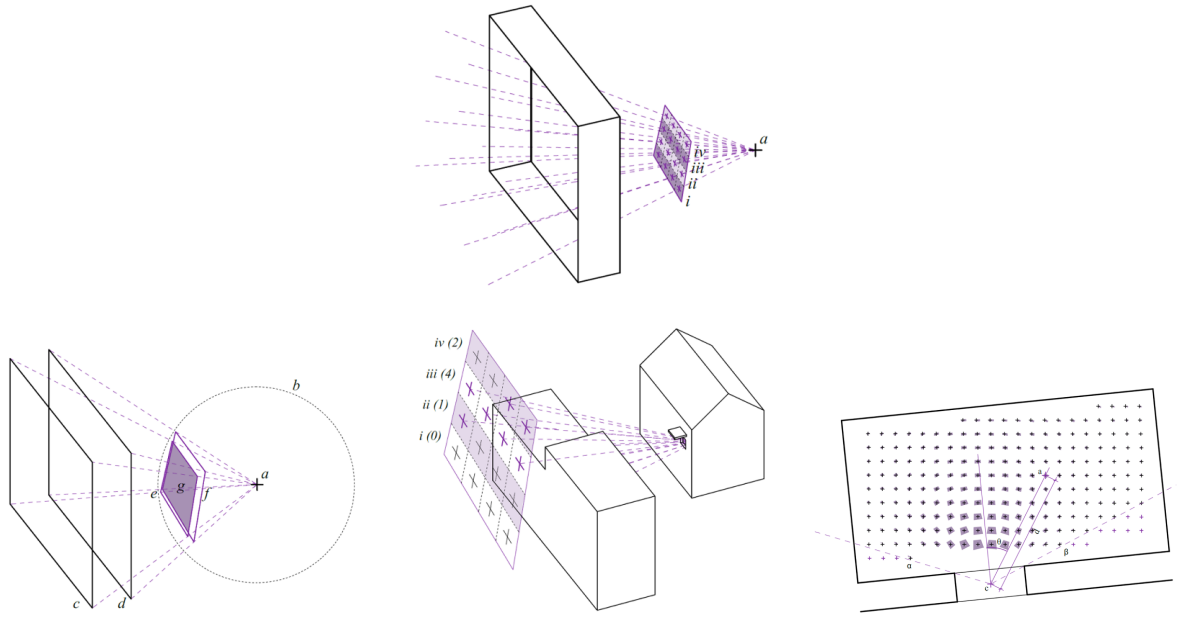
Externally reflected component (ERC) features.

- *External Obstruction Factor* (excluded): Quantifies obstruction of the sky dome by surrounding geometry, typically via ray intersections subdividing the window into quadrants. This feature was

excluded because no external obstructions were present in the dataset; all cases assumed unobstructed façades.

Internally reflected component (IRC) features.

- *Room Average Solid Angle*: The mean solid angle of the window across all sensors in the room. This provides a global measure of overall daylight penetration and acts as a room-level prior.
- *Distance to Window*: The Euclidean distance between a sensor point and the centre of the window. This captures geometric attenuation of light intensity with depth into the space.
- *Angle Relative to Window Normal*: The angle θ between the vector from sensor to window centre and the outward-facing window normal. This encodes the degree of obliqueness in the sensor's view of the aperture, with larger values corresponding to reduced daylight access.
- *Window Head Height*: The vertical distance between the floor plane and the top of the window. This feature governs how deeply daylight can penetrate into the room by controlling the height of the luminous opening.



(a) Solid angle calculation for direct skylight. The window aperture is projected onto a sphere centred at the sensor, with the overlap area determining the visible portion of sky.

(b) External obstruction encoding (not used). The aperture is subdivided into vertical stripes and assessed for visible sky fraction. This feature is omitted here as no external obstructions were modelled.

(c) IRC features. Distance to the window, angular deviation from the normal, and window head height capture attenuation and internal reflection effects.

Figure C.4: Feature encoding diagrams from Dieguez et al. (2025). Features are grouped by DF component (direct, external, internal). [19]

The surrogate model was implemented as a feedforward ANN with a fully connected MLP structure. Dieguez et al. [19] originally proposed a five-hidden-layer configuration, which was adopted here as the reference implementation. To complement this, a simplified variant with two hidden layers of 32 units each (identical to the Raw baseline) was also trained. This dual setup made it possible to disentangle the influence of the physically informed Dieguez features from the effects of model capacity.

The original network reported by Dieguez et al. is defined as follows:

- Layer 1: Dense layer with 16 neurons, ReLU activation function
- Layer 2: Dense layer with 32 neurons, Sigmoid activation function
- Layer 3: Dense layer with 16 neurons, ReLU activation function
- Layer 4: Dense layer with 4 neurons, ReLU activation function

- Layer 5: Dense layer with 2 neurons, Sigmoid activation function
- Output layer: Dense layer with 1 neuron, Sigmoid activation function

Both the original and simplified versions were trained using the Adam optimiser with mean squared error (MSE) as the loss function. All input features were standardised prior to training, and dropout layers were inserted between hidden layers to improve generalisation. As the original publication does not specify a dropout rate, a conventional value of $p = 0.2$ was assumed, consistent with related ANN-based daylight surrogates [17, 18]. In all cases, the output layer consisted of a single regression neuron producing the DF prediction for each sensor.

The simplified variant reduced the parameter count substantially compared to the original five-layer network (1361 parameters), providing a clearer assessment of whether performance gains stem from the richness of the Dieguez feature set or from the depth of the architecture. Full architectural specifications for both versions are summarised in Table C.4.

C.2.4. Summary features and architecture

The three benchmark ANN baselines span a spectrum of input complexity and inductive bias. The Raw model employs only four direct geometric descriptors (x, y, width, WWR), serving as a minimal reference point. The Le-Thanh model introduces relational descriptors that abstract sensor–window geometry into distances, angles, and ray-based occlusion measures, resulting in 52 features per sensor. Finally, the Dieguez model incorporates physically grounded features linked to the three canonical DF components (direct, external, internal). This feature set explicitly encodes wall thickness effects through the solid angle union, as well as room-level priors such as the average solid angle. A comprehensive overview of all input features used across the three models is given in Table C.3.

Table C.3: Overview of raw and derived input features used across different models.

Feature Name	Description	Scope	Derived	From Literature
x	Sensor position in the horizontal X-axis (room coordinate system)	Point		
y	Sensor position in the horizontal Y-axis (room coordinate system)	Point		
Width	Horizontal room width (long axis)	Global		
WWR	Window-to-wall ratio	Global		
$\mathbf{x}_{\text{thanh}}$	Radial distances to obstacles along evenly spaced rays around each sensor	Point	✓	Le-Thanh et al.
$\mathbf{d}_{\text{thanh}}$	Euclidean distances from each sensor to window corners	Point	✓	Le-Thanh et al.
$\mathbf{w}_{\text{thanh}}$	Angles between the Y-axis and sensor-to-window corner vectors	Point	✓	Le-Thanh et al.
Solid Angle	Projected solid angle subtended by the window at the sensor	Point	✓	Dieguez et al.
Aspect Ratio	Projected shape ratio (width-to-height) of the window from the sensor's view	Point	✓	Dieguez et al.
Solid Angle Tilt	Vertical tilt of the projected window shape in spherical view	Point	✓	Dieguez et al.
Room Avg. Solid Angle	Average solid angle across all sensors in the room	Global	✓	Dieguez et al.
Distance to Window	Straight-line distance from sensor to window center point	Point	✓	Dieguez et al.
Angle rel. to Window Normal	Angle between the sensor-to-window vector and window surface normal	Point	✓	Dieguez et al.
Window Head Height	Distance from the floor to the top of the window	Global		Dieguez et al.

In parallel with these feature sets, each model was paired with a multilayer perceptron (MLP) architec-

ture. The Raw and simplified Dieguez baselines adopted a lightweight two-layer structure, ensuring comparability of feature effects under identical network capacity. The Le-Thanh model followed the best-performing three-layer design reported in the original study, while the full Dieguez model used its original five-layer configuration. All networks were trained with the Adam optimiser, mean squared error (MSE) loss, and dropout regularisation ($p = 0.2$). Table C.4 summarises the input dimensions, hidden layers, activation functions, and output structure for each baseline.

Table C.4: Overview of MLP architectures used for each benchmark model. All networks employ the Adam optimiser with mean squared error (MSE) loss.

Model	Input Size	Hidden Layers (neurons, activation)	Regularisation	Output
Raw baseline	4	[32 ReLU, 32 ReLU]	Dropout $p = 0.2$	1 (linear)
Le-Thanh baseline	52	[60 ReLU, 30 ReLU, 30 ReLU]	Dropout $p = 0.2$	1 (linear)
Dieguez (original)	9	[16 ReLU, 32 Sigmoid, 16 ReLU, 4 ReLU, 2 Sigmoid]	Dropout $p = 0.2$	1 (linear)
Dieguez (simplified)	9	[32 ReLU, 32 ReLU]	Dropout $p = 0.2$	1 (linear)

C.3. Graph Construction

C.3.1. Homogeneous edge features

In the homogeneous graph, edges connect sensor nodes according to an eight-connected grid. Each edge is described by a set of geometric descriptors that capture pairwise distances, orientation, and relative aperture information. These features are listed below.

Euclidean distance. For displacement $\mathbf{d} = (dx, dy, dz)$ between two sensor points, the Euclidean distance is

$$d_{\text{euclid}} = \|\mathbf{d}\|_2. \tag{C.2}$$

This serves as the most direct measure of spatial separation.

Squared distance. The squared distance is

$$d^2 = \|\mathbf{d}\|_2^2, \tag{C.3}$$

which avoids the square root and amplifies larger separations.

Normalised distance. To remove scale-dependence, the distance is normalised by the room footprint diagonal $d_{\text{diag}} = \sqrt{\text{width}^2 + \text{depth}^2}$:

$$d_{\text{norm}} = \frac{\|\mathbf{d}\|_2}{d_{\text{diag}}}. \tag{C.4}$$

This ensures distances remain comparable across differently sized rooms.

Coordinate differences. The raw horizontal displacements are included as

$$\Delta x, \quad \Delta y. \tag{C.5}$$

These encode directional information in the plane of the sensor grid.

Normalised direction vectors. To capture relative orientation, the direction cosines are computed as

$$\hat{d}_x = \frac{\Delta x}{\|\mathbf{d}\|_2}, \quad \hat{d}_y = \frac{\Delta y}{\|\mathbf{d}\|_2}. \tag{C.6}$$

These normalised direction components are invariant to distance, isolating orientation effects.

Solid angle difference. Each sensor edge stores the difference in solid angle subtended by the window:

$$\Delta SA = SA_i - SA_j, \tag{C.7}$$

where SA_i and SA_j are the solid angles seen from sensors i and j. This encodes how much the visibility of the aperture changes across the edge.

Distance-to-window difference. Similarly, the difference in distance-to-window is

$$\Delta DW = DW_i - DW_j, \tag{C.8}$$

with DW_i the shortest Euclidean distance from sensor i to the aperture plane. This feature captures how quickly aperture proximity changes across the grid.

Absolute distance-to-window difference. The absolute value of this difference is also included:

$$|\Delta DW| = |DW_i - DW_j|. \tag{C.9}$$

This discards directional information, focusing purely on magnitude differences.

Relative angular alignment. Finally, relative orientation of window-facing directions is encoded as

$$\cos(\Delta\phi) = \cos(\phi_i - \phi_j), \tag{C.10}$$

where ϕ_i denotes the azimuthal angle between the sensor-to-window vector and a fixed reference axis. This feature measures angular coherence of window-facing directions between neighbouring sensors.

C.3.2. Heterogeneous edge features

For heterogeneous graphs, each edge is enriched with a set of invariant geometric descriptors. The first three descriptors are identical to the edge features used in the homogeneous graph representation (. Section C.3.1) and are therefore not explained again:

- Euclidean distance d_{euclid},
- Squared distance d^2,
- Normalised distance d_{norm}.

The remaining features extend this basis to capture orientation and aspect-ratio invariance. These are described below.

Scaled 3D distance. To ensure invariance with respect to room proportions, the displacement vector

$$\mathbf{d} = (dx,\, dy,\, dz)$$

is anisotropically scaled by the room spans $(\text{width}, \text{depth}, \text{height})$. The scaled displacement is

$$\mathbf{d}_s = \left(\tfrac{dx}{\text{width}}, \tfrac{dy}{\text{depth}}, \tfrac{dz}{\text{height}}\right).$$

The scaled 3D distance is then

$$d_{3d,s} = \|\mathbf{d}_s\|_2. \tag{C.11}$$

This feature is added to remove sensitivity to anisotropic scaling (e.g. elongated rooms), effectively mapping the room into a unit cube.

Scaled horizontal distance. The horizontal component of the scaled displacement is

$$d_{xy,s} = \sqrt{\left(\tfrac{dx}{\text{width}}\right)^2 + \left(\tfrac{dy}{\text{depth}}\right)^2}. \tag{C.12}$$

This feature isolates planimetric separation, which is often more influential for daylight transmission than vertical displacement.

Scaled vertical separation. The absolute vertical difference, normalised by room height, is given by

$$d_{z,s} = \left| \frac{dz}{\text{height}} \right|. \tag{C.13}$$

This feature highlights elevation differences between sensors and apertures, which play a key role in light penetration.

Squared direction cosines. Let $\hat{\mathbf{d}} = \mathbf{d}_s / \|\mathbf{d}_s\|$ denote the normalised scaled displacement. The squared directional cosines with respect to each axis are

$$\cos^2 \theta_x = \hat{d}_x^2, \quad \cos^2 \theta_y = \hat{d}_y^2, \quad \cos^2 \theta_z = \hat{d}_z^2. \tag{C.14}$$

These capture orientation cues in a rotation-invariant way, ensuring that the sign of the direction does not affect the feature values.

Directional projections. If a wall or window normal $\mathbf{n}$ is available, a local frame is constructed from the unit vectors

$$\mathbf{u} = (0, 0, 1), \quad \mathbf{t} = \mathbf{u} \times \mathbf{n}, \quad \mathbf{n},$$

scaled and normalised in the same way as $\mathbf{d}_s$. The displacement orientation is then projected onto these axes:

$$d_{\text{dot},t} = \hat{\mathbf{d}} \cdot \hat{\mathbf{t}}, \quad d_{\text{dot},u} = \hat{\mathbf{d}} \cdot \hat{\mathbf{u}}, \quad d_{\text{dot},n} = \hat{\mathbf{d}} \cdot \hat{\mathbf{n}}. \tag{C.15}$$

These projections provide interpretable geometric cues: alignment along the wall tangent, vertical alignment with the global up vector, and alignment with the wall normal. They help the model distinguish between light transfer paths that differ in orientation but not in distance.

C.3.3. Graph statistics

Table **??** summarises the structural properties of the constructed graphs across all dataset splits. Both homogeneous and heterogeneous graphs exhibit substantial variability in size, reflecting the diversity of room geometries. On average, heterogeneous graphs contain slightly more nodes than their homogeneous counterparts due to the additional window nodes, and they also feature a markedly higher edge count as a result of the denser connectivity introduced by aperture–sensor relations. The Transformation subset (formerly the ablation set) shows consistently smaller graphs, since transformations are applied only to a reduced set of base cases. These statistics highlight the balance between dataset diversity and computational tractability that guided the graph construction.

Table C.5: Detailed graph statistics across dataset splits. "All" aggregates Train, Val, and Transformation sets only.

Graph type	Split	#Graphs	Avg. nodes	Min. nodes	Max. nodes	Avg. edges	Min. edges	Max. edges
Homogeneous	All	236	80.35	9	196	1091.15	80	2808
	Train	180	82.7	9	196	1127.1	80	2808
	Val	20	83.8	16	196	1142.8	168	2808
	Transformation	36	66.7	36	100	882.7	440	1368
Heterogeneous	All	236	85.35	14	201	1496.93	129	3792
	Train	180	87.7	14	201	1544.7	129	3792
	Val	20	88.8	21	201	1565.5	252	3792
	Transformation	36	71.7	41	105	1220.0	624	1872

As shown in Table **??**, the Transformation split contains fewer nodes on average than the training and validation sets. This is a consequence of the selection strategy: only mid-range widths were chosen for the transformed rooms, excluding both very narrow and very wide configurations. As a result, the Transformation split avoids the extremes in room size that would otherwise lead to particularly small or particularly large graphs.

The dependence of graph size on room width arises from the way the sensor grid is constructed. The nominal target spacing between sensors is approximately $0.5\,\text{m}$ in both plan directions, but this value is adjusted slightly so that an integer number of sensors fits exactly within the room width. Consequently, the effective spacing Δx becomes a piecewise function of width, with small deviations around $0.5\,\text{m}$.

As shown in Fig. C.5, this leads to narrow intervals of reduced spacing (denser grids and more nodes) and wider intervals with slightly larger spacing (sparser grids and fewer nodes).

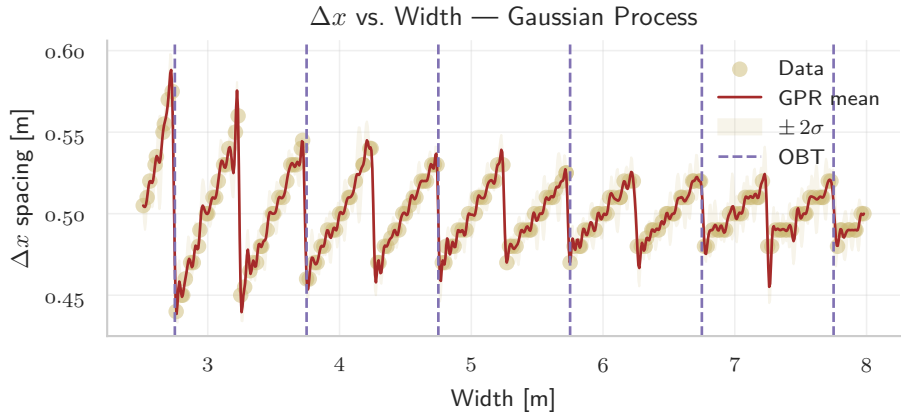


Figure C.5: Predicted sensor spacing Δx versus room width, with training and validation graphs combined for fitting. The GPR model (red line) with 2σ uncertainty band (shaded) reveals intervals of reduced spacing, which directly increases the average number of nodes observed in the OBT split. Vertical dashed lines indicate the widths used in the OBT set.

C.4. Operator-specific hyperparameters

This appendix summarises the hyperparameter search spaces explored for each operator during BO. Only operator-specific settings are listed; global training parameters (learning rate, batch size, dropout, etc.) were shared across all operators.

GENConv

Parameter	Range / Choices
Aggregation	{add, mean, max, softmax, softmax_sg, power}
t (softmax temperature)	[0.5, 2.0] (learnable or fixed)
p (power aggregation)	[0.5, 2.0] (learnable or fixed)
Message normalization	{on, off}, with learnable scale
Expansion factor	{2, …, 6}
Num layers per block	{1, 2, 3}
Bias	{True, False}

Table C.6: Hyperparameter search space for GENConv.

PNAConv

Parameter	Range / Choices
Aggregators	Combinations of {mean, max, sum}
Scalers	Subsets of {identity, amplification, attenuation, linear, inverse linear}
Towers	{1, 2, 3, 4} (divisibility constraint on hidden size)
Pre-layers / Post-layers	{1, 2}
Divide input	{True, False}

Table C.7: Hyperparameter search space for PNAConv.

NNConv

Parameter	Range / Choices
Edge MLP depth	{1, 2}
Edge MLP hidden size	{8, 16, 24, 32}
Aggregation	{mean, add}
Root weight	{True, False}
Bias	{True, False}

Table C.8: Hyperparameter search space for NNConv.

SplineConv

Parameter	Range / Choices
Pseudo-coordinates dimension d	{2, 3}
Kernel size	{3–6} (constrained by d)
Spline degree	{1, 2, 3}
Open spline	{True, False}
Aggregation	{mean, max, add}
Root weight	{True, False}
Bias	{True, False}
Use FFN block	{True, False}, with hidden mult. {2–4}

Table C.9: Hyperparameter search space for SplineConv.

GCNPlus

Parameter	Range / Choices
Residual connections	{True, False}
Batch normalization	{True, False}
Feed-forward network	{True, False}, with hidden mult. {2–4}
Normalization	{True, False}
Add self-loops	Fixed = False

Table C.10: Hyperparameter search space for GCNPlus.

C.5. Simulation and Implementation Details

This section summarises the computational settings used for the daylight simulations, the software environment used to construct and train the models, and the hardware on which the experiments were run. All configurations were kept consistent across the project to ensure reproducibility.

C.5.1. Radiance and Honeybee Simulation Settings

All DF simulations were performed using the standard Honeybee DF recipe, which internally relies on Radiance's implementation of the CIE standard overcast sky. No modifications were made to the default sky model, glass material, or surface reflectances supplied by Honeybee. Opaque surfaces therefore used the default internal reflectance values typically applied in Honeybee's DF workflow, and glazing was modelled using the default Honeybee glass primitive. Walls were modelled with a uniform thickness of $0.2\,\mathrm{m}$.

Radiance solver parameters (e.g. ambient bounces, ambient divisions, ray-tracing accuracy settings) were those automatically selected by the Honeybee DF recipe. These defaults are optimised for DF computations and provide a consistent balance between accuracy and runtime across all geometries in the dataset.

C.5.2. Rhino and Grasshopper Environment

All geometry construction and model export operations were performed in Rhino 8 using Grasshopper 1.22.0 and Ladybug Tools 1.8. The parametric Grasshopper definitions generated the room geometries, window configurations, and sensor grids, and passed them to Honeybee for Radiance simulation. No custom modifications were made to the Honeybee DF schema, and the same workflow was used for all dataset tiers to maintain uniformity.

C.5.3. Machine Learning Frameworks

All machine-learning experiments were conducted in Python 3.11 using PyTorch 2.8.0 with CUDA 12.6 and the corresponding PyTorch Geometric stack. Graph learning models, batching, and heterogeneous message passing were implemented using PyTorch Geometric with the CUDA 12.6–compatible wheels (torch-scatter, torch-sparse, torch-cluster, torch-spline-conv) obtained from the official `data.pyg.org` distribution. Bayesian optimisation of feature masks, operator variants, and architectural parameters used Optuna 3.5 or later. All experiments relied exclusively on stable, GPU-accelerated releases to ensure consistency across local and cloud-based environments.

C.5.4. Hardware Setup

Most simulations, preprocessing operations, and all model training runs were executed on a laptop equipped with an NVIDIA Quadro T2000 with Max-Q Design GPU and 32 GB of system RAM. This configuration was used for dataset generation, baseline model training, graph construction, and the full evaluation on the Final Test Dataset.

Bayesian optimisation experiments, which require repeated training of many candidate configurations, were performed on Google Colab using an NVIDIA A100 GPU with 40 GB of VRAM, available through a paid subscription. This environment provided the computational throughput necessary for large-scale search over model hyperparameters and operator variants.

C.5.5. Reproducibility and Random Seeds

Reproducibility was ensured through consistent use of fixed random seeds across all stages of the workflow. For the training and validation datasets, a seed of 42 was used for geometry sampling, sensor-grid generation, and graph construction. For the Tier 0–2 subsets of the Final Test Dataset, sampling used seed 43 to prevent overlap with the training distribution. All other deterministic sampling stages (e.g. Tier 3–5 generation, augmentation steps) used seed 42 unless stated otherwise.

Whenever multiple training repetitions were required, including Bayesian optimisation, mask selection, and final model retraining, the following seed set was always used:

$$\{40, 41, 42, 43, 44\}.$$

Seeds were applied consistently to Python's random module, NumPy, PyTorch, and Optuna trial samplers to the extent supported by each library. Radiance simulations were not subject to stochastic variation.

All simulation, preprocessing, and training code was version-controlled, and no stateful operations outside the controlled seed management were used.

D

Supplementary Result Details

D.1. Benchmarks

D.1.1. Inputs

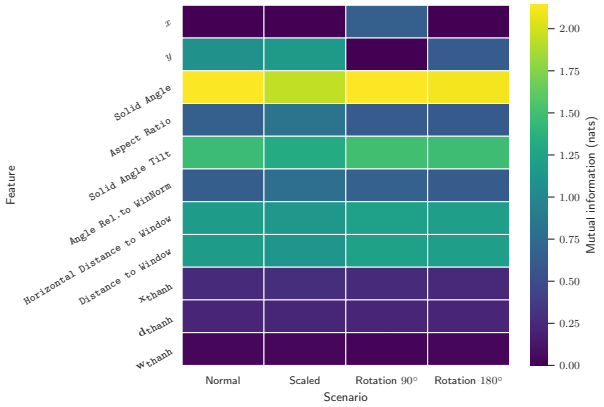


Figure D.1: Mutual information (MI) of all candidate features across geometric transformations (normal, scaled, $90°$ rotation, $180°$ rotation). The analysis shows that x, y, and the Le-Thanh encodings (x_{thanh}, d_{Thanh}, w_{Thanh}) are consistently less informative compared to other descriptors such as *Solid Angle*, *solidAngleTilt*, and *Distance2Window*.

D.1.2. Training Benchmarks

To assess the stability of the benchmark ANN feature sets, each configuration was trained with five different random seeds (40–44). Figures D.2–D.5 show the per-epoch training and validation MSE curves for all seeds. Solid lines denote training error, dashed lines denote validation error, and colors correspond to seeds.

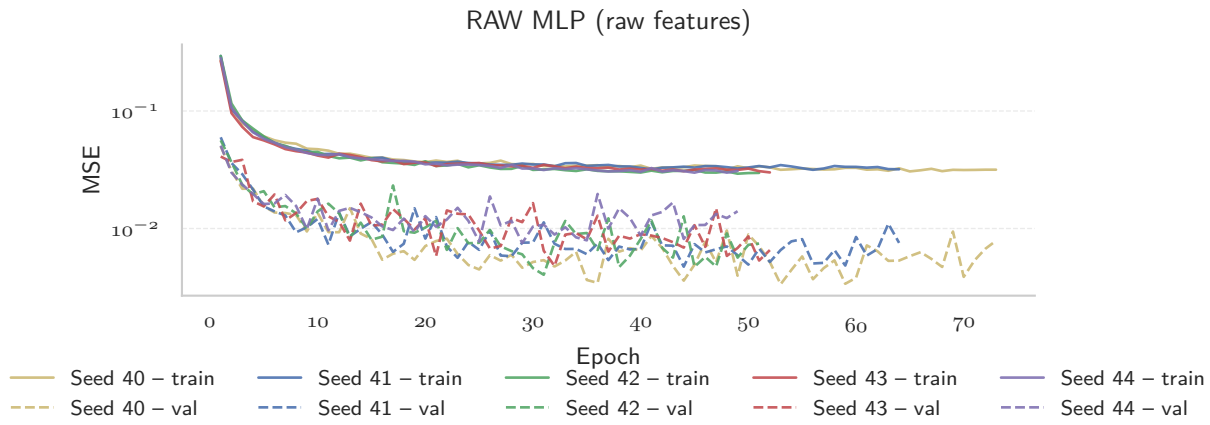


Figure D.2: Training history of the RAW MLP on raw feature inputs. Each colored curve corresponds to a seed (40–44). Validation curves exhibit minor variability but overall converge smoothly.

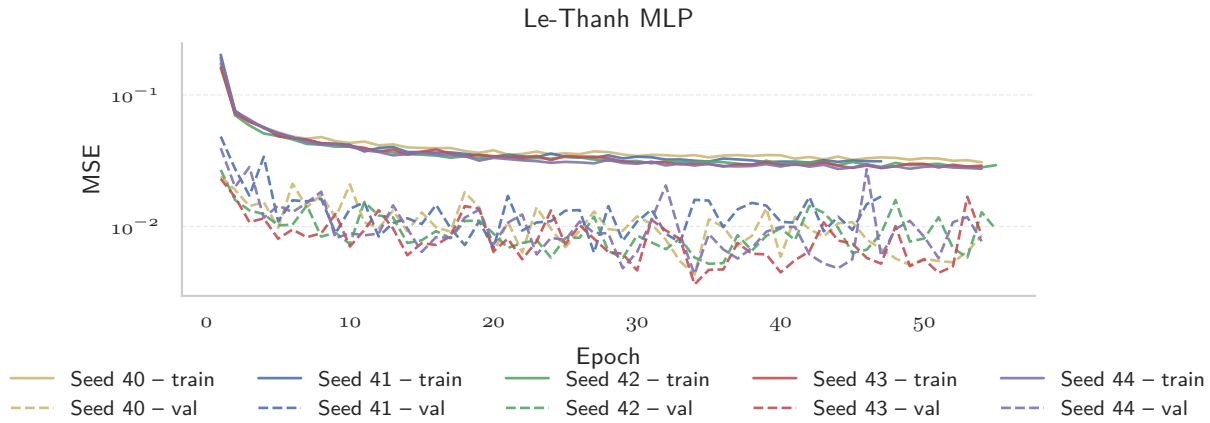


Figure D.3: Training history of the Le-Thanh MLP. Convergence behaviour is consistent across seeds, with validation MSE stabilising after 20–30 epochs.

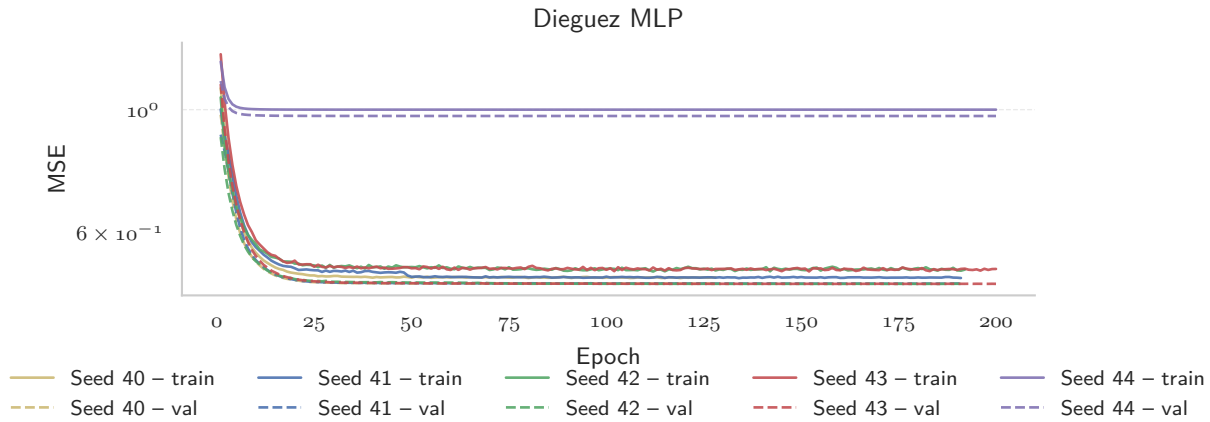


Figure D.4: Training history of the Dieguez MLP on Dieguez features. For most seeds the model converges steadily, but for seed 44 both training and validation losses plateau near the baseline predictor (MSE ≈ 1.0). This indicates that the model failed to move away from predicting the mean of the standardized targets. Such behaviour is expected occasionally under certain initializations, and is why results are reported as averages over multiple seeds.

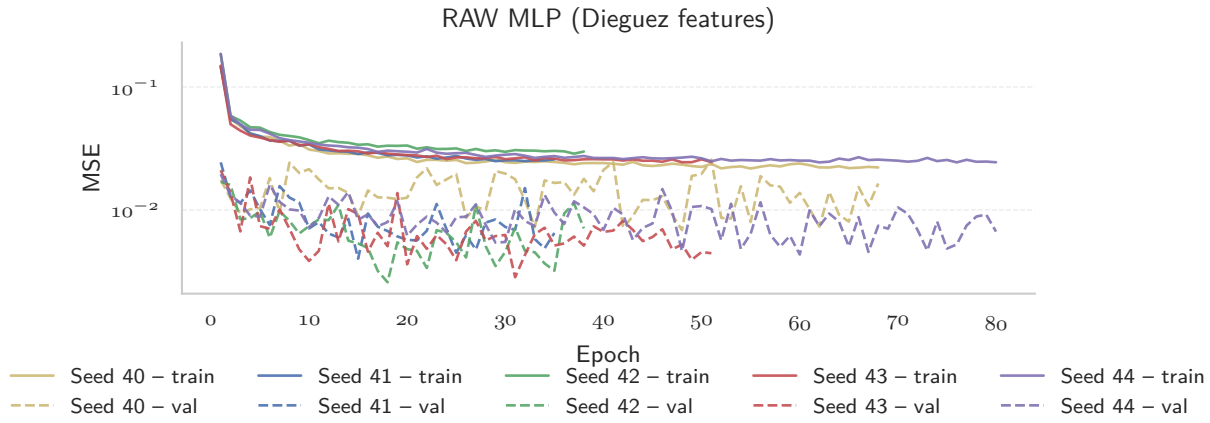


Figure D.5: Training history of the RAW MLP when trained on Dieguez features. Variability between seeds is larger than for the raw feature case, but all runs achieve satisfactory convergence.

Overall, these training histories demonstrate that the benchmark ANNs converge reliably under repeated training, with the exception of isolated runs (e.g. Dieguez seed 44). This underscores the importance of reporting mean and standard deviation across seeds rather than relying on a single best run.

D.1.3. Benchmark Results

Table D.1: Benchmark results across feature sets and transformations.

	Normal			Scaled			Rotation 90°			Rotation 180°		
Model	RMSE	MaxAbs	SSIM	RMSE	MaxAbs	SSIM	RMSE	MaxAbs	SSIM	RMSE	MaxAbs	SSIM
Dieguez	2.25	7.44	0.69	2.53	3.40	0.36	2.24	7.38	0.69	2.25	7.42	0.69
Le-Thanh	0.24	1.01	0.99	0.37	2.01	0.95	7.82	30.81	-0.16	5.51	15.86	-0.12
Raw	0.28	1.01	0.99	0.29	1.70	0.97	4.61	12.58	-0.06	5.44	13.16	-0.49
Raw+Dieguez	0.34	1.36	0.99	0.26	1.42	0.97	0.40	1.71	0.99	0.34	1.47	0.99

D.2. Feature Ablation

D.2.1. Homogeneous

Phase 1: Initial feature removal

Table D.2: Phase 1 Pareto front results with feature masks (mean ± standard deviation over 5 seeds).

index	studies	# hits	mask	RMSE	Max Abs	SSIM
0	0	1	[0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0]	0.494 ± 0.086	2.412 ± 0.702	0.932 ± 0.020
1	0	1	[0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 1]	0.499 ± 0.304	2.295 ± 1.309	0.960 ± 0.018
2	0	1	[0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0]	0.794 ± 0.188	2.394 ± 0.442	0.899 ± 0.051
3	0	1	[0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0]	0.529 ± 0.116	2.027 ± 0.400	0.917 ± 0.056
4	0	1	[0 0 1 0 1 0 0 1 0 1 0 0 1 0 0 0 1 0 1]	0.770 ± 0.125	2.713 ± 0.320	0.909 ± 0.044
5	0	1	[0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0]	0.731 ± 0.157	2.669 ± 0.267	0.883 ± 0.040
6	1	1	[0 0 1 0 0 1 0 1 0 0 0 0 1 0 0 1 1 0 1]	0.533 ± 0.076	2.028 ± 0.369	0.948 ± 0.013
7	1	1	[0 0 1 0 0 1 0 1 1 0 0 0 1 0 0 1 1 0 1]	0.522 ± 0.058	2.418 ± 0.575	0.960 ± 0.004
8	2	1	[0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1]	0.499 ± 0.202	1.869 ± 0.384	0.946 ± 0.042
9	2	1	[0 0 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 0 1]	0.466 ± 0.161	2.228 ± 0.653	0.942 ± 0.029
10	2	1	[0 0 1 1 0 1 1 1 0 0 0 0 1 0 0 0 1 0 1]	0.551 ± 0.094	2.178 ± 0.296	0.943 ± 0.014
11	2	1	[0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1]	0.600 ± 0.096	2.448 ± 0.144	0.938 ± 0.017
12	3	2	[1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 1]	0.606 ± 0.087	2.542 ± 0.503	0.953 ± 0.004
13	3	1	[1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0]	0.558 ± 0.101	2.006 ± 0.544	0.951 ± 0.010
14	3	1	[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 1]	0.429 ± 0.126	1.606 ± 0.369	0.931 ± 0.047
15	3	1	[0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 1 1]	0.427 ± 0.140	2.061 ± 1.017	0.947 ± 0.025
16	4	1	[1 0 1 0 0 0 0 0 1 1 0 0 1 0 0 0 1 0 1]	0.733 ± 0.199	2.900 ± 0.717	0.937 ± 0.017
17	4	1	[1 0 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0 1]	0.665 ± 0.082	2.248 ± 0.353	0.943 ± 0.012
18	4	1	[1 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 1 0 1]	0.631 ± 0.129	2.470 ± 0.204	0.949 ± 0.009
19	4	1	[1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 1]	0.585 ± 0.164	2.249 ± 0.424	0.919 ± 0.040
20	4	1	[1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1]	0.598 ± 0.091	2.390 ± 0.385	0.900 ± 0.051

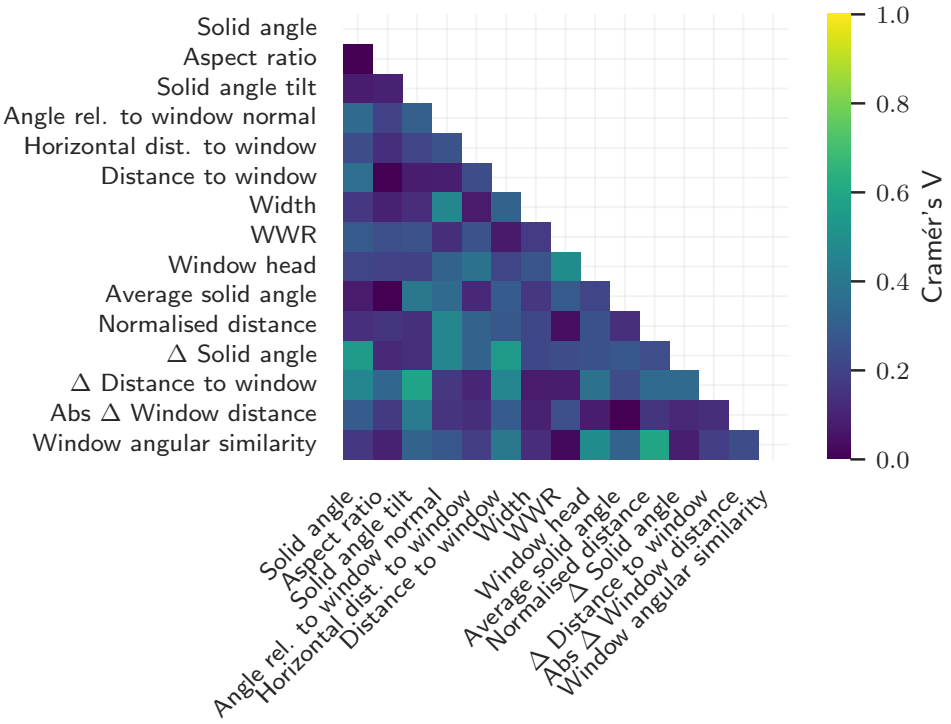


Figure D.6: Phase 1: feature-pair redundancy computed over Pareto-optimal trials. The heatmap shows the lower triangle of Cramér's V between feature inclusion indicators (1 = selected, 0 = not selected). Values near 1 (yellow) indicate strong co-selection dependence (high redundancy), while values near 0 (purple) indicate weak association. The diagonal is one by definition.

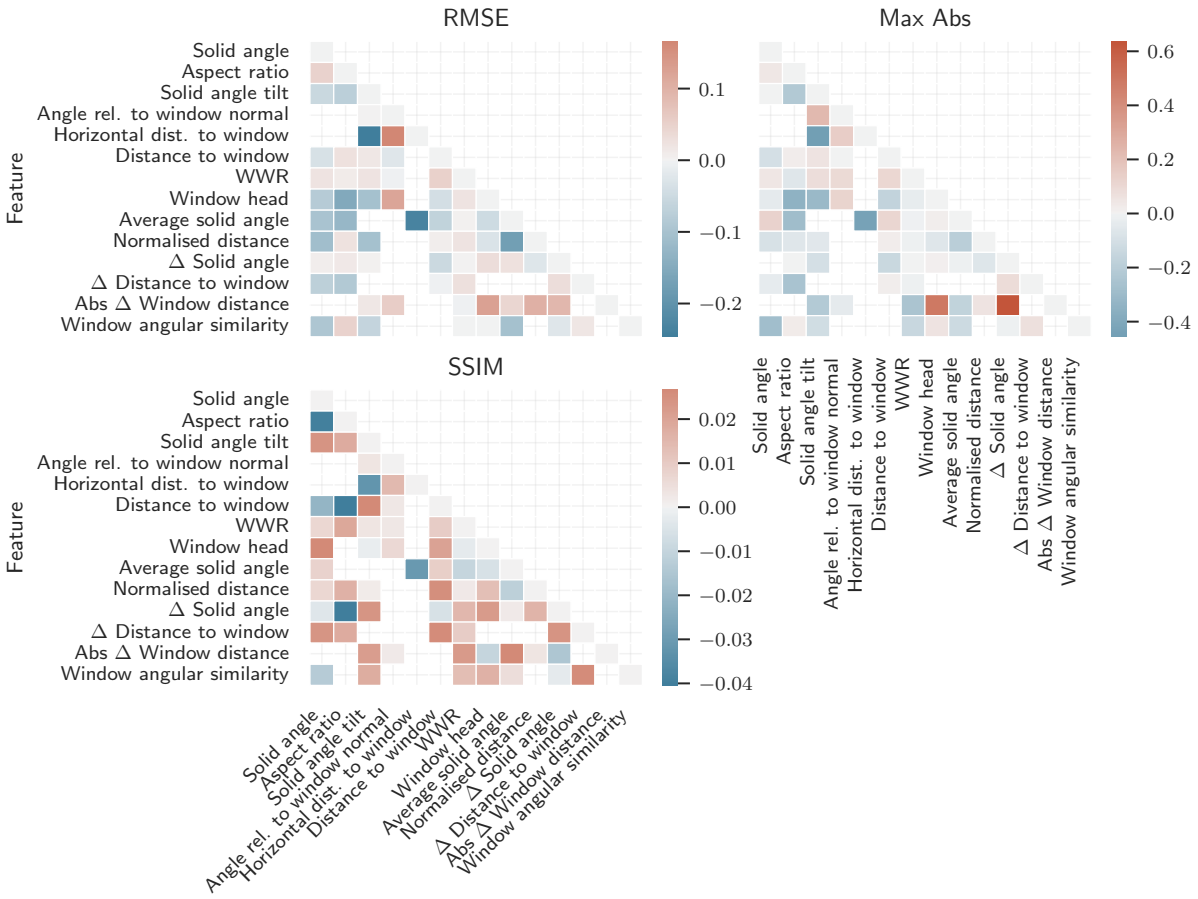


Figure D.7: Phase 1: pairwise feature synergy for the homogeneous model. Each panel shows the lower triangle of the pairwise *synergy* matrix for a metric (left: RMSE, right: Max Abs, bottom: SSIM). Colour is centred at zero in all panels. *Positive (reddish) values* indicate that using the two features together performs **better than the average of using them separately** (lower error for RMSE/Max Abs, higher value for SSIM) — desirable, complementary pairs. *Negative (bluish) values* indicate **negative synergy / redundancy** — the pair adds little or hurts performance relative to the singles. Diagonal entries are zero by design. Note that each panel uses its own colour range (different metric scales) but shares the same zero centre.

Phase 2: Mask selection

Table D.3: Homogeneous: Top 17 Pareto-optimal feature masks ranked by composite scores.

Index	Feature Mask	n_{hit}	RMSE	MaxAbs	SSIM	$\text{Score}_{\text{combo}}$	$\text{Score}_{\text{even}}$
4	[0,1,1,1,1,0,1,0,1,0,1]	2	0.451	1.183	0.894	0.0319	0.0527
6	[1,0,0,0,1,0,1,0,1,0,1]	1	0.467	1.954	0.910	0.0862	0.0923
5	[0,1,1,1,1,1,1,1,1,0,1]	1	0.480	2.504	0.848	0.2392	0.2569
0	[0,1,0,1,1,1,1,1,1,1,1]	1	0.543	2.011	0.862	0.2547	0.2011
8	[0,1,0,0,1,0,1,0,1,0,0]	2	0.535	2.050	0.889	0.2560	0.2078
10	[1,1,0,0,0,1,0,1,0,1,1]	1	0.570	1.748	0.901	0.2781	0.1906
13	[1,1,0,1,0,1,1,1,1,1,1]	1	0.567	1.884	0.814	0.3071	0.2438
3	[0,1,1,1,0,1,1,1,0,0,0]	1	0.533	2.325	0.840	0.3110	0.2772
9	[1,0,1,0,0,1,0,1,0,0,0]	1	0.561	2.091	0.890	0.3158	0.2434
11	[1,1,0,1,0,0,1,0,1,0,1]	1	0.579	2.286	0.901	0.3843	0.2978
1	[0,1,0,1,1,0,1,0,1,0,1]	1	0.580	2.196	0.892	0.3846	0.2973
2	[0,1,1,0,0,1,1,1,1,1,1]	1	0.540	2.386	0.893	0.4068	0.3663
7	[1,0,0,1,0,0,0,1,0,0,1]	1	0.591	2.228	0.892	0.4131	0.3171
12	[1,1,0,0,0,0,0,1,0,0,1]	2	0.602	2.425	0.851	0.4200	0.3188
15	[1,1,1,0,1,1,0,0,0,1,1]	1	0.574	2.385	0.833	0.4394	0.3748
16	[1,1,1,1,0,1,0,1,0,1,1]	1	0.616	2.463	0.911	0.5061	0.3892
14	[1,1,1,0,1,0,1,0,1,0,1]	1	0.699	3.425	0.899	0.8651	0.6932

Although WWR appears in the majority of shortlisted masks, its marginal contribution analysis indicates a slight negative effect on average performance. This apparent contradiction reflects redundancy: WWR is highly correlated with other geometric descriptors such as distance to window and solid angle. In masks where these descriptors are already present, adding WWR provides little new information and can even destabilise training. Conversely, masks without WWR can still achieve high robustness if they retain complementary descriptors (e.g. Mask 2), while weaker feature sets without WWR (e.g. Mask 3) perform poorly. Hence, the contribution analysis should be interpreted as marginal utility given other features, not as absolute feature importance.

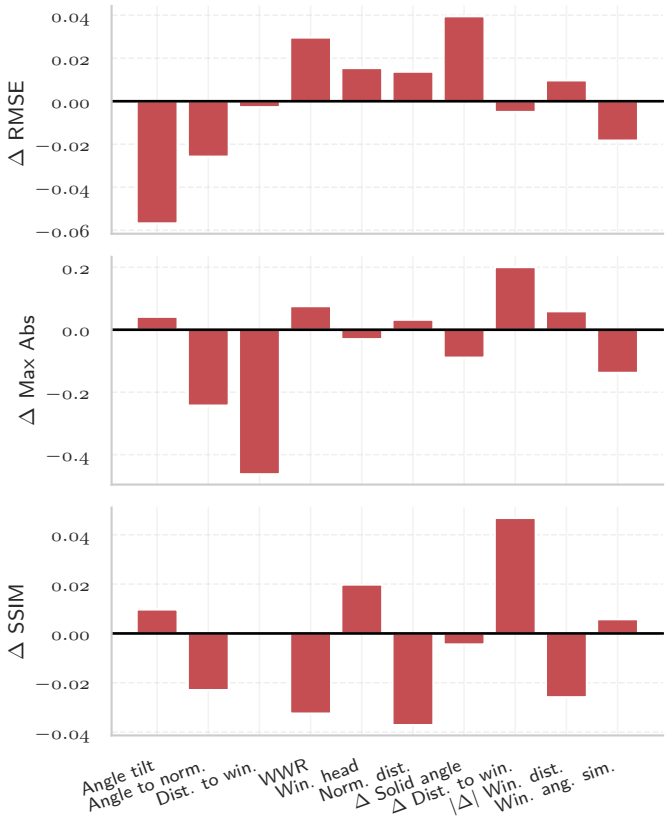


Figure D.8: Marginal performance contribution of each feature in the homogeneous Phase 2 Bayesian optimisation runs. Bars represent the mean change (Δ) in each performance metric when a given feature is included, relative to its exclusion across Pareto-optimal configurations. Positive values indicate performance improvement for all metrics (i.e., lower RMSE and MaxAbs, higher SSIM). In contrast to the broader Stage 1 search, no single descriptor exhibits a uniformly positive effect across all metrics. This reflects the interdependent nature of the refined feature set, where predictive performance emerges from synergistic combinations rather than from the influence of individual features alone. The remaining variability therefore captures minor local trade-offs within an already well-performing region of the search space, rather than strong additive effects of specific features.

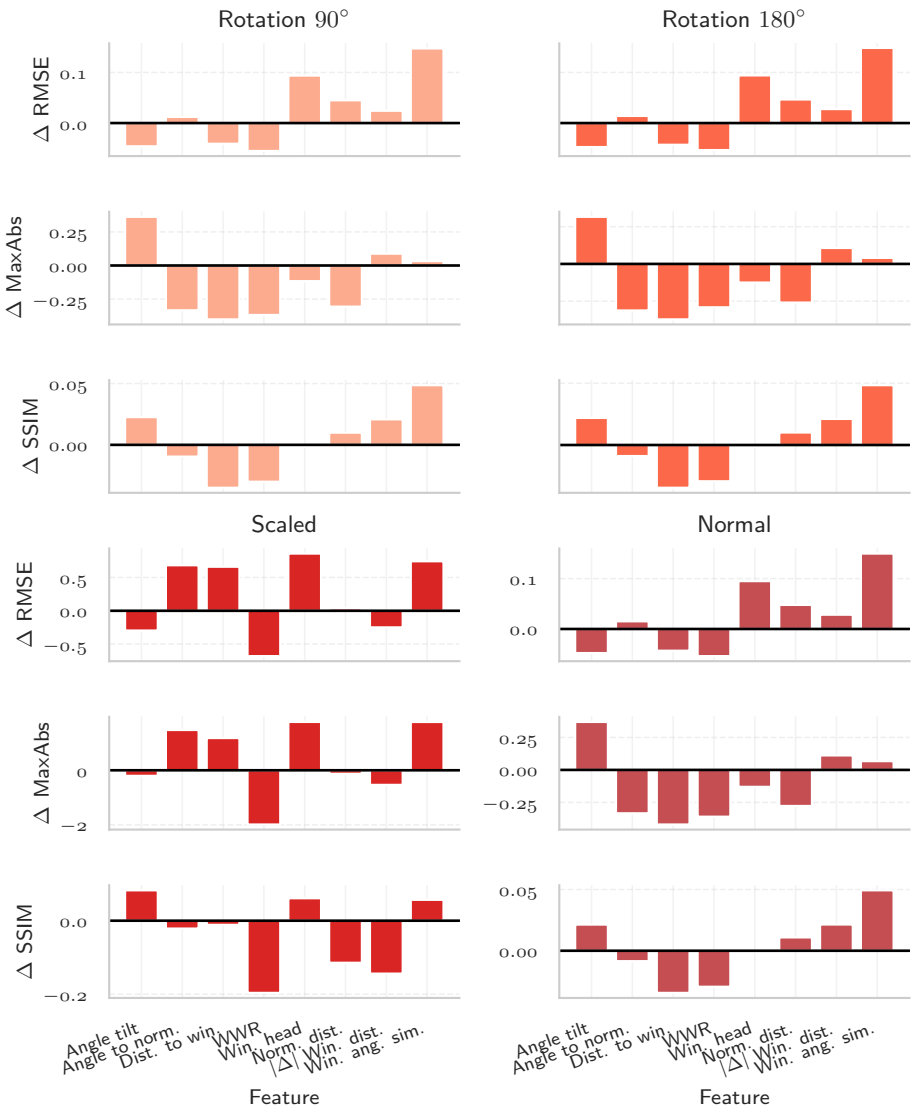


Figure D.9: Per-transformation contributions of homogeneous edge features for RMSE, MaxAbs, and SSIM across the top six re-evaluated masks. Panels correspond to rotation ($90°$, $180°$), scaling, and overall aggregation.

D.2.2. Heterogeneous graphs: single ablation run

Supplementary: Contribution analysis of heterogeneous features

The marginal contribution analysis in Figure D.10 provides a quantitative validation of the frequency-based findings presented in Section 5.1.4. Each bar represents the mean change in performance metrics (ΔRMSE, ΔMaxAbs, ΔSSIM) when a given feature is included, averaged across the Pareto-optimal configurations.

Consistent with the frequency results, Euclidean and normalised distance exhibit small but positive contributions to accuracy and stability, confirming their role as the most informative distance-based descriptors. In contrast, squared distance shows near-zero or negative Δ values across all metrics, reinforcing its redundancy relative to the other distance terms. Scaled distances display transformation-dependent behaviour: scaled 3D distance and vertical separation occasionally improve RMSE but degrade structural similarity, while scaled horizontal distance shows the opposite pattern. Among the angular relations, $\vec{d}\cdot\vec{t}$ and $\vec{d}\cdot\vec{n}$ provide modest improvements to RMSE and MaxAbs, though their effects remain secondary to the dominant distance-based features.

No single descriptor exhibits a consistently positive impact across all metrics, indicating that the predictive strength of the heterogeneous model arises primarily from the joint interaction of complementary descriptors rather than from any individual feature. The analysis therefore confirms that edge-level predictive performance depends on the collective balance between distance magnitude, orientation, and projection relations, rather than on isolated geometric quantities.

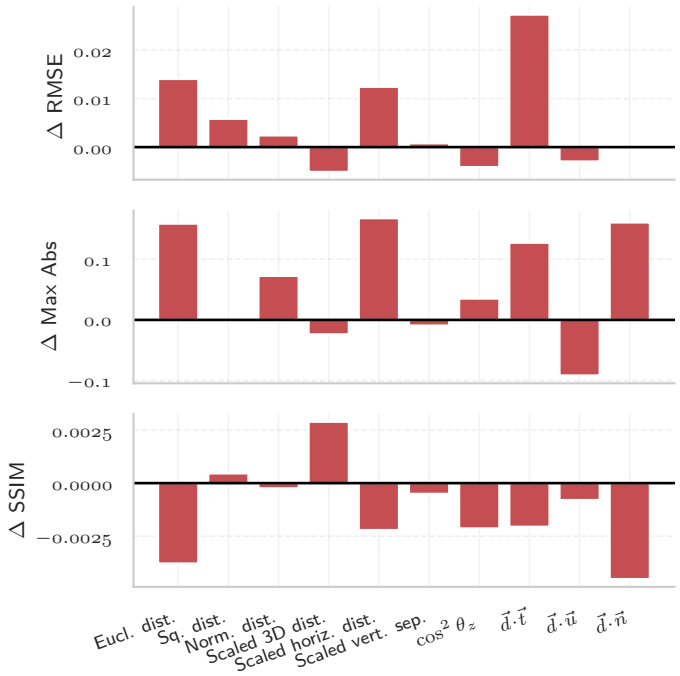


Figure D.10: Overall contributions of heterogeneous edge features to RMSE, MaxAbs, and SSIM within the Pareto-optimal subset. Positive values denote improved performance when the feature is included.

Table D.4: Heterogeneous: Top 20 Pareto-optimal feature masks ranked by composite scores.

Index	Feature Mask	n_{hit}	RMSE	MaxAbs	SSIM	$\text{Score}_{\text{combo}}$	$\text{Score}_{\text{even}}$
0	[1,1,1,1,0,1,0,0,0,1,1,1]	1	0.159	0.689	0.983	0.0026	0.0056
1	[1,1,1,0,1,1,0,0,0,1,1,1]	2	0.161	0.726	0.971	0.0347	0.0369
2	[1,1,1,0,1,0,0,0,0,1,0,1]	1	0.161	0.790	0.987	0.0609	0.0636
3	[1,1,1,0,1,1,0,0,1,1,1,1]	1	0.164	0.827	0.982	0.1139	0.1055
4	[1,1,1,1,1,0,0,0,0,1,0,1]	2	0.177	0.753	0.982	0.1924	0.1351
5	[1,0,1,0,1,0,0,0,1,1,1,1]	2	0.175	0.837	0.978	0.2154	0.1699
6	[1,1,1,0,1,0,0,0,1,1,0,1]	1	0.181	0.746	0.975	0.2259	0.1553
7	[1,0,0,0,1,0,0,0,1,1,1,1]	1	0.180	0.857	0.979	0.2749	0.2103
8	[1,0,1,0,0,0,0,0,0,1,1,1]	1	0.182	0.892	0.982	0.3045	0.2353
9	[0,1,0,1,1,0,0,0,0,1,0,1]	1	0.189	0.830	0.984	0.3438	0.2437
10	[1,0,1,0,1,1,0,0,1,0,1,1]	1	0.195	0.829	0.981	0.3920	0.2734
11	[1,0,1,0,1,0,0,0,0,1,1,1]	1	0.187	1.004	0.985	0.4021	0.3202
12	[1,0,1,1,1,1,0,0,0,1,0,1]	1	0.187	1.008	0.983	0.4070	0.3243
13	[1,0,0,0,1,0,0,0,0,1,1,1]	1	0.187	1.081	0.983	0.4373	0.3598
14	[0,1,1,1,1,0,0,0,0,1,0,0]	1	0.187	1.108	0.987	0.4560	0.3767
15	[1,0,1,1,1,1,0,0,1,0,1,1]	1	0.208	1.058	0.988	0.6194	0.4626
16	[0,1,1,0,0,1,0,0,0,1,1,1]	1	0.200	1.235	0.986	0.6357	0.5151
17	[1,1,0,1,0,0,0,0,1,1,1,1]	1	0.200	1.289	0.985	0.6614	0.5434
18	[1,0,1,1,0,0,0,0,1,0,0,1]	1	0.215	1.027	0.983	0.6711	0.4875
19	[0,1,1,0,1,0,0,0,0,1,1,1]	1	0.217	1.161	0.985	0.7559	0.5698

Per-transformation contribution analysis

This section presents a detailed breakdown of feature contributions across transformation scenarios for the heterogeneous edge-descriptor model. The analysis builds on the six shortlisted and re-evaluated masks discussed in the main text, providing complementary insight into how individual features affect predictive accuracy (RMSE, MaxAbs) and structural similarity (SSIM) under different geometric transformations. The purpose of this appendix is to confirm the consistency of the feature-level trends identified in the main chapter.

The results show clear transformation-dependent patterns. Squared distance, despite its frequent inclusion in Pareto-optimal masks, consistently degrades performance across all metrics and transformations, confirming its redundancy relative to the Euclidean and normalised distance encodings. Scaled 3D distance and scaled vertical separation improve predictions under rotations but reduce accuracy when scaling transformations are applied, whereas scaled horizontal distance shows the opposite behaviour. These trade-offs suggest that different transformation types emphasise distinct geometric dependencies, particularly in how depth and alignment are represented within the edge attributes. Among the angular terms, $\vec{d}\cdot\vec{t}$ and $\vec{d}\cdot\vec{n}$ exhibit stable positive effects under rotations, while their influence diminishes under scaling. Overall, scaling transformations exert a stronger influence on model robustness than rotations, as reflected by the broader value ranges across all metrics.

An aggregated contribution plot is provided in Figure D.12, confirming the same general trends. It reinforces the weak and often detrimental role of squared distance and the conditional importance of scaled and angular descriptors, without introducing additional insights beyond those of the per-transformation analysis. For this reason, the aggregate figure is reported here primarily for completeness.

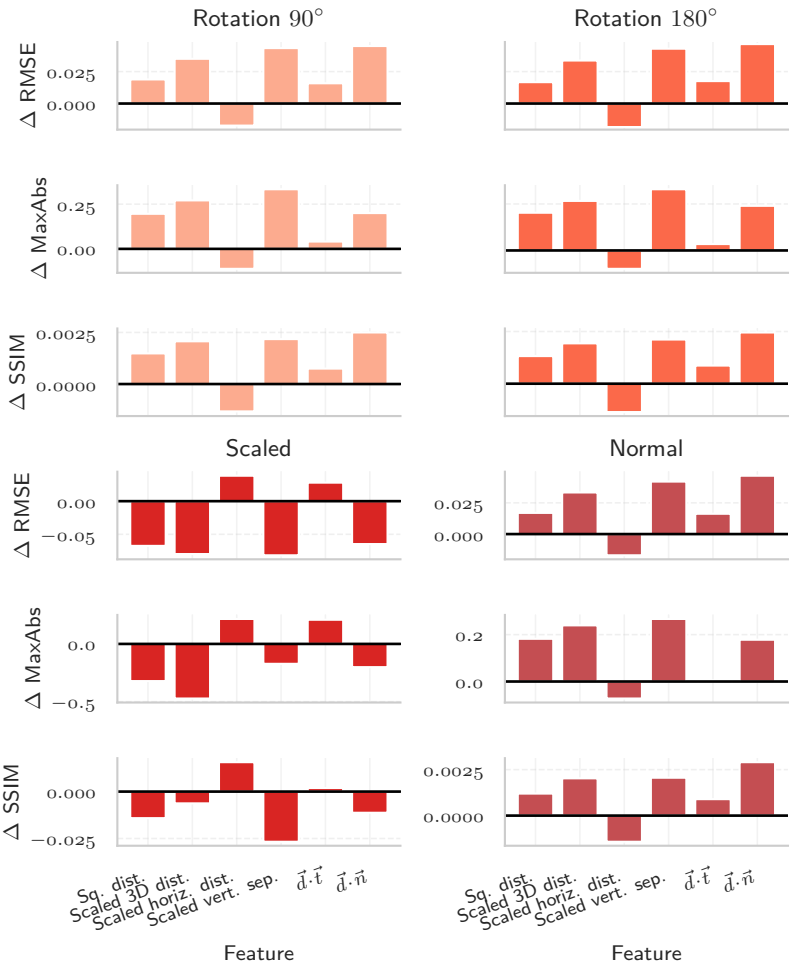


Figure D.11: Per-transformation contributions of heterogeneous edge features for RMSE, MaxAbs, and SSIM across the top six re-evaluated masks. Panels correspond to rotation ($90°$, $180°$), scaling, and overall aggregation. Positive values indicate performance improvement (lower RMSE and MaxAbs, higher SSIM) when the feature is included.

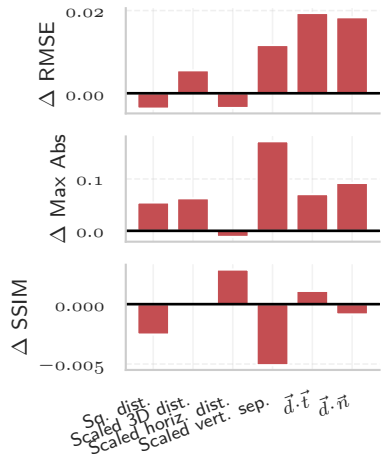


Figure D.12: Aggregate contribution analysis of heterogeneous edge features across all transformation scenarios. The plot confirms the limited role of squared distance and the transformation-specific relevance of scaled and angular relations.

D.2.3. Final feature model comparison

To complement the RMSE comparison presented in the main text, Figures D.13 and D.14 report the corresponding results for MaxAbs and SSIM across all transformation scenarios. Both metrics confirm the same overall trend: the heterogeneous model consistently outperforms the homogeneous configuration, achieving lower absolute error and higher structural similarity under normal, scaled, and rotated conditions. The improvement is particularly pronounced for the scaling transformation, where the heterogeneous formulation maintains accuracy and structural fidelity despite the geometric distortion.

Together, these results corroborate the robustness of the heterogeneous feature representation and demonstrate that its advantage extends beyond average prediction accuracy (RMSE) to include peak error control and spatial consistency in the predicted daylight distributions.

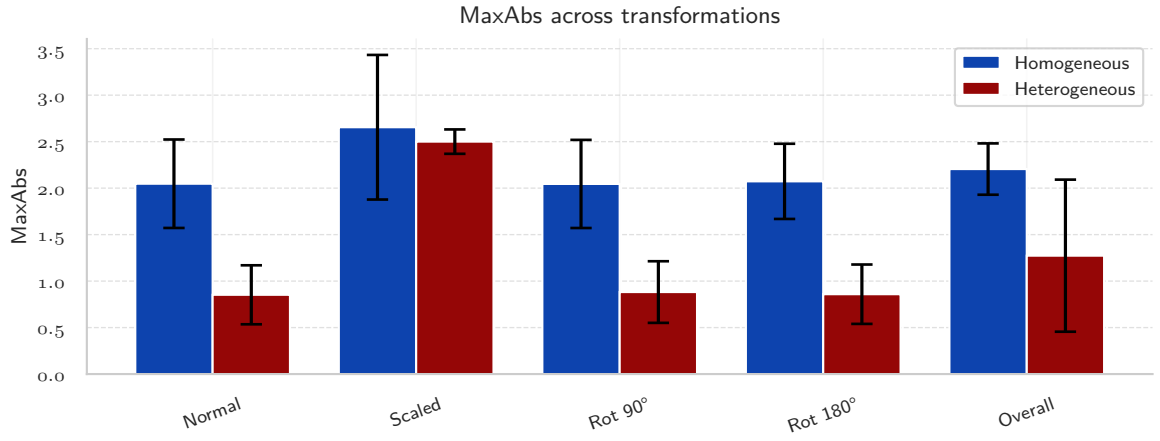


Figure D.13: Maximum absolute error (MaxAbs) for the final homogeneous and heterogeneous feature configurations across transformations. Bars show mean values; error bars indicate one standard deviation across seeds.

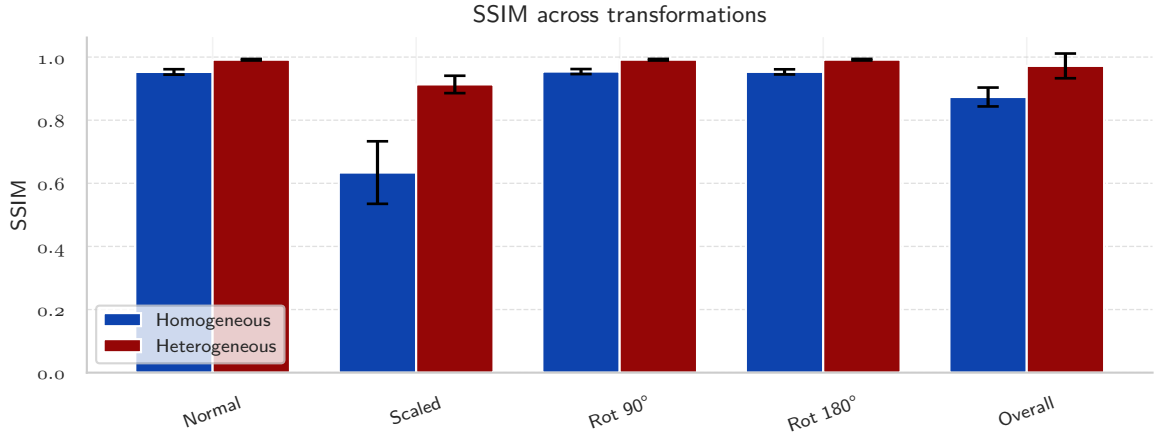


Figure D.14: SSIM for the final homogeneous and heterogeneous feature configurations across transformations. Bars show mean values; error bars indicate one standard deviation across seeds.

D.3. Architecture and Operator Design

D.3.1. Violin plots

The following figures provide an overview of the categorical and discrete hyperparameter distributions resulting from the Bayesian optimisation studies described in Section 5.2. Each violin represents the relative frequency of parameter selections among the best-performing trials across five independent Optuna runs for a given operator. Only discrete hyperparameters (e.g. aggregation type, normalisation scheme, activation, residual connection) are visualised. Continuous parameters such as `learning rate`, `weight_decay`, and `dropout` are omitted because their continuous ranges make direct frequency comparison uninformative. These plots are included here to document the optimisation landscape for

reproducibility and completeness.

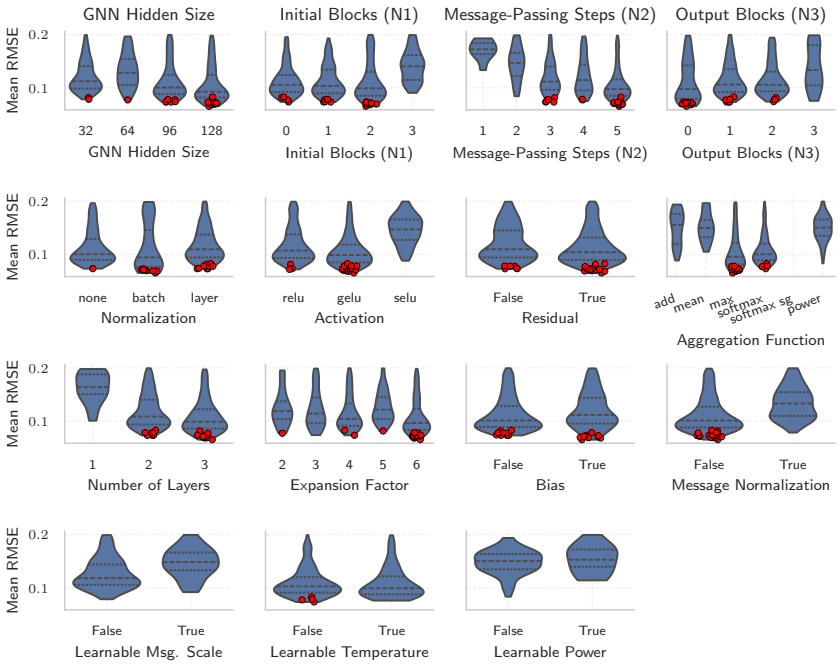


Figure D.15: Distribution of selected hyperparameter values for the homogeneous GENConv operator after Bayesian optimisation. Each violin represents the categorical frequency of the best-performing trials across five Optuna studies. Continuous parameters such as `learning rate`, `weight_decay`, and `dropout` are not shown.

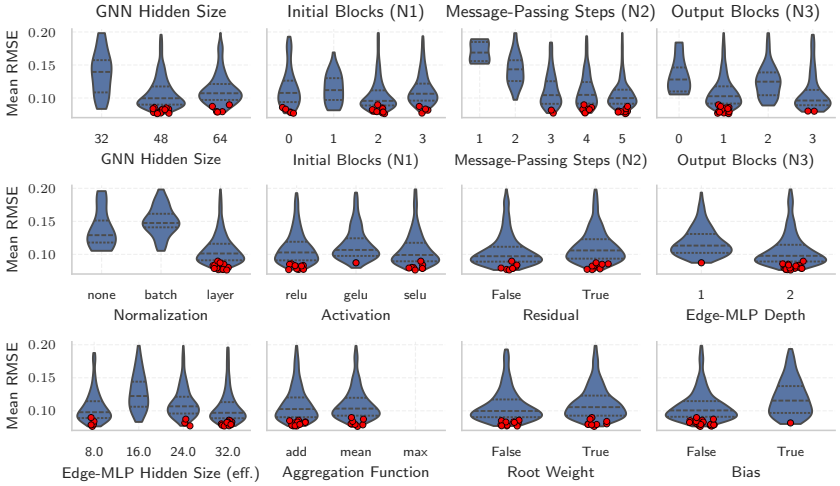


Figure D.16: Distribution of categorical hyperparameter selections for the homogeneous NNConv operator. Shown are only discrete search dimensions; continuous parameters (`lr`, `weight_decay`, `dropout`) are excluded.

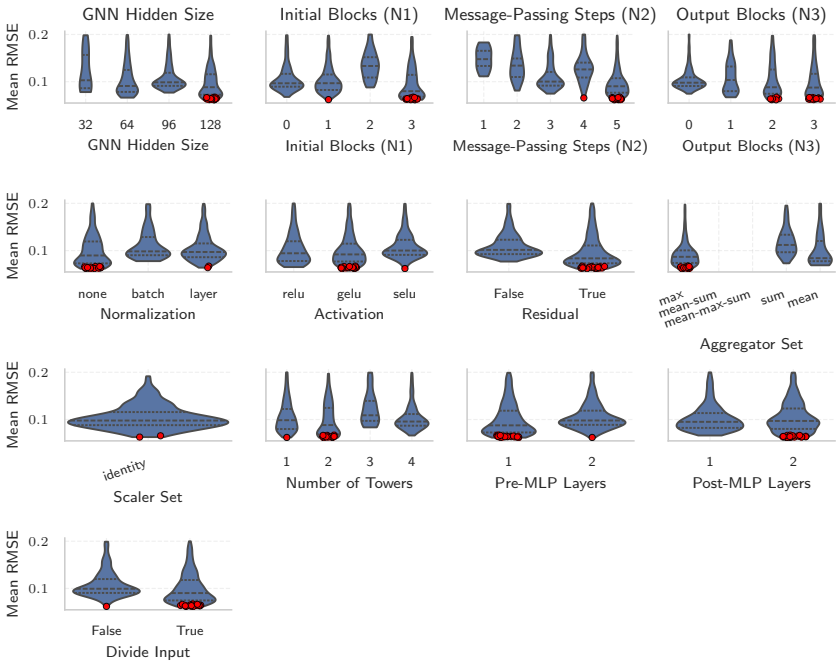


Figure D.17: Bayesian optimisation outcomes for discrete hyperparameters of the homogeneous PNAeConv operator. Continuous variables are omitted for clarity.

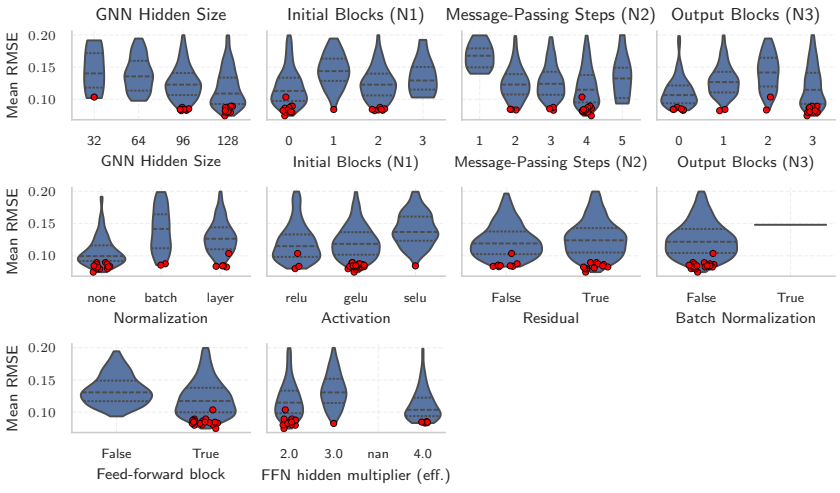


Figure D.18: Distribution of discrete hyperparameter values selected for the homogeneous GCN+ operator. Continuous parameters are omitted.

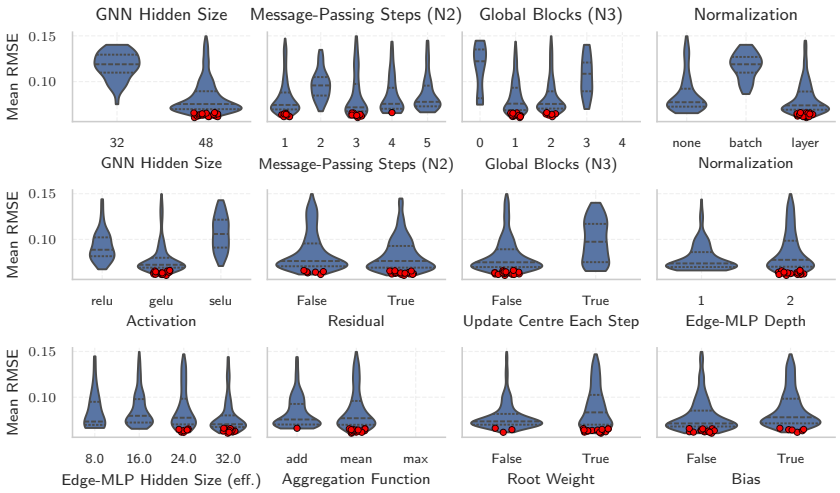


Figure D.19: Frequency distribution of categorical hyperparameter choices for the heterogeneous NNConv operator across five Optuna runs. Continuous parameters (`lr`, `weight_decay`, `dropout`) are excluded.

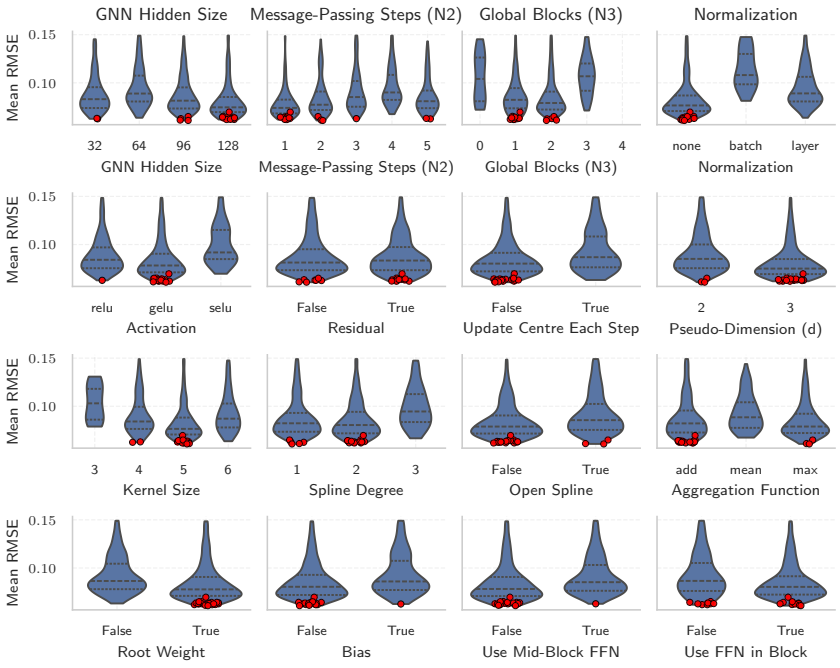


Figure D.20: Violin plot of categorical hyperparameter selections for the heterogeneous SplineConv operator. Continuous optimisation variables are not shown.

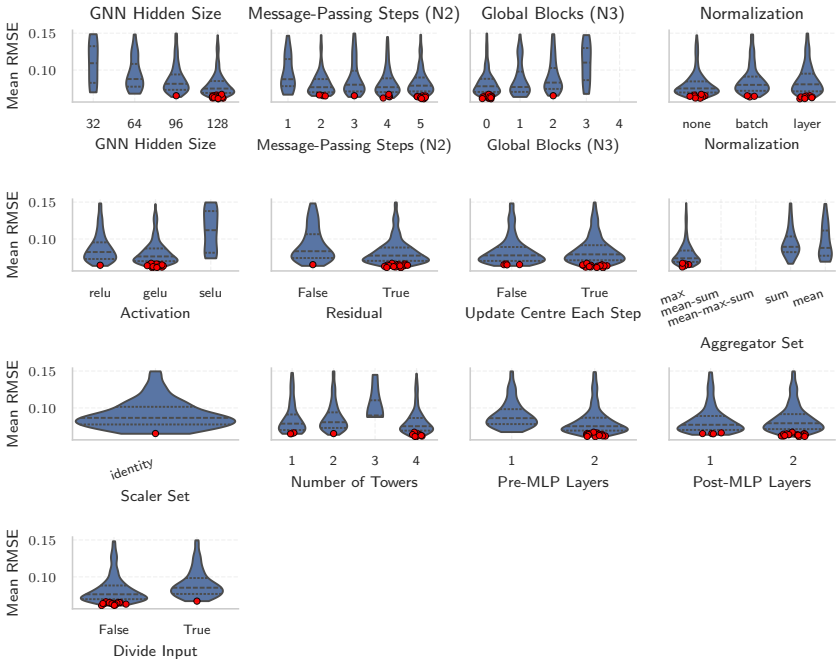


Figure D.21: Distributions of discrete hyperparameter values obtained for the heterogeneous PNAConv operator after Bayesian optimisation. Continuous search dimensions are excluded for clarity.

D.3.2. Elite Configurations

The tables in this section report the elite configurations for each GNN operator, defined as the single best-performing Optuna trial according to the validation rmse_mean. These settings represent the highest observed performance for each operator during optimisation, but they were not used in the final comparative experiments. Instead, the final models were based on a more stable and interpretable combination of hyperparameters drawn from the top-performing set of trials, prioritising consistency, architectural simplicity, and cross-operator comparability over marginal gains in validation error. The elite configurations documented here should therefore be viewed as upper-bound reference results rather than the chosen working configurations.

Table D.5: Elite (single best trial) shared hyperparameters for homogeneous and heterogeneous GNN operators. These configurations correspond to the best-performing Optuna trial per operator, not the final chosen settings used in subsequent experiments.

Graph type	Operator	n	N_1	N_2	N_3	Norm.	Act.	Res.	U.C.	Dropout	LR	Weight decay
Homo.	NNConv	48	2	5	2	layer	selu	False	–	0.0	8.99×10^{-3}	4.02×10^{-6}
Homo.	GENConv	128	2	5	1	batch	gelu	True	–	0.0	5.42×10^{-4}	1.97×10^{-6}
Homo.	SplineConv	96	3	5	2	none	gelu	True	–	0.0	6.04×10^{-4}	5.13×10^{-6}
Homo.	PNAConv	128	3	5	3	none	gelu	True	–	0.0	2.38×10^{-3}	1.65×10^{-6}
Homo.	GCN+	128	1	4	4	none	gelu	True	–	0.0	2.04×10^{-3}	1.68×10^{-6}
Hetero.	NNConv	48	–	3	2	layer	gelu	True	False	0.0	7.32×10^{-4}	3.27×10^{-6}
Hetero.	GENConv	96	–	3	1	layer	gelu	True	True	0.0	5.03×10^{-4}	2.22×10^{-6}
Hetero.	SplineConv	96	–	2	3	none	gelu	False	False	0.0	5.82×10^{-3}	2.84×10^{-6}
Hetero.	PNAConv	128	–	5	1	layer	gelu	True	True	0.05	1.17×10^{-3}	2.39×10^{-6}

Table D.6: Elite (single best trial) operator-specific hyperparameters for heterogeneous GNN operators. Values correspond to the best-performing Optuna trial per operator and are not necessarily identical to the final chosen configurations.

(a) NNConv

Param	Value
Edge-MLP depth	2
Edge-MLP size	16
Aggregation	mean
Root weight	True
Bias	False

(b) GENConv

Param	Value
Aggregation	`softmax`
# Layers	2
Expansion	5
Bias	False
Msg norm	False

(c) SplineConv

Param	Value
d_{pseudo}	2
Kernel size	5
Degree	1
Open spline	True
Aggregation	max
Root weight	True
Bias	False
Mid-block FFN	False
FFN in block	True

(d) PNAConv

Param	Value
Aggregators	`mean_max`
Scalers	`identity_amp`
Towers	4
Pre-MLP layers	2
Post-MLP layers	2
Divide input	False

Table D.7: Elite (single best trial) operator-specific hyperparameters for homogeneous GNN operators. Values correspond to the best-performing Optuna trial per operator and are not necessarily identical to the final chosen configurations.

(a) NNConv

Param	Value
Edge-MLP depth	2
Edge-MLP size	8
Aggregation	mean
Root weight	True
Bias	False

(b) GENConv

Param	Value
Aggregation	`max`
# Layers	3
Expansion	6
Bias	True
Msg norm	False

(c) SplineConv

Param	Value
d_{pseudo}	3
Kernel size	5
Degree	1
Open spline	False
Aggregation	max
Root weight	False
Bias	True
Mid-block FFN	False
FFN in block	False

(d) PNAConv

Param	Value
Aggregators	`max`
Scalers	`identity_att`
Towers	2
# Pre-MLP	1
# Post-MLP	2
Divide input	True

(e) GCNPlus

Param	Value
Use BN	False
Use FFN	True
Norm. ad.	True
Self-loops	False

D.4. Selection Results Models

This appendix complements the operator-selection results presented in Section 5.3 by reporting the full numerical metrics for all GNN operators across the four geometric transformations used for evaluation. Tables D.8–D.10 provide the mean and standard deviation of *RMSE*, *MaxAbs*, and *SSIM* for each operator under *Normal*, *Rot. 90°*, *Rot. 180°*, and *Scaled* conditions. These values correspond to the same runs summarised visually in the main text and allow direct comparison of absolute error levels, variability across seeds, and sensitivity to transformation type.

Table D.11 then aggregates these results into a compact robustness summary, reporting worst-case errors, CVaR_2, and transformation-averaged performance, together with a composite rank score used to support the model-selection process. While the qualitative trends—particularly the convergence of heterogeneous operators under rotation and their divergence under scaling—are discussed in the main text, the full quantitative results are included here for completeness and reproducibility.

Table D.8: RMSE (mean ± std) for all operators under geometric transformations.

Model	Normal	Rot. 90°	Rot. 180°	Scaled
Heterogeneous				
NNConv	0.0665 ± 0.0081	0.0650 ± 0.0072	0.0634 ± 0.0065	0.5070 ± 0.2580
GENConv	0.0675 ± 0.0103	0.0658 ± 0.0092	0.0666 ± 0.0070	0.3507 ± 0.0907
SplineConv	0.0662 ± 0.0099	0.0621 ± 0.0071	0.0635 ± 0.0087	0.4863 ± 0.1579
PNAConv	0.0663 ± 0.0096	0.0639 ± 0.0066	0.0651 ± 0.0080	0.7787 ± 0.1062
Homogeneous				
NNConv	0.2709 ± 0.0911	0.2707 ± 0.0933	0.2693 ± 0.0900	0.6070 ± 0.1772
GENConv	0.5940 ± 0.1274	0.5932 ± 0.1280	0.5910 ± 0.1229	0.4672 ± 0.1285
SplineConv	0.4614 ± 0.1040	0.4591 ± 0.1008	0.4573 ± 0.1001	0.3682 ± 0.1046
PNAConv	0.4397 ± 0.0765	0.4368 ± 0.0782	0.4361 ± 0.0719	0.5419 ± 0.1047
GCN+	0.4846 ± 0.1020	0.4883 ± 0.1003	0.4814 ± 0.0970	0.4289 ± 0.0989

Table D.9: MaxAbs (mean ± std) for all operators under geometric transformations.

Model	Normal	Rot. 90°	Rot. 180°	Scaled
Heterogeneous				
NNConv	0.1950 ± 0.0394	0.1846 ± 0.0359	0.1918 ± 0.0421	2.0648 ± 1.2264
GENConv	0.1961 ± 0.0399	0.1928 ± 0.0469	0.1892 ± 0.0307	1.4076 ± 0.4524
SplineConv	0.1876 ± 0.0278	0.1829 ± 0.0393	0.1887 ± 0.0409	2.1283 ± 0.8872
PNAConv	0.1921 ± 0.0383	0.1816 ± 0.0361	0.1853 ± 0.0375	3.0583 ± 0.5896
Homogeneous				
NNConv	0.6288 ± 0.2099	0.6207 ± 0.2295	0.6272 ± 0.2322	2.1692 ± 0.8479
GENConv	1.3954 ± 0.3737	1.3939 ± 0.3878	1.4059 ± 0.3904	1.5399 ± 0.3702
SplineConv	1.2965 ± 0.3134	1.3093 ± 0.2981	1.3026 ± 0.3263	1.0612 ± 0.4504
PNAConv	0.9471 ± 0.2103	0.9422 ± 0.2382	0.9604 ± 0.2211	2.0099 ± 0.4667
GCN+	1.2014 ± 0.3721	1.2241 ± 0.3684	1.2085 ± 0.3859	1.3760 ± 0.2311

Table D.10: SSIM (mean ± std) for all operators under geometric transformations.

Model	Normal	Rot. 90°	Rot. 180°	Scaled
Heterogeneous				
NNConv	0.99963 ± 0.00009	0.99963 ± 0.00013	0.99966 ± 0.00010	0.96923 ± 0.02344
GENConv	0.99962 ± 0.00012	0.99963 ± 0.00014	0.99964 ± 0.00012	0.98202 ± 0.00887
SplineConv	0.99963 ± 0.00011	0.99966 ± 0.00010	0.99966 ± 0.00012	0.97736 ± 0.01476
PNAConv	0.99964 ± 0.00009	0.99965 ± 0.00010	0.99967 ± 0.00008	0.94803 ± 0.02033
Homogeneous				
NNConv	0.99558 ± 0.00284	0.99553 ± 0.00295	0.99564 ± 0.00278	0.94424 ± 0.03063
GENConv	0.97914 ± 0.01187	0.97935 ± 0.01225	0.97939 ± 0.01164	0.96282 ± 0.01605
SplineConv	0.98956 ± 0.00619	0.98968 ± 0.00633	0.98976 ± 0.00607	0.96915 ± 0.01544
PNAConv	0.98919 ± 0.00552	0.98939 ± 0.00591	0.98942 ± 0.00536	0.96171 ± 0.01855
GCN+	0.98779 ± 0.00686	0.98769 ± 0.00710	0.98798 ± 0.00670	0.96298 ± 0.01353

Table D.11: Robustness summary across geometric transformations, including worst-case errors, CVaR-2, averages over transformations, and a composite rank score.

Model	Worst RMSE	Worst MaxAbs	Worst SSIM Loss	$CVaR_2$ RMSE	Avg. RMSE	Avg. MaxAbs	Avg. SSIM	Rank Score
HET. GENConv	0.3507	1.4076	0.01798	0.2091	0.1377	0.4964	0.99523	1.4
HET. SplineConv	0.4863	2.1283	0.02264	0.2763	0.1696	0.6719	0.99408	3.2
HOMO. SplineConv	0.4614	1.3093	0.03085	0.4602	0.4365	1.2424	0.98454	3.8
HET. NNConv	0.5070	2.0648	0.03077	0.2868	0.1755	0.6591	0.99204	4.0
HOMO. GCN+	0.4883	1.3760	0.03702	0.4865	0.4708	1.2525	0.98161	5.2
HOMO. PNAConv	0.5419	2.0099	0.03829	0.4908	0.4636	1.2149	0.98243	6.6
HET. PNAConv	0.7787	3.0583	0.05197	0.4225	0.2435	0.9043	0.98675	6.8
HOMO. GENConv	0.5940	1.5399	0.03718	0.5936	0.5614	1.4338	0.97518	7.0
HOMO. NNConv	0.6070	2.1692	0.05576	0.4390	0.3545	1.0115	0.98274	7.0

E

Extended Evaluation on the Final Test Dataset

E.1. Supplementary Metrics: MaxAbs and SSIM

This appendix complements the RMSE-based evaluation presented in Chapter 6 by reporting additional performance metrics for the same experiments. The *maximum absolute error* (MaxAbs) quantifies the largest pointwise deviation between predicted and reference daylight factor values, highlighting localised failure cases, while the *structural similarity index measure* (SSIM) captures the spatial agreement of illumination patterns across the sensor grid. Both metrics are computed over the same set of five random-seed repetitions described in Section 4.8. All figures follow the same tier grouping and visual format as their RMSE counterparts for direct comparison.

E.1.1. Tiers 0-2

Table E.1: Quantitative performance (mean ± SD) for Tiers 0–2 of the Final Test Dataset. Values are averaged over five independent training repetitions. Best metrics per variant are **bold**

Variant	Model	RMSE	MaxAbs	SSIM
Tier 0 – Base (Square)				
Base (Square)	WindowGraphNet	0.28 ± 0.06	0.88 ± 0.26	0.991 ± 0.004
	Raw ANN	**0.25 ± 0.03**	0.79 ± 0.10	**0.994 ± 0.001**
	Le-Thanh ANN	0.26 ± 0.03	**0.75 ± 0.06**	**0.994 ± 0.001**
	Dieguez ANN	2.51 ± 0.51	6.61 ± 1.48	0.575 ± 0.292
	Simple Dieguez ANN	0.56 ± 0.06	1.31 ± 0.11	0.981 ± 0.003
Tier 1 – Geometric Transformations				
Rotated 180°	WindowGraphNet	**0.28 ± 0.06**	**0.87 ± 0.25**	**0.992 ± 0.004**
	Raw ANN	5.62 ± 0.05	11.38 ± 0.05	−0.599 ± 0.005
	Le-Thanh ANN	7.50 ± 1.86	16.21 ± 2.39	0.251 ± 0.205
	Dieguez ANN	2.51 ± 0.51	6.61 ± 1.47	0.576 ± 0.292
	Simple Dieguez ANN	0.56 ± 0.06	1.32 ± 0.11	0.981 ± 0.003
Rotated 90°	WindowGraphNet	**0.28 ± 0.06**	**0.88 ± 0.26**	**0.991 ± 0.004**
	Raw ANN	4.68 ± 0.05	10.65 ± 0.08	−0.051 ± 0.010
	Le-Thanh ANN	8.57 ± 1.08	26.18 ± 1.76	−0.051 ± 0.056
	Dieguez ANN	2.50 ± 0.51	6.61 ± 1.47	0.574 ± 0.292
	Simple Dieguez ANN	0.62 ± 0.06	1.74 ± 0.31	0.979 ± 0.004
Scaled × 2	WindowGraphNet	0.66 ± 0.13	1.75 ± 0.19	0.879 ± 0.052
	Raw ANN	**0.35 ± 0.04**	1.30 ± 0.17	**0.966 ± 0.009**
	Le-Thanh ANN	0.38 ± 0.04	**1.25 ± 0.20**	0.953 ± 0.014
	Dieguez ANN	2.47 ± 0.15	3.56 ± 0.68	0.317 ± 0.152
	Simple Dieguez ANN	0.65 ± 0.24	1.50 ± 0.29	0.883 ± 0.073
Tier 2 – Rectangular and Scaled Variants				
Rectangular (Wide)	WindowGraphNet	2.09 ± 0.12	5.42 ± 0.38	0.792 ± 0.022
	Raw ANN	**0.62 ± 0.08**	**2.58 ± 0.28**	**0.985 ± 0.005**
	Le-Thanh ANN	3.22 ± 0.23	8.44 ± 0.70	0.485 ± 0.075
	Dieguez ANN	2.85 ± 0.55	8.34 ± 1.23	0.555 ± 0.280
	Simple Dieguez ANN	1.52 ± 0.07	6.47 ± 0.14	0.940 ± 0.008
Rectangular (Tall)	WindowGraphNet	1.19 ± 0.13	3.68 ± 0.30	0.771 ± 0.067
	Raw ANN	**0.39 ± 0.06**	**1.00 ± 0.17**	**0.848 ± 0.035**
	Le-Thanh ANN	4.89 ± 0.31	10.46 ± 0.28	0.259 ± 0.038
	Dieguez ANN	2.91 ± 0.22	6.67 ± 1.43	0.295 ± 0.082
	Simple Dieguez ANN	1.28 ± 0.09	4.69 ± 0.19	0.822 ± 0.039
Scaled × 5 (Square)	WindowGraphNet	3.05 ± 1.79	5.20 ± 2.18	0.119 ± 0.080
	Raw ANN	**1.54 ± 0.40**	**3.72 ± 0.91**	**0.189 ± 0.124**
	Le-Thanh ANN	2.33 ± 0.37	4.34 ± 0.35	−0.083 ± 0.020
	Dieguez ANN	3.49 ± 0.48	4.29 ± 0.95	0.010 ± 0.012
	Simple Dieguez ANN	1.89 ± 1.02	2.54 ± 0.99	0.179 ± 0.189

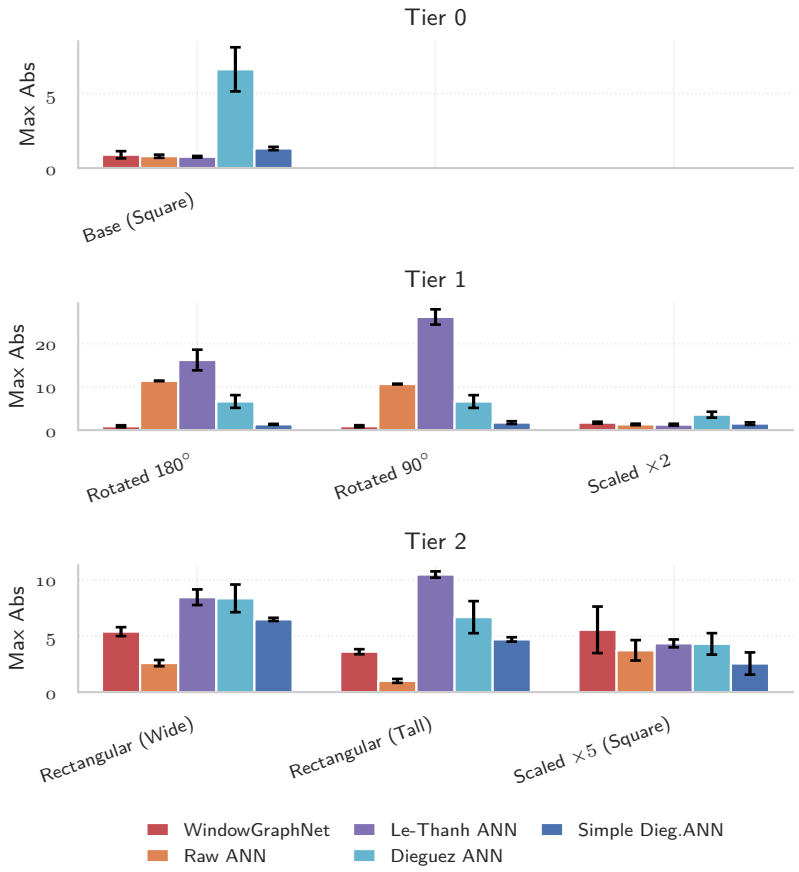


Figure E.1: Maximum absolute error (MaxAbs) of *WindowGraphNet* and ANN baselines for Tiers 0–2 of the Final Test Dataset. The figure mirrors the RMSE layout of Figure 6.1, covering in-distribution and rectangular transformations. Bars show mean values; error bars indicate one standard deviation across repetitions.

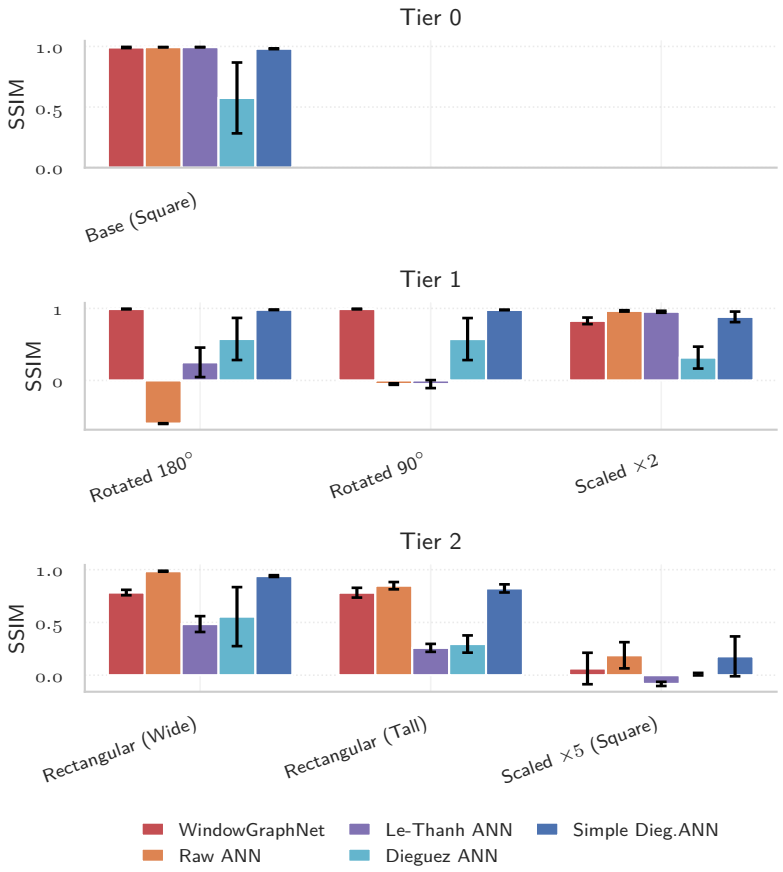


Figure E.2: Structural similarity index measure (SSIM) of *WindowGraphNet* and ANN baselines for the first three tiers of the Final Test Dataset. Higher values indicate closer spatial correspondence between predicted and true DF distributions.

E.1.2. Tiers 3-5

Table E.2: Quantitative performance (mean ± SD) for Tiers 3–5 of the Final Test Dataset. Values are averaged over five independent training repetitions.

Tier	Model	RMSE	MaxAbs	SSIM
Tier 3 – Rectangular Offset Windows				
	WindowGraphNet	**1.93 ± 0.07**	4.58 ± 0.21	0.677 ± 0.019
	Raw ANN	2.97 ± 0.04	8.18 ± 0.18	0.481 ± 0.007
	Le-Thanh ANN	2.74 ± 0.20	7.01 ± 0.53	0.359 ± 0.102
	Dieguez ANN	2.41 ± 0.35	4.95 ± 1.26	0.448 ± 0.236
	Simple Dieguez ANN	1.99 ± 0.14	**4.52 ± 0.54**	**0.789 ± 0.013**
Tier 4 – L-shaped Partial Self-Occlusion				
	WindowGraphNet	**1.78 ± 0.16**	**4.26 ± 0.36**	**0.862 ± 0.011**
	Raw ANN	4.85 ± 0.01	11.21 ± 0.03	−0.050 ± 0.005
	Le-Thanh ANN	7.54 ± 1.28	19.95 ± 2.36	−0.022 ± 0.043
	Dieguez ANN	2.99 ± 0.47	6.36 ± 1.27	0.530 ± 0.247
	Simple Dieguez ANN	6.16 ± 1.12	13.18 ± 2.54	0.606 ± 0.041
Tier 5 – L-shaped Deep Self-Occlusion				
	WindowGraphNet	**2.63 ± 0.54**	5.90 ± 1.25	**0.461 ± 0.039**
	Raw ANN	1.89 ± 0.17	**5.81 ± 0.36**	0.131 ± 0.032
	Le-Thanh ANN	3.08 ± 1.00	8.62 ± 1.30	0.119 ± 0.195
	Dieguez ANN	2.99 ± 0.02	3.89 ± 0.35	0.300 ± 0.127
	Simple Dieguez ANN	3.69 ± 0.70	9.87 ± 2.43	0.437 ± 0.060

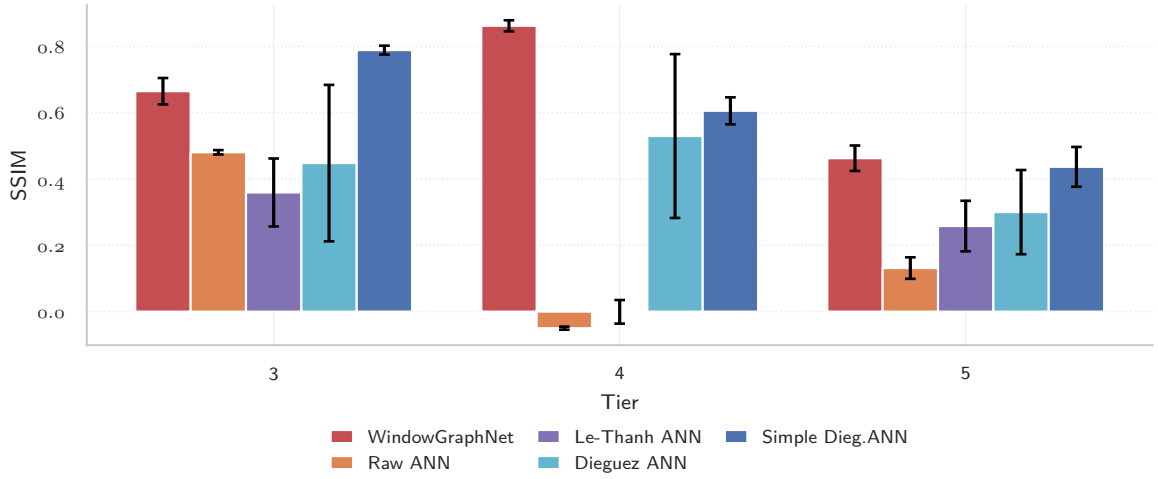


Figure E.3: Structural similarity index measure (SSIM) for *WindowGraphNet* and ANN baselines across the final tiers, assessing the preservation of spatial light-distribution patterns under occlusion and asymmetry.

Table E.3: RMSE performance by window placement index (g) for Tiers 4 and 5. Indices $g = 3, 4$ correspond to Tier 4 (windows on long façades with partial self-occlusion), while $g = 0, 1, 2, 5$ correspond to Tier 5 (windows on short façades with deep self-occlusion). Values are reported as mean ± SD across five training repetitions.

Tier	Window Index (g)	Model	RMSE (mean ± SD)
Tier 4 – Long-façade windows (partial self-occlusion)			
4	3	*Dieguez ANN*	2.99 ± 0.53
4	3	*WindowGraphNet*	1.43 ± 0.12
4	3	*Le-Thanh ANN*	7.06 ± 0.91
4	3	*Raw ANN*	4.64 ± 0.03
4	3	*Simple Dieguez ANN*	6.31 ± 1.06
4	4	*Dieguez ANN*	2.99 ± 0.39
4	4	*WindowGraphNet*	2.16 ± 0.23
4	4	*Le-Thanh ANN*	8.07 ± 1.67
4	4	*Raw ANN*	5.08 ± 0.03
4	4	*Simple Dieguez ANN*	6.01 ± 1.23
Tier 5 – Short-façade windows (deep self-occlusion)			
5	5	*Dieguez ANN*	2.96 ± 0.10
5	5	*WindowGraphNet*	1.76 ± 0.32
5	5	*Le-Thanh ANN*	3.31 ± 1.01
5	5	*Raw ANN*	2.17 ± 0.07
5	5	*Simple Dieguez ANN*	2.70 ± 0.48
5	0	*Dieguez ANN*	3.06 ± 0.10
5	0	*WindowGraphNet*	3.16 ± 0.67
5	0	*Le-Thanh ANN*	2.23 ± 1.12
5	0	*Raw ANN*	1.55 ± 0.02
5	0	*Simple Dieguez ANN*	4.25 ± 1.01
5	1	*Dieguez ANN*	3.01 ± 0.05
5	1	*WindowGraphNet*	3.39 ± 0.90
5	1	*Le-Thanh ANN*	3.36 ± 1.40
5	1	*Raw ANN*	1.96 ± 0.32
5	1	*Simple Dieguez ANN*	5.37 ± 1.04
5	2	*Dieguez ANN*	2.93 ± 0.07
5	2	*WindowGraphNet*	2.33 ± 0.38
5	2	*Le-Thanh ANN*	3.21 ± 0.56
5	2	*Raw ANN*	1.82 ± 0.26
5	2	*Simple Dieguez ANN*	2.65 ± 0.50

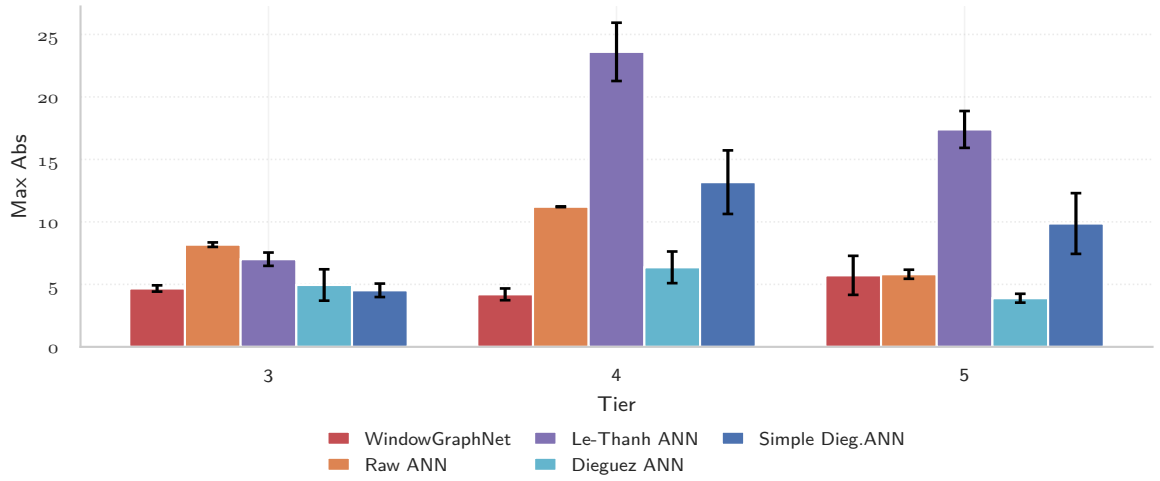


Figure E.4: Maximum absolute error (MaxAbs) of *WindowGraphNet* and ANN baselines for Tiers 3–5, corresponding to offset-window and L-shaped configurations with varying levels of self-occlusion.

E.2. Model Size and Capacity Analysis

A comparison of the total parameter counts reveals substantial differences in model capacity between the four ANN baselines and the heterogeneous GNN. Table E.4 summarises the number of trainable parameters and architectural characteristics of each network.

While all models were trained under comparable optimisation schedules, their parameter budgets differ by two orders of magnitude, which directly affects their ability to fit and generalise across the available training data.

Table E.4: Model sizes and architectural characteristics of the ANN and GNN models.

Model	Architecture Type	Hidden Layers	Parameters
Raw ANN	Fully connected (shallow)	2	1,249
Le-Thanh ANN	Fully connected (medium)	3	5,971
Dieguez ANN	Deep alternating MLP	6	1,281
Simple Dieguez ANN	Fully connected (shallow)	2	1,345
WindowGraphNet (GENConv)	Heterogeneous GNN (shared GEN layers)	3 message-passing steps	301,345

E.2.1. Capacity Distribution

The four ANN models are extremely compact, each containing between one and six hidden layers and fewer than 6,000 trainable parameters. Such small models are computationally efficient and converge rapidly, but their representational capacity is limited. Among them, the *Le-Thanh ANN* is the largest, with roughly five times the parameters of the other MLPs due to its wider intermediate layers.

In contrast, the heterogeneous GNN contains approximately 301k parameters—about two orders of magnitude more than the ANNs. However, the distribution of capacity is highly uneven: more than 99% of its parameters reside within the four relation-specific GENConv modules. The linear input stems and output head together contribute fewer than 1,000 parameters, confirming that the bulk of model complexity lies in the message-passing layers.

E.2.2. Parameter Sharing and Effective Depth

Despite its higher nominal capacity, *WindowGraphNet* maintains efficient weight usage through parameter sharing across propagation steps. The three message-passing iterations reuse the same GENConv weights, expanding the receptive field without increasing the parameter count. If each propagation step were unshared, the total parameter count would rise from approximately 301k to over 900k. This sharing mechanism balances expressive power with regularisation, allowing deeper spatial reasoning at constant model size.

E.2.3. Memory Footprint and Computational Load

All models operate in single-precision (32-bit floating-point) arithmetic. Under this representation, the ANN models require between 5 kB and 25 kB of memory to store their parameters, whereas *Window-GraphNet* occupies roughly 1.2 MB. Although this increase results in higher computational cost during training and inference, it remains lightweight by contemporary deep-learning standards. The added complexity is justified by the GNN's ability to model spatial dependencies that are inaccessible to independent per-sensor architectures.

F

Standardisation of Graphs

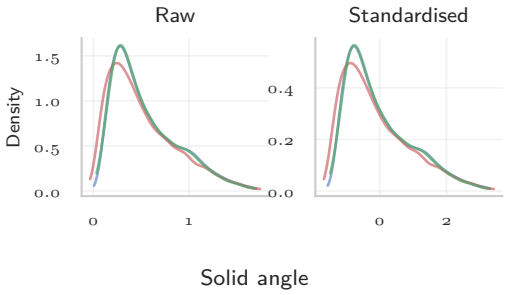
F.1. Feature distributions

To compare how each feature behaves across dataset splits (Train, Val, Ablation), we plot kernel–density estimates (KDEs) for the *raw* (left) and *standardised* (right) versions of every feature. A single shared legend is included elsewhere in the appendix; all figures below only show the distributions.

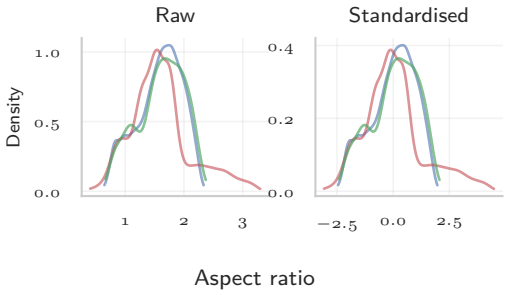
Figure F.1: Shared legend for the feature–distribution plots in this section.

F.2. Homogeneous features

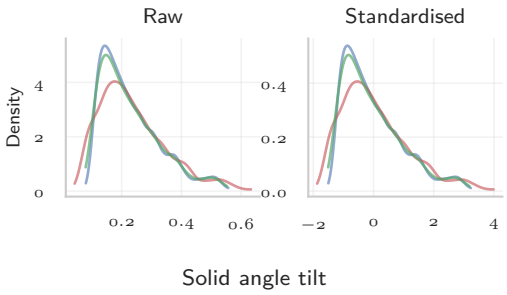
F.2.1. Node features



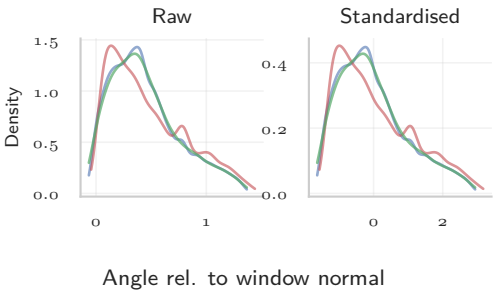
(a) Homogeneous node — Solid angle.



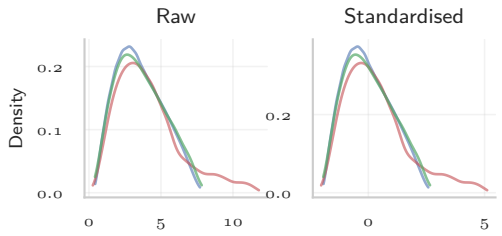
(b) Homogeneous node — Aspect ratio.



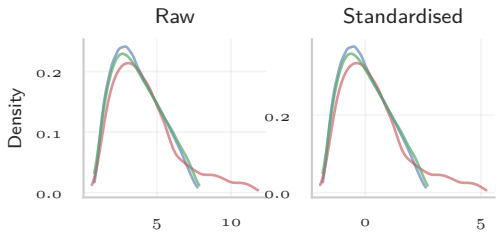
(a) Homogeneous node — Solid angle tilt.



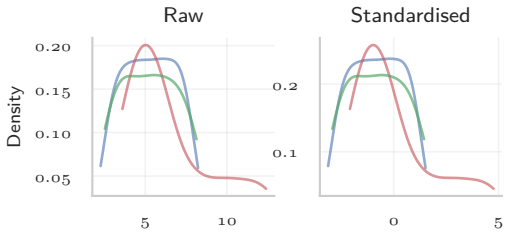
(b) Homogeneous node — Angle rel. to window normal.



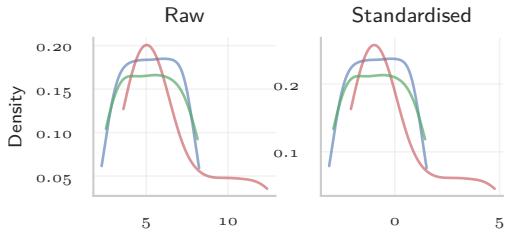
(a) Homogeneous node — Horizontal distance to window.



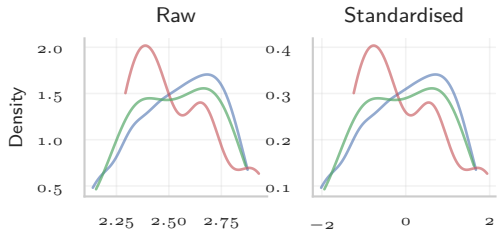
(b) Homogeneous node — Distance to window.



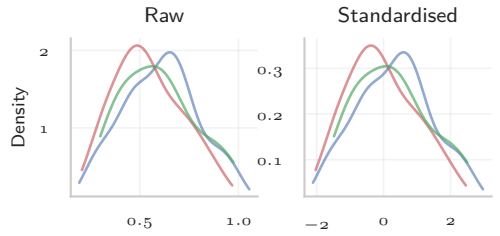
(a) Homogeneous node — Width.



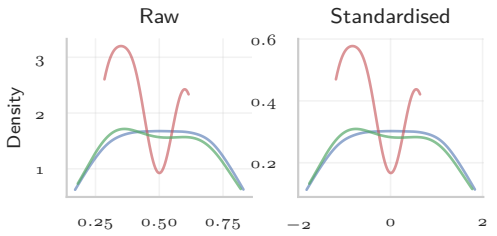
(b) Homogeneous node — Depth.



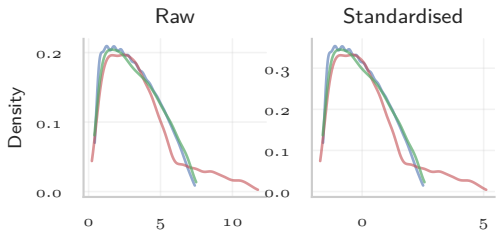
(a) Homogeneous node — Window head.



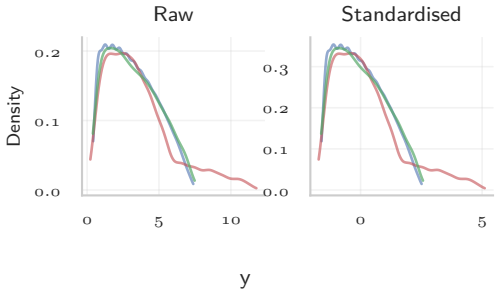
(b) Homogeneous node — Average solid angle.



(a) Homogeneous node — Window-to-wall ratio (WWR).

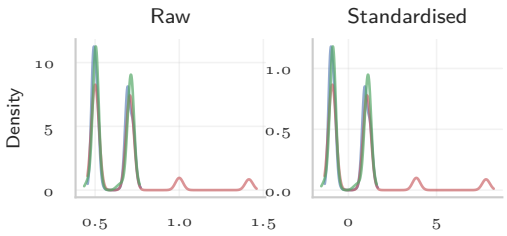


(b) Homogeneous node — x coordinate.

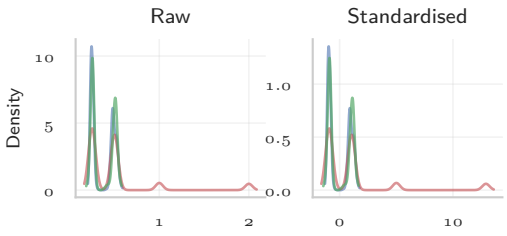


(a) Homogeneous node — y coordinate.

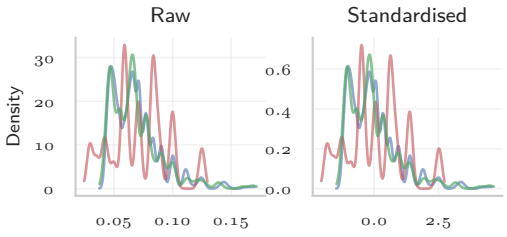
F.2.2. Edge features



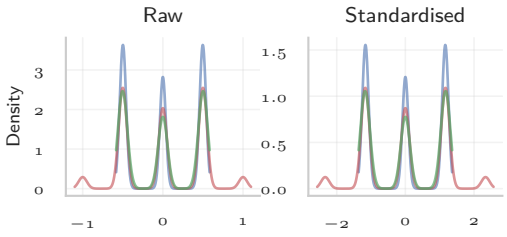
(a) Homogeneous edge — Euclidean distance.



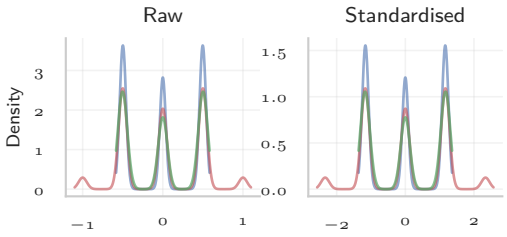
(b) Homogeneous edge — Squared distance.



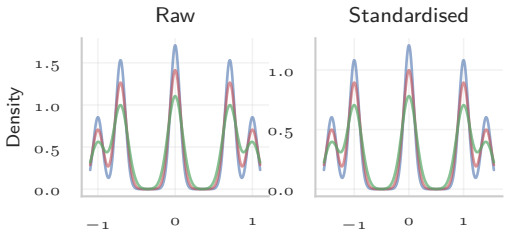
(a) Homogeneous edge — Normalised distance.



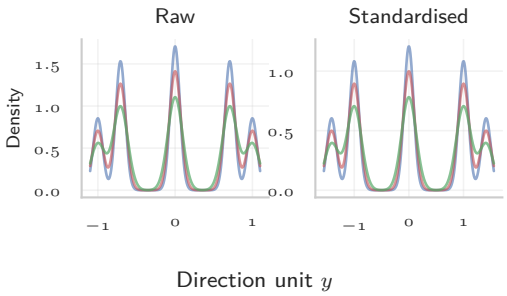
(b) Homogeneous edge — Δ position (x).



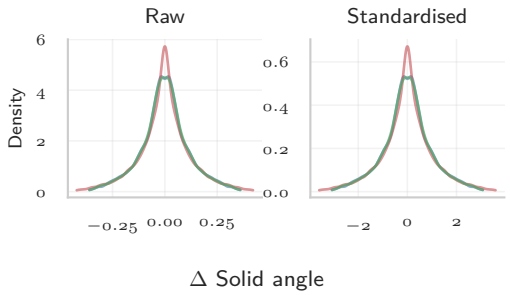
(a) Homogeneous edge — Δ position (y).



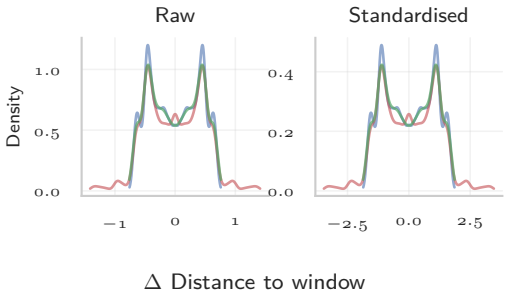
(b) Homogeneous edge — Unit direction (x).



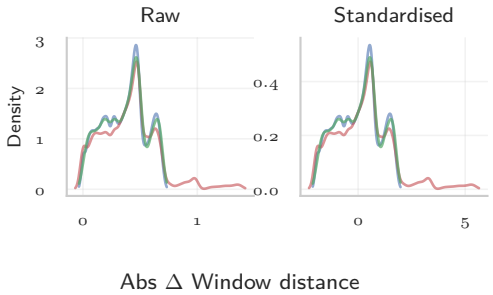
(a) Homogeneous edge — Unit direction (y).



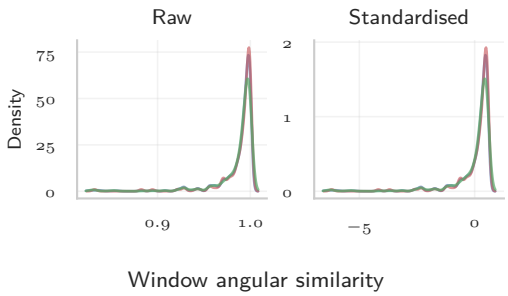
(b) Homogeneous edge — Δ solid angle.



(a) Homogeneous edge — Δ distance to window.

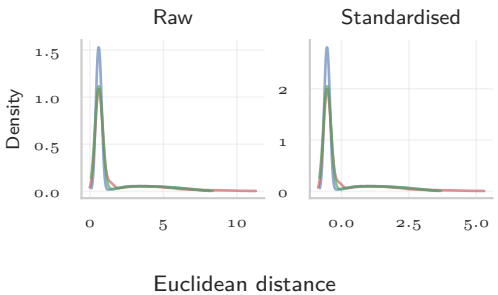


(b) Homogeneous edge — Abs. Δ window distance.

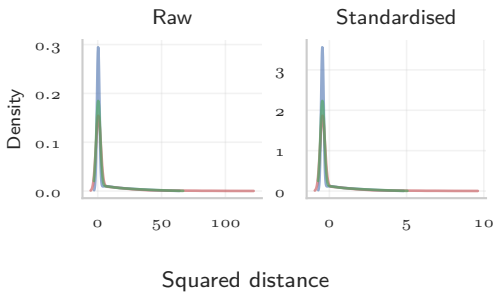


(a) Homogeneous edge — Window angular similarity.

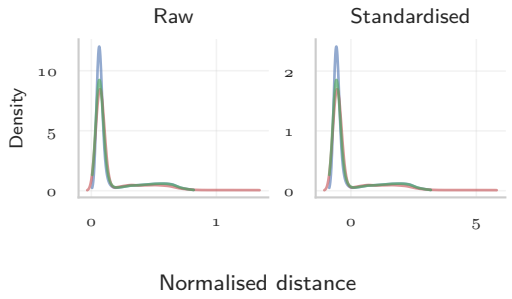
F.3. Heterogeneous features



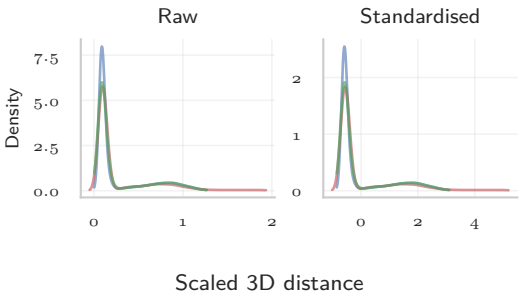
(a) Heterogeneous edge — Euclidean distance.



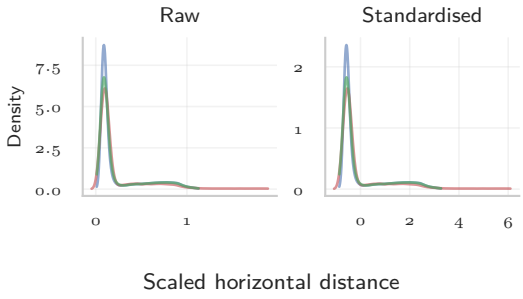
(b) Heterogeneous edge — Squared distance.



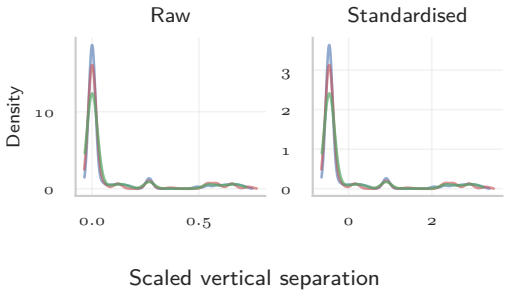
(a) Heterogeneous edge — Normalised distance.



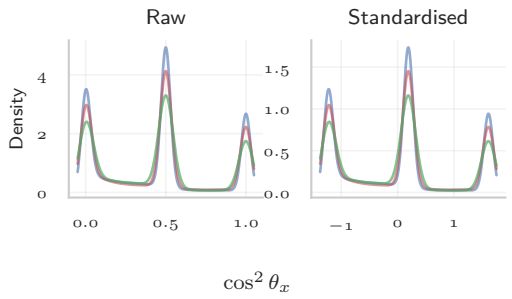
(b) Heterogeneous edge — Scaled 3D distance.



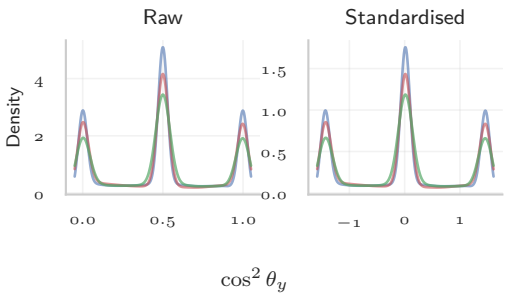
(a) Heterogeneous edge — Scaled horizontal distance.



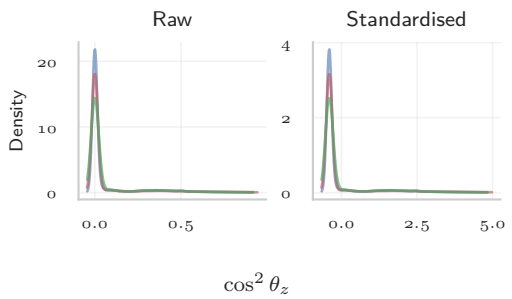
(b) Heterogeneous edge — Scaled vertical separation.



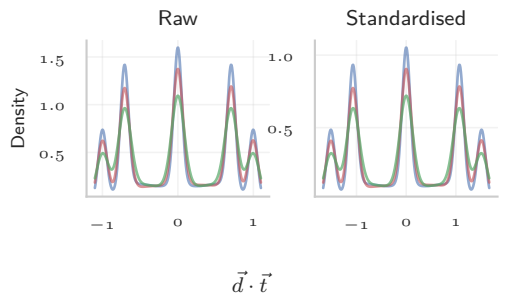
(a) Heterogeneous edge — $\cos^2\theta_x$.



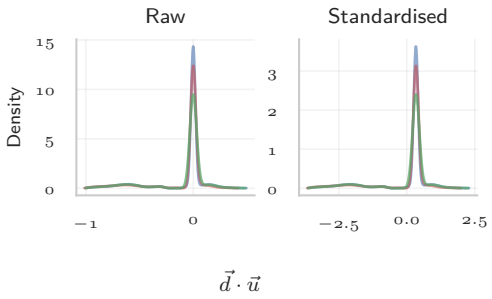
(b) Heterogeneous edge — $\cos^2\theta_y$.



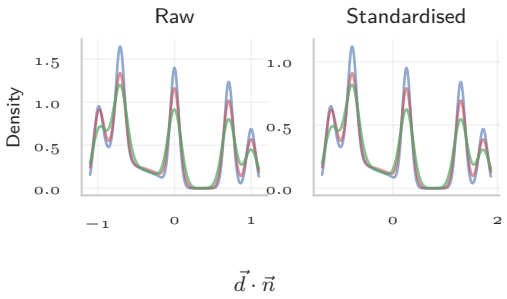
(a) Heterogeneous edge — $\cos^2\theta_z$.



(b) Heterogeneous edge — $\vec{d}\cdot\vec{t}$.



(a) Heterogeneous edge — $\vec{d} \cdot \vec{u}$.



(b) Heterogeneous edge — $\vec{d} \cdot \vec{n}$.