

A Graph Neural Network approach for wake prediction in offshore wind farms using Large Eddy Simulations

Master's thesis

C.W. van den Berg

Delft University of Technology

TU Delft

A Graph Neural Network approach for wake prediction in offshore wind farms using Large Eddy Simulations

Master's thesis

by

C.W. van den Berg

To obtain the degree of Master of Science
at the Delft University of Technology

Student number: 5328063

Thesis committee:		
	Dr. S. J. Pfenninger-Lee	TU Delft, Supervisor
	Dr. ir. A. Ghorbani	TU Delft, Second Supervisor
	Dr. ir. R. A. Verzijlbergh	TU Delft, Advisor

Cover: Wind Turbines by TabNad | Shutterstock

TU Delft

Acknowledgements

First of all, I would like to thank Dr. Remco Verzijlbergh for his guidance throughout this thesis. He helped me select this topic, which initially concerned a field I knew relatively little about. During our biweekly meetings, he encouraged me not to move too quickly towards results, but to first make sure that I properly understood the underlying concepts and methods. His feedback also helped me stay on track throughout the project and was particularly valuable when interpreting some of the figures and results produced by the models. I greatly appreciated his guidance and the freedom he gave me to explore different directions throughout the research.

I would also like to thank Dr. Stefan Pfenninger-Lee and Dr. Amineh Ghorbani for being part of my thesis committee. I completed the formation of my committee relatively late in the process, and I am therefore especially grateful that they were willing to make the time to support and evaluate my thesis.

Finally, I would like to thank everyone who supported me throughout the process of completing this thesis.

Executive Summary

The large-scale development of offshore wind is a complex socio-technical challenge in which technological performance, economic viability, government policy, energy markets, and the decisions of multiple actors are strongly interconnected. Governments use policy and tender procedures to shape offshore wind development, while developers and investors must make long-term design, bidding, and investment decisions under considerable uncertainty. These decisions depend on expectations of future energy production and project revenues. Technical modelling uncertainty can therefore propagate beyond engineering calculations into financial assessments, tender strategies, and ultimately the viability and deployment of offshore wind projects. Improving the information available to decision-makers is consequently important for both individual projects and the wider transition towards a low-carbon electricity system.

Wake prediction represents one important source of uncertainty within this decision-making system. Wind turbines reduce downstream wind speeds, affecting the production of neighbouring turbines and therefore the expected energy yield of a wind farm. Engineering wake models such as Jensen are computationally efficient but rely on simplifying assumptions, while high-fidelity Large Eddy Simulations (LES) provide substantially more detailed representations of wake behaviour at much greater computational cost. This creates a trade-off between the accuracy of information available for decision-making and the computational effort required to generate it, particularly when many layouts or scenarios must be evaluated.

This thesis investigates whether recent advances in artificial intelligence can reduce this trade-off by using a Graph Neural Network (GNN) trained directly on high-fidelity LES data. The central research question is: *To what extent can a Graph Neural Network, trained on LES data, improve the accuracy of offshore wind-farm wake predictions while maintaining computational efficiency compared to traditional engineering wake models?*

A turbine-interaction GNN was developed in which turbines are represented as nodes and potential wake interactions as directed edges. Through message passing, the model learns how upstream turbines influence downstream wind-speed deficits. The architecture was first verified using synthetic Jensen data and subsequently evaluated using the WINS50 LES dataset. The experiments investigated heterogeneous inflow information, turbulence intensity and atmospheric-stability indicators, graph construction, and alternative message-passing architectures. Finally, a multi-farm GNN was trained on 25 wind farms and evaluated on the unseen *future_052* layout, with Jensen/NOJ as the principal engineering-model baseline.

The experiments demonstrate that the GNN can successfully learn turbine–turbine wake relationships. The LES experiments further showed that additional model information and complexity do not automatically improve prediction accuracy. Local wind speed provided useful information about heterogeneous conditions, while local wind direction offered limited additional benefit. Turbulence intensity generally improved performance, and exploratory experiments indicated that atmospheric-stability information may provide further predictive value. Graph construction also revealed a trade-off between accuracy and computational cost: wider interaction corridors improved predictions, whereas full connectivity provided only a small additional improvement at substantially greater computational cost. Similarly, an attention-based architecture did not outperform the edge-conditioned NNConv model. These findings indicate that a relatively simple, physically informed graph representation can efficiently capture relevant wake interactions.

The strongest result was obtained in the final cross-farm experiment. When trained on 25 wind farms and evaluated on the excluded *future_052* layout, the GNN achieved a mean absolute error of 0.7888 m/s compared with 1.0859 m/s for Jensen, representing an approximately 27% reduction in prediction error. The GNN also exhibited virtually no mean prediction bias. At the same time, including graph construction, evaluation of the complete test set was approximately 4.5 times faster than the Jensen implementation. The results therefore demonstrate that wake relationships learned from computationally expensive LES simulations can be transferred to an unseen wind-farm layout while retaining computationally efficient evaluation.

Importantly, these improvements can affect the information on which offshore wind decisions are based. An illustrative valuation case translated predicted wind speeds into annual energy production (AEP) and discounted lifetime revenues. For the considered wind farm, the LES reference resulted in an AEP of approximately 5318 GWh. The GNN overestimated AEP by only approximately 0.6%, corresponding to a discounted lifetime revenue error of approximately EUR 22.8 million under the case-study assumptions. The Jensen model produced a substantially larger energy-production and revenue error. Although this does not constitute a complete financial valuation, it demonstrates how technical modelling errors can propagate into economically significant differences in project assessment.

The contribution of this thesis therefore extends beyond improving a technical wake-prediction metric. More accurate and computationally efficient wake predictions can improve one important source of information within a wider system of interconnected technical, economic, and policy decisions. For developers and investors, this can support more reliable comparisons of layouts, expected revenues, investment alternatives, and tender strategies, while efficient evaluation enables a larger number of scenarios to be considered under uncertainty. For governments, better-informed project assessments can contribute to a more robust basis for the large-scale development of offshore wind capacity.

Within the scope of this thesis, LES-trained turbine-interaction GNNs provide a promising bridge between high-fidelity physical simulation and the computational efficiency required for repeated wind-farm assessment. The demonstrated combination of improved predictive accuracy, efficient evaluation, and transfer to an unseen wind-farm layout shows the potential to make high-fidelity wake information more accessible for practical decision-making. With further validation, such models could provide a more reliable technical basis for wind-farm design, project valuation, investment assessment, and tender decisions, ultimately supporting more robust offshore wind development within the wider energy transition.

Contents

Executive Summary ii

1 Introduction and research framework 1
- 1.1 Introduction . . . 1
- 1.2 Literature selection and synthesis . . . 2
  - 1.2.1 Search strategy . . . 2
  - 1.2.2 Thematic synthesis . . . 3
    - 1.2.2.1 Explanation of core tables . . . 5
- 1.3 Comparative analysis of the literature . . . 5
  - 1.3.1 Overview of key themes . . . 5
  - 1.3.2 Engineering wake models as the baseline for wind-farm prediction . . . 5
  - 1.3.3 Data-driven wake modelling enabled by high-fidelity LES data . . . 6
  - 1.3.4 Graph-based modelling of wind farms . . . 6
  - 1.3.5 Hybrid and physics-informed approaches . . . 7
  - 1.3.6 Summary of insights . . . 7
  - 1.3.7 Research gaps . . . 7
- 1.4 Research design and methodology . . . 7
  - 1.4.1 Sub-questions, methods, data requirements and analysis tools . . . 8
    - 1.4.1.1 Overview table . . . 9

2 Graph neural network theory 10
- 2.1 Introduction . . . 10
- 2.2 Toy model for turbine interaction learning . . . 10
- 2.3 Toy model extension to multiple turbines . . . 11
  - 2.3.1 Edge representation . . . 11
  - 2.3.2 Edge interaction function . . . 12
  - 2.3.3 Message aggregation . . . 12
  - 2.3.4 Activation function choice . . . 13
- 2.4 General message passing formulation . . . 13

3 Model implementation 15
- 3.1 Introduction . . . 15
- 3.2 Data preparation . . . 15
- 3.3 Graph construction . . . 16
  - 3.3.1 Node features . . . 16
  - 3.3.2 Edge construction . . . 17
  - 3.3.3 Edge features . . . 17
  - 3.3.4 Temporal graph structure . . . 17
  - 3.3.5 Target variable . . . 17
  - 3.3.6 Design choices . . . 17
- 3.4 Model architecture . . . 17
  - 3.4.1 Edge-conditioned message passing . . . 18
  - 3.4.2 NNConv layers . . . 18
  - 3.4.3 Prediction head . . . 18
  - 3.4.4 Model depth and interpretation . . . 19
  - 3.4.5 Model configuration . . . 19
- 3.5 Training setup . . . 19
  - 3.5.1 Data split . . . 19
  - 3.5.2 Normalisation . . . 19
  - 3.5.3 Loss function . . . 20
  - 3.5.4 Optimisation . . . 20
  - 3.5.5 Early stopping . . . 20

3.5.6 Evaluation metric . . . 20

4 Results: Model verification and validation **22**
4.1 Verification . . . 22
4.1.1 Synthetic benchmark setup . . . 22
4.1.2 Training and evaluation setup . . . 23
4.1.3 Verification results . . . 23
4.1.3.1 Scaling with wind farm size . . . 24
4.1.3.2 Computational performance . . . 24
4.1.4 Generalisation and robustness . . . 25
4.1.4.1 Wind-speed extrapolation . . . 26
4.1.4.2 Generalisation to unseen turbine spacing . . . 27
4.1.4.3 Generalisation to staggered wind farm layouts . . . 28
4.1.5 Error analysis . . . 28
4.2 Model validation . . . 29
4.2.1 Experimental setup . . . 29
4.2.2 Results: mean inflow . . . 30
4.2.3 Results: maximum inflow . . . 31
4.2.4 Results: local inflow . . . 32
4.3 Conclusion . . . 33

5 Results: GNN model refinement **34**
5.1 Heterogeneous LES inflow conditions . . . 34
5.1.1 LES-based experimental setup . . . 34
5.1.2 Results: influence of heterogeneous inflow representations . . . 35
5.1.2.1 Performance across inflow wind-speed regimes . . . 35
5.1.2.2 Spatial distribution of local inflow benefits . . . 36
5.2 Influence of turbulence intensity on wake prediction . . . 38
5.2.1 Experimental setup . . . 38
5.2.2 Aggregate performance impact of turbulence intensity . . . 39
5.2.3 Performance across turbulence-intensity regimes . . . 40
5.2.4 Performance across wind-speed regimes . . . 42
5.2.5 Spatial distribution of turbulence-intensity benefits . . . 42
5.2.5.1 Wake exposure and blockage hypothesis . . . 44
5.2.6 Alternative atmospheric stability measures . . . 44
5.3 Influence of graph construction choices and attention mechanisms . . . 45
5.3.1 Experimental setup . . . 45
5.3.2 Sensitivity to graph construction choices . . . 46
5.3.3 Attention-based message passing . . . 47
5.3.4 Computational costs . . . 48
5.4 Conclusion . . . 49

6 Results: Cross-farm validation and financial case study **50**
6.1 Introduction . . . 50
6.2 Cross-farm generalisation experiment . . . 50
6.3 Generalisation to an unseen wind-farm layout . . . 51
6.3.1 Computational performance . . . 51
6.3.2 Spatial error distribution . . . 52
6.4 Illustrative valuation case study . . . 52
6.4.1 Methodology . . . 53
6.4.2 Results . . . 54
6.4.3 Conclusion . . . 54

7 Conclusion and discussion **55**
7.1 Key findings and interpretation . . . 55
7.2 Scientific contribution and implications . . . 56
7.3 Practical implications . . . 57
7.4 Limitations . . . 58
7.5 Future research . . . 58

A Literature review overview **63**

B Verification run overview **64**

C Table Chapter 5 **65**

D Overview of Selected Wind Farms and Turbine Counts **67**

E The use of AI tools **68**

List of Figures

1.1 PRISMA Flow Diagram of the selection process 3

2.1 Illustration of the message passing mechanism in a wind farm graph. Upstream turbines T_1 and T_2 send messages to a downstream turbine T_3. Incoming messages are aggregated to form M_3, which is used by the node update function to update the turbine state. 14

4.1 Test MAE as a function of training set size for different hidden dimensions, shown separately for the two target formulations. Performance improves rapidly with increasing dataset size up to approximately 10^4 samples, after which gains diminish. The influence of model capacity is comparatively limited. 23
4.2 Runtime as a function of dataset size for the ΔU target formulation with hidden dimension 64. The total runtime is dominated by the training phase. 25
4.3 Predicted turbine-level wind speeds compared to Jensen reference values for the in-distribution and out-of-distribution test regimes. The dashed line indicates perfect agreement. 26
4.4 Training and test layouts used for the turbine-spacing generalisation experiment. The model is trained on five regular spacing configurations and evaluated on unseen interpolation, dense extrapolation, and sparse extrapolation cases. 27
4.5 Absolute prediction error as a function of downstream wake distance for the synthetic 4×4 wind farm. The solid line shows the median absolute error, the shaded region indicates the interquartile range (25–75 percentile), and the bars show the number of wake-interaction samples per downstream-distance bin. 29
4.6 Turbine-level wind speed predictions for a representative timestep using mean inflow. Both the GNN and the Jensen model reproduce similar large-scale wake patterns across the wind farm. 31
4.7 Absolute prediction error per turbine for the mean inflow case. Errors are relatively small and evenly distributed, with no strong systematic bias for either model. 31
4.8 Turbine-level wind speed predictions for a representative timestep using maximum inflow. The Jensen model exhibits a clear overestimation compared to the LES reference, while the GNN remains closer to the reference. 32
4.9 Absolute prediction error per turbine for the maximum inflow case. The Jensen model shows significantly larger and more structured errors, whereas the GNN remains relatively robust. 32

5.1 Mean test MAE as a function of global inflow wind-speed bin for the four inflow representations. The upper panel shows the mean test MAE, while the lower panel shows the number of test timesteps in each bin. 36
5.2 Distribution of global inflow wind directions across the training, validation, and test sets. The chronological dataset split leads to non-identical directional distributions between subsets, which influences the wake interaction patterns encountered during evaluation. . 37
5.3 Spatial difference in turbine-level test MAE between the local-local and global-global inflow representations. Negative values indicate improved performance when local inflow information is included. Turbine positions are shown relative to the wind-farm extent and normalised by rotor diameter. 37
5.4 Test MAE across all inflow and turbulence-intensity configurations. Error bars indicate standard deviation over three random seeds. 40
5.5 Prediction error as a function of turbulence-intensity regime for the local-global inflow representation. Error bars indicate standard deviation across the three random seeds. . . 41
5.6 Prediction error as a function of inflow wind-speed regime for the local-global inflow representation. Error bars indicate standard deviation across three random seeds. 42

5.7 Spatial distribution of turbine-level MAE improvement for the local-global inflow representation when turbulence intensity is included. Positive values indicate reduced prediction error relative to the baseline without turbulence intensity. Results are averaged over 1314 unseen test timesteps and three random seeds. 43
5.8 Distribution of global wind directions in the training, validation, and test sets. The test-set distribution is used to support interpretation of the spatial distribution of turbulence-intensity-related error reductions. 43
5.9 Average turbine-level MAE improvement obtained by adding global turbulence intensity to the local-global inflow configuration, grouped by wake exposure. Wake exposure is defined as the average number of upstream turbines within the graph wake corridor over all test timesteps. Positive values indicate reduced prediction error relative to the No-TI baseline. 44
5.10 Influence of graph construction choices on predictive accuracy. Error bars indicate variation across three random seeds. The fully connected configuration represents a single exploratory run. 46
5.11 Comparison of predictive performance between NNConv and GAT across graph construction choices. Error bars indicate variation across three random seeds. Hatched bars represent exploratory single-seed fully connected runs. 47
5.12 Average training runtime for NNConv and GAT across graph construction configurations. Fully connected results represent single exploratory runs. 48

6.1 Spatial distribution of turbine-level MAE for the multi-farm GNN and Jensen wake model on the unseen `future_052` wind farm. The wind-direction distribution for the full test period is shown on the right. 52

List of Tables

1.1 Core methodological comparison of wake modelling literature . . . 4
1.2 Overview of sub-questions, data requirements, sources, and analysis tools. . . . 9

3.1 Model configuration. . . . 19
3.2 Training configuration. . . . 21

4.1 Scaling behaviour for increasingly large synthetic wind farm layouts using the ΔU target formulation and hidden dimension 64. . . . 24
4.2 Average performance for different hidden dimensions using the ΔU target formulation, computed over the 10^4 and 10^5 sample experiments. . . . 25
4.3 In-distribution and out-of-distribution test MAE for the ΔU target formulation. . . . 26
4.4 Prediction accuracy for unseen turbine-spacing configurations. . . . 27
4.5 Prediction accuracy for unseen staggered wind farm layouts. . . . 28
4.6 Performance comparison using mean inflow wind speed. . . . 30
4.7 Performance comparison using maximum inflow wind speed. . . . 31
4.8 Performance of the extended GNN model with local inflow information, comparing two target formulations. . . . 33

5.1 Overview of the considered inflow representations. . . . 35
5.2 Predictive performance for different LES-based inflow representations compared with the Jensen baseline. . . . 35
5.3 Overview of the evaluated inflow and turbulence-intensity configurations. . . . 38
5.4 Aggregate predictive performance across inflow and turbulence-intensity configurations (mean ± standard deviation over three seeds). . . . 39
5.5 Distribution of mean local 2020 turbulence-intensity bins across the training and test sets. 41
5.6 Predictive performance for different atmospheric-stability representations compared with the Jensen baseline. . . . 45
5.7 Computational cost comparison of representative model configurations. . . . 48

6.1 Performance comparison on the unseen `future_052` wind farm. . . . 51
6.2 Computational runtime comparison for the future_052 test set (8760 hourly timesteps and 68 turbines). Runtime statistics are based on 100 repeated runs. . . . 51
6.3 Assumptions used in the illustrative valuation case study. . . . 54
6.4 Case-study results for `future_052` using the IEA 15 MW reference turbine power curve. . 54

A.1 Full table of reviewed wake-modelling literature . . . 63

B.1 Overview of all verification runs for different dataset sizes, hidden dimensions, and target formulations. Runtime components are reported separately for Jensen generation, graph construction, and training. . . . 64

C.1 Per-seed performance results for all inflow and turbulence-intensity configurations. . . . 65

D.1 Wind farms included in the Chapter 6 cross-farm generalisation experiment. . . . 67

List of Abbreviations and Symbols

Abbreviations

AEP	Annual Energy Production
AI	Artificial Intelligence
ANN	Artificial Neural Network
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
GNN	Graph Neural Network
LES	Large Eddy Simulation
MAE	Mean Absolute Error
ML	Machine Learning
MSE	Mean Squared Error
NOJ	N.O. Jensen wake model
RANS	Reynolds-Averaged Navier–Stokes
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
SCADA	Supervisory Control and Data Acquisition
TI	Turbulence Intensity

Symbols

D	Rotor diameter / reference rotor diameter
N	Number of turbine nodes
T	Number of timesteps
U	Wind speed
U_∞	Free-stream wind speed
U_i^{2020}	Turbine-level wind speed in the 2020 reference scenario
U_i^{2050}	Turbine-level wind speed in the 2050 scenario
U_i^{Jensen}	Turbine-level wind speed predicted by the Jensen model
U_{global}	Global inflow wind speed
ΔU_i	Wake-induced wind-speed difference at turbine i
$\Delta \hat{U}_i$	Predicted wake-induced wind-speed difference at turbine i
θ	Wind direction
Δx_{ij}	Difference in x-coordinate between turbines i and j
Δy_{ij}	Difference in y-coordinate between turbines i and j
x_{ij}^{down}	Downwind distance between turbines i and j
x_{ij}^{cross}	Crosswind distance between turbines i and j
x_{min}	Minimum downstream interaction distance
$x_{\text{max}}^{\text{down}}$	Maximum downstream interaction distance
$x_{\text{max}}^{\text{cross}}$	Maximum crosswind interaction distance
$\mathcal{G}$	Wind-farm graph
$\mathcal{V}$	Set of graph nodes (turbines)
$\mathcal{E}$	Set of directed graph edges
$\mathcal{N}_j$	Set of neighbouring turbines connected to turbine j
$\mathbf{h}_i^{(k)}$	Hidden/node state of turbine i at layer k
$\mathbf{e}_{ij}$	Edge-feature vector between turbines i and j
$\mathbf{m}_{ij}^{(k)}$	Message from turbine i to turbine j at layer k
$\mathbf{M}_j^{(k)}$	Aggregated incoming message at turbine j
$\Psi^{(k)}$	Learnable edge network at layer k
ϕ	Node-update function
d	Hidden dimension
$\mathbf{W}$	Learnable weight matrix
$\mathbf{b}$	Learnable bias vector
$\mathcal{L}$	Loss function
η	Learning rate
$\boldsymbol{\mu}_x$	Mean of the training node features
$\boldsymbol{\sigma}_x$	Standard deviation of the training node features
μ_y	Mean of the training targets
σ_y	Standard deviation of the training targets
$\tilde{\mathbf{h}}_i$	Normalised node-feature vector
$\widetilde{\Delta U_i}$	Normalised wake-induced wind-speed difference
$P(U_{i,t})$	Turbine power at wind speed $U_{i,t}$
Δt	Duration of one timestep
P_{el}	Electricity price
PV	Present value of lifetime revenues
r	Discount rate
n	Project lifetime in years

1

Introduction and research framework

1.1. Introduction

As the global energy transition accelerates, offshore wind energy is expected to play a key role in achieving large-scale decarbonization of electricity systems. In particular, the North Sea is projected to host a rapidly increasing number of offshore wind farms in the coming decades (Taylor Wessing, 2024). While this expansion is essential for meeting climate targets, it also involves large financial investments whose economic viability strongly depends on accurate long-term energy yield predictions (IEA, 2019). Recent offshore wind tenders in the German North Sea have illustrated the uncertainty surrounding such investments, where aggressive bidding strategies and changing market conditions raised concerns regarding long-term project profitability (Clean Energy Wire, 2025; Reuters, 2023).

Offshore wind development is therefore not only a technical challenge, but also a complex decision-making problem involving interactions between governments, developers, investors, and energy markets (International Energy Agency, 2025). Governments use offshore wind auctions and tender procedures to allocate development rights and pursue renewable-energy targets, while developers must make long-term strategic decisions under substantial uncertainty regarding future market conditions and project performance (International Energy Agency, 2025; Reuters, 2023). The outcomes of these interconnected decisions influence both the financial viability of individual projects and the pace and scale at which offshore wind capacity can be deployed as part of the energy transition (Clean Energy Wire, 2025; International Energy Agency, 2025).

One important source of technical uncertainty in offshore wind development is the prediction of wake effects within and between wind farms (Bentsen et al., 2022). Wind turbine wakes reduce wind speeds and increase turbulence downstream, resulting in power losses for affected turbines (Ainslie, 1988; Jensen, 1983). As offshore wind farms continue to increase in size and density, accurate wake modelling becomes increasingly important for estimating energy production (Taylor Wessing, 2024). These predictions are used throughout the offshore wind development process to support decisions on wind-farm design, project valuation, and investment planning (IEA, 2019; Reuters, 2023). Small modelling inaccuracies can lead to substantial deviations in predicted annual energy production, potentially affecting the financial viability of large offshore wind projects (IEA, 2019; Reuters, 2023).

Engineering wake models are commonly used in industry due to their low computational cost and simplicity (Bastankhah & Porté-Agel, 2014; Jensen, 1983). However, these models rely on strong simplifying assumptions and often struggle to accurately represent wake interactions in large offshore wind farm clusters (Krutova et al., 2020; Levick et al., 2022). In contrast, high-fidelity simulations such as Large Eddy Simulations (LES) provide detailed and physically realistic representations of wake dynamics, but their high computational cost limits their practical applicability (Kochkov et al., 2021; Santoni et al., 2024).

Recent advances in artificial intelligence, particularly in deep learning for scientific and engineering applications, offer a potential alternative modelling approach (Ye et al., 2025; Zhou et al., 2020). This follows a broader trend in AI, where neural-network surrogates are increasingly used to approximate computationally expensive physical simulations while maintaining much of their predictive accuracy

(Kochkov et al., 2021; Z. Li et al., 2021). Graph Neural Networks (GNNs), in particular, are well suited for modelling wind farms due to their ability to represent relational interactions between turbines (Gilmer et al., 2017; S. Li et al., 2023). Large-scale datasets such as the WINS50 LES dataset enable the training and evaluation of such data-driven wake models under realistic offshore conditions (Whiffle, 2024).

The aim of this thesis is to investigate whether a Graph Neural Network trained on WINS50 LES data can serve as a viable alternative for predicting wake-induced wind speed deficits in offshore wind farms compared to established engineering wake models. The guiding research question is:

"To what extent can a Graph Neural Network, trained on LES data, improve the accuracy of offshore wind-farm wake predictions while maintaining computational efficiency compared to traditional engineering wake models?"

This thesis is structured as follows. First, the theoretical background on wind turbine wakes, engineering wake models, and graph-based machine learning is presented. Next, the dataset and methodology are described, followed by a quantitative comparison between the GNN and traditional wake models. Finally, the results are discussed and conclusions are drawn.

1.2. Literature selection and synthesis

This literature review adopts a structured approach to identify and analyse academic sources relevant to offshore wind-farm wake modelling and machine-learning-based prediction. The selection and screening of studies follow the PRISMA framework to ensure transparency and consistency.

1.2.1. Search strategy

The Scopus database was used to identify peer-reviewed publications published between 2018 and 2026. A structured search query was applied to titles, abstracts, and keywords to capture studies on wind-farm wake modelling, including engineering wake models, data-driven surrogate models, and machine-learning-based approaches. Wildcard operators (*) were used in the search string to include multiple word variants (e.g., model, models, modelling). The complete search query is reported in Figure 1.1.

In addition to the database search, relevant studies were identified through backward and forward snowballing of reference lists, manual screening of key journals and conference proceedings, and the inclusion of influential papers foundational to the field. All records, regardless of source, were subjected to the same screening and eligibility criteria.

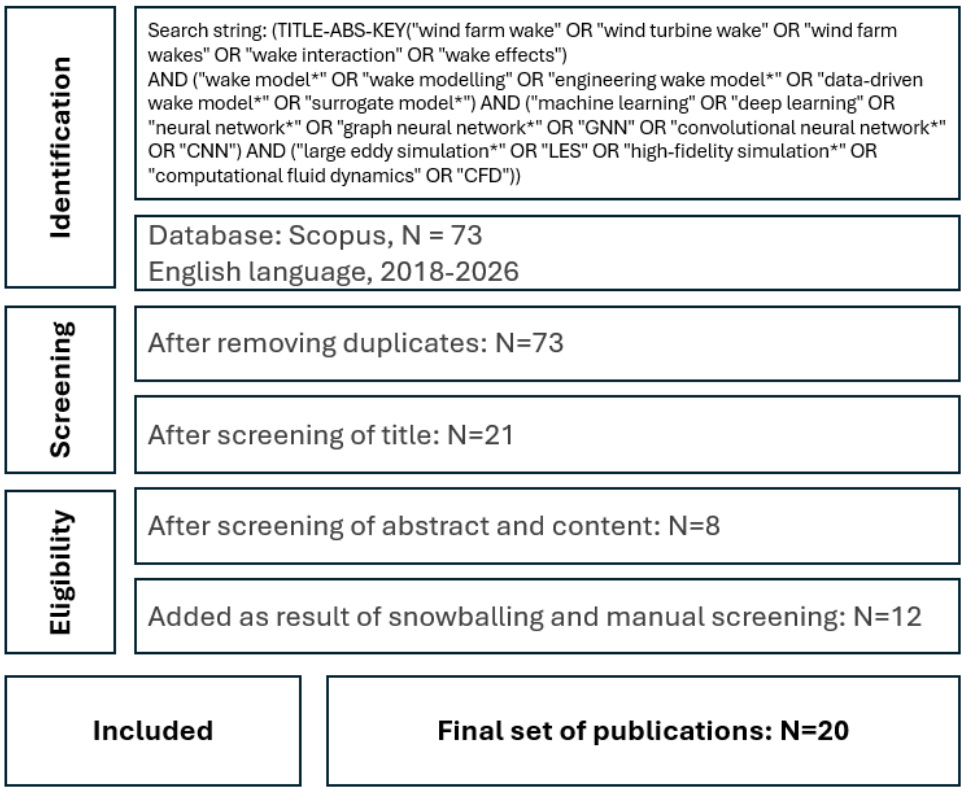


Figure 1.1: PRISMA Flow Diagram of the selection process

The database search yielded 73 records, which were screened based on title and abstract, resulting in 8 studies meeting the inclusion criteria. An additional 12 relevant publications were identified through snowballing and targeted searches. Together, this resulted in a final set of 20 peer-reviewed studies included in the literature review. A complete list of the included studies is provided in Appendix A.

1.2.2. Thematic synthesis

A thematic synthesis was conducted to compare the core methodological characteristics of the selected literature on wind-farm wake modelling and surrogate prediction. The comparison focuses strictly on four aspects that are central to the research question: (i) model type, (ii) whether machine learning is used, (iii) whether LES data are used for training, and (iv) the main computational claims made by the authors. Table 1.1 summarises these aspects.

Table 1.1: Core methodological comparison of wake modelling literature

Authors	Model type	Domain representation	Uses LES (training)?	Computational claim
Jensen (1983)	Top-hat engineering wake model	Interaction-based	No	Extremely fast, low-fidelity baseline
Ainslie (1988)	Eddy-viscosity wake model	Field-based	No	Fast, improved physical realism
Bastankhah & Porté-Agel (2014)	Gaussian engineering wake model	Field-based	No	Engineering standard, low computational cost
Larsen et al. (2008)	Dynamic wake meandering model	Interaction-based	No	Improved far-wake realism at low cost
Krutova et al. (2020)	Gaussian wake models (LES evaluation)	Field-based	No	Very fast, accuracy limited by model assumptions
Ye et al. (2025)	Review of data-driven wake models	N/A	No	Not applicable (survey paper)
Ti et al. (2021)	Feed-forward ANN	Field-based	No (RANS)	Seconds-level wake and power prediction
Rui Li et al. (2022)	Bi-CNN	Field-based	Yes	Real-time wake-field prediction
Bentsen et al. (2022)	Attention-based GNN	Interaction-based	No (FLORIS)	Real-time farm-level prediction
Li et al. (2023)	Mesh-based GNN	Field-based	No (RANS)	End-to-end flow surrogate, faster than CFD
Santoni et al. (2024)	CNN (engineering–LES hybrid)	Hybrid	Yes	Engineering-level cost with LES-like mean-field accuracy
Zhang et al. (2025)	CNN (multi-fidelity)	Field-based	Yes	Orders faster than LES wake prediction
Wang et al. (2025)	DWM	Field-based	Yes	Fast and physically interpretable surrogate
Duthé et al. (2023)	GNN	Interaction-based	No (PyWake)	Real-time capable, engineering-informed
Hou et al. (2026)	Two-stage hybrid GNN	Interaction-based	No (PyWake)	Real-time surrogate for wake-affected farms
Liu et al. (2024)	Physics-guided 3D GNN	Interaction-based	No (SCADA)	Improved accuracy with physics-guided, low-cost inference
Schøler et al. (2025)	Graph Neural Operator	Interaction-based	No	Efficient turbine–wake interaction inference
Zongyi Li et al. (2021)	Fourier Neural Operator	Field-based	No (DNS)	Orders-of-magnitude speedup vs solvers
Kochkov et al. (2021)	ML-accelerated CFD	Field-based	Yes	40–80× speedup vs DNS/LES
Bleeg (2020)	GNN (CFD.ML)	Interaction-based	No (RANS)	CFD-consistent turbine interaction prediction with engineering-level runtime

1.2.2.1. Explanation of core tables

Table 1.1 summarises the wake-modelling literature most relevant to this thesis. The studies can be broadly divided into three groups. The first group consists of the classical wake-engineering models of Jensen, Ainslie, Bastankhah and Larsen, which form the methodological foundation of modern wake modelling and provide important reference points throughout this thesis. The second group contains review and comparative studies that synthesise existing wake-modelling approaches and help position recent developments within the broader literature. The third group comprises machine-learning-based wake-surrogate models, including CNNs, GNNs and neural operators, which are most closely related to the approach proposed in this work. Together, these studies provide the context needed to evaluate whether an LES-trained graph neural network can combine the physical fidelity of high-fidelity wake simulations with the computational efficiency of interaction-based surrogate models.

The **model type** categorises each approach according to its underlying methodology, ranging from classical engineering wake models to data-driven architectures such as convolutional neural networks, graph neural networks, and neural operators. The **domain representation** distinguishes how wake effects are represented within the model. Field-based approaches learn wake behaviour directly from spatial flow fields, typically using CNNs. Interaction-based approaches model turbines as nodes connected through wake interactions, typically using GNNs. Hybrid approaches combine both representations. The column **Uses LES (training)?** identifies whether LES data are employed directly for model training, thereby separating high-fidelity learning approaches from methods trained on Reynolds-Averaged Navier-Stokes (RANS) simulations, engineering wake models, or operational data. While LES explicitly resolves the largest turbulent flow structures, RANS predicts only time-averaged flow quantities and is therefore computationally less demanding but generally less accurate (Baungaard et al., 2022). Finally, the **computational claim** summarises how each study positions its method in terms of computational cost relative to high-fidelity CFD simulations and conventional engineering wake models.

Taken together, the comparison reveals a systematic divide in the literature. Field-based machine-learning models trained on high-fidelity flow data can retain detailed spatial wake information, while graph-based models provide a flexible representation of turbine interactions that can accommodate varying wind-farm layouts and offer low computational inference costs (R. Li et al., 2022; Schøler et al., 2025). This gap motivates the central research question of this thesis: whether a graph-based model trained directly on LES data can combine the physical fidelity of LES-based surrogates with the scalability of turbine-level interaction models, and thereby outperform traditional engineering wake models in offshore wind-farm applications.

1.3. Comparative analysis of the literature

1.3.1. Overview of key themes

The reviewed literature addresses a common objective: predicting wake effects in wind farms with sufficient physical fidelity while maintaining computational efficiency for design and operation. To structure the analysis, this chapter reviews four major strands of wake-modelling research: classical engineering wake models, data-driven wake models trained on high-fidelity simulation data, graph-based approaches for turbine-interaction modelling, and hybrid or physics-informed methods. Together, these strands capture the main methodological developments in the field and provide the basis for identifying the research gaps addressed in this thesis.

These modelling approaches are compared in terms of their underlying methodology, their use of high-fidelity data such as LES, their representation of wake interactions, and the trade-off between computational efficiency and predictive accuracy.

1.3.2. Engineering wake models as the baseline for wind-farm prediction

Engineering wake models form the traditional baseline for wind-farm wake prediction due to their low computational cost and analytical transparency (Shakoor et al., 2016). Early formulations, such as the Jensen wake model, describe wake expansion using simplified top-hat velocity deficit assumptions and linear growth laws (Jensen, 1983). Despite their simplicity, such models remain widely used for layout optimisation and early-stage planning.

Subsequent developments sought to improve physical realism. Ainslie introduced an eddy-viscosity-based formulation to model wake recovery (Ainslie, 1988), while Bastankhah and Porté-Agel proposed a Gaussian wake model derived from momentum conservation, in which the wake velocity deficit is

assumed to follow a Gaussian distribution across the wake cross-section (Bastankhah & Porté-Agel, 2014). Wake meandering models further extend these approaches by accounting for large-scale wake displacement driven by atmospheric turbulence (Larsen et al., 2008).

Despite their computational efficiency, engineering wake models rely on simplifying assumptions and empirical parameterisations, which can reduce accuracy under complex atmospheric conditions, strong wake interactions, and large wind-farm layouts (Krutova et al., 2020; Santoni et al., 2024). LES is therefore commonly used as a benchmarking reference to assess their limitations rather than as a training source (Krutova et al., 2020). These limitations motivate the development of data-driven wake models that seek to retain computational efficiency while improving predictive accuracy relative to traditional engineering wake models (Ye et al., 2025).

1.3.3. Data-driven wake modelling enabled by high-fidelity LES data

The increasing availability of high-fidelity simulation data has enabled a growing body of research on purely data-driven wake modelling. Grid-based neural networks, most notably convolutional neural networks (CNNs), dominate this line of work. When trained on LES data, these models can reproduce complex wake structures and velocity deficit patterns that are difficult to capture using analytical or engineering formulations (R. Li et al., 2022; Zhang et al., 2025).

Multiple studies show that LES-trained CNN models substantially outperform engineering wake models in predictive accuracy, while achieving inference times that are orders of magnitude lower than those of CFD simulations (Santoni et al., 2024). This combination of accuracy and efficiency makes them appealing surrogates for applications that require repeated wake evaluations.

Despite these advantages, field-based models generally rely on fixed spatial grids to represent the flow field. This can reduce their flexibility when applied to wind farms with different layouts or turbine configurations (Ye et al., 2025). In addition, evidence of robust generalisation across atmospheric conditions and wind-farm scales remains limited (Ye et al., 2025). Consequently, although LES-trained CNNs demonstrate the potential of high-fidelity data-driven wake modelling, their application to large and diverse wind-farm layouts remains challenging (Ye et al., 2025).

1.3.4. Graph-based modelling of wind farms

Graph-based learning has emerged as a promising approach for wind-farm modelling because it can explicitly represent turbine interactions (Bentsen et al., 2022). In these models, turbines are represented as nodes and wake relationships are encoded through directed edges, where connections are established between upstream and downstream turbines according to the wind direction and relative turbine positions (Bentsen et al., 2022). GNNs have been shown to capture inter-turbine dependencies efficiently and to scale well to large wind farms (Bentsen et al., 2022; Hou et al., 2026; S. Li et al., 2023; Liu et al., 2024).

Two graph-based representations can be distinguished in the wake-modelling literature. Some studies model turbines as graph nodes, whereas others construct graphs directly on the flow field using spatial mesh points as nodes (Bentsen et al., 2022; S. Li et al., 2023). The latter, such as the approach of Li et al. (S. Li et al., 2023), are conceptually closer to field-based surrogates than to interaction-based wake models.

At the turbine-interaction level, GNNs have been trained on outputs from engineering wake models (Duthé et al., 2023, 2024) or on steady-state RANS CFD simulations (Bleeg, 2020). For example, Bleeg (Bleeg, 2020) proposed a graph neural network surrogate model trained on RANS simulations to predict turbine interaction losses in wind farms. Subsequent work has evaluated related industrial implementations, such as the CFD.ML model, against operational wind-farm data (Levick et al., 2022). These studies indicate that graph-based interaction models can achieve engineering-level inference speed while matching or exceeding classical wake models in power prediction for large wind farms. However, because the training targets are derived from engineering models, RANS CFD, or operational data, existing graph-based approaches do not directly learn wake physics from LES. This distinction may be important because wake development and recovery are strongly influenced by atmospheric turbulence and atmospheric stability (Platis et al., 2022). LES data explicitly capture many of these effects, whereas engineering-model outputs and steady-state RANS simulations represent them only in a simplified manner (Platis et al., 2022).

Consequently, while graph-based models offer advantages in scalability and structural representation

(S. Li et al., 2023), a direct comparison of GNN-based wake models trained on LES data with traditional engineering wake models against LES reference data remains largely unexplored.

1.3.5. Hybrid and physics-informed approaches

Hybrid and physics-informed approaches combine traditional engineering wake models with machine-learning techniques to improve predictive accuracy while maintaining low computational cost (Santoni et al., 2024; Wang et al., 2025). Examples include symbolic regression coupled with dynamic wake meandering (Wang et al., 2025) and CNN-based augmentation of engineering wake models, both of which achieve improved accuracy at near-engineering computational cost (Santoni et al., 2024).

Related work by Howland and Dabiri shows that physics-informed models trained on operational SCADA data can outperform classical engineering wake models in turbine-level power prediction (Howland & Dabiri, 2019). While these approaches improve the accuracy of engineering wake models, they remain fundamentally different from models that learn wake behaviour directly from LES data.

More broadly, developments in physics-informed machine learning demonstrate that significant computational speed-ups are possible without abandoning physical constraints (Kochkov et al., 2021; Z. Li et al., 2021). Together, these approaches provide an intermediate reference point between purely engineering models and fully data-driven surrogates.

1.3.6. Summary of insights

The reviewed literature highlights several complementary approaches to wind-farm wake modelling. Engineering wake models remain widely used because of their computational efficiency and interpretability, but their accuracy can deteriorate under complex atmospheric conditions and strong wake interactions (Krutova et al., 2020; Shakoor et al., 2016). Data-driven models trained on LES data demonstrate that high-fidelity wake behaviour can be learned directly from simulation data, often achieving higher predictive accuracy than traditional engineering approaches (R. Li et al., 2022; Santoni et al., 2024). However, these models are typically field-based and rely on fixed spatial representations, which can limit their flexibility across different wind-farm layouts and scales (R. Li et al., 2022; Zhang et al., 2025).

Graph-based approaches address this limitation by representing wake interactions directly between turbines, providing a scalable framework for wind farms (Bleeg, 2020; S. Li et al., 2023). Existing turbine-interaction GNNs have shown promising performance, but are generally trained on engineering-model outputs, RANS CFD simulations, or operational data rather than LES (Bleeg, 2020; Duthé et al., 2023; S. Li et al., 2023). Hybrid and physics-informed approaches occupy an intermediate position, combining machine-learning techniques with established wake-modelling frameworks to improve predictive accuracy while retaining much of the computational efficiency and physical structure of engineering models (Duthé et al., 2024; Howland & Dabiri, 2019; Santoni et al., 2024).

Overall, the literature reveals a trade-off between physical fidelity, computational efficiency, and scalability. While substantial progress has been made across all three areas, no single modelling paradigm has yet emerged as a dominant solution for high-fidelity and computationally efficient wake prediction at the wind-farm scale.

1.3.7. Research gaps

Despite recent advances, key gaps remain in the literature. To the best of the author's knowledge, no systematic studies train turbine-interaction-based graph wake models directly on LES data and evaluate them against traditional engineering wake models under consistent conditions. Moreover, side-by-side comparisons across modelling paradigms using a common high-fidelity reference are limited, and robustness across atmospheric conditions and wind-farm configurations remains insufficiently explored.

These gaps motivate the central research question addressed in this thesis:

> *To what extent can a Graph Neural Network, trained on LES data, improve the accuracy of offshore wind-farm wake predictions while maintaining computational efficiency compared to traditional engineering wake models?*

1.4. Research design and methodology

This research evaluates whether a GNN trained on LES data can outperform a traditional engineering wake model in the prediction of wake-induced wind-speed deficits in offshore wind farms while

maintaining practical computational efficiency. The GNN is compared against the Jensen wake model, implemented through the PyWake NOJ framework, which serves as the engineering-model baseline throughout this thesis.

The analysis is based on the WINS50 dataset developed by Whiffle, which contains high-fidelity LES simulations of offshore wind farms (Whiffle, 2024). These LES data are used both to train the GNN and as the reference against which model predictions are evaluated. Using a common LES reference enables a consistent comparison between modelling approaches under identical inflow conditions and wind-farm configurations.

A turbine-level interaction framework is adopted in which turbines are represented as graph nodes and wake interactions are represented as directed connections between turbines. The proposed GNN is implemented using PyTorch and PyTorch Geometric, allowing wake interactions to be learned directly from LES-derived data (PyTorch, 2026). Model performance is evaluated using quantitative error metrics and compared with the predictions obtained from the Jensen wake model.

Together, these elements provide a transparent and reproducible framework for assessing whether LES-trained graph-based models can combine the predictive accuracy associated with high-fidelity simulations with the computational efficiency required for practical wind-farm applications.

1.4.1. Sub-questions, methods, data requirements and analysis tools

To address the main research question in a structured and transparent manner, the research is divided into five sub-questions. Each sub-question focuses on a specific aspect of the proposed framework, ranging from model design and the representation of heterogeneous inflow conditions to graph construction, performance evaluation, and scalability analysis. Together, these sub-questions define the methodological pathway through which the proposed Graph Neural Network is developed, evaluated, and compared with traditional engineering wake models. An overview of the sub-questions and their corresponding methods, data requirements, sources, and analysis tools is provided in Table 1.2.

SQ1: How can a Graph Neural Network be designed to predict wake-induced wind-speed deficits? This sub-question focuses on defining a turbine-level GNN architecture, including node and edge features, message-passing structure, and output targets derived from LES data.

SQ2: How can a GNN represent heterogeneous flow conditions within large offshore wind farms? This sub-question examines how local variations in wind speed and wind direction across a wind farm can be incorporated into a graph-based modelling framework. The resulting model configurations are compared to evaluate the impact of heterogeneous inflow representations on wake-prediction performance.

SQ3: How can atmospheric stability be incorporated through turbulence intensity in GNN-based wake prediction? This sub-question investigates the use of turbulence intensity as a representation of atmospheric stability within the graph-based framework. The analysis evaluates whether including turbulence intensity as an additional input feature improves the prediction of wake-induced wind-speed deficits.

SQ4: How do graph construction choices and attention mechanisms affect wake prediction accuracy? This sub-question investigates the impact of different graph topologies and message-passing mechanisms on the ability of the GNN to prioritise dominant wake interactions.

SQ5: How computationally efficient and scalable is the GNN-based wake model compared with a traditional engineering wake model? This sub-question assesses both the computational efficiency and scalability of the proposed approach. Training and evaluation costs are analysed as wind-farm size and graph connectivity increase, while the evaluation runtime of the trained GNN is compared with the Jensen engineering wake model.

1.4.1.1. Overview table

By explicitly linking each sub-question to a specific modelling decision or evaluation step, this section defines a transparent and reproducible research pathway that aligns with the overall research objective.

Sub-question	Data requirements	Sources	Tools
SQ1	Turbine layout; inflow conditions; wake-induced wind-speed and power deficits	WINS50 LES dataset	Python; PyTorch; PyTorch Geometric
SQ2	Local and global wind-speed and wind-direction information; turbine–turbine interaction patterns	WINS50 LES dataset	Python; data visualisation libraries
SQ3	Turbulence intensity; corresponding wake-induced wind-speed deficits	WINS50 LES dataset	Python; statistical analysis tools
SQ4	Alternative graph topologies; edge definitions; attention mechanisms	Internally defined model variants	PyTorch Geometric
SQ5	Training and evaluation time; wind-farm size; graph connectivity; Jensen runtime	Internally generated benchmarks; Jensen model	Python; profiling tools

Table 1.2: Overview of sub-questions, data requirements, sources, and analysis tools.

2

Graph neural network theory

2.1. Introduction

Before implementing the full GNN model for wake prediction, it is useful to first illustrate the fundamental learning mechanism of message passing in a simplified setting.

A GNN can represent a wind farm as a graph in which turbines are nodes and wake interactions are represented as directed edges (Zhou et al., 2020). While the full model used later in this thesis incorporates heterogeneous inflow conditions and LES-derived data, the underlying learning mechanism can be demonstrated using a minimal conceptual example.

To build intuition for how turbine interactions can be learned from data, a small toy model is constructed. This simplified model illustrates how wake influence can be represented as messages exchanged between turbines and how the strength of these interactions can be learned through gradient-based optimisation.

The following sections first introduce a two-turbine toy model and then extend the formulation to a small wind farm layout. This chapter therefore provides the conceptual foundation of the graph-based wake interaction model used later in this thesis.

2.2. Toy model for turbine interaction learning

To illustrate the basic learning mechanism of a GNN, we first construct a minimal toy model consisting of two turbines aligned with the wind direction. Turbine 1 is assumed to experience the free-stream wind speed, while turbine 2 is affected by the wake generated by turbine 1.

Let the free-stream wind speed be denoted by U_∞. The hidden representation of each turbine is initialized as

$$h_1 = U_\infty, \qquad h_2 = U_\infty. \tag{2.1}$$

The interaction between the turbines is modeled as a simple message-passing step. Turbine 1 sends a message to turbine 2 that is proportional to its hidden state:

$$m = w\, h_1, \tag{2.2}$$

where w is a learnable parameter representing the strength of the wake deficit interaction. In practice, the model learns negative interaction weights so that the message reduces the downstream wind speed, representing the wake deficit.

The predicted wind speed at turbine 2 is then given by

$$\hat{U}_2 = h_2 + m. \tag{2.3}$$

To train the model, the prediction is compared to the true wind speed at turbine 2, denoted U_2^{true}. The loss function is defined as the squared error

$$\mathcal{L} = (\hat{U}_2 - U_2^{\text{true}})^2. \tag{2.4}$$

The parameter w is updated using gradient descent. The gradient of the loss with respect to w is

$$\frac{\partial \mathcal{L}}{\partial w} = 2(\hat{U}_2 - U_2^{\text{true}})h_1. \tag{2.5}$$

The parameter update rule becomes

$$w_{t+1} = w_t - \eta \frac{\partial \mathcal{L}}{\partial w_t}, \tag{2.6}$$

where η denotes the learning rate.

Although extremely simplified, this toy example illustrates the core idea of message passing in graph neural networks: interactions between turbines are represented as messages exchanged between nodes, and the model learns the interaction strength directly from data. Similar message passing formulations are used in recent machine-learning-based wake-modelling approaches (Duthé et al., 2023; Hou et al., 2026; S. Li et al., 2023).

2.3. Toy model extension to multiple turbines

To illustrate how the message passing mechanism scales to a wind farm, the toy model is extended from two turbines to a small array of sixteen turbines arranged in a 4×4 grid. Each turbine is treated as a node in a graph and potential wake interactions are represented as directed edges between turbines.

The hidden state of each turbine i is initialized with the free-stream wind speed

$$h_i = U_\infty, \qquad i = 1, \ldots, N \tag{2.7}$$

where $N = 16$ denotes the number of turbines.

2.3.1. Edge representation

Wake interactions between turbines are determined by their relative geometric position with respect to the wind direction. For each ordered turbine pair (i, j), the downwind and crosswind distances are computed as

$$x_{ij}^{\text{down}} = \frac{\Delta x_{ij} \cos\theta + \Delta y_{ij} \sin\theta}{D}, \tag{2.8}$$

$$x_{ij}^{\text{cross}} = \frac{-\Delta x_{ij} \sin\theta + \Delta y_{ij} \cos\theta}{D}, \tag{2.9}$$

where D denotes the rotor diameter and θ the wind direction. These normalized distances correspond to geometric quantities commonly used in classical engineering wake models, where wake deficits depend primarily on downstream distance and lateral offset (Ainslie, 1988; Bastankhah & Porté-Agel, 2014; Jensen, 1983).

A directed edge from turbine i to turbine j is created only if turbine j lies downstream of turbine i within a predefined wake corridor,

$$0 < x_{ij}^{\text{down}} \leq x_{\max}, \qquad |x_{ij}^{\text{cross}}| \leq y_{\max}. \tag{2.10}$$

This restriction ensures that interactions are only created when a physical wake interaction is geometrically plausible.

2.3.2. Edge interaction function

For each edge, a small neural network computes the interaction strength based on the relative turbine geometry. The first step computes a linear combination of the geometric inputs,

$$z_{ij} = a_1 x_{ij}^{\text{down}} + a_2 |x_{ij}^{\text{cross}}| + a_3, \tag{2.11}$$

where a_1, a_2, and a_3 are learnable parameters controlling the influence of downwind and crosswind distance. The absolute value of the crosswind distance is used because wake interactions are assumed to be symmetric with respect to the wake centreline. This is consistent with commonly used Gaussian wake formulations, such as the model of Bastankhah and Porté-Agel (2014), where the velocity deficit depends on the squared lateral distance from the wake centreline, resulting in identical wake effects for equal positive and negative crosswind offsets (Bastankhah & Porté-Agel, 2014).

A nonlinear activation function is required to enable the interaction network to represent nonlinear relationships between turbine geometry and wake influence (Glorot et al., 2011). Without an activation function, the interaction function would remain purely linear, limiting the model's ability to capture threshold-like wake behaviour. In this work, the rectified linear unit (ReLU) activation function is used (Glorot et al., 2011). The rationale for this choice is discussed below.

$$h_{ij} = \max(0, z_{ij}). \tag{2.12}$$

The resulting value is transformed into an edge weight

$$w_{ij} = a_4 h_{ij} + a_5, \tag{2.13}$$

where a_4 and a_5 are additional learnable parameters. Together, the parameters $a_1, \ldots, a_5$ form a small neural network that maps turbine geometry to a wake interaction strength.

The message transmitted from turbine i to turbine j is defined as

$$m_{ij} = h_i \, w_{ij}, \tag{2.14}$$

meaning that the wake influence depends both on the upstream wind state h_i and on the learned interaction weight.

2.3.3. Message aggregation

The predicted wind speed at turbine j is obtained by aggregating the incoming messages from upstream turbines,

$$\hat{U}_j = h_j + \sum_{i \in \mathcal{N}_j} m_{ij}, \tag{2.15}$$

where $\mathcal{N}_j$ denotes the set of turbines that have an edge pointing toward turbine j.

The model parameters are learned by minimizing the discrepancy between predicted and observed turbine wind speeds. Gradients are propagated through the interaction function and message passing operations to update the parameters using gradient descent.

This formulation corresponds directly to the message passing mechanism used in graph neural networks (Gilmer et al., 2017). Each turbine node receives information from neighboring turbines through edge messages, which are computed using a learnable interaction function of the turbine geometry. The node state is then updated by aggregating these incoming messages, allowing the model to represent the combined wake influence of multiple upstream turbines.

Similar graph-based representations of turbine interactions have recently been applied in data-driven wind farm modelling and wake prediction (Bentsen et al., 2022; Hou et al., 2026; S. Li et al., 2023).

2.3.4. Activation function choice

For completeness, the choice of activation function used in the interaction network is briefly discussed. The rectified linear unit (ReLU) activation function is used because it activates wake interactions only when the geometric combination z_{ij} becomes positive. If $z_{ij} < 0$, the activation becomes zero and no interaction is transmitted, effectively filtering out geometrically implausible wake influences.

Alternative activation functions are possible but are less directly aligned with the physical interpretation of wake interactions in this context. For example, sigmoid and hyperbolic tangent functions map inputs to bounded ranges, which results in smooth, continuous responses across all input values (Glorot et al., 2011). While this can be beneficial in general neural network settings, it makes it less straightforward to represent threshold-like behaviour, where wake interactions are only relevant within specific geometric configurations. In contrast, the ReLU activation introduces a natural thresholding effect, which better reflects the conditional nature of wake interactions (Dubey et al., 2022).

ReLU therefore provides a simple and interpretable mechanism for activating turbine interactions only when geometric conditions indicate a potential wake influence.

2.4. General message passing formulation

The toy examples introduced above illustrate the basic idea of learning turbine interactions through message passing. More generally, graph neural networks operate by iteratively exchanging information between nodes connected by edges (Gilmer et al., 2017; Zhou et al., 2020). A wind farm can therefore be represented as a graph

$$G = (V, E), \tag{2.16}$$

where V denotes the set of turbine nodes and E the set of directed edges representing potential wake interactions between turbines.

Each turbine i is associated with a node state $h_i^{(k)}$ at message passing layer k. In the initial layer, this state contains the input features of the turbine, such as inflow conditions or geometric information. In the wind farm setting considered in this thesis, these features include turbine geometry and inflow conditions such as wind speed and wind direction.

$$h_i^{(0)} = x_i. \tag{2.17}$$

Edges may additionally carry feature vectors e_{ij} describing the geometric relationship between turbines i and j, such as normalised downwind and crosswind distances.

The message passing mechanism consists of two main operations: *message computation* and *message aggregation*. For each directed edge (i, j), a message is computed according to the general message passing framework proposed by Gilmer et al. (2017),

$$m_{ij}^{(k)} = \psi\left(h_i^{(k)}, e_{ij}\right), \tag{2.18}$$

where $\psi(\cdot)$ denotes a learnable message function, typically implemented as a neural network.

All incoming messages to turbine j are then aggregated,

$$M_j^{(k)} = \sum_{i \in \mathcal{N}_j} m_{ij}^{(k)}, \tag{2.19}$$

where $\mathcal{N}_j$ denotes the set of turbines that send messages to turbine j.

The turbine state is subsequently updated using another learnable function,

$$h_j^{(k+1)} = \phi\left(h_j^{(k)}, M_j^{(k)}\right), \tag{2.20}$$

where $\phi(\cdot)$ represents the node update function. This operation combines the current turbine state with the aggregated wake influence from neighboring turbines.

By stacking multiple message passing layers, information can propagate through the wind farm graph, allowing the model to represent complex multi-turbine wake interactions (Zhou et al., 2020). The toy models presented in the previous sections can be interpreted as simplified instances of this general message passing framework, where the interaction function and node update rules are reduced to simple linear operations.

This formulation forms the basis of the graph neural network models used later in this thesis for data-driven wake prediction.

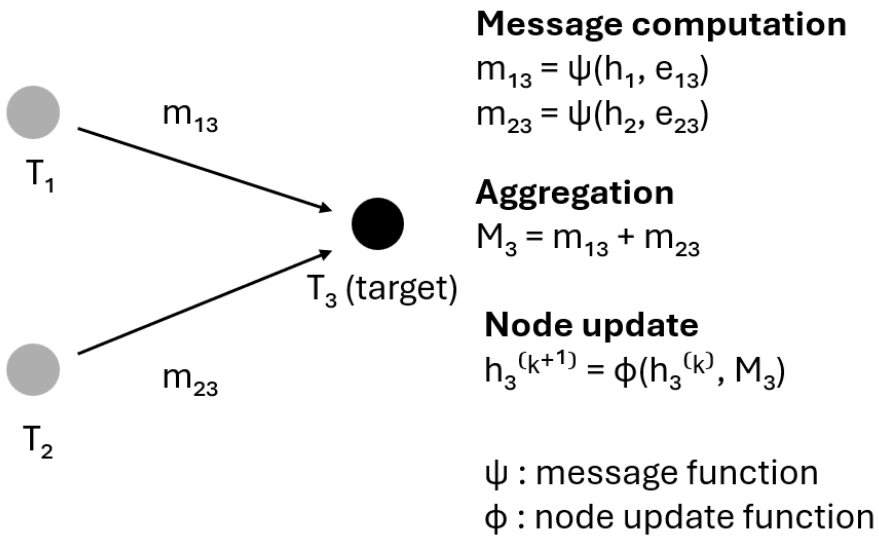


Figure 2.1: Illustration of the message passing mechanism in a wind farm graph. Upstream turbines T_1 and T_2 send messages to a downstream turbine T_3. Incoming messages are aggregated to form M_3, which is used by the node update function to update the turbine state.

Figure 2.1 illustrates the message passing mechanism for a simple three-turbine example. Upstream turbines transmit messages to a downstream turbine based on their current node states and the geometric relationship between turbines. These incoming messages are aggregated and used to update the downstream turbine state.

In a full wind farm graph, the same message passing procedure is applied simultaneously to all turbines, allowing the model to represent complex multi-turbine wake interactions across the entire farm. The next chapter builds on this formulation by presenting the full graph neural network architecture and its implementation for predicting turbine wind speeds under LES-derived inflow conditions.

3

Model implementation

3.1. Introduction

This chapter presents the implementation of the GNN for wake prediction, building on the conceptual framework introduced in Chapter 2. The objective is to translate the physics-informed message passing formulation into a concrete, data-driven model, enabling wake interactions to be learned directly from high-fidelity simulation data.

The implementation consists of four components: data preparation, graph construction, model architecture, and training setup. First, the WINS50 LES dataset is processed and a supervised learning problem is defined. Second, a directed graph representation of the wind farm is constructed, where turbines are modelled as nodes and wake interactions as edges derived from the prevailing wind direction. Third, the GNN architecture is specified using edge-conditioned message passing. Finally, the training procedure is described, including data splitting, normalisation, and optimisation.

Together, these steps define a reproducible pipeline for learning wake interactions from high-fidelity simulation data.

3.2. Data preparation

The WINS50 dataset is a high-fidelity LES dataset covering offshore wind conditions over the North Sea, produced by Whiffle as part of the WINS50 project (Whiffle, 2024). The dataset provides temporally resolved meteorological fields and turbine-level quantities for two scenarios: a 2020 reference scenario and a hypothetical 2050 scenario.

The data used in this study consists of two components: (i) gridded meteorological fields from the 2020 scenario and (ii) turbine-level time series from both the 2020 and 2050 scenarios. The field data provides spatially resolved atmospheric variables on a structured grid, including wind speed, wind direction, and turbulence-related quantities (Whiffle, 2024). The turbine data provides time series of flow conditions at individual turbine locations, including hub-height wind speed, wind direction, and turbulence intensity (Whiffle, 2024).

The 2020 scenario includes all wind turbines operational within the simulation domain and serves as a baseline case. The 2050 scenario extends this layout with additional future wind farms. Both scenarios are simulated over the same meteorological year, enabling a controlled comparison in which the primary difference is the presence of additional turbines and their associated wake interactions (Whiffle, 2024).

The WINS50 dataset provides different output products with varying temporal resolutions. In particular, turbine-level quantities are available at a 5-minute resolution, while the gridded meteorological field outputs used in this study are provided at an hourly resolution (Whiffle, 2024).

As a result, a temporal mismatch exists between the turbine and field datasets. This is resolved by matching each turbine timestep to the nearest available field timestep before defining the learning target.

The prediction target is defined as the difference in turbine-level wind speed between the 2050 and 2020 scenarios:

$$\Delta U_i = U_i^{2050} - U_i^{2020}, \tag{3.1}$$

where U_i^{2050} denotes the turbine-level wind speed in the 2050 scenario and U_i^{2020} represents the corresponding reference wind speed from the 2020 scenario. The resulting difference ΔU_i is used as the prediction target for the GNN.

This formulation is chosen over predicting absolute wind speeds, as it reduces large-scale inflow variability and emphasises wake-induced differences. Additionally, the verification experiments in Chapter 4 showed that predicting ΔU resulted in lower prediction errors than predicting absolute wind speeds. Consequently, the learning problem is directed towards modelling turbine–turbine interactions rather than atmospheric variability. By using the 2020 scenario as a consistent inflow reference, the model is encouraged to focus on the additional wake interactions introduced in the 2050 layout, rather than learning spurious correlations related to changing environmental conditions.

For wind farms that are only present in the 2050 scenario, the 2020 wind field at the corresponding locations is used as a proxy for the inflow conditions in the absence of these additional wakes, while still reflecting the influence of the existing 2020 wind farm layout. As an initial case study, the wind farm *doggerbankSW_east* is selected, as it is the smallest wind farm in the 2050 scenario, consisting of 12 turbines. This reduces computational complexity and allows for a clearer analysis of the learned wake interactions before scaling to larger wind farm clusters.

Wind speed and wind direction from the 2020 scenario are obtained from the gridded field data at a height of 150 m, which is adopted as the turbine hub height in this study. The WINS50 field outputs are provided at a discrete set of vertical levels, allowing hub-height conditions to be used directly without the need for vertical interpolation. This avoids introducing additional sources of uncertainty and ensures that all input variables are obtained directly from the LES data.

The gridded wind fields are interpolated to turbine locations using bilinear interpolation based on their (x, y) coordinates.

A global wind direction is used throughout this thesis to provide a consistent reference frame for defining wake interactions. At each timestep, this direction is computed using circular averaging over the spatial domain. This avoids local wake-induced distortions and ensures that the graph structure is defined based on the large-scale inflow conditions.

The resulting model inputs consist of turbine-specific wind speed values obtained through interpolation of the 2020 wind field and a global wind direction derived from the same field, with ΔU_i as the prediction target.

To avoid potential spin-up effects at the start of the simulation, a contiguous subset of $T = 1200$ timesteps (indices 200–1400) is selected, resulting in 1200 graph samples.

3.3. Graph construction

This section specifies the graph representation used in the implementation, building on the message passing framework introduced in Chapter 2. Each timestep is represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with node and edge features derived from the WINS50 data.

3.3.1. Node features

Each turbine node i is assigned a feature vector

$$\mathbf{h}_i = \left[U_i^{2020},\ \sin(\theta^{2020}),\ \cos(\theta^{2020}) \right], \tag{3.2}$$

where U_i^{2020} denotes the interpolated 2020 wind speed at turbine i, and θ^{2020} represents the global wind direction derived from the 2020 wind field. Both quantities are used as a consistent reference for the inflow conditions. The wind direction is encoded using sine and cosine to ensure continuity across the $0°/360°$ boundary. Spatial coordinates are not included as node features, as their effect is already captured through edge features describing relative turbine geometry.

3.3.2. Edge construction

Directed edges are constructed at each timestep based on the geometric relationship between turbine pairs under the prevailing wind direction. For each ordered turbine pair (i, j), the downwind and crosswind distances are computed relative to the global wind direction, following the formulation introduced in Chapter 2. An edge $(i \rightarrow j)$ is included if turbine j lies within a predefined wake corridor,

$$x_{\min} < x_{ij}^{\text{down}} < x_{\max}^{\text{down}} \quad \text{and} \quad \left| x_{ij}^{\text{cross}} \right| < x_{\max}^{\text{cross}}, \tag{3.3}$$

where $x_{\min} = 0.5D$, $x_{\max}^{\text{down}} = 45D$, and $x_{\max}^{\text{cross}} = 3D$, with D denoting the reference diameter used for distance normalisation and graph construction. This physics-based filtering ensures that only upstream–downstream wake interactions are included, explicitly enforcing the directional nature of wake propagation. In contrast to dense or fully connected graph representations, where all turbine pairs are allowed to exchange information, this work restricts edges to geometrically plausible wake interactions. This reflects the hypothesis that incorporating prior physical knowledge into the graph structure leads to more physically meaningful representations of turbine interactions while reducing spurious connections and improving computational efficiency. Within the general message passing framework (Gilmer et al., 2017), this corresponds to embedding inductive bias directly in the graph connectivity.

3.3.3. Edge features

Each directed edge $(i \rightarrow j)$ is assigned a feature vector

$$\mathbf{e}_{ij} = \left[\frac{x_{ij}^{\text{down}}}{D}, \frac{x_{ij}^{\text{cross}}}{D} \right], \tag{3.4}$$

which encodes the relative turbine geometry in rotor-diameter units. These features provide a compact, physically interpretable description of turbine interactions, relating wake effects directly to downstream distance and lateral offset.

3.3.4. Temporal graph structure

A separate graph is constructed for each timestep. While turbine positions remain fixed, both node features and edge connectivity vary over time due to changes in inflow wind speed and global wind direction. As the wind direction changes, the wake corridor rotates accordingly, resulting in a time-dependent set of directed edges. The resulting dataset therefore consists of a sequence of $T = 1200$ dynamic graphs.

3.3.5. Target variable

The node-level prediction target ΔU_i is defined in Section 3.2 and attached to each graph instance.

3.3.6. Design choices

Several modelling choices are made to align the graph structure with the underlying wake physics. First, the use of a global wind direction ensures a consistent reference frame for defining upstream–downstream relationships. Second, a physics-based wake corridor is used instead of a dense graph representation, explicitly restricting interactions to geometrically plausible wake paths and avoiding spurious connections between unrelated turbines. Third, edge features are limited to normalised downwind and crosswind distances, providing a minimal yet physically meaningful description of turbine interactions.

These design choices are inspired by classical engineering wake models, in which wake effects depend strongly on the relative position of turbines with respect to the wind direction. Both the Jensen and Bastankhah and Porté-Agel wake models describe wake behaviour as a function of downstream and lateral position, motivating the use of $(x_{\text{down}}/D, x_{\text{cross}}/D)$ as interaction features (Bastankhah & Porté-Agel, 2014; Jensen, 1983).

3.4. Model architecture

This section describes the graph neural network architecture used to predict wake-induced wind speed differences at turbine level. Building on the graph representation defined in Section 3.3, the model maps

node and edge features to node-level predictions of ΔU_i. The model is implemented in Python using the PyTorch deep learning framework together with the PyTorch Geometric library (PyTorch, 2026).

3.4.1. Edge-conditioned message passing

The architecture follows the message passing formulation introduced in Chapter 2. For each directed edge (i, j), a message is computed as

$$\mathbf{m}_{ij}^{(k)} = \Psi^{(k)}(\mathbf{e}_{ij})\ \mathbf{h}_i^{(k)}, \tag{3.5}$$

where $\mathbf{h}_i^{(k)}$ denotes the node state of turbine i at layer k, $\mathbf{e}_{ij}$ the edge feature vector, and $\Psi^{(k)}$ a learnable edge network that maps edge features to a weight matrix. This formulation allows interaction strengths between turbines to depend explicitly on their relative geometry, directly linking learned messages to wake-relevant spatial features.

Incoming messages are aggregated using mean pooling,

$$\mathbf{M}_j^{(k)} = \frac{1}{|\mathcal{N}_j|} \sum_{i \in \mathcal{N}_j} \mathbf{m}_{ij}^{(k)}, \tag{3.6}$$

For simplicity, the node update is written in terms of aggregated messages only,

$$\mathbf{h}_j^{(k+1)} = \mathrm{ReLU}\left(\mathbf{M}_j^{(k)}\right). \tag{3.7}$$

In practice, the NNConv operator additionally applies learnable transformations to node features and implicitly combines self-information with aggregated neighbor messages. The above formulation is therefore a simplified representation of the underlying operation.

3.4.2. NNConv layers

The model consists of two NNConv layers. NNConv is an edge-conditioned message passing operator in which the messages exchanged between nodes depend explicitly on the edge features connecting them (Simonovsky & Komodakis, 2017). This makes it particularly suitable for wake modelling, as the strength of turbine interactions is expected to depend on their relative geometry, specifically the downstream and crosswind distances.

Each NNConv layer uses a small multilayer perceptron (MLP) as the edge network $\Psi^{(k)}$ (Simonovsky & Komodakis, 2017). The edge network takes the two-dimensional edge feature vector $\mathbf{e}_{ij}$ as input and outputs a weight matrix that transforms the source node features (Simonovsky & Komodakis, 2017). As a result, the interaction strength between turbines is not fixed but learned as a function of their relative position (Simonovsky & Komodakis, 2017).

The first NNConv layer captures direct wake interactions from immediate upstream neighbours, while the second layer enables information to propagate further through the graph, allowing indirect multi-turbine wake effects to be represented. After each layer, a rectified linear unit (ReLU) activation function is applied.

Within this formulation, the graph structure determines which turbine pairs can exchange information, while the edge network learns how geometric relationships between turbines translate into wake-induced interaction strengths. In the implemented model, this mechanism is realised using the `NNConv` operator from the PyTorch Geometric library, with all parameters learned through gradient-based optimisation (PyTorch, 2026).

3.4.3. Prediction head

After the message passing layers, the final node representations are mapped to a scalar prediction using a feed-forward output network,

$$\Delta\hat{U}_i = \mathbf{W}_2\,\mathrm{ReLU}\left(\mathbf{W}_1\mathbf{h}_i^{(2)} + \mathbf{b}_1\right) + \mathbf{b}_2, \tag{3.8}$$

where $\mathbf{h}_i^{(2)}$ denotes the node representation after the second message passing layer, and $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, and $\mathbf{b}_2$ are learnable parameters. The output $\Delta\hat{U}_i$ represents the predicted wake-induced wind speed difference at turbine i, completing the mapping from inflow conditions and turbine interactions to wake-induced deficits.

3.4.4. Model depth and interpretation

Using two message passing layers allows the model to represent both direct and indirect wake interactions, enabling information to propagate beyond immediate neighbours and capture multi-turbine wake effects.

The architecture therefore combines a physics-informed graph structure with a flexible, data-driven interaction model. The graph determines which turbine pairs can interact, while the NNConv layers learn how strongly these interactions affect downstream wind speed deficits.

3.4.5. Model configuration

The hidden dimension is set to $d = 32$ for now, balancing model capacity with the limited size of the training dataset. The overall model is intentionally kept compact to match the size of the dataset and to maintain interpretability. The total number of trainable parameters is small by design, reflecting the limited size of the training dataset and the goal of constructing an interpretable baseline model. A summary of the model configuration is provided in Table 3.1.

Table 3.1: Model configuration.

Component	Setting
Message passing layers	2
Hidden dimension	32
Edge network	Linear → ReLU → Linear
Aggregation	Mean
Activation	ReLU
Node features	3
Edge features	2
Output	Scalar per node

3.5. Training setup

This section describes the training procedure used to optimise the graph neural network, including data splitting, normalisation, loss function, and optimisation settings.

3.5.1. Data split

The initial testing consists of $T = 1200$ graph samples, each corresponding to a single timestep. A chronological split is applied to preserve the temporal structure of the dataset. The first 70% of timesteps are used for training, the subsequent 15% for validation, and the final 15% for testing, resulting in 840 training samples, 180 validation samples, and 180 test samples.

A chronological split is preferred over a random split because observations in time-series data are generally not independent (Hyndman & Athanasopoulos, 2021). Randomly assigning timesteps to the training and test sets may place highly correlated observations in both sets, leading to overly optimistic estimates of model performance (Hyndman & Athanasopoulos, 2021). Therefore, the model is evaluated on observations that occur later in time than those used for training.

3.5.2. Normalisation

Node features and target values are normalised using statistics computed from the training set only, ensuring that information from future observations is not used during model development (Hyndman & Athanasopoulos, 2021). During evaluation, predictions are transformed back to physical units (m/s).

$$\tilde{\mathbf{h}}_i = \frac{\mathbf{h}_i - \boldsymbol{\mu}_x}{\boldsymbol{\sigma}_x}, \qquad \Delta\tilde{U}_i = \frac{\Delta U_i - \mu_y}{\sigma_y} \tag{3.9}$$

where $\boldsymbol{\mu}_x$, $\boldsymbol{\sigma}_x$, μ_y, and σ_y are computed from the training set. Edge features are not normalised because downstream and crosswind distances are already expressed in rotor-diameter units. Normalisation helps the optimisation process by ensuring that variables with different numerical ranges contribute on comparable scales during training (Goodfellow et al., 2016).

3.5.3. Loss function

The model is trained by minimising the mean squared error (MSE) between the normalised predictions and targets,

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(\Delta\hat{\tilde{U}}_i - \Delta\tilde{U}_i \right)^2, \tag{3.10}$$

where N denotes the total number of turbine nodes in a batch. The use of MSE encourages the model to penalise larger prediction errors more strongly, which is desirable for accurately capturing strong wake deficits (Goodfellow et al., 2016).

3.5.4. Optimisation

Model parameters are optimised using the Adam optimiser, an adaptive gradient-based optimisation algorithm that adjusts the learning rate for each parameter during training (Kingma & Ba, 2015). A learning rate of $\eta = 10^{-3}$ is used, as this is a commonly used default value for Adam and provided stable convergence during preliminary model testing. Training is performed in mini-batches of 16 graph samples and run for a maximum of 200 epochs, with each epoch representing one complete pass through the training dataset.

3.5.5. Early stopping

To prevent overfitting, early stopping is applied based on the validation performance (Goodfellow et al., 2016). Validation performance is monitored using the mean absolute error (MEA), as it provides a more interpretable measure of error in physical units. Training is terminated if the validation MAE does not improve for 15 consecutive epochs. The model parameters corresponding to the lowest validation MAE are retained for final evaluation.

3.5.6. Evaluation metric

Model performance is evaluated using the MAE computed on denormalised predictions,

$$\mathrm{MAE} = \frac{1}{N} \sum_{i=1}^{N} \left| \Delta\hat{U}_i - \Delta U_i \right|, \tag{3.11}$$

expressed in metres per second. This metric provides a physically interpretable measure of prediction error at turbine level.

A summary of the training configuration is provided in Table 3.2.

Table 3.2: Training configuration.

Parameter	**Value**
Training samples	840
Validation samples	180
Test samples	180
Data split	Chronological (70/15/15)
Optimiser	Adam
Learning rate	10^{-3}
Batch size	16
Maximum epochs	200
Early stopping patience	15 epochs
Loss function	Mean squared error (MSE)
Evaluation metric	Mean absolute error (MAE)

4

Results: Model verification and validation

4.1. Verification

In Chapter 3, the GNN framework for wake prediction was developed, including the graph construction procedure, model architecture, and training pipeline. However, the implementation itself has not yet been verified. Before applying the framework to complex LES-based wind farm data, it is first necessary to establish whether the model is capable of correctly learning known wake-interaction behaviour in a controlled setting.

This chapter therefore verifies the implemented framework using the Jensen wake model, specifically the NOJ formulation in PyWake, as a reference. By generating synthetic wind farm layouts and randomly sampled inflow conditions, the wake-learning problem can be isolated from the additional complexity of realistic atmospheric variability, heterogeneous inflow fields, and LES-specific noise.

The GNN is trained to reproduce turbine-level wind speeds predicted by the Jensen model, allowing direct comparison between the learned approximation and the known engineering wake model. This provides a controlled benchmark for assessing whether the graph-based message-passing framework captures the intended wake interactions.

Beyond predictive accuracy, the experiments analyse the effect of dataset size, model capacity, and wind farm size, as well as computational performance and generalisation to unseen inflow conditions and modified wind farm layouts. Finally, a downstream-distance error analysis is performed to assess whether prediction accuracy remains consistent across both short- and longer-range wake interactions.

Together, these experiments provide a systematic verification of the implemented framework and establish confidence in the technical correctness of the modelling approach before moving to the substantially more complex LES-based experiments in the following chapters.

4.1.1. Synthetic benchmark setup

A synthetic benchmark is constructed to verify the correctness of the implemented model. In contrast to Chapter 3, where the model is trained on LES data, this setup uses an artificial wind farm layout and randomly generated inflow conditions. This allows full control over the input space and isolates the wake-learning problem.

A 2×2 wind farm is considered, consisting of four turbines arranged in a regular grid. The turbines are spaced $7D$ apart in the streamwise direction and $5D$ in the lateral direction. This layout is simple enough to remain interpretable, while still producing wake interactions.

For each timestep, the ambient inflow wind speed is sampled uniformly between 4 and 14 m/s, and the wind direction is sampled uniformly over $[0°, 360°)$. This results in a wide range of wake configurations and exposes the model to diverse interaction patterns.

The graph representation follows the same physics-based construction as described in Chapter 3. Node

features consist of a global inflow wind speed and wind direction that are identical for all turbines within a timestep. As a result, no turbine-specific inflow variability is present, allowing the verification experiment to focus solely on the learned wake interactions. Edges are defined using the wake-corridor criterion and corresponding geometric features.

For each timestep, turbine-level wind speeds are computed using the Jensen wake model. These values serve as ground truth targets for training and evaluation.

4.1.2. Training and evaluation setup

The GNN is trained to approximate the turbine-level wind speeds predicted by the Jensen model. Two target formulations are considered.

First, the model predicts the absolute turbine-level wind speed,

$$U_i^{\text{Jensen}}. \tag{4.1}$$

Second, the model predicts the wake-induced deviation from the inflow,

$$\Delta U_i = U_i^{\text{Jensen}} - U_{\text{global}}. \tag{4.2}$$

In this case, the absolute wind speed is reconstructed as

$$U_i^{\text{Jensen}} = U_{\text{global}} + \Delta U_i. \tag{4.3}$$

This comparison assesses whether separating global inflow conditions from local wake effects provides a clearer learning signal.

A dataset of synthetic timesteps is generated, with independent samples corresponding to different inflow conditions. The model is trained and evaluated using the procedure described in Chapter 3. Performance is assessed using the MAE between predicted and reference turbine-level wind speeds.

4.1.3. Verification results

The predictive performance of the GNN is first evaluated in the controlled synthetic Jensen benchmark as a function of training set size and model capacity. Figure 4.1 shows the test MAE for different hidden dimensions and for both target formulations.

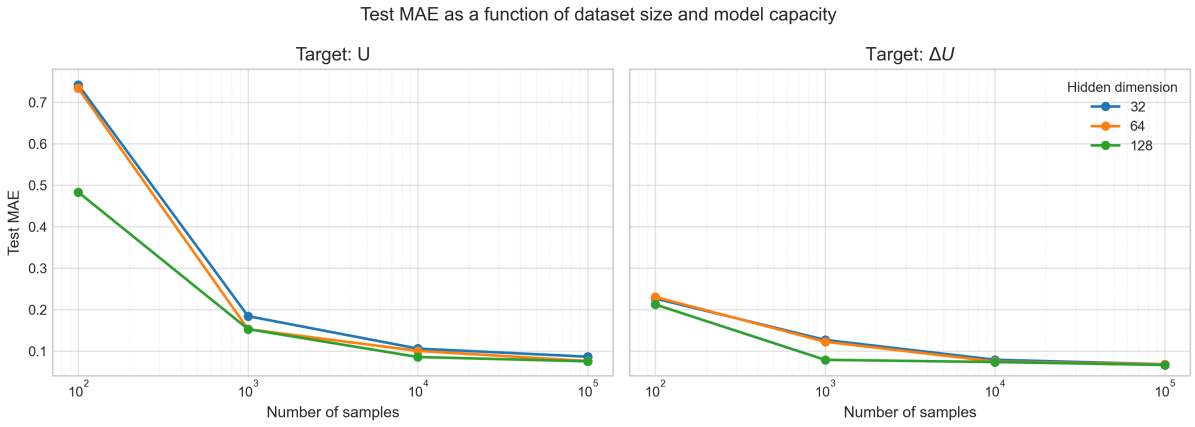


Figure 4.1: Test MAE as a function of training set size for different hidden dimensions, shown separately for the two target formulations. Performance improves rapidly with increasing dataset size up to approximately 10^4 samples, after which gains diminish. The influence of model capacity is comparatively limited.

Several observations can be made. First, predictive performance improves strongly as the number of training samples increases from 10^2 to 10^4. Beyond this point, additional improvements become relatively small, indicating diminishing returns. This behaviour is consistent with the relatively limited complexity of the Jensen wake model: once the dominant wake-interaction patterns have been learned, additional synthetic samples provide comparatively little new information.

Second, the influence of model capacity is modest compared to the effect of dataset size. Increasing the hidden dimension leads to only minor improvements in predictive accuracy, indicating that even relatively compact models are already sufficiently expressive for this controlled approximation task.

Third, the ΔU target formulation consistently yields lower and more stable errors than direct prediction of absolute turbine wind speed. This supports the hypothesis that separating global inflow conditions from local wake-induced deviations provides a clearer learning signal, allowing the model to focus more directly on the wake interactions it is intended to learn.

Overall, these results demonstrate that the implemented GNN is capable of reproducing the Jensen wake model with low prediction error in a controlled synthetic setting. A complete overview of all verification experiments, including runtime breakdowns and detailed performance metrics, is provided in Appendix B.

4.1.3.1. Scaling with wind farm size

To assess whether the implemented framework remains effective for larger wind farms with more complex wake interactions, additional experiments were performed for increasingly large synthetic layouts ranging from 2×2 up to 10×10 turbines. For consistency, all experiments in this scaling analysis use the ΔU target formulation and a hidden dimension of 64, since the previous results showed that this target formulation consistently outperformed direct prediction of absolute wind speed while providing a practical balance between model complexity and predictive accuracy.

Table 4.1: Scaling behaviour for increasingly large synthetic wind farm layouts using the ΔU target formulation and hidden dimension 64.

Layout	Turbines	Samples	Total time	Jensen	Graph	Training	Test MAE
2×2	4	10 000	21 m 13 s	3 m 32 s	11 s	17 m 25 s	0.0745
4×4	16	10 000	35 m 57 s	3 m 46 s	24 s	31 m 21 s	0.0771
6×6	36	10 000	74 m 18 s	4 m 52 s	58 s	68 m 02 s	0.0813
8×8	64	10 000	148 m 11 s	6 m 11 s	1 m 42 s	139 m 04 s	0.0854
10×10	100	10 000	276 m 16 s	8 m 06 s	2 m 09 s	265 m 34 s	0.0888

Several clear trends emerge from Table 4.1. As wind farm size increases, the total runtime rises substantially. This increase is primarily driven by the training phase, which becomes progressively more expensive as the number of turbines, wake interactions, and graph edges increases. Jensen-based data generation and graph construction also become slower for larger layouts, although their contribution remains comparatively small relative to the total training cost.

At the same time, predictive performance remains remarkably stable despite the substantial increase in graph size and interaction complexity. Even for the 10×10 layout containing 100 turbines and dense wake interactions, the model maintains a test MAE below 0.1 m/s. This indicates that the implemented GNN framework is capable of reproducing the Jensen wake model accurately not only for small synthetic toy layouts, but also for substantially larger wind farms with many interacting wakes.

The reported runtimes should primarily be interpreted as indicative scaling behaviour rather than exact benchmark values, since each experiment was executed only once and may be affected by normal fluctuations in hardware utilisation. Nevertheless, the overall trend is clear: computational cost increases strongly with wind farm size, while predictive accuracy remains consistently high.

Overall, these results indicate that the implemented framework scales effectively in terms of predictive performance, even as the underlying graph complexity increases substantially.

4.1.3.2. Computational performance

To further analyse computational behaviour, Figure 4.2 shows the runtime breakdown as a function of dataset size for the ΔU target formulation with hidden dimension 64.



Figure 4.2: Runtime as a function of dataset size for the ΔU target formulation with hidden dimension 64. The total runtime is dominated by the training phase.

The total runtime increases strongly with dataset size and is consistently dominated by the training phase. Jensen-based data generation grows more gradually and remains significantly smaller than the training cost, while graph construction contributes only a minor fraction of the total runtime.

This indicates that the primary computational bottleneck of the framework lies in model training rather than data preparation. While this introduces a substantial upfront computational cost, the resulting trained model can subsequently serve as a fast learned surrogate for repeated wake predictions.

To further assess the effect of model capacity, Table 4.2 compares average runtime and predictive accuracy for different hidden dimensions using the ΔU target formulation. The reported values are averaged over the 10^4 and 10^5 sample experiments.

Table 4.2: Average performance for different hidden dimensions using the ΔU target formulation, computed over the 10^4 and 10^5 sample experiments.

Hidden dimension	Avg. total runtime	Avg. test MAE
32	1 h 02 m 27 s	0.0734
64	1 h 48 m 53 s	0.0711
128	2 h 05 m 39 s	0.0698

Increasing the hidden dimension from 32 to 64 improves predictive performance, although this comes at a substantial increase in computational cost. Increasing the hidden dimension further to 128 yields only a very small additional reduction in MAE despite further increasing runtime, indicating clear diminishing returns in model capacity for this verification task.

It should be noted that these experiments are performed in a relatively controlled synthetic setting with limited wake complexity. In this context, even smaller hidden representations are sufficient to capture most relevant wake-interaction patterns. For this reason, a hidden dimension of 64 is selected as a practical compromise between computational cost and predictive accuracy for the verification experiments and the later LES-based analyses.

4.1.4. Generalisation and robustness

To further assess whether the implemented GNN learns general wake relationships rather than only reproducing specific training configurations, additional robustness experiments are performed. These tests evaluate extrapolation to unseen wind-speed ranges and generalisation to modified wind farm layouts.

4.1.4.1. Wind-speed extrapolation

To assess robustness outside the training range, an out-of-distribution (OOD) test is performed in which the GNN is trained only on inflow wind speeds between 6 and 12 m/s. Performance is then evaluated on one in-distribution test set and two OOD test sets covering lower (4–6 m/s) and higher (12–14 m/s) wind speeds. Wind direction remains uniformly sampled over $[0°, 360°)$.

Table 4.3: In-distribution and out-of-distribution test MAE for the ΔU target formulation.

Test regime	Wind speed range	MAE [m/s]
In-distribution	6–12 m/s	0.0726
OOD low	4–6 m/s	0.0551
OOD high	12–14 m/s	0.0938

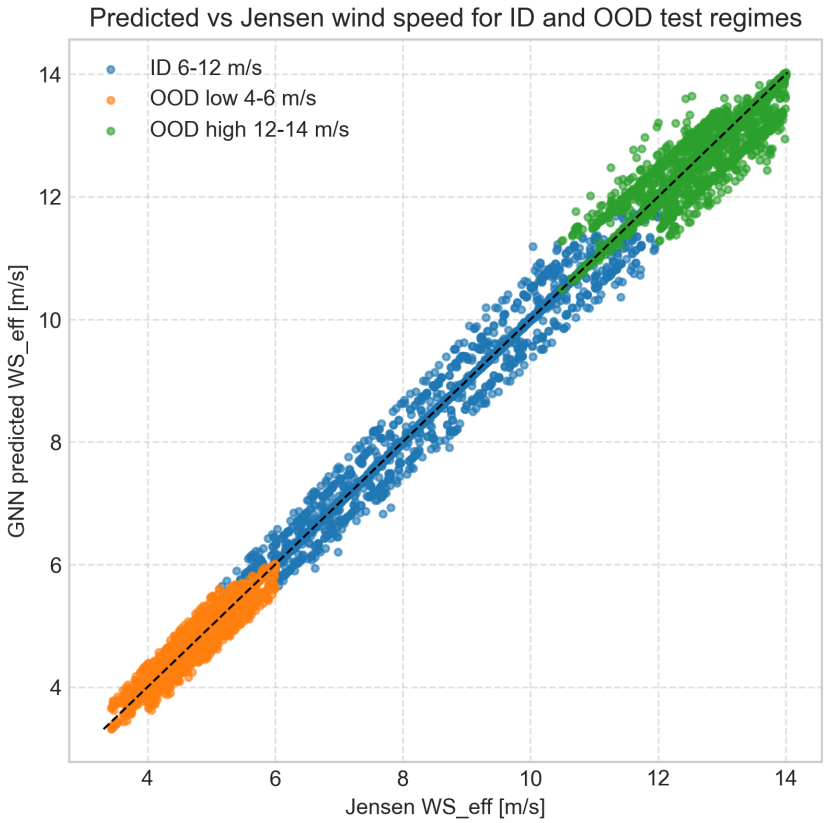


Figure 4.3: Predicted turbine-level wind speeds compared to Jensen reference values for the in-distribution and out-of-distribution test regimes. The dashed line indicates perfect agreement.

The results show that the GNN remains close to the Jensen reference outside the wind-speed range used during training. The low-speed OOD regime even achieves a slightly lower MAE than the in-distribution case. This does not necessarily mean that low-speed extrapolation is easier in a general sense, but is likely caused by the smaller absolute wake deficits at lower inflow speeds, which reduce the magnitude of the absolute prediction error.

The highest error occurs in the high-speed OOD regime, where the MAE increases to 0.0938 m/s. This indicates that extrapolation beyond the upper end of the training range is more challenging, likely because higher inflow speeds lead to larger absolute wake-induced differences. Nevertheless, the error remains below 0.1 m/s, suggesting that the model captures a reasonably general relationship between inflow speed and wake effects rather than only interpolating within the training distribution.

4.1.4.2. Generalisation to unseen turbine spacing

To evaluate whether the GNN generalises beyond a single fixed wind farm geometry, an additional experiment is performed in which the model is trained on multiple 4×4 layouts with different turbine spacings. Five spacing configurations are included in the training data: $6D \times 4D$, $7D \times 5D$, $8D \times 5D$, $9D \times 6D$, and $10D \times 7D$. The model is then evaluated on four test regimes: an in-distribution test set consisting of unseen samples from the training spacing families, an interpolation case with an unseen intermediate spacing of $8.5D \times 5.5D$, a dense extrapolation case with smaller spacing of $5D \times 3D$, and a sparse extrapolation case with larger spacing of $11D \times 7D$. Figure 4.4 shows the considered layouts.



Figure 4.4: Training and test layouts used for the turbine-spacing generalisation experiment. The model is trained on five regular spacing configurations and evaluated on unseen interpolation, dense extrapolation, and sparse extrapolation cases.

To provide a baseline for this experiment, the same type of layout transfer was first tested using a model trained on only a single spacing configuration. When trained on a fixed $7D \times 5D$ layout, the model achieved a low in-distribution MAE of 0.0864 m/s on the same layout, but the error increased to 0.1997 m/s when evaluated on an unseen $9D \times 6D$ layout. This shows that a model trained on one fixed geometry can reproduce that layout well, but does not automatically generalise to different turbine spacings.

The results of the multi-spacing generalisation experiment are summarised in Table 4.4.

Table 4.4: Prediction accuracy for unseen turbine-spacing configurations.

Test regime	Spacing	MAE [m/s]
In-distribution	training spacings	0.1100
OOD interpolation	$8.5D \times 5.5D$	0.1083
OOD dense extrapolation	$5D \times 3D$	0.3279
OOD sparse extrapolation	$11D \times 7D$	0.1052

The results of the multi-spacing generalisation experiment are summarised in Table 4.4.

The model achieves similar prediction errors for the in-distribution, interpolation, and sparse extrapolation cases, with MAE values of 0.1100, 0.1083, and 0.1052 m/s, respectively. This indicates that the GNN can generalise to unseen spacing configurations within and slightly beyond the range represented during

training. The low error for the sparse extrapolation case suggests that the learned relationship between turbine geometry and wake effects remains transferable to layouts with larger turbine spacings.

In contrast, the dense extrapolation case ($5D \times 3D$) results in a substantially higher MAE of 0.3279 m/s. This configuration contains stronger wake interactions and more overlapping wakes than any layout observed during training, making the Jensen mapping more difficult to learn and generalise.

Overall, the results suggest that training on multiple turbine spacings substantially improves geometric generalisation. The model performs well for interpolation and moderate extrapolation, but performance deteriorates when applied to substantially denser layouts that are not represented in the training data.

4.1.4.3. Generalisation to staggered wind farm layouts

To further test geometric generalisation, the model trained on regular 4×4 layouts is evaluated on staggered wind farm layouts. In these layouts, alternating turbine rows are shifted horizontally by half a grid spacing. This changes the relative wake interaction patterns while keeping the number of turbines and the general spacing scale comparable to the regular training layouts.

Three staggered test cases are considered: a staggered $8D \times 5D$ layout, a staggered $8D \times 5D$ layout with small random shifts in turbine positions of up to $0.5D$, and a denser staggered $6D \times 4D$ layout.

Table 4.5: Prediction accuracy for unseen staggered wind farm layouts.

Test regime	Layout	MAE [m/s]
In-distribution	regular training spacings	0.1100
OOD staggered	$8D \times 5D$, half-row offset	0.3819
OOD staggered + jitter	$8D \times 5D$, half-row offset + perturbations	0.3463
OOD dense staggered	$6D \times 4D$, half-row offset	0.4544

The results show a clear performance drop when the model is evaluated on staggered layouts. While the in-distribution test set achieves an MAE of 0.1100 m/s, the staggered $8D \times 5D$ layout increases the error to 0.3819 m/s. The staggered layout with additional random shifts in turbine positions yields a similar MAE of 0.3463 m/s, indicating that the primary challenge is the staggered layout structure itself rather than the exact turbine positions.

The dense staggered $6D \times 4D$ case produces the highest error, with an MAE of 0.4544 m/s. This layout combines an unseen staggered structure with stronger wake overlap and a higher density of wake interactions.

Overall, these results show that geometric generalisation is more limited for changes in layout structure than for changes in regular turbine spacing. While the GNN generalises well across regular grid spacings, staggered layouts introduce wake interaction patterns that are not represented in the training data. This makes accurate generalisation more challenging for the GNN.

4.1.5. Error analysis

The previous sections evaluated the overall prediction accuracy of the GNN. To gain further insight into model behaviour, the prediction error is analysed as a function of downstream wake distance. This analysis is performed on the synthetic 4×4 wind farm, which contains a wider range of wake interaction distances than the original 2×2 layout.

Figure 4.5 shows the median absolute prediction error and interquartile range (25–75 percentile) for different downstream wake distances. The number of wake-interaction samples in each distance bin is also shown.

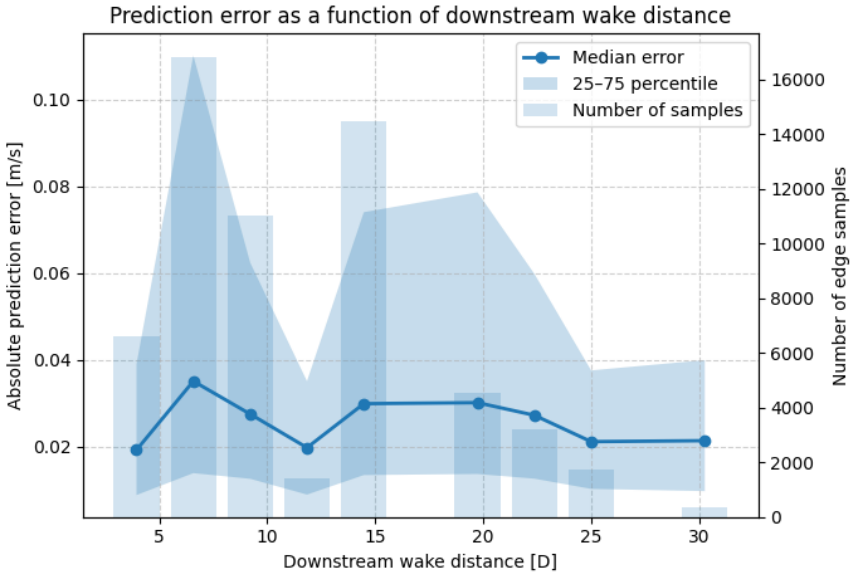


Figure 4.5: Absolute prediction error as a function of downstream wake distance for the synthetic 4×4 wind farm. The solid line shows the median absolute error, the shaded region indicates the interquartile range (25–75 percentile), and the bars show the number of wake-interaction samples per downstream-distance bin.

Overall, the prediction error remains low across the full range of downstream wake distances, with median absolute errors below 0.04 m/s in all bins. No clear trend of increasing error with downstream distance is observed, suggesting that the GNN reproduces both nearby and longer-range wake interactions with comparable accuracy.

Some variation is visible between distance bins, particularly in the interquartile range. However, the number of samples differs substantially between bins, meaning that part of this variability is likely related to differences in the available wake-interaction samples rather than systematic changes in model performance.

These results indicate that the implemented GNN captures the Jensen wake behaviour consistently across a broad range of downstream interaction distances.

4.2. Model validation

Following the verification of the GNN implementation in the previous section, this section validates its predictive performance under LES-derived wind-farm conditions. The GNN is compared with the Jensen wake model and a no-wake baseline to assess its ability to reproduce wake-induced wind-speed effects in a more realistic setting.

The validation first considers a controlled comparison in which the GNN and Jensen model receive the same uniform inflow information. Alternative definitions of the uniform inflow velocity are then considered to assess the sensitivity of both models to this assumption. Finally, a separate experiment evaluates the GNN using more detailed local inflow information.

Together, these experiments validate the predictive accuracy of the GNN and establish a baseline for the more detailed model developments considered in subsequent sections.

4.2.1. Experimental setup

The evaluation is performed on a subset of the WINS50 dataset, focusing on the wind farm *doggerbankSW_east*, consisting of 12 turbines. A total of $T = 1200$ timesteps are selected by aligning turbine-level data with the nearest available LES field data.

For each timestep, both the GNN and the Jensen wake model receive identical input information. A single uniform inflow wind speed is defined based on interpolated 2020 wind field values at turbine locations. Two variants are considered:

- **Mean inflow:** the spatial average wind speed over all turbine locations.

- **Maximum inflow:** the maximum wind speed over all turbine locations.

In addition, a single global wind direction is computed from the 2020 field data using circular averaging. Together with the fixed turbine layout, these quantities form the complete input for both models.

For the controlled benchmark experiments, the GNN is trained under these constrained conditions, while the engineering baseline is evaluated using the NOJ implementation of the Jensen model in PyWake. As in Chapter 4, both an absolute wind-speed formulation, U_i^{2050}, and a wake-difference formulation,

$$\Delta U_i = U_i^{2050} - U_i^{2020}$$

are considered throughout the experiments. For the ΔU_i formulation, absolute wind speeds are reconstructed before evaluation.

Model performance is assessed on a held-out test set using the mean absolute error (MAE) and root mean squared error (RMSE), both expressed in metres per second. In addition to the GNN and Jensen model, a simple no-wake baseline is included.

4.2.2. Results: mean inflow

The results for the baseline configuration, in which the uniform inflow is defined as the spatial average wind speed, are summarised in Table 4.6.

Table 4.6: Performance comparison using mean inflow wind speed.

Model	MAE [m/s]	RMSE [m/s]
GNN (global $U^{2020} \rightarrow U^{2050}$)	0.5162	0.6099
GNN (global $U^{2020} \rightarrow \Delta U$)	0.3692	0.4397
Jensen (NOJ)	0.5606	0.6422
Uniform baseline	0.5856	0.6689

The GNN achieves the lowest prediction error across both target formulations. When predicting the absolute wind speed U^{2050}, the improvement over the Jensen model is modest, and both models outperform the uniform baseline. This indicates that modelling wake interactions improves turbine-level predictions even when only limited inflow information is available.

However, when using the alternative ΔU formulation, the prediction error decreases substantially. This suggests that learning wake-induced differences provides a clearer learning signal, even under simplified inflow conditions, consistent with the findings of the previous section.

The relatively small difference between the GNN and Jensen model for the U^{2050} case suggests that, under mean inflow conditions, both models primarily capture the same large-scale wake structures. In contrast, the improved performance of the ΔU formulation indicates that the GNN is able to more effectively isolate and learn turbine–turbine wake interactions, rather than reproducing the dominant background flow.

To further analyse the spatial structure of the predictions, Figure 4.6 and Figure 4.7 show turbine-level wind speeds and corresponding absolute errors for a representative timestep.

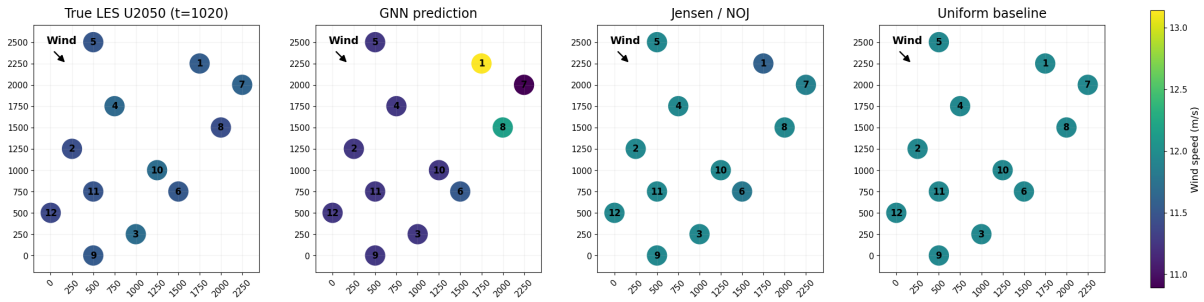


Figure 4.6: Turbine-level wind speed predictions for a representative timestep using mean inflow. Both the GNN and the Jensen model reproduce similar large-scale wake patterns across the wind farm.

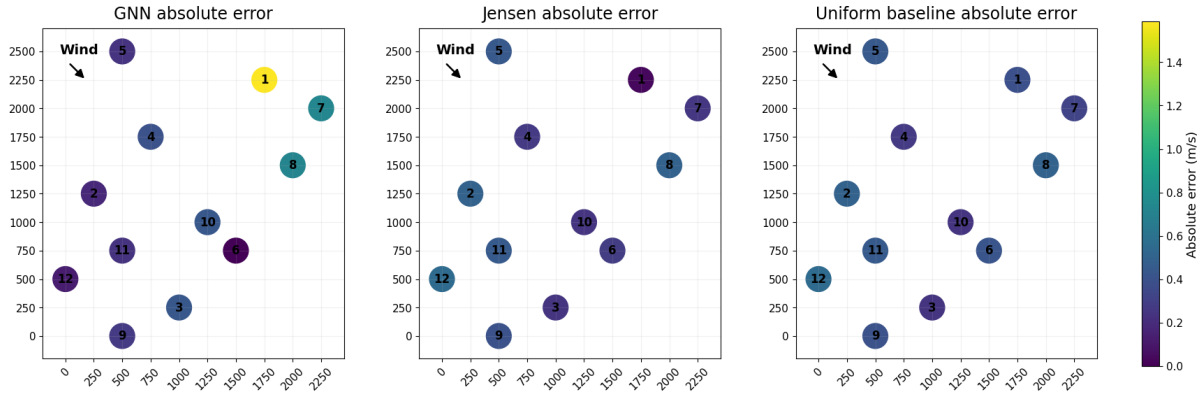


Figure 4.7: Absolute prediction error per turbine for the mean inflow case. Errors are relatively small and evenly distributed, with no strong systematic bias for either model.

The figures show that both the GNN and the Jensen model reproduce the main wake patterns present in the LES reference. The corresponding error distributions are relatively small and spatially uniform, indicating that neither model exhibits a strong systematic bias under mean inflow conditions. That being said, there is one clear outlier in the GNN error at turbine 1, where the model overestimates the wind speed. A possible explanation is that the predefined wake corridor does not capture all relevant upstream wake interactions affecting this turbine, causing the local wake deficit to be underestimated.

4.2.3. Results: maximum inflow

The results for the alternative configuration, in which the uniform inflow is defined as the maximum wind speed at turbine locations, are summarised in Table 4.7.

Table 4.7: Performance comparison using maximum inflow wind speed.

Model	MAE [m/s]	RMSE [m/s]
GNN (global $U^{2020} \rightarrow U^{2050}$)	0.4623	0.6799
GNN (global $U^{2020} \rightarrow \Delta U$)	0.3762	0.4598
Jensen (NOJ)	0.7218	0.8109
Uniform baseline	0.7578	0.8430

The GNN achieves the lowest prediction error across both target formulations. When predicting the absolute wind speed U^{2050}, the model clearly outperforms the Jensen model, whose performance deteriorates significantly under maximum inflow conditions. In particular, the Jensen model exhibits a systematic overestimation of turbine wind speeds across the wind farm.

This behaviour can be explained by the strong dependence of the Jensen model on the prescribed inflow velocity. Since the model scales linearly with the inflow, an overestimation of the free-stream velocity

directly translates into an overestimation of turbine wind speeds. In contrast, the GNN is able to learn non-linear corrections and therefore remains significantly less sensitive to the choice of inflow definition.

When using the alternative ΔU formulation, the prediction error decreases substantially compared to the U^{2050} case. This indicates that learning wake-induced differences improves robustness under inflow misspecification, as the model focuses on deviations relative to a reference state rather than absolute wind-speed levels.

To further analyse the spatial structure of the predictions, Figure 4.8 and Figure 4.9 show turbine-level wind speeds and corresponding absolute errors for a representative timestep.

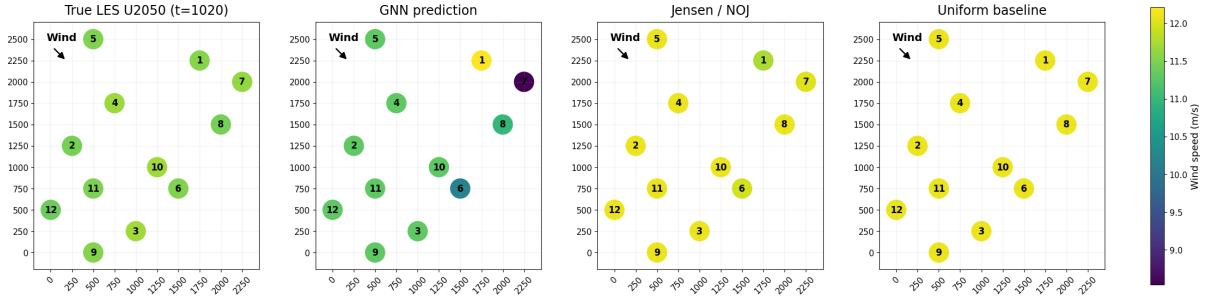


Figure 4.8: Turbine-level wind speed predictions for a representative timestep using maximum inflow. The Jensen model exhibits a clear overestimation compared to the LES reference, while the GNN remains closer to the reference.

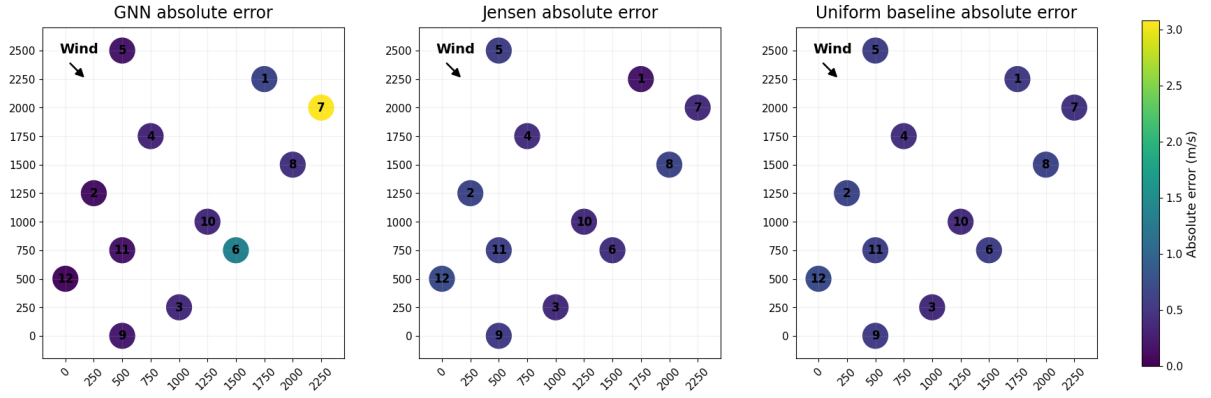


Figure 4.9: Absolute prediction error per turbine for the maximum inflow case. The Jensen model shows significantly larger and more structured errors, whereas the GNN remains relatively robust.

The figures show that the Jensen model overestimates turbine-level wind speeds more strongly than the GNN. This results in larger and more spatially structured errors across the wind farm. In contrast, the GNN predictions remain closer to the LES reference, indicating that the model is less sensitive to the maximum-inflow assumption. Similar to the mean inflow case, there is one clear outlier in the GNN error distribution. However, whereas the largest error under mean inflow occurred at turbine 1, the largest error under maximum inflow occurs at turbine 7, where the model overestimates the wind speed. This suggests that the simplified inflow definition influences which local wake interactions are most difficult for the GNN to capture accurately.

4.2.4. Results: local inflow

In the previous experiments, the GNN was deliberately restricted to a simplified input setting with a single uniform inflow wind speed and a global wind direction. While this enabled a controlled comparison with the Jensen model, it also limited the amount of information available to the GNN.

In this section, the full GNN model as introduced in Chapter 3 is evaluated. In contrast to the baseline configuration, this model incorporates spatially varying inflow information at the turbine level. Specifically, the local 2020 wind speed at each turbine is used as an input feature.

The performance of the extended GNN model under both formulations is summarised in Table 4.8.

Table 4.8: Performance of the extended GNN model with local inflow information, comparing two target formulations.

Model	MAE [m/s]	RMSE [m/s]
GNN (local $U^{2020} \rightarrow U^{2050}$)	0.5239	0.6309
GNN (local $U^{2020} \rightarrow \Delta U$)	0.3880	0.4709

The results show that simply increasing the amount of input information does not automatically improve model performance in this setting. When predicting U^{2050}, the extended GNN performs similarly to the simplified uniform inflow setup, despite having access to spatially varying inflow features.

In contrast, the consistently stronger performance of the ΔU formulation suggests that, in this experimental setting, the formulation of the learning task has a larger influence on predictive performance than the additional local inflow information considered here. This indicates that structuring the prediction problem around wake-induced differences may be more effective than simply increasing input complexity.

4.3. Conclusion

This chapter verified and validated the proposed GNN framework through controlled synthetic experiments and LES-derived wind-farm data. The synthetic verification showed that the implemented graph construction, message-passing architecture, and training pipeline can accurately learn known wake interactions represented by the Jensen model. The model also showed generalisation beyond the training conditions, although performance deteriorated for substantially different and denser wind-farm layouts.

The subsequent validation under LES-derived conditions showed that the GNN can achieve competitive or improved predictive performance compared with the Jensen wake model. Across both stages, predicting the wake-induced wind-speed difference ΔU proved more effective than directly predicting absolute wind speed. The validation further showed that the GNN was relatively robust to simplified inflow assumptions, while the addition of local inflow information alone did not clearly improve performance.

Overall, these results provide confidence in the technical implementation of the GNN and support its further development for more representative offshore wind-farm conditions. The following chapters therefore investigate richer inflow representations, atmospheric information, and alternative representations of turbine interactions.

5

Results: GNN model refinement

5.1. Heterogeneous LES inflow conditions

Chapter 4 verified the implemented GNN in a controlled synthetic setting and subsequently validated its predictive performance against the Jensen wake model under simplified LES-derived conditions. However, this validation only included a limited exploration of spatially varying inflow information, leaving the individual contributions of local wind speed and wind direction unclear.

Offshore wind-farm inflow conditions are spatially heterogeneous due to atmospheric variability and large-scale flow gradients (Platis et al., 2022). This section therefore extends the validation by systematically evaluating four LES-derived inflow representations from the WINS50 dataset: global-global, local-global, global-local, and local-local combinations of wind speed and wind direction.

The objective is to determine whether incorporating heterogeneous inflow information improves turbine-level wake prediction and which components provide the greatest predictive value.

5.1.1. LES-based experimental setup

The experiments in this chapter are performed on a subset of the WINS50 dataset using the wind farm *future_052*, which is only present in the 2050 scenario. This wind farm was selected because of its relatively large size (68 turbines) and limited interference from neighbouring wind farms, making it suitable for analysing internal wake interactions under heterogeneous inflow conditions.

As in the previous chapters, turbine-level data from the 2050 scenario are combined with atmospheric flow-field information from the 2020 reference scenario. However, unlike the earlier benchmark experiments, which relied on nearest-timestep matching between turbine observations and LES flow fields, the present analysis is restricted to timesteps for which LES field data are directly available. Since the atmospheric flow fields are available at hourly resolution, only the corresponding hourly turbine observations are retained, resulting in a dataset of 8760 temporally aligned timesteps over a full year. This avoids temporal mismatch between turbine-level observations and inflow conditions.

The wake-difference formulation is used as the prediction target throughout this chapter:

$$\Delta U_i = U_i^{2050} - U_i^{2020},$$

since Chapters 3 and 4 showed that predicting wake-induced deviations provides a more effective learning signal than directly predicting absolute turbine wind speed.

For each timestep, a directed turbine interaction graph is constructed using the wake-corridor framework introduced in Chapter 3, ensuring that only physically plausible wake interactions are represented.

Four inflow representations are evaluated, differing in whether wind speed and wind direction are represented globally or locally at turbine level. These configurations are summarised in Table 5.1.

Global inflow quantities are computed through spatial aggregation over turbine locations, while local inflow quantities are obtained by interpolating LES flow-field values directly at individual turbine coordinates.

Table 5.1: Overview of the considered inflow representations.

Configuration	Wind speed	Wind direction
Global-global	Global average	Global average
Local-global	Local turbine-level	Global average
Global-local	Global average	Local turbine-level
Local-local	Local turbine-level	Local turbine-level

The dataset is divided chronologically into training, validation, and test subsets using a 70%/15%/15% split to avoid temporal leakage. Model performance is evaluated using mean absolute error (MAE) and root mean squared error (RMSE), both expressed in metres per second.

5.1.2. Results: influence of heterogeneous inflow representations

The predictive performance of the four considered inflow representations is summarised in Table 5.2. All GNN-based configurations outperform the Jensen baseline, indicating that the learned models provide a clear improvement over the classical engineering wake model under heterogeneous LES inflow conditions.

Table 5.2: Predictive performance for different LES-based inflow representations compared with the Jensen baseline.

Configuration	MAE [m/s]	RMSE [m/s]
Jensen baseline	0.8611	1.1216
Global-global	0.6790	0.9004
Local-global	0.6774	0.9035
Global-local	0.6800	0.9046
Local-local	0.6659	0.8893

The results show that incorporating heterogeneous LES-derived inflow information leads to only modest differences in predictive performance between the considered inflow representations. While the local-local configuration achieves the lowest prediction error, all four GNN configurations perform very similarly and substantially outperform the Jensen baseline.

The improvement of the local-local configuration relative to the global-global configuration is approximately 1.9% in terms of MAE. Given the small magnitude of these differences, the results should be interpreted with caution and do not support strong conclusions regarding the superiority of a particular inflow representation. Nevertheless, the results suggest that combining local wind-speed and local wind-direction information provides a slight performance advantage under heterogeneous inflow conditions.

5.1.2.1. Performance across inflow wind-speed regimes

To further analyse under which inflow conditions heterogeneous information is most beneficial, the test-set prediction error is evaluated as a function of the global inflow wind speed. For each test timestep, the turbine-level absolute prediction errors are averaged over all turbines, resulting in a single timestep-level MAE value. These timestep-level errors are subsequently grouped into 2 m/s wind-speed bins based on the global 2020 inflow wind speed. Figure 5.1 shows the resulting mean test MAE per bin for the four inflow configurations, while the lower panel reports the number of test timesteps in each bin.

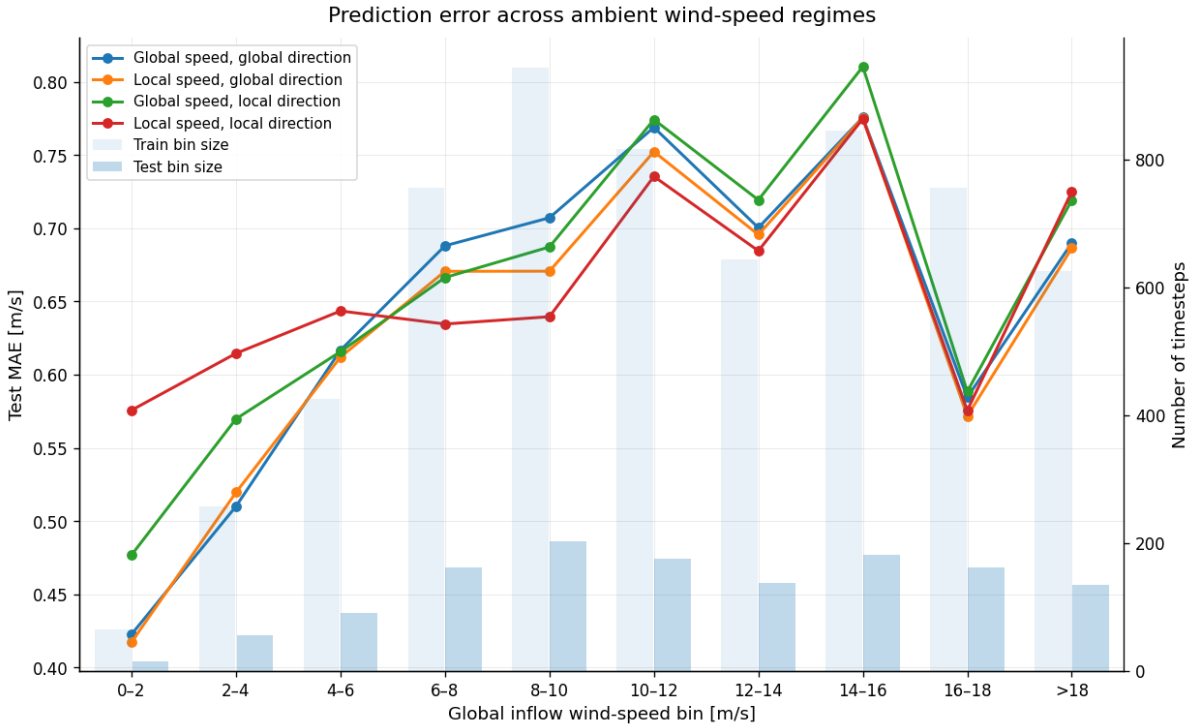


Figure 5.1: Mean test MAE as a function of global inflow wind-speed bin for the four inflow representations. The upper panel shows the mean test MAE, while the lower panel shows the number of test timesteps in each bin.

The wind-speed-bin analysis shows that prediction errors are not uniformly distributed across inflow regimes. For all configurations, the MAE generally increases towards the intermediate and higher wind-speed bins, with the largest errors occurring approximately between 10 and 16 m/s. Although the global representations perform relatively well at low wind speeds, the limited number of samples in the lowest wind-speed bins should be considered when interpreting these results.

The relative performance of the four inflow representations varies across wind-speed regimes, with no single configuration consistently outperforming the others. The local-local configuration performs relatively poorly at lower wind speeds, but achieves the lowest errors across much of the intermediate and highest wind-speed ranges. In contrast, the global-global configuration performs well at low wind speeds, but becomes one of the weaker configurations in parts of the intermediate range.

Overall, the results suggest that the impact of heterogeneous inflow information depends on the inflow regime. However, the differences between the configurations remain relatively small, and therefore strong conclusions regarding the importance of specific local inflow features should be drawn with caution.

5.1.2.2. Spatial distribution of local inflow benefits

While the wind-speed-bin analysis provides insight into the inflow regimes under which heterogeneous information improves predictive performance, it does not reveal whether these improvements are distributed uniformly across the wind farm. To investigate this, a spatial analysis of turbine-level prediction errors is performed.

Because turbine positions are fixed while inflow direction varies over time, spatial error patterns should be interpreted in the context of the directional characteristics of the dataset. Figure 5.2 shows the distribution of global inflow wind directions for the training, validation, and test sets. Due to the chronological dataset split, the directional distributions differ between subsets, with the test set containing a stronger concentration of southerly to south-easterly inflow conditions.

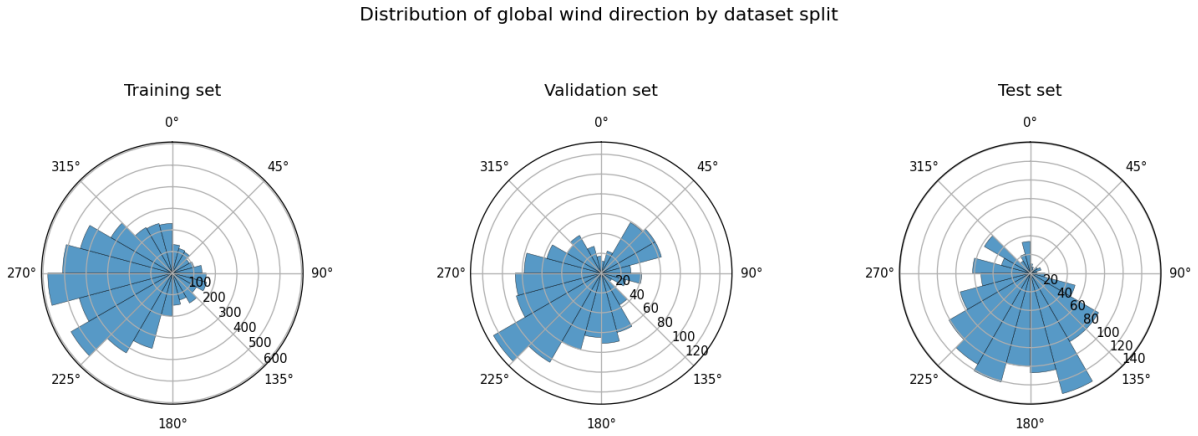


Figure 5.2: Distribution of global inflow wind directions across the training, validation, and test sets. The chronological dataset split leads to non-identical directional distributions between subsets, which influences the wake interaction patterns encountered during evaluation.

To examine whether local inflow information improves prediction performance uniformly across the wind farm, Figure 5.3 shows the turbine-level difference in test MAE between the local-local and global-global configurations. Negative values indicate turbines for which the inclusion of local inflow information reduces prediction error relative to the global baseline.

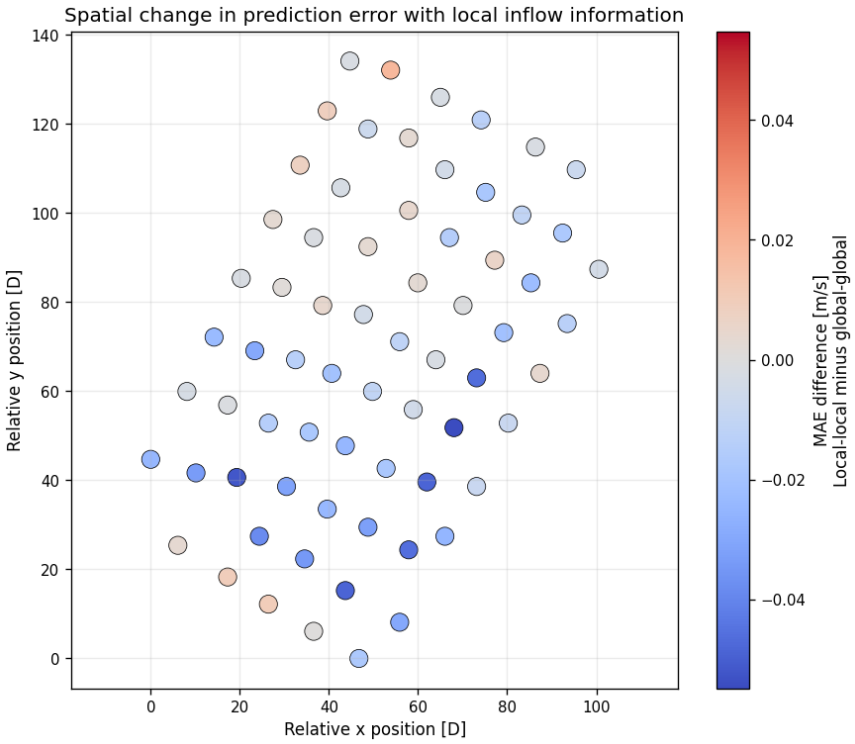


Figure 5.3: Spatial difference in turbine-level test MAE between the local-local and global-global inflow representations. Negative values indicate improved performance when local inflow information is included. Turbine positions are shown relative to the wind-farm extent and normalised by rotor diameter.

The spatial distribution confirms that the benefit of local inflow information is not uniform across the wind farm. Although the turbine-level differences remain relatively small, generally within approximately ±0.05 m/s, a clear spatial pattern is visible. Interestingly, the largest improvements occur primarily in the second row of turbines relative to the dominant inflow directions, rather than at the most upstream or deepest downstream turbines. A possible explanation is that these turbines are exposed to both local inflow variability and the onset of wake interactions, allowing the additional local information to provide the greatest benefit. Further downstream, the performance differences become smaller and occasionally

favour the global-global configuration.

Overall, the results suggest that the benefit of heterogeneous LES-derived inflow information is spatially concentrated rather than uniformly distributed across the wind farm. Local inflow information appears most beneficial in specific regions where it complements the graph-based wake representation, rather than providing a universal improvement across all turbines.

5.2. Influence of turbulence intensity on wake prediction

The previous section showed that heterogeneous LES-derived inflow information can improve turbine-level wake prediction. However, wind speed and wind direction do not fully describe the atmospheric conditions influencing wake behaviour. In particular, turbulence intensity (TI) affects wake mixing and recovery and may therefore provide additional predictive information. TI is a dimensionless measure of wind-speed fluctuations relative to the mean wind speed; in the considered LES data, values range approximately between 0 and 1, with a mean of 0.063.

This section evaluates TI as an additional node feature for the four previously introduced inflow representations, considering both global and local TI information. The objective is to determine whether TI improves predictive performance and whether its value depends on the underlying inflow representation. In addition, the Obukhov length and Richardson number are briefly evaluated as alternative measures of atmospheric stability.

5.2.1. Experimental setup

This section uses the same LES-based experimental framework as the previous section. The wind farm *future_052*, temporally aligned hourly WINS50 dataset, prediction target, graph construction procedure, model architecture, and chronological train/validation/test split are kept unchanged. This ensures that the results remain directly comparable to the previous section, with turbulence intensity as the only additional atmospheric input.

As before, the wake-difference formulation is used as the prediction target:

$$\Delta U_i = U_i^{2050} - U_i^{2020},$$

where U_i^{2050} denotes the effective wind speed at turbine i in the 2050 scenario, and U_i^{2020} represents the corresponding 2020 reference inflow condition.

The four inflow representations from Chapter 5 are retained: global-global, local-global, global-local, and local-local. These differ in whether wind speed and wind direction are represented as global spatial averages or turbine-specific local quantities.

To evaluate the effect of turbulence intensity, three turbulence-intensity input setups are considered for each inflow representation:

- **No-TI:** turbulence intensity is not included as an input feature;
- **Global-TI:** one turbulence-intensity value per timestep is included as an additional node feature for all turbines;
- **Local-TI:** turbine-specific turbulence-intensity values are included as additional node features.

This results in twelve experimental configurations, summarised in Table 5.3.

Table 5.3: Overview of the evaluated inflow and turbulence-intensity configurations.

Inflow configuration	No-TI	Global-TI	Local-TI
Global-global	✓	✓	✓
Local-global	✓	✓	✓
Global-local	✓	✓	✓
Local-local	✓	✓	✓

Without turbulence intensity, each node feature vector consists of wind speed and the sine and cosine representation of wind direction. Including turbulence intensity increases the node feature dimension from three to four by adding either the global or local turbulence-intensity value.

All experiments are repeated using three random seeds to assess variability due to stochastic training effects. Model performance is evaluated using MAE and RMSE, both expressed in meters per second. In the tables and figures below, the reported uncertainty represents the standard deviation across the three random seeds.

5.2.2. Aggregate performance impact of turbulence intensity

The aggregate predictive performance of all twelve configurations is summarised in Table 5.4. Figure 5.4 visualises the corresponding test MAE values.

Table 5.4: Aggregate predictive performance across inflow and turbulence-intensity configurations (mean ± standard deviation over three seeds).

Configuration	TI setup	MAE [m/s]	RMSE [m/s]
Jensen baseline	–	0.8611	1.1216
Global-global	No-TI	0.679 ± 0.015	0.900 ± 0.022
Global-global	Global-TI	0.669 ± 0.006	0.904 ± 0.009
Global-global	Local-TI	0.653 ± 0.005	0.880 ± 0.018
Local-global	No-TI	0.677 ± 0.003	0.903 ± 0.004
Local-global	Global-TI	0.649 ± 0.009	0.878 ± 0.018
Local-global	Local-TI	0.657 ± 0.004	0.885 ± 0.007
Global-local	No-TI	0.680 ± 0.013	0.905 ± 0.016
Global-local	Global-TI	0.650 ± 0.008	0.877 ± 0.023
Global-local	Local-TI	0.652 ± 0.005	0.879 ± 0.006
Local-local	No-TI	0.666 ± 0.008	0.889 ± 0.008
Local-local	Global-TI	0.668 ± 0.016	0.900 ± 0.014
Local-local	Local-TI	0.660 ± 0.014	0.885 ± 0.016

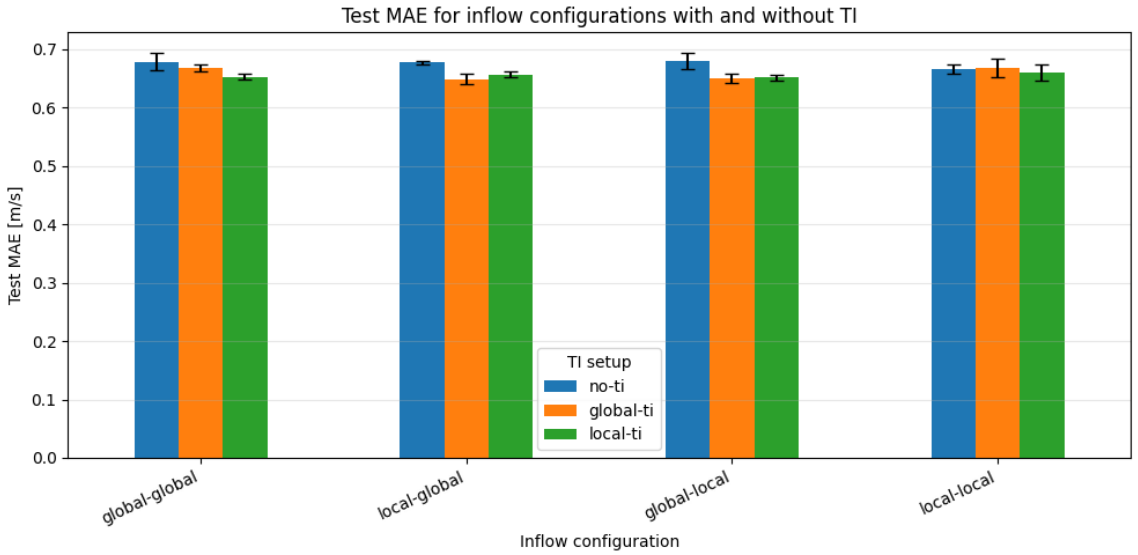


Figure 5.4: Test MAE across all inflow and turbulence-intensity configurations. Error bars indicate standard deviation over three random seeds.

The results show that turbulence intensity generally improves predictive performance, but that the magnitude of the improvement depends on the underlying inflow representation. For the global-global, local-global, and global-local configurations, adding turbulence intensity reduces the mean MAE by approximately 1.5–4.4% relative to the corresponding No-TI baseline. The strongest improvements are obtained for the local-global and global-local configurations with global turbulence intensity, and for the global-global configuration with local turbulence intensity.

The local-local configuration behaves differently. Because this setup already contains both local wind speed and local wind direction, it represents the most detailed inflow description before turbulence intensity is added. In this case, global turbulence intensity slightly worsens performance, while local turbulence intensity provides only a small improvement of less than 1%. This suggests that, once the model already receives detailed local inflow information, turbulence intensity adds limited extra information.

A notable result is that global turbulence intensity often performs as well as, or better than, local turbine-level turbulence intensity. This suggests that the main value of turbulence intensity may be in describing the overall atmospheric mixing conditions across the wind farm, rather than differences in turbulence between individual turbines. Local TI may also introduce additional spatial variability that is not always useful for the current graph formulation, especially when local wind-speed information is already included.

5.2.3. Performance across turbulence-intensity regimes

The aggregate results only show the average performance improvement across the full test set. However, this does not reveal whether the benefit of including turbulence intensity is consistent across all conditions, or concentrated in specific turbulence regimes. To examine this, model performance is analysed across different turbulence-intensity ranges.

Because the local-global configuration showed the strongest and most consistent performance gains, it is used as the representative case for the detailed diagnostic analyses in the remainder of this chapter.

For each test timestep, turbine-level absolute prediction errors are first averaged across all turbines to obtain a single timestep-level MAE. These timestep-level errors are subsequently grouped into turbulence-intensity bins based on the mean local 2020 turbulence intensity across the wind farm. Figure 5.5 shows the result for the local-global inflow representation.

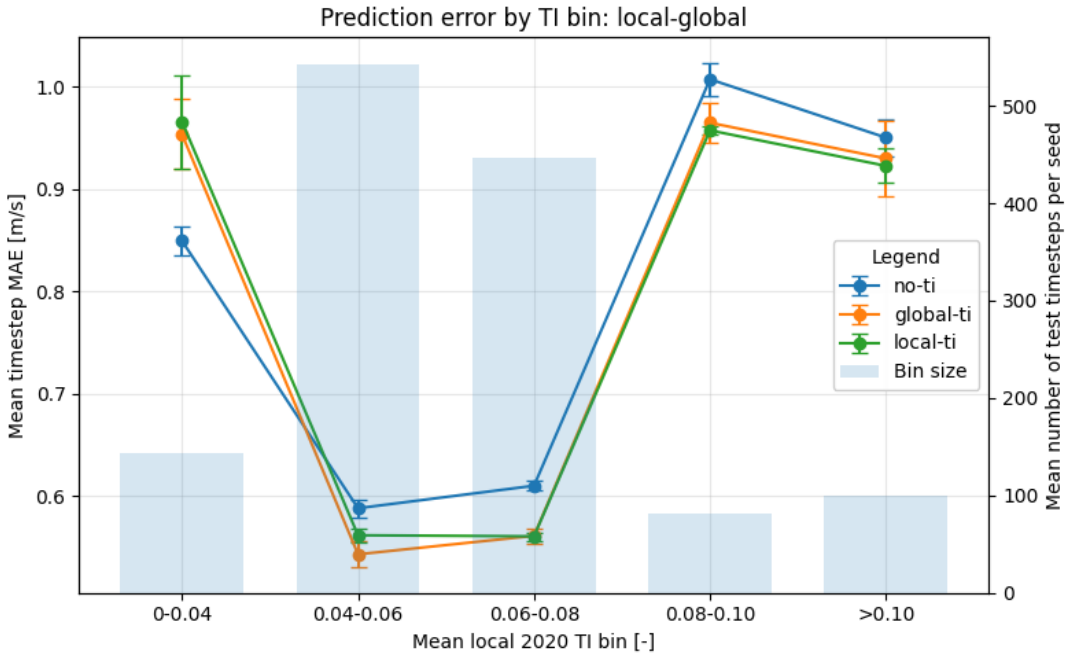


Figure 5.5: Prediction error as a function of turbulence-intensity regime for the local-global inflow representation. Error bars indicate standard deviation across the three random seeds.

The turbulence-intensity bins are unevenly populated, with most test timesteps concentrated in the intermediate turbulence ranges between 0.04 and 0.08, while both the lowest and highest turbulence conditions contain substantially fewer samples. This should be considered when interpreting the results.

A clear pattern nevertheless emerges. Across all turbulence regimes except the lowest (<0.04), including turbulence intensity reduces the prediction error relative to the baseline model without turbulence intensity. The most notable exception is the lowest turbulence regime, where the baseline model performs substantially better than both turbulence-intensity variants. This suggests that explicitly including turbulence intensity does not necessarily improve predictions under low-turbulence conditions, and may even introduce additional complexity that the current model does not use effectively.

All three models also achieve markedly lower errors in the 0.04–0.08 range compared with the other turbulence regimes. Since Figure 5.5 shows that these bins contain substantially more test samples, one possible explanation is that these conditions are simply better represented in the dataset overall, making them easier for the models to learn. Table 5.5 confirms that the training data are similarly concentrated in the same intermediate turbulence ranges.

Table 5.5: Distribution of mean local 2020 turbulence-intensity bins across the training and test sets.

TI bin [-]	Train [%]	Test [%]
0–0.04	26.6	11.0
0.04–0.06	35.6	41.3
0.06–0.08	27.8	33.9
0.08–0.10	4.9	6.2
> 0.10	5.1	7.6

Interestingly, this does not explain the poor performance in the lowest turbulence regime. Although the 0–0.04 bin still represents a substantial share of the training data, prediction errors remain high in the test set. This suggests that the difficulty is not solely caused by limited data availability, but may reflect a genuine limitation of the current model under low-turbulence conditions.

Across most turbulence regimes, the difference between global and local turbulence-intensity inputs remains limited. This supports the interpretation from the aggregate results: turbulence intensity appears to be most useful as a descriptor of the overall atmospheric mixing state, rather than as a highly local turbine-specific correction.

5.2.4. Performance across wind-speed regimes

The predictive value of turbulence intensity is also analysed as a function of inflow wind speed. As in Chapter 5, timestep-level MAE values are grouped into global 2020 wind-speed bins. Figure 5.6 shows the resulting error patterns for the local-global inflow representation.

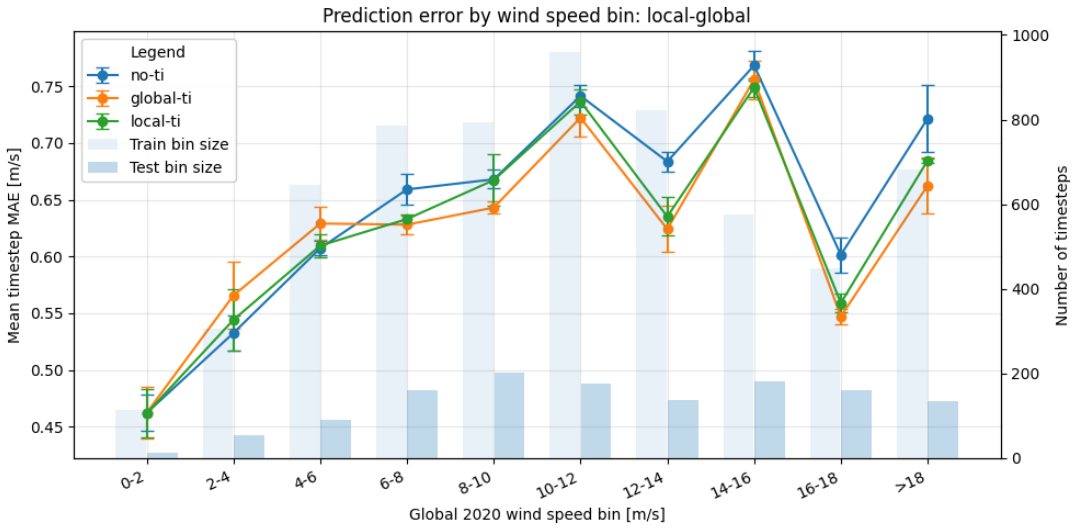


Figure 5.6: Prediction error as a function of inflow wind-speed regime for the local-global inflow representation. Error bars indicate standard deviation across three random seeds.

At low wind speeds, the differences between the three turbulence-intensity setups remain relatively small, indicating that turbulence intensity provides limited additional benefit in this regime.

At moderate and higher wind speeds, the benefit of including turbulence intensity becomes more apparent. From approximately 12 m/s onwards, both turbulence-intensity configurations generally outperform the baseline model without turbulence intensity, with the largest differences occurring in the 12–4 m/s and 16-18 m/s ranges.

Interestingly, the MAE does not increase monotonically with wind speed, but decreases in the 12–14 m/s and 16–18 m/s bins for all three models. This pattern is not fully explained by the distribution of training and test samples across wind-speed bins, indicating that additional factors beyond simple data coverage likely influence model performance.

Nevertheless, the overall pattern indicates that turbulence intensity is most beneficial at moderate to high wind speeds, where wake interactions are expected to be more pronounced, and less consistently beneficial at low wind speeds.

5.2.5. Spatial distribution of turbulence-intensity benefits

To examine whether the benefits of turbulence intensity are distributed uniformly across the wind farm, a spatial analysis is performed using the local-global configuration, which showed the strongest aggregate performance improvement from turbulence-intensity inclusion. For each turbine, the mean absolute error (MAE) is computed over all 1314 unseen test timesteps and averaged across the three random seeds. The turbine-level improvement is defined relative to the corresponding No-TI baseline:

$$\Delta\text{MAE}_i = \text{MAE}_{i,\text{No-TI}} - \text{MAE}_{i,\text{TI}},$$

where positive values indicate reduced prediction error when turbulence intensity is included.

Figure 5.7 shows the resulting spatial improvement maps for both global and local turbulence-intensity inputs.

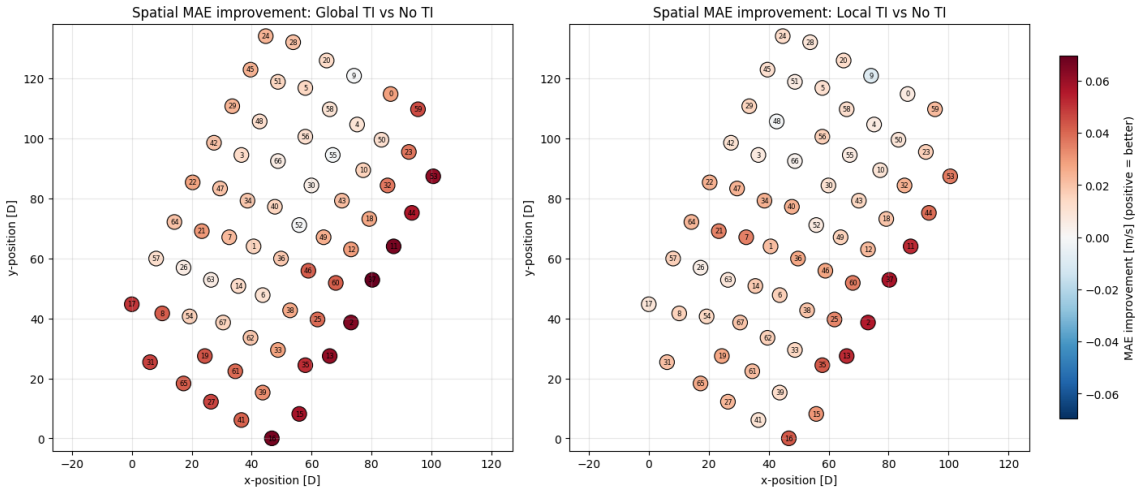


Figure 5.7: Spatial distribution of turbine-level MAE improvement for the local-global inflow representation when turbulence intensity is included. Positive values indicate reduced prediction error relative to the baseline without turbulence intensity. Results are averaged over 1314 unseen test timesteps and three random seeds.

The results show that the predictive benefit of turbulence intensity is clearly not spatially uniform. Some turbines exhibit substantial reductions in prediction error, whereas others show only limited improvement or slight deterioration.

To support interpretation of these spatial patterns, Figure 5.8 shows the distribution of global wind directions in the training, validation, and test sets.

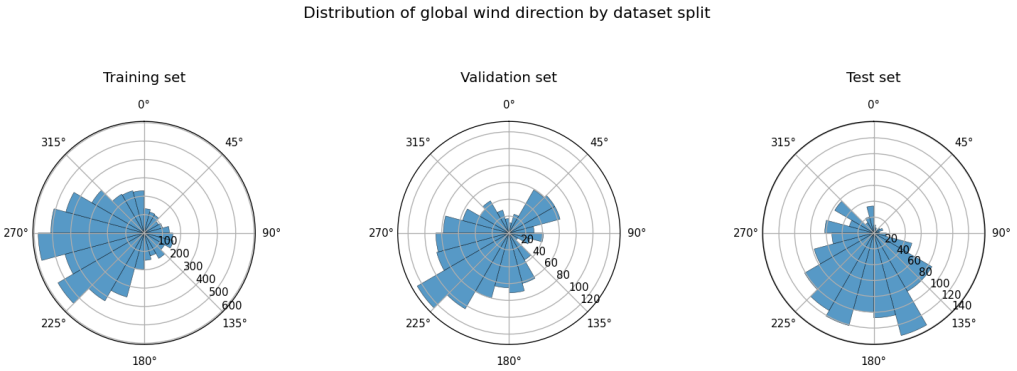


Figure 5.8: Distribution of global wind directions in the training, validation, and test sets. The test-set distribution is used to support interpretation of the spatial distribution of turbulence-intensity-related error reductions.

The test-set wind-direction distribution is clearly non-uniform, meaning that some turbines are more frequently exposed to incoming free-stream conditions, while others are more often positioned in wake-affected regions of the wind farm. Interestingly, some of the strongest performance improvements are observed for turbines located at the inflow boundary of the wind farm. Since these turbines are exposed to free-stream flow and do not experience upstream wake effects, the observed improvements cannot be explained solely by improved modelling of wake recovery.

A plausible explanation is that turbulence intensity helps compensate for information that is not fully captured by the 2020 free-stream wind speed used as input. In particular, the 2020 wind speed may overestimate the actual inflow experienced by front-row turbines due to blockage effects caused by the presence of the wind farm. Turbulence intensity may therefore provide additional information about the atmospheric inflow conditions entering the farm, improving predictions even for turbines that are not directly affected by wakes. This interpretation is supported by the earlier aggregate results,

which showed that global turbulence intensity often performed nearly as well as local turbine-specific turbulence intensity. Together, these findings suggest that turbulence intensity captures broader inflow characteristics that are relevant throughout the wind farm, rather than only local wake-recovery processes.

5.2.5.1. Wake exposure and blockage hypothesis

The spatial analysis suggested that some of the largest turbulence-intensity-related improvements occur for turbines located near the inflow boundary of the wind farm. Since these turbines are not strongly affected by upstream wake interactions, the observed improvements cannot be explained solely by improved modelling of wake recovery. To investigate this observation further, an additional wake-exposure analysis was performed.

For each turbine, the average number of upstream turbines within the graph wake corridor was computed across all test timesteps. This quantity was used as a simple measure of wake exposure. Turbines were subsequently grouped into low, medium, and high wake-exposure categories based on their average number of upstream turbines.

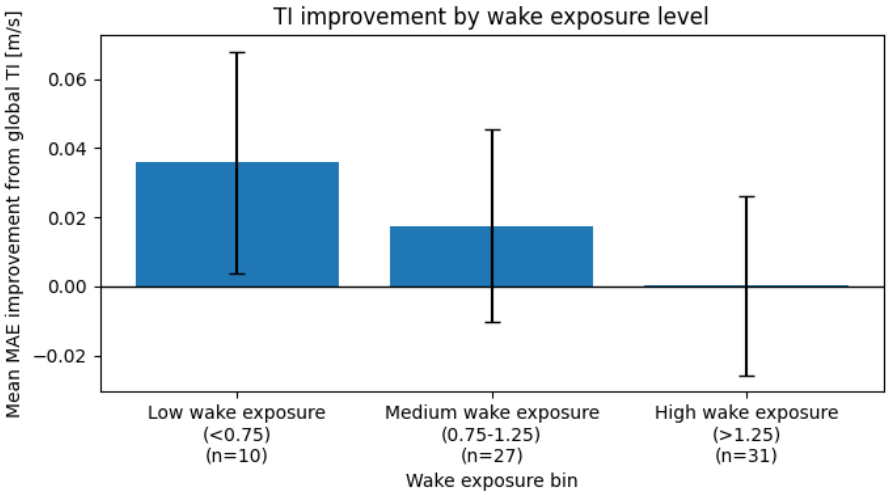


Figure 5.9: Average turbine-level MAE improvement obtained by adding global turbulence intensity to the local-global inflow configuration, grouped by wake exposure. Wake exposure is defined as the average number of upstream turbines within the graph wake corridor over all test timesteps. Positive values indicate reduced prediction error relative to the No-TI baseline.

Figure 5.9 shows a clear decrease in turbulence-intensity-related improvement as wake exposure increases. Turbines with low wake exposure exhibit an average MAE reduction of 0.036 m/s, compared to 0.018 m/s for medium wake exposure and almost no improvement for highly wake-exposed turbines. This trend is further supported by a moderate negative correlation between wake exposure and turbulence-intensity-related improvement ($r = -0.45$). In addition, a separate front-row analysis showed that turbines receiving no upstream wake interactions exhibited an average MAE reduction of 0.061 m/s when turbulence intensity was included, whereas wake-exposed turbines showed a slight average deterioration of 0.008 m/s.

Together, these results indicate that the predictive benefit of turbulence intensity is largest for turbines exposed to incoming free-stream conditions and decreases as wake exposure increases. This behaviour suggests that turbulence intensity primarily provides information about the atmospheric inflow conditions entering the wind farm rather than improving the modelling of wake recovery alone. One possible explanation is that turbulence intensity helps compensate for effects not fully captured by the 2020 free-stream wind speed used as model input, such as wind-farm blockage. Although blockage effects are not directly measured in the present study, the observed relationship between wake exposure and turbulence-intensity benefit provides supporting evidence for a blockage-related interpretation.

5.2.6. Alternative atmospheric stability measures

In addition to TI, atmospheric stability can be characterised using other measures, such as the Obukhov length and Richardson number. Both provide information on whether atmospheric conditions promote or suppress turbulent mixing. The inverse Obukhov length is directly available in WINS50 as an hourly

spatial field and describes the relative influence of buoyancy and wind shear on turbulence (American Meteorological Society, n.d.-b). The Richardson number is not directly available in WINS50, but can be calculated from vertical differences in temperature and wind conditions (American Meteorological Society, n.d.-a). Here, it is calculated using the LES temperature, pressure, wind-speed, and wind-direction fields at 200 and 300 m, surrounding the turbine hub height of 250 m.

To investigate whether these stability measures provide useful information for wake prediction, both local and global representations of inverse Obukhov length and Richardson number are compared with the previously evaluated TI representations. Their predictive performance is presented in Table 5.6.

Table 5.6: Predictive performance for different atmospheric-stability representations compared with the Jensen baseline.

Configuration	MAE [m/s]	RMSE [m/s]
Jensen baseline	0.8611	1.1216
Local-global + No atmospheric stability	0.677	0.903
Local-global + Global-TI	0.649	0.878
Local-global + Local-TI	0.657	0.885
Local-global + Global Obukhov	0.707	0.915
Local-global + Local Obukhov	0.698	0.896
Local-global + Global Richardson	0.651	0.860
Local-global + Local Richardson	0.661	0.875
Local-global + Global Richardson + Global Obukhov + Local-TI	0.6386	0.8454

The results show that both Obukhov representations perform worse than the model without additional atmospheric-stability information. Among the individual stability measures, global TI achieves the lowest MAE, while global Richardson achieves the lowest RMSE. Interestingly, combining global Richardson, global Obukhov, and local TI results in the best overall performance, suggesting that these measures may provide complementary information when used together.

5.3. Influence of graph construction choices and attention mechanisms

The proposed GNN relies on a physics-inspired graph structure in which turbine interactions are restricted to a predefined wake corridor. While this provides an interpretable and computationally efficient representation, the sensitivity of predictive performance to these connectivity assumptions remains unclear.

In addition, the baseline model uses edge-conditioned convolution (NNConv), which incorporates relative turbine geometry through edge features. Alternative message-passing mechanisms, such as graph attention, may reduce dependence on predefined graph structure by learning the relative importance of neighbouring turbines.

This section therefore evaluates the influence of graph construction choices on predictive performance and computational cost, and compares the baseline NNConv architecture with a Graph Attention Network (GAT). The objective is to determine whether denser connectivity or attention-based message passing provides sufficient benefits to justify the additional model complexity.

5.3.1. Experimental setup

All experiments in this chapter use the same LES-derived dataset, training procedure, and evaluation framework as the best-performing configuration identified in Chapter 5.2. This ensures that observed differences can be attributed directly to graph construction or message passing choices.

The baseline configuration uses local wind speed, global wind direction, and local turbulence intensity

as node features, combined with the NNConv architecture introduced in Chapter 3. Graph connectivity follows the original wake corridor definition, with directed edges created for turbines within a downstream interaction range of $45D$ and a lateral corridor width of $\pm 3D$.

Two experiment groups are considered. First, the wake corridor geometry is modified to evaluate sensitivity to graph construction assumptions. Second, the NNConv architecture is replaced by GAT to assess whether attention-based message passing improves predictive performance.

For all experiments, the same chronological training, validation, and test split, normalisation procedure, optimisation settings, and evaluation metrics are maintained. Predictive performance is assessed using denormalised turbine-level MAE, while computational cost is evaluated through training runtime comparisons.

5.3.2. Sensitivity to graph construction choices

To evaluate sensitivity to graph construction assumptions, the baseline wake corridor was systematically modified while keeping all other model components unchanged. Four alternative corridor-based configurations were tested: a narrower lateral corridor ($\pm 2D$), a wider corridor ($\pm 5D$), a shorter downstream range ($30D$), and a longer downstream range ($60D$). In addition, a fully connected directed graph was included as an exploratory extreme case, removing all physics-based connectivity filtering.

The corridor-based experiments were repeated across three random seeds, while the fully connected case was evaluated for a single representative seed due to its substantially higher computational cost.

Figure 5.10 shows the resulting predictive performance.

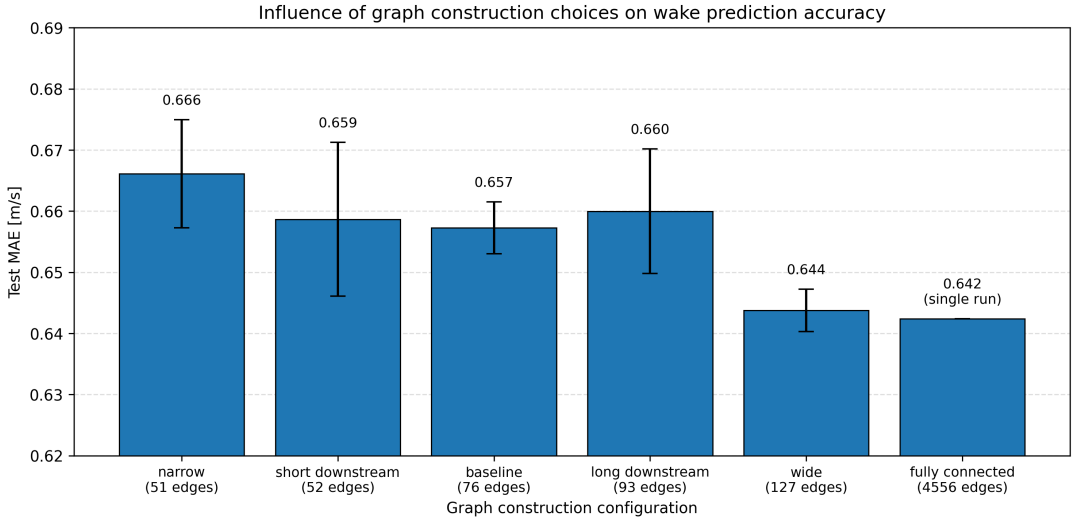


Figure 5.10: Influence of graph construction choices on predictive accuracy. Error bars indicate variation across three random seeds. The fully connected configuration represents a single exploratory run.

Restricting the graph too aggressively reduced predictive performance. The narrow wake corridor increased average test MAE from 0.657 m/s in the baseline model to 0.666 m/s, suggesting that excluding lateral interactions removes useful predictive information.

Changing the downstream interaction range had only limited impact. Reducing the cutoff to $30D$ produced near-baseline performance (0.659 m/s), while extending it to $60D$ slightly increased error (0.660 m/s). This suggests that, within the present setup, the baseline downstream range already captures most relevant wake interactions.

The strongest improvement was obtained by widening the lateral corridor to $\pm 5D$, reducing average test MAE to 0.644 m/s. This suggests that the original baseline graph may be somewhat too restrictive. However, while the widened corridor consistently performed best across seeds, the absolute performance differences remained modest.

The fully connected exploratory graph achieved a similar MAE of 0.642 m/s, despite increasing graph density from 76 to 4556 edges per timestep. Since this result is based on a single seed and required

substantially higher computational cost, it should be interpreted cautiously.

Overall, the results suggest moderate sensitivity to graph construction assumptions. While graph design influences predictive performance, the differences remain limited, with less than 4% variation between the best and worst multi-seed configurations. Among the robust experiments, the widened wake corridor provided the strongest and most consistent performance, suggesting that modestly increasing graph density improves predictive accuracy, while denser graph representations offer diminishing returns.

5.3.3. Attention-based message passing

The previous section showed that predictive performance depends only moderately on graph construction assumptions. This raises the question of whether a more flexible message passing architecture can reduce this dependence further.

To evaluate this, the baseline NNConv architecture was replaced by a graph attention network (GAT), while keeping the dataset, node features, training procedure, and graph construction configurations unchanged. Unlike NNConv, which explicitly incorporates edge geometry through edge-conditioned transformations, GAT dynamically weights neighbouring turbines based on learned attention scores.

Figure 5.11 compares predictive performance across the tested graph configurations for both architectures.

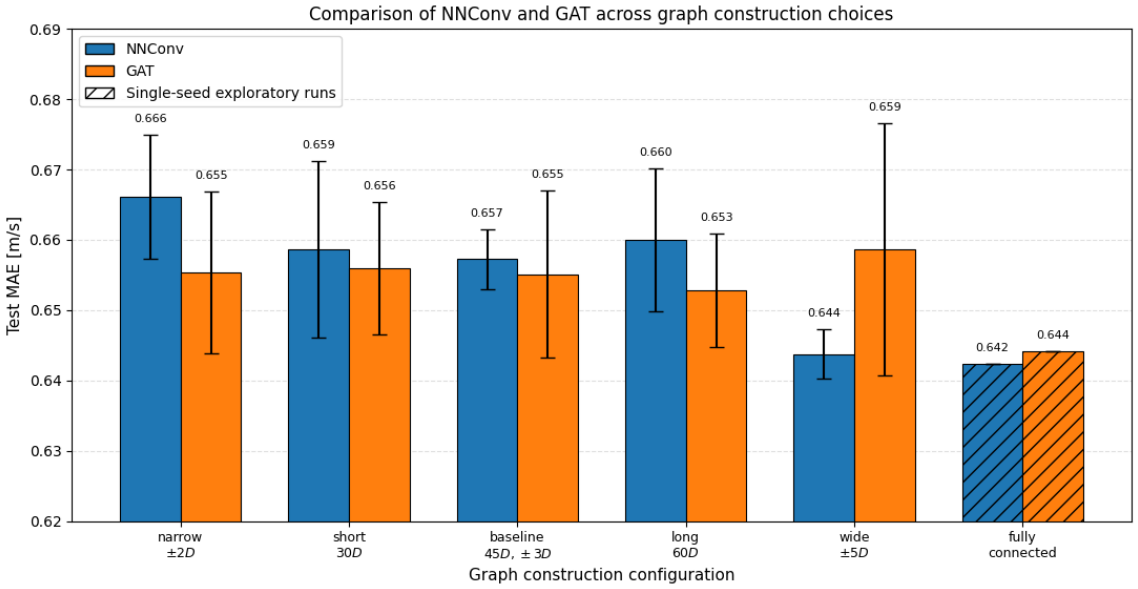


Figure 5.11: Comparison of predictive performance between NNConv and GAT across graph construction choices. Error bars indicate variation across three random seeds. Hatched bars represent exploratory single-seed fully connected runs.

Replacing NNConv with GAT did not improve predictive performance. Across the corridor-based configurations, GAT achieved average test MAEs between 0.653 and 0.659 m/s, while NNConv ranged between 0.644 and 0.666 m/s.

The main difference lies in sensitivity to graph design. NNConv performance varied more clearly with graph construction, with restrictive configurations degrading performance and the widened corridor producing the strongest results. In contrast, GAT remained comparatively stable across all corridor-based graphs, suggesting that attention reduces dependence on manually specified connectivity assumptions by dynamically downweighting less relevant neighbours.

However, this increased flexibility did not translate into better accuracy. The best GAT configuration remained above the best NNConv result. This may reflect the stronger physics-informed inductive bias of NNConv for the present wake prediction problem, although differences in architectural tuning or implementation choices may also have contributed.

The fully connected exploratory experiments produced nearly identical predictive performance, with test MAEs of 0.642 m/s for NNConv and 0.644 m/s for GAT. Since these results are based on single-seed runs only, they should be interpreted cautiously.

Overall, the results suggest that the tested GAT implementation improves robustness to graph construction choices, but does not outperform NNConv under the present experimental setup. The computational implications are examined in the following section.

5.3.4. Computational costs

While predictive accuracy is the primary evaluation criterion, computational efficiency is also an important practical consideration when selecting graph construction strategies and message passing architectures. Increasing graph density or adopting more flexible architectures may improve predictive performance, but can substantially increase training cost and reduce scalability.

Figure 5.12 compares the average training runtimes of the tested NNConv and GAT configurations.

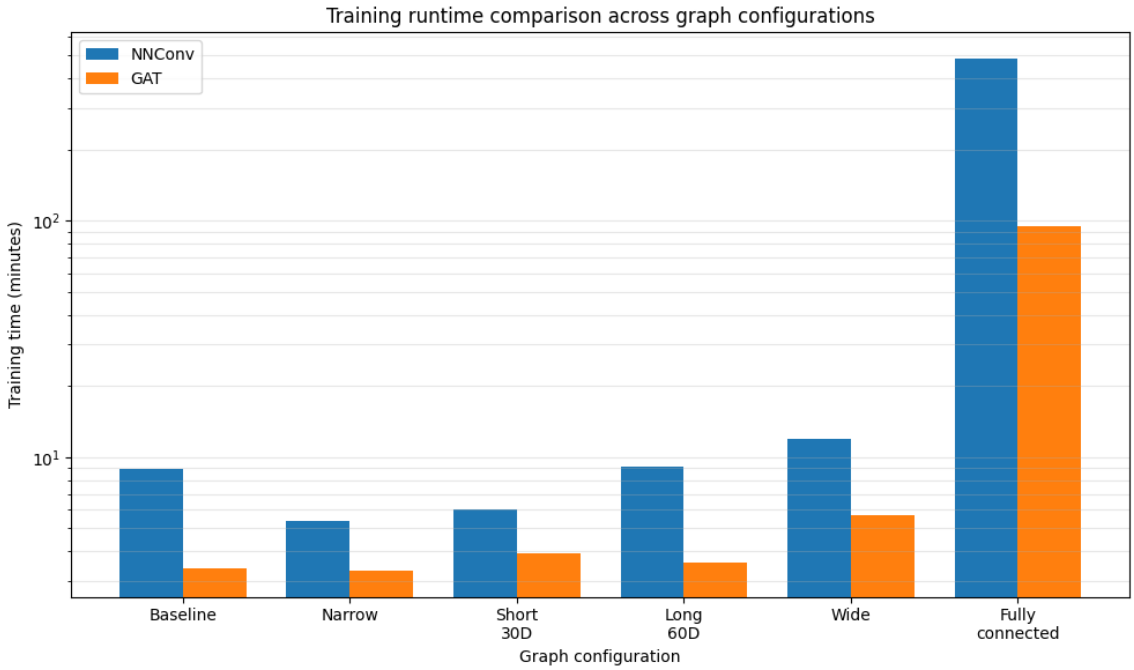


Figure 5.12: Average training runtime for NNConv and GAT across graph construction configurations. Fully connected results represent single exploratory runs.

Table 5.7: Computational cost comparison of representative model configurations.

Configuration	Avg. edges	Training time (min)	Test MAE (m/s)
NNConv baseline	76	8.9	0.657
NNConv wide	127	11.9	0.644
NNConv fully connected	4556	488.2	0.642
GAT baseline	76	3.4	0.655
GAT wide	127	5.7	0.659
GAT fully connected	4556	95.0	0.644

For the corridor-based graph configurations, computational costs remained modest for both architectures. NNConv required approximately 5–12 minutes depending on graph density, while GAT consistently trained more than twice as fast, reducing runtime from an average of 8.9 to 3.4 minutes for the baseline configuration.

NNConv training time increased more strongly with graph density, reflecting its use of edge-specific transformations during message passing. GAT showed weaker scaling, since attention computes scalar

importance weights rather than full edge-conditioned transformations.

This difference became particularly clear in the fully connected exploratory experiments. Increasing graph density from 76 to 4556 edges increased NNConv training time from approximately 9 minutes to over 488 minutes, while GAT required only 95 minutes for the same dense graph.

However, this computational advantage did not translate into improved predictive performance. Fully connected GAT achieved a test MAE of 0.644 m/s, nearly identical to fully connected NNConv (0.642 m/s), while both provided only negligible improvement over the much cheaper widened NNConv configuration (0.644 m/s).

This suggests that fully connected graph representations are computationally inefficient for the present problem. The widened physics-inspired NNConv configuration provides the most favourable trade-off between predictive performance and computational cost.

5.4. Conclusion

This chapter investigated how inflow representation, turbulence intensity, atmospheric stability, and graph design influence the predictive performance of the proposed GNN. The results show that increasing the level of input detail does not necessarily lead to substantial improvements. Local inflow information provided modest benefits that varied across operating conditions and turbine locations, while turbulence intensity generally improved predictions when the available inflow representation was less detailed. Its limited additional value for the most detailed inflow configurations suggests that part of this information is already captured by local wind-speed and wind-direction inputs. Although the individual atmospheric-stability measures provided mixed results, combining Richardson number, Obukhov length, and turbulence intensity achieved the best overall predictive performance.

The graph experiments similarly showed that additional model complexity does not automatically improve predictive accuracy. A moderately widened physics-inspired graph improved performance, whereas fully connected graphs provided negligible additional benefit at substantially higher computational cost. The tested GAT architecture was less sensitive to graph construction and computationally cheaper, but did not outperform NNConv in predictive accuracy.

Overall, the results favour a relatively simple and physically motivated model configuration rather than maximising input or graph complexity. These findings provide the basis for selecting the GNN configuration used in the subsequent cross-farm generalisation analysis.

6

Results: Cross-farm validation and financial case study

6.1. Introduction

Wind-farm wake models are ultimately intended to support the design and assessment of new offshore wind-farm projects (Shakoor et al., 2016). As a result, predictive performance on previously unseen wind-farm layouts is of particular practical importance. A model that performs well only on layouts encountered during training may have limited value when applied to new developments with different geometric characteristics.

Generalisation across wind-farm layouts is especially relevant for data-driven approaches, which may implicitly learn layout-specific patterns rather than transferable wake-interaction behaviour (S. Li et al., 2023; Schøler et al., 2025). The ability to transfer learned wake-interaction behaviour to previously unseen wind farms is therefore an important consideration for the practical deployment of machine-learning-based wake models.

This chapter investigates the extent to which training on multiple wind-farm layouts improves out-of-distribution generalisation to a previously unseen wind farm. Model performance on the unseen test case is compared against the Jensen wake model as a conventional engineering baseline.

To illustrate the practical relevance of the results, a simplified case study is included that links wake-prediction accuracy to project valuation.

6.2. Cross-farm generalisation experiment

To evaluate cross-farm generalisation, a multi-farm training strategy was adopted in which a single GNN model was trained on data from multiple offshore wind farms and evaluated on the previously unseen wind farm `future_052`. The test wind farm was excluded entirely from the training and validation data of the multi-farm model, allowing its performance to be evaluated on a wind-farm layout not encountered during model fitting.

The training dataset consists of all wind farms that are present exclusively in the 2050 WINS50 scenario and contain between 50 and 150 turbines. Applying these selection criteria resulted in a total of 25 training wind farms. The lower limit excludes very small wind farms that are less representative of large-scale offshore developments, while the upper limit avoids substantially larger wind farms that would considerably increase computational cost and training time. Furthermore, the selected turbine-count range remains broadly comparable to that of the test wind farm. A complete overview of all selected wind farms and their corresponding turbine counts is provided in Appendix D.

For this experiment, the 5-minute turbine data were aggregated to hourly means before alignment with the hourly LES field data. Each hourly turbine value therefore represents the mean of the twelve 5-minute observations within the corresponding hour. This hourly-mean alignment is used throughout the remainder of this thesis. A random 85/15 train-validation split was applied independently to each training wind farm.

For consistency with the findings of Chapter 5, the local-global configuration with local turbulence intensity was selected as the model input representation. The graph structure was constructed using the wake corridor identified in the previous chapter, with downstream and crosswind interaction limits of 45D and 5D respectively. Model performance on the unseen test farm is additionally compared against the Jensen wake model as a conventional engineering baseline.

6.3. Generalisation to an unseen wind-farm layout

Table 6.1 presents the predictive performance of the multi-farm GNN and the Jensen wake model on the previously unseen `future_052` wind farm. Since this wind farm was excluded entirely from the training and validation data of the multi-farm model, the results evaluate the model's ability to generalise to a wind-farm layout not encountered during model fitting.

Table 6.1: Performance comparison on the unseen `future_052` wind farm.

Model	MAE [m/s]	RMSE [m/s]	Bias [m/s]
Multi-farm GNN	0.7888	1.0755	-0.0000
Jensen (NOJ)	1.0859	1.4616	0.7286

The multi-farm GNN achieved a test MAE of 0.7888 m/s and an RMSE of 1.0755 m/s. In comparison, the Jensen wake model obtained a higher MAE of 1.0859 m/s and an RMSE of 1.4616 m/s. Relative to the engineering baseline, this corresponds to reductions of approximately 27% and 26% in MAE and RMSE, respectively.

Furthermore, the GNN exhibited virtually zero prediction bias, whereas the Jensen model systematically overpredicted wind speeds, resulting in a positive bias of 0.7286 m/s. This indicates that the GNN not only reduces the magnitude of prediction errors but also provides substantially more balanced wind-speed estimates across the wind farm.

It should be noted that the hourly-mean alignment introduced in this chapter affects the reported error magnitudes. For example, the Jensen MAE increases from 0.8611 m/s under the previous alignment to 1.0859 m/s under the hourly-mean alignment. A similar effect may apply to the GNN. Therefore, the multi-farm GNN error cannot be directly compared with the GNN errors reported in the previous chapters.

6.3.1. Computational performance

Table 6.2 compares the computational time required to evaluate the complete `future_052` test set. Runtime statistics are based on 100 repeated runs.

Table 6.2: Computational runtime comparison for the future_052 test set (8760 hourly timesteps and 68 turbines). Runtime statistics are based on 100 repeated runs.

Model	Mean runtime [s]	Std. [s]
Multi-farm GNN (inference only)	13.92	1.66
Multi-farm GNN (graph construction + inference)	54.99	1.79
Jensen (NOJ)	244.77	6.72

The inference-only runtime of the GNN is approximately 14 seconds, while the complete workflow including graph construction requires approximately 55 seconds. In comparison, the Jensen model requires approximately 245 seconds. Even when graph construction is included, the GNN remains approximately 4.5 times faster than the Jensen model while simultaneously reducing the MAE by 27% and the RMSE by 26%.

Training the multi-farm GNN on the 25 training wind farms required approximately 1357 minutes (22.6 hours). However, this cost is incurred only once and is therefore excluded from the runtime comparison, which focuses on prediction time for new wind-farm data.

6.3.2. Spatial error distribution

While Table 6.1 demonstrates that the multi-farm GNN achieves lower overall prediction errors than the Jensen wake model, it does not reveal how these errors are distributed throughout the wind farm. Figure 6.1 therefore presents the turbine-level MAE for both models across the `future_052` layout.

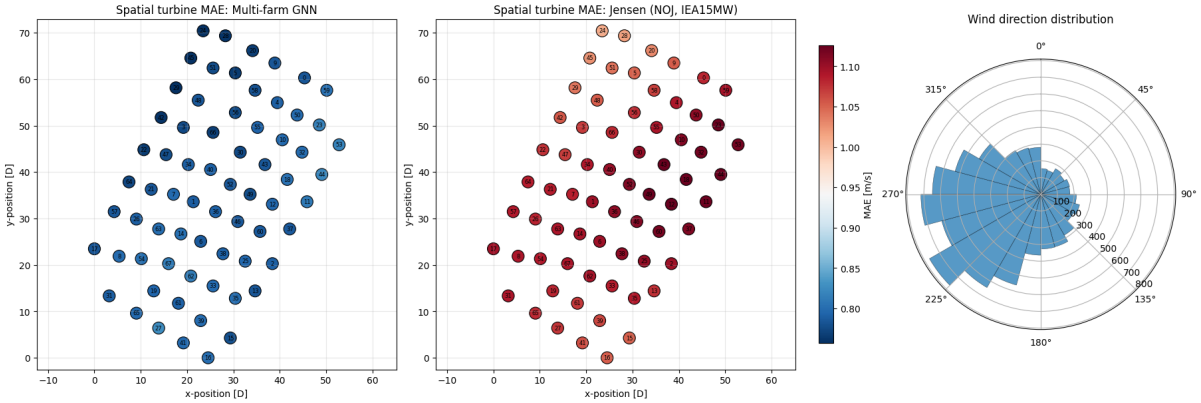


Figure 6.1: Spatial distribution of turbine-level MAE for the multi-farm GNN and Jensen wake model on the unseen `future_052` wind farm. The wind-direction distribution for the full test period is shown on the right.

Several observations can be made from Figure 6.1. First, the GNN consistently achieves lower prediction errors across all turbines in the wind farm. Whereas the Jensen model exhibits turbine-level MAE values that frequently exceed 1.1 m/s, the majority of GNN errors remain below 0.9 m/s. This indicates that the improvement reported in Table 6.1 is not driven by a small subset of turbines, but is instead distributed across the wind farm.

Second, the spatial error patterns suggest that the GNN is able to capture layout-dependent wake interactions more effectively than the engineering wake model. Although both models exhibit some variation in performance between turbines, the Jensen model shows substantially larger errors throughout the farm, particularly for turbines located deeper within the array where wake interactions become more complex.

Finally, the wind-direction distribution shows that although south-westerly to westerly inflow directions are dominant during the test period, a broad range of other wind directions is also represented. The spatial error patterns therefore reflect model performance across multiple wake configurations rather than being restricted to a single prevailing inflow direction.

6.4. Illustrative valuation case study

The previous sections showed that the multi-farm GNN achieves substantially lower prediction errors than the Jensen wake model on a previously unseen wind-farm layout. However, statistical metrics such as MAE and RMSE do not directly indicate whether these improvements are practically relevant for wind-farm development. To provide additional context, an illustrative valuation case study is therefore performed.

Wake models are commonly used to estimate future energy production during wind-farm planning and assessment (Meyer Forsting et al., 2021). Since project revenues depend directly on electricity generation, errors in wake-loss predictions can lead to errors in estimated project value. Even relatively small differences in annual energy production may translate into substantial financial differences over the lifetime of a large offshore wind farm.

The purpose of this case study is not to provide a detailed economic assessment, but rather to illustrate the potential practical implications of improved wake-model accuracy. The LES results are treated as the reference, while the multi-farm GNN and Jensen model are used to generate alternative estimates of annual energy production. These production estimates are then translated into indicative revenue and project-value estimates using a simplified set of economic assumptions.

Comparing the resulting valuation errors provides an indication of how improvements in wake-model accuracy may affect planning and investment decisions in offshore wind-farm development.

6.4.1. Methodology

The case study uses the unseen `future_052` wind farm as a representative example. The objective is to show how wake-model prediction errors can propagate from wind-speed estimates to energy-production estimates and, finally, to indicative economic outcomes.

For each model, turbine-level wind-speed estimates are obtained for all turbines and all hourly timesteps. The LES turbine wind speeds are treated as the reference. The GNN and Jensen model estimates are then compared against this reference. Since the GNN predicts the wind-speed difference relative to the 2020 inflow conditions, the predicted absolute turbine wind speed is first reconstructed as

$$U_{2050,pred} = U_{2020} + \Delta U_{pred}. \tag{6.1}$$

The resulting turbine-level wind speeds are converted into electrical power using the tabulated power curve of the IEA 15 MW Offshore Reference Wind Turbine. This curve has a rated power of 15 MW and covers wind speeds from 3 to 25 m/s (Allen et al., 2020; Gaertner et al., 2020). Annual energy production is then calculated by summing power production over all turbines and all timesteps:

$$AEP = \sum_{t=1}^{T} \sum_{i=1}^{N} P(U_{i,t}) \Delta t, \tag{6.2}$$

where $P(U_{i,t})$ is the power output of turbine (i) at timestep (t), (N=68) is the number of turbines, (T=8760) is the number of hourly timesteps, and $\Delta t = 1$ hour.

The AEP estimate obtained from the LES data is treated as the reference value. The GNN and Jensen AEP estimates are compared against this reference to quantify how wake-model prediction errors affect estimated annual energy production.

To illustrate the financial implications, the AEP estimates are converted into annual revenue estimates using a fixed electricity price:

$$Revenue = AEP \cdot P_{el}, \tag{6.3}$$

where (AEP) denotes annual energy production and P_{el} is the assumed electricity price.

Finally, annual revenues are converted into discounted lifetime revenue estimates using a fixed project lifetime and discount rate:

$$PV = Revenue \cdot \frac{1 - (1 + r)^{-n}}{r}, \tag{6.4}$$

where PV is the present value of lifetime revenues, r is the discount rate, and n is the project lifetime in years (Finance Formulas, 2024). This calculation accounts for the fact that revenues received in future years have a lower present value than revenues received today.

The calculation therefore follows the sequence:

Wind speed → Power → AEP → Annual revenue → Discounted lifetime revenue

The economic assumptions used in the case study are summarised in Table 6.3. An electricity price of 60 EUR/MWh is assumed as a representative long-term value for illustrative purposes, consistent with projected offshore wind electricity values reported by DNV (Norwegian Offshore Wind, 2024). The turbine power curve and project lifetime are based on the IEA 15 MW Offshore Reference Wind Turbine (Allen et al., 2020; Gaertner et al., 2020). A discount rate of 7% is adopted, which is broadly consistent with values commonly used in offshore wind economic assessments (Freyman & Arup, 2023). Although these assumptions are simplified, they provide a realistic basis for illustrating how wake-model prediction errors can propagate into economic outcomes.

Table 6.3: Assumptions used in the illustrative valuation case study.

Parameter	Value	Source
Power curve	IEA 15 MW reference turbine	(Gaertner et al., 2020)
Electricity price	60 EUR/MWh	(Norwegian Offshore Wind, 2024)
Project lifetime	25 years	(Gaertner et al., 2020)
Discount rate	7%	(Freyman & Arup, 2023)

6.4.2. Results

Table 6.4 presents the resulting AEP, annual revenue, and discounted lifetime revenue estimates for the LES reference, the multi-farm GNN, and the Jensen wake model.

Table 6.4: Case-study results for `future_052` using the IEA 15 MW reference turbine power curve.

Model	AEP [GWh]	AEP error [%]	Revenue [M EUR/yr]	PV [B EUR]	PV error [M EUR]
LES reference	5318.1	0.0	319.1	3.719	0.0
GNN	5350.8	0.6	321.0	3.741	22.8
Jensen	5966.0	12.2	358.0	4.171	453.0

The LES reference results in an estimated AEP of 5318.1 GWh/year and an annual revenue of 319.1 million EUR/year. Using the simplified discounted-cash-flow calculation, this corresponds to a discounted lifetime revenue of 3.719 billion EUR.

The GNN slightly overestimates AEP by 0.6%, corresponding to an annual revenue error of approximately 2.0 million EUR. When discounted at 7% over the assumed 25-year project lifetime, this results in a present-value revenue error of approximately 22.8 million EUR.

The Jensen model shows a substantially larger overestimation. Its AEP estimate is 12.2% higher than the LES reference, corresponding to an annual revenue error of 38.9 million EUR/year and a discounted lifetime revenue error of 453.0 million EUR. These results illustrate how relatively small differences in wake-model accuracy can translate into substantial differences in projected revenues and valuation estimates when propagated through the turbine power curve and accumulated over the lifetime of an offshore wind project.

6.4.3. Conclusion

The case study shows that improved wind-speed prediction can translate into substantially more accurate energy-production and revenue estimates. The GNN's near-zero wind-speed bias results in a considerably smaller AEP error than the Jensen model, whose positive bias is amplified through the nonlinear turbine power curve.

These results should not be interpreted as a full financial valuation, as factors such as capital and operational expenditure, financing, electricity-price uncertainty, curtailment, and degradation are not considered. The calculated values therefore represent discounted lifetime revenue estimates rather than complete project values.

Nevertheless, the results demonstrate the practical relevance of wake-model accuracy. Differences in predictive performance propagate into economically meaningful differences in estimated long-term revenue, highlighting the importance of accurate wake modelling for offshore wind investment and tendering decisions.

7

Conclusion and discussion

7.1. Key findings and interpretation

This thesis investigated whether a Graph Neural Network trained directly on high-fidelity LES data can improve offshore wind-farm wake prediction while maintaining the computational efficiency required for practical application. The main findings are summarised below by explicitly answering each of the five sub-questions, followed by an answer to the main research question.

SQ1: How can a Graph Neural Network be designed to predict wake-induced wind-speed deficits?

A turbine-interaction GNN can predict wake-induced wind-speed deficits by representing turbines as nodes and wake interactions as directed edges, with message passing used to learn how upstream turbines influence downstream conditions. The verification experiments demonstrated that this representation can successfully learn turbine–turbine wake interactions, accurately reproducing the behaviour of the Jensen wake model under controlled conditions. The model also generalised well to unseen wind-speed ranges and regular turbine spacings, although performance deteriorated for structurally different staggered layouts. This shows that the proposed graph representation is capable of learning wake relationships, while its ability to generalise depends on the diversity of interaction patterns represented during training.

SQ2: How can a GNN represent heterogeneous flow conditions within large offshore wind farms?

Heterogeneous flow conditions can be represented by incorporating local turbine-level wind information as node features. However, the LES experiments showed that increasing the amount of spatially resolved information does not automatically improve predictive performance. Different combinations of local and global wind speed and wind direction produced relatively similar overall errors, while their relative performance varied across wind-speed regimes. Local wind speed provided useful spatial information, whereas local wind direction offered limited additional benefit. The results therefore indicate that heterogeneous inflow information can be incorporated effectively into the graph representation, but that additional spatial detail does not necessarily translate directly into greater predictive accuracy.

SQ3: How can atmospheric stability be incorporated through turbulence intensity in GNN-based wake prediction?

Atmospheric conditions can be incorporated by adding turbulence intensity as an additional node feature. Including TI generally improved predictive accuracy, while global TI often performed as well as local turbine-level TI, indicating limited additional value from its spatially resolved representation. Further analysis suggested that TI was particularly useful in improving the representation of blockage effects. Additional experiments showed that the Richardson number achieved performance comparable to TI, whereas Obukhov length alone did not improve prediction accuracy. Combining Richardson number, Obukhov length, and TI resulted in the best overall performance, suggesting that different atmospheric-stability indicators may contain complementary predictive information.

SQ4: How do graph construction choices and attention mechanisms affect wake prediction accuracy?

Graph construction affects the balance between predictive accuracy and computational efficiency. Widening the lateral interaction corridor improved prediction accuracy, indicating that relevant turbine

interactions extend beyond narrowly defined downstream corridors. A fully connected graph provided only a small additional improvement in accuracy while increasing computational cost substantially. Furthermore, the attention-based architecture did not outperform the edge-conditioned NNConv model. These results show that physically informed graph construction provides an effective way of restricting interactions while retaining relevant wake information, and that increasing graph density or architectural complexity does not necessarily lead to better wake predictions.

SQ5: How computationally efficient and scalable is the GNN-based wake model compared with a traditional engineering wake model?

The computational experiments showed that the cost of the GNN increases with wind-farm size, particularly during training as the number of turbines and graph interactions increases. Predictive accuracy nevertheless remained relatively stable across the synthetic scaling experiments. Once trained, however, the GNN provided computationally efficient evaluation. In the final multi-farm experiment, including graph construction, the GNN evaluated the complete *future_052* test set approximately 4.5 times faster than the Jensen implementation. These results demonstrate that the main computational limitation of the proposed approach lies in its upfront training cost, whereas repeated evaluation with a trained model remains computationally efficient as wind-farm size increases and can outperform the implemented engineering-model baseline in evaluation speed.

Main research question: To what extent can a Graph Neural Network, trained on LES data, improve the accuracy of offshore wind-farm wake predictions while maintaining computational efficiency compared to traditional engineering wake models?

The results demonstrate that an LES-trained turbine-interaction GNN can substantially improve wake-prediction accuracy relative to the Jensen engineering wake model while retaining computationally efficient evaluation. When trained on 25 wind farms and evaluated on the completely excluded *future_052* layout, the GNN achieved an MAE of 0.7888 m/s compared with 1.0859 m/s for Jensen, corresponding to an approximately 27% reduction in prediction error. The GNN also exhibited virtually no mean prediction bias. At the same time, including graph construction, evaluation of the complete test set was approximately 4.5 times faster than the Jensen implementation. The proposed approach therefore demonstrates that high-fidelity LES information can be transferred to a computationally efficient turbine-interaction model that, within the evaluated case, outperforms the traditional engineering-model baseline in both predictive accuracy and evaluation speed.

The demonstrated performance on an unseen wind-farm layout is particularly important because it shows that the learned wake relationships are transferable beyond the layouts used for training. However, this result should not yet be interpreted as evidence of universal cross-farm generalisation. The final evaluation was performed on one completely excluded wind farm, and broader validation across additional layouts, meteorological years, turbine configurations, engineering wake models, and operational measurements is still required. Within the scope of this thesis, however, the results provide strong evidence that LES-trained GNNs can combine improved predictive accuracy with computationally efficient evaluation, supporting their potential as reusable surrogate wake models for offshore wind-farm applications.

7.2. Scientific contribution and implications

The main scientific contribution of this thesis is the development and evaluation of a turbine-interaction GNN trained directly on high-fidelity LES data. Previous turbine-interaction graph-based wake models have primarily relied on engineering-model outputs, lower-fidelity simulations, or operational data. This thesis demonstrates that wake behaviour learned directly from LES can be incorporated into a graph-based surrogate model and transferred to a wind-farm layout excluded from model training.

A second contribution is the systematic evaluation of modelling choices that are particularly relevant when constructing LES-trained graph models, including the representation of heterogeneous inflow conditions, atmospheric stability, and turbine connectivity. Rather than identifying a universally optimal configuration, the experiments clarify which forms of additional physical information provide useful predictive information and where additional model complexity offers limited benefit. In particular, the results show that greater spatial detail, denser graph connectivity, and more flexible message-passing mechanisms do not necessarily translate into improved predictive performance.

Finally, the cross-farm framework demonstrates the potential for LES-trained surrogate models to move beyond layout-specific applications. By training on multiple wind farms and successfully evaluating

the model on a completely excluded layout, this thesis demonstrates that wake relationships learned from computationally expensive LES simulations can transfer to a new wind-farm configuration within the evaluated setting. This positions the proposed approach between high-fidelity simulation and conventional engineering wake models: high-fidelity LES is used to learn wake behaviour during model development, while the trained graph model provides comparatively low-cost evaluation for new wind-farm conditions. Scientifically, this suggests that information generated through computationally expensive LES simulations can be reused across wind-farm configurations rather than remaining confined to the layouts for which the simulations were originally performed.

7.3. Practical implications

The results of this thesis have practical implications for offshore wind-farm assessment and decision-making. Wake-model predictions influence expected energy production and therefore feed directly into project valuation, investment decisions, wind-farm design, and tender strategies. The valuation case study demonstrates the potential magnitude of this effect. Relative to the LES reference, the GNN overestimated AEP by approximately 0.6%, corresponding to a discounted lifetime revenue difference of approximately EUR 22.8 million, whereas the Jensen model overestimated AEP by approximately 15.1%, corresponding to approximately EUR 560.2 million under the assumptions of the case study. This illustrates that differences in wake-model accuracy can translate into economically significant differences in the information available to decision-makers.

The computational performance of the GNN is particularly relevant for applications requiring repeated wake evaluations, such as wind-farm layout optimization and scenario analysis. These applications require many alternative turbine configurations or operating conditions to be evaluated, making the trade-off between predictive accuracy and computational cost practically important. Once trained, the proposed GNN can evaluate new wind-farm conditions rapidly while retaining the higher predictive accuracy demonstrated in the cross-farm experiment. LES-trained GNNs could therefore complement existing engineering models by transferring information from computationally expensive high-fidelity simulations into a model suitable for repeated evaluations, without requiring a new LES simulation for every considered configuration.

More broadly, offshore wind development takes place within a system in which technical, economic, policy, and market decisions are strongly interconnected. Developers must make long-term investment and design decisions based on uncertain estimates of future energy production and revenues, while investors assess the associated financial risks and governments determine the conditions under which offshore wind capacity is developed through policies and competitive tender procedures. Wake-model uncertainty therefore does not remain confined to a technical modelling exercise: it can propagate into economic assessments and subsequently influence decisions made by multiple actors. Improving the accuracy of wake predictions can reduce one source of uncertainty within this wider decision-making process and provide a more reliable technical basis for comparing alternative projects, layouts, and investment strategies.

At the same time, improved wake predictions do not remove the broader uncertainties surrounding offshore wind development. Electricity prices, financing conditions, policy changes, and future wind conditions remain important determinants of project viability. The practical value of an improved wake model should therefore be understood as improving one component of the information available to decision-makers rather than eliminating uncertainty from the overall system. Its computational efficiency is important in this context because it allows more layouts and scenarios to be evaluated, potentially enabling decision-makers to explore a wider range of alternatives and better understand the consequences of uncertainty before committing to long-term decisions.

These findings should not yet be interpreted as demonstrating an industry-ready replacement for established wake-modelling tools, as broader validation against operational measurements and additional wind farms is still required. However, within the scope of this thesis, the results demonstrate that an LES-trained GNN can substantially improve wake-prediction accuracy while providing computationally efficient evaluation. The valuation case further shows that these improvements can translate into substantially more accurate estimates of energy production and project value. The proposed approach therefore has the potential to provide decision-makers with higher-quality information at comparatively low computational cost, supporting more informed design, investment, and tender decisions. In this way, LES-trained GNNs could contribute not only to improved wake modelling, but ultimately to more robust

decision-making in the large-scale development of offshore wind energy.

7.4. Limitations

Several limitations should be considered when interpreting the results of this thesis.

- **Limited data coverage and validation.** The LES experiments are based on a single meteorological year from the WINS50 dataset, while many of the detailed model-development experiments are performed on a single wind farm. Although the final experiment extends training to 25 wind farms, only one wind farm is used for the final out-of-distribution evaluation. The extent to which this demonstrated cross-farm transfer generalises to other meteorological years, turbine configurations, and substantially different wind-farm layouts therefore remains uncertain. Furthermore, the GNN is trained and evaluated against LES data rather than operational wind-farm measurements. The results demonstrate the model's ability to learn and transfer LES-derived wake behaviour, but do not yet establish whether the same improvements are achieved under real offshore wind-farm conditions. Finally, the temporal alignment between the hourly LES fields and 5-minute turbine data was not consistent across all experiments, with hourly averaging introduced only in the later chapters. Consequently, absolute error metrics cannot always be directly compared across chapters, as part of the observed differences may result from the change in temporal alignment rather than from the investigated modelling choices alone.
- **Model development and hyperparameter selection.** Several modelling choices, including the initial wake-corridor dimensions, learning rate, and other training parameters, were selected based on preliminary experimentation rather than through a systematic and fully documented hyperparameter optimisation procedure. Additional testing during model development indicated that alternative learning-rate settings could further improve predictive performance. The reported GNN performance should therefore not be interpreted as the optimal performance achievable with the proposed architecture. Moreover, much of the model development and configuration selection was performed using *future_052*. Although the final multi-farm experiment excluded this wind farm entirely from training and validation, earlier design decisions were informed by experiments on this same farm. The final cross-farm evaluation should therefore not be interpreted as a completely independent assessment of all modelling choices.
- **Experimental and benchmark uncertainty.** Not all experiments were repeated across multiple random seeds, and computationally expensive configurations were evaluated with fewer repetitions. Neural-network performance can vary with random initialisation and optimisation, meaning that small differences between configurations should be interpreted cautiously. Furthermore, all experiments were run on a standard laptop, meaning that the reported runtimes should be interpreted with some caution. In addition, Jensen/NOJ is used as the principal engineering-model benchmark. The results therefore demonstrate an improvement relative to this specific baseline and do not establish superiority over all available engineering wake models.
- **Fixed assumptions in graph construction.** Graph connectivity is determined using predefined wake-corridor thresholds and a single global wind direction, even when local wind-direction information is provided as node input. The resulting topology therefore cannot adapt directly to local flow variations, turbulence conditions, or atmospheric stability. The graph formulation also assumes a fixed reference rotor diameter of 126 m and does not explicitly account for differences in turbine dimensions. Although the graph-sensitivity experiments showed that the model is relatively robust to reasonable changes in connectivity, these assumptions constrain the physical representation of wake interactions.

7.5. Future research

The findings and limitations of this thesis suggest several directions for future research.

- **Broader cross-farm and operational validation.** The most important next step is to determine whether the demonstrated cross-farm performance extends to a broader range of wind farms and to real offshore operating conditions. Future studies should evaluate LES-trained GNNs on multiple wind farms that are kept entirely separate from both model development and training, covering a wider range of farm sizes, turbine spacings, layout structures, turbine types, and meteorological conditions. In addition, validation against operational wind-farm measurements, such as SCADA data, would determine whether the improvements demonstrated relative to WINS50 LES data

also transfer to real offshore wind-farm conditions. Such validation would provide a substantially stronger basis for assessing the generalisability and practical applicability of LES-trained GNN wake models.

- **Improved representation of atmospheric conditions.** The observed benefit of turbulence intensity suggests that atmospheric conditions contain useful information beyond wind speed and direction. Additional exploratory experiments with atmospheric-stability indicators showed that the Richardson number performed comparably to TI, while Obukhov length alone did not improve predictive performance. Interestingly, combining Richardson number, Obukhov length, and TI resulted in the best overall performance, suggesting that these variables may provide complementary information. However, these atmospheric-stability indicators were investigated only to a limited extent. More systematic research into alternative stability representations and model configurations may reveal additional predictive value. Future work could also investigate whether atmospheric conditions should influence not only node features but also wake-interaction strength or graph connectivity.
- **Adaptive and physics-informed graph construction.** Rather than relying on fixed wake-corridor dimensions and a single global wind direction, future GNNs could use graph structures that adapt to local wind direction, turbulence conditions, atmospheric stability, and turbine characteristics. This could preserve the computational advantages of sparse graphs while providing a more physically representative description of wake interactions. Such adaptive graph structures may be particularly relevant when transferring the model to wind farms with substantially different layouts or turbine configurations.
- **Model optimisation and broader benchmarking.** Future research should include systematic hyperparameter optimisation, repeated training across more random seeds, and comparisons with additional engineering wake models. Additional testing during model development indicated that alternative learning-rate settings could further improve predictive performance, suggesting that the results reported in this thesis may not yet represent the full predictive potential of the proposed architecture. Comparisons with more advanced engineering wake models would also establish whether the demonstrated accuracy advantage extends beyond the Jensen/NOJ baseline.

Overall, future research should determine whether the accuracy and computational advantages demonstrated in this thesis can be maintained across a broader range of wind farms and, importantly, under real-world operating conditions. Broader cross-farm and operational validation would provide the next step towards practical application. Nevertheless, this thesis demonstrates that wake relationships learned from computationally expensive LES simulations can be transferred to an unseen wind-farm layout while outperforming the Jensen baseline in both predictive accuracy and evaluation speed. This establishes LES-trained turbine-interaction GNNs as a promising approach for combining high-fidelity wake information with the computational efficiency required for large-scale offshore wind-farm assessment.

Bibliography

Ainslie, J. F. (1988). Calculating the flowfield in the wake of wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, *27*(1–3), 213–224. https://doi.org/https://doi.org/10.1016/0167-6105(88)90037-2

Allen, C., Viselli, A., Dagher, H., Goupee, A., Gaertner, E., Abbas, N., Hall, M., & Barter, G. (2020). *Definition of the UMaine VolturnUS-S reference platform developed for the IEA Wind 15-megawatt offshore reference wind turbine* (tech. rep.). International Energy Agency. https://www.nrel.gov/docs/fy20osti/76773.pdf

American Meteorological Society. (n.d.-a). Gradient richardson number [Accessed: 2026-08-30]. https://glossary.ametsoc.org/wiki/gradient-richardson-number/

American Meteorological Society. (n.d.-b). Obukhov length [Accessed: 2026-08-30]. https://glossary.ametsoc.org/wiki/obukhov-length/

Bastankhah, M., & Porté-Agel, F. (2014). A new analytical model for wind-turbine wakes. *Renewable Energy*, *70*, 116–123. https://doi.org/https://doi.org/10.1016/j.renene.2014.01.002

Baungaard, M., van der Laan, M. P., & Kelly, M. (2022). RANS modeling of a single wind turbine wake in the unstable surface layer. *Wind Energy Science*, *7*(2), 783–800. https://doi.org/10.5194/wes-7-783-2022

Bentsen, L. Ø., Warakagoda, N. D., Stenbro, R., & Engelstad, P. (2022). Wind park power prediction: Attention-based graph networks and deep learning to capture wake losses. *Journal of Physics: Conference Series*, *2265*(2), 022035. https://doi.org/https://doi.org/10.1088/1742-6596/2265/2/022035

Bleeg, J. (2020). A graph neural network surrogate model for the prediction of turbine interaction loss. *Journal of Physics: Conference Series*, *1618*(6), 062054. https://doi.org/https://doi.org/10.1088/1742-6596/1618/6/062054

Clean Energy Wire. (2025, May 27). *Major offshore wind projects in germany at stake as companies seek opt-out – media report*. Retrieved May 28, 2026, from https://www.cleanenergywire.org/news/major-offshore-wind-projects-germany-stake-companies-seek-opt-out-media-report

Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, *503*, 92–108. https://doi.org/https://doi.org/10.1016/j.neucom.2022.06.111

Duthé, G., Santos, F. d. N., Abdallah, I., Réthoré, P.-E., Weijtjens, W., Chatzi, E., & Devriendt, C. (2023). Local flow and loads estimation on wake-affected wind turbines using graph neural networks and pywake. *Journal of Physics: Conference Series*, *2505*(1), 012014. https://doi.org/10.1088/1742-6596/2505/1/012014

Duthé, G., Santos, F. d. N., Abdallah, I., Weijtjens, W., Devriendt, C., & Chatzi, E. (2024). Flexible multi-fidelity framework for load estimation of wind farms through graph neural networks and transfer learning. *Data-Centric Engineering*, *5*, e29. https://doi.org/10.1017/dce.2024.35

Finance Formulas. (2024). *Present value of annuity*. Retrieved June 3, 2026, from https://financeformulas.net/Present_Value_of_Annuity.html

Freyman, M., & Arup. (2023). *Levelised cost of energy for floating offshore wind* (tech. rep.). Department for Energy Security and Net Zero (DESNZ), UK Government. Retrieved June 3, 2026, from https://assets.publishing.service.gov.uk/media/655371f7019bd600149f1ffa/floating-offshore-wind-lcoe-report_.pdf

Gaertner, E., Rinker, J., Sethuraman, L., Zahle, F., Anderson, B., Barter, G., Abbas, N., Meng, F., Bortolotti, P., Skrzypinski, W., Scott, G., Feil, R., Bredmose, H., Dykes, K., Sheilds, M., Allen, C., & Viselli, A. (2020). *Definition of the IEA 15-megawatt offshore reference wind turbine* (tech. rep.). International Energy Agency. https://www.nrel.gov/docs/fy20osti/75698.pdf

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message passing for quantum chemistry. *Proceedings of the 34th International Conference on Machine Learning*, *70*, 1263–1272. https://doi.org/https://doi.org/10.48550/arXiv.1704.01212

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In G. Gordon, D. Dunson, & M. Dudík (Eds.), *Proceedings of the fourteenth international conference on artificial intelligence and*

statistics (aistats) (pp. 315–323, Vol. 15). PMLR. https://proceedings.mlr.press/v15/glorot11a.html

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* [http://www.deeplearningbook.org]. MIT Press.

Hou, G., Zhang, F., Huang, C., & Huang, T. (2026). Multivariate modeling on wake-affected wind farms by two-stage hybrid graph neural network. *Applied Energy*, *402*, 127018. https://doi.org/https://doi.org/10.1016/j.apenergy.2025.127018

Howland, M. F., & Dabiri, J. O. (2019). Wind farm modeling with interpretable physics-informed machine learning. *Energies*, *12*(14), 2716. https://doi.org/10.3390/en12142716

Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd ed.). OTexts. https://otexts.com/fpp3/

IEA. (2019). *Offshore wind outlook 2019* (tech. rep.). IEA. Retrieved May 28, 2026, from https://www.iea.org/reports/offshore-wind-outlook-2019

International Energy Agency. (2025). *The netherlands 2024* (tech. rep.). International Energy Agency. Paris. Retrieved September 12, 2026, from https://www.iea.org/reports/the-netherlands-2024

Jensen, N. O. (1983, November). *A note on wind generator interaction* (tech. rep. No. Risø-M-2411). Risø National Laboratory. Roskilde, Denmark. https://backend.orbit.dtu.dk/ws/files/55857682/ris_m_2411.pdf

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*. https://doi.org/10.48550/arXiv.1412.6980

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., & Hoyer, S. (2021). Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, *118*(21), e2101784118. https://doi.org/10.1073/pnas.2101784118

Krutova, M., Bakhoday Paskyabi, M., Nielsen, F. G., & Reuder, J. (2020). Evaluation of gaussian wake models under different atmospheric stability conditions: Comparison with large eddy simulation results. *Journal of Physics: Conference Series*, *1669*(1), 012016. https://doi.org/10.1088/1742-6596/1669/1/012016

Larsen, G. C., Madsen, H. A., Thomsen, K., & Larsen, T. J. (2008). Wake meandering: A pragmatic approach. *Wind Energy*, *11*(4), 377–395. https://doi.org/10.1002/we.267

Levick, T., Neubert, A., Friggo, D., Downes, P., Ruisi, R., & Bleeg, J. (2022). Validating the next generation of turbine interaction models. *Journal of Physics: Conference Series*, *2257*(1), 012010. https://doi.org/10.1088/1742-6596/2257/1/012010

Li, R., Zhang, J., & Zhao, X. (2022). Dynamic wind farm wake modeling based on a bilateral convolutional neural network and high-fidelity les data. *Energy*, *258*, 124845. https://doi.org/https://doi.org/10.1016/j.energy.2022.124845

Li, S., Zhang, M., & Piggott, M. D. (2023). End-to-end wind turbine wake modelling with deep graph representation learning. *Applied Energy*, *339*, 120928. https://doi.org/https://doi.org/10.1016/j.apenergy.2023.120928

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*. https://doi.org/10.48550/arXiv.2010.08895

Liu, Y., Qiu, Y., Feng, Y., Zhou, H., Xiong, X.-L., & Zhu, C. (2024). A novel physical-guided 3d graph neural network for accurate wind farm power forecasting. *SSRN Preprint*. https://doi.org/http://dx.doi.org/10.2139/ssrn.4990406

Meyer Forsting, A., Rathmann, O., van der Laan, M. P., Troldborg, N., Gribben, B., Hawkes, G., & Branlard, E. (2021). Verification of induction zone models for wind farm annual energy production estimation. *Journal of Physics: Conference Series*, *1934*, 012023. https://doi.org/10.1088/1742-6596/1934/1/012023

Norwegian Offshore Wind. (2024). *Energy transition outlook: Falling costs for floating wind*. Retrieved June 3, 2026, from https://www.norwegianoffshorewind.no/news/energy-transition-outlook-falling-costs-for-floating-wind

Platis, A., Hundhausen, M., Lampert, A., Emeis, S., & Bange, J. (2022). The role of atmospheric stability and turbulence in offshore wind-farm wakes in the german bight. *Boundary-Layer Meteorology*, *182*(3), 441–469. https://doi.org/10.1007/s10546-021-00668-4

PyTorch. (2026). Pytorch documentation. https://docs.pytorch.org/docs/main/index.html

Reuters. (2023, July 13). *How big oil won the bidding for german offshore wind sites*. Retrieved May 28, 2026, from https://www.reuters.com/business/energy/how-big-oil-won-bidding-german-offshore-wind-sites-2023-07-13/

Santoni, C., Zhang, D., Zhang, Z., Samaras, D., Sotiropoulos, F., & Khosronejad, A. (2024). Toward ultra-efficient high-fidelity predictions of wind turbine wakes: Augmenting the accuracy of engineering models with machine learning. *Physics of Fluids*, *36*, 065159. https://doi.org/https://doi.org/10.1063/5.0213321

Schøler, J. P., Rasmussen, F. P. W., Quick, J., & Réthoré, P.-E. (2025). Graph neural operator for windfarm wake flow [Preprint, under review]. *Wind Energy Science Discussions*. https://doi.org/https://doi.org/10.5194/wes-2025-261

Shakoor, R., Hassan, M. Y., Raheem, A., & Wu, Y.-K. (2016). Wake effect modeling: A review of wind farm layout optimization using jensen's model. *Renewable and Sustainable Energy Reviews*, *58*, 1048–1059. https://doi.org/10.1016/j.rser.2015.12.229

Simonovsky, M., & Komodakis, N. (2017). Dynamic edge-conditioned filters in convolutional neural networks on graphs. *arXiv preprint arXiv:1704.02901*. https://doi.org/10.48550/arXiv.1704.02901

Taylor Wessing. (2024, August 22). *Offshore wind in germany and europe*. Retrieved May 28, 2026, from https://www.taylorwessing.com/en/insights-and-events/insights/2024/08/offshore-wind-in-deutschland-und-europa

Wang, D., Feng, D., Zhou, K., Chen, Y., Liao, S.-J., & Chen, S. (2025). Symbolic regression-enhanced dynamic wake meandering: Fast and physically consistent wind turbine wake modelling. *Journal of Fluid Mechanics*, *1025*, A56. https://doi.org/https://doi.org/10.1017/jfm.2025.10947

Whiffle. (2024). Whiffle wins50 open data on aws: Large eddy simulation data of the winds of the north sea in 2050 [Accessed: 2026-03-23].

Ye, M., Li, M., Liu, M., Xiao, C., & Wan, D. (2025). Overview of data-driven models for wind turbine wake flows. *Journal of Marine Science and Application*, *24*, 1–20. https://doi.org/https://doi.org/10.1007/s11804-025-00683-8

Zhang, D., Santoni, C., Zhang, Z., Samaras, D., & Khosronejad, A. (2025). Adaptive multitask neural network for high-fidelity wake flow modeling of wind farms. *Energies*, *18*(11), 2897. https://doi.org/https://doi.org/10.3390/en18112897

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57–81. https://doi.org/https://doi.org/10.1016/j.aiopen.2021.01.001

A

Literature review overview

Table A.1: Full table of reviewed wake-modelling literature

Title	**Authors**	**Year**
A note on wind generator interaction	Jensen	1983
Calculating the flowfield in the wake of wind turbines	Ainslie	1988
Wake meandering: A pragmatic approach	Larsen et al.	2008
A new analytical model for wind-turbine wakes	Bastankhah & Porté-Agel	2014
Evaluation of Gaussian wake models under different atmospheric stability conditions: Comparison with large eddy simulation results	Krutova et al.	2020
Artificial Neural Networks based wake model for power prediction of wind farm	Ti et al.	2021
Fourier Neural Operator for Parametric Partial Differential Equations	Zongyi Li et al.	2021
Machine learning–accelerated computational fluid dynamics	Kochkov et al.	2021
Dynamic wind farm wake modeling based on a Bilateral Convolutional Neural Network and high-fidelity LES data	Rui Li et al.	2022
Wind Park Power Prediction: Attention-Based Graph Networks and Deep Learning to Capture Wake Losses	Bentsen et al.	2022
Validating the Next Generation of Turbine Interaction Models	Levick et al.	2022
End-to-end wind turbine wake modelling with deep graph representation learning	Li et al.	2023
Local flow and loads estimation on wake-affected wind turbines using graph neural networks and PyWake	Duthé et al.	2023
Toward ultra-efficient high-fidelity predictions of wind turbine wakes: Augmenting the accuracy of engineering models with machine learning	Santoni et al.	2024
A Novel Physical-Guided 3D Graph Neural Network for Accurate Wind Farm Power Forecasting	Liu et al.	2024
Overview of Data-Driven Models for Wind Turbine Wake Flows	Ye et al.	2025
Adaptive Multitask Neural Network for High-Fidelity Wake Flow Modeling of Wind Farms	Zhang et al.	2025
Symbolic regression-enhanced dynamic wake meandering: fast and physically consistent wind turbine wake modelling	Wang et al.	2025
Graph Neural Operator for wind farm wake flow	Schøler et al.	2025
Multivariate modeling on wake-affected wind farms by two-stage hybrid graph neural network	Hou et al.	2026

B

Verification run overview

This appendix provides a complete overview of all verification experiments presented in Chapter 4. The table includes the dataset size, hidden dimension, target formulation, runtime breakdown, and resulting model performance metrics for each run.

Table B.1: Overview of all verification runs for different dataset sizes, hidden dimensions, and target formulations. Runtime components are reported separately for Jensen generation, graph construction, and training.

Run	Samples	Hidden	Target	Total time	Jensen	Graph	Training	Best val. MAE	Test MAE
1	100	32	U	21.0 s	0.91 s	0.05 s	18.73 s	0.468721	0.742031
2	1000	32	U	85.0 s	6.27 s	0.39 s	77.36 s	0.188298	0.183942
3	10000	32	U	947.0 s	56.26 s	2.66 s	884.87 s	0.099976	0.105767
4	100000	32	U	3024.0 s	322.62 s	20.07 s	2670.98 s	0.088403	0.086181
5	100	64	U	15.8 s	0.65 s	0.03 s	12.72 s	0.443585	0.733659
6	1000	64	U	198.0 s	20.56 s	1.16 s	174.83 s	0.150571	0.152588
7	10000	64	U	1019.0 s	202.50 s	14.15 s	793.38 s	0.094312	0.100144
8	100000	64	U	4705.0 s	362.90 s	19.59 s	4306.79 s	0.077985	0.075818
9	100	128	U	20.0 s	0.63 s	0.04 s	17.73 s	0.471565	0.483106
10	1000	128	U	122.0 s	6.19 s	0.26 s	113.97 s	0.150914	0.152709
11	10000	128	U	1350.0 s	90.14 s	1.14 s	1255.27 s	0.079527	0.085406
12	100000	128	U	8284.0 s	361.60 s	17.74 s	7889.47 s	0.077285	0.075006
13	100	32	ΔU	3.0 s	0.45 s	0.02 s	1.26 s	0.193327	0.226614
14	1000	32	ΔU	80.0 s	5.65 s	0.66 s	72.29 s	0.119904	0.126724
15	10000	32	ΔU	496.0 s	57.36 s	2.05 s	434.23 s	0.073180	0.078771
16	100000	32	ΔU	6998.0 s	378.70 s	15.82 s	6592.01 s	0.070322	0.067984
17	100	64	ΔU	5.7 s	0.71 s	0.03 s	3.35 s	0.198212	0.230464
18	1000	64	ΔU	53.0 s	5.56 s	0.58 s	46.20 s	0.116046	0.122305
19	10000	64	ΔU	1273.0 s	212.39 s	10.80 s	1045.26 s	0.069001	0.074509
20	100000	64	ΔU	11793.0 s	315.66 s	22.28 s	11432.02 s	0.069941	0.067599
21	100	128	ΔU	7.6 s	1.08 s	0.04 s	4.40 s	0.191111	0.212169
22	1000	128	ΔU	173.0 s	7.04 s	0.21 s	165.13 s	0.084548	0.078616
23	10000	128	ΔU	1181.0 s	52.62 s	2.31 s	1122.64 s	0.067961	0.073360
24	100000	128	ΔU	13896.0 s	341.00 s	17.61 s	13521.33 s	0.068469	0.066176

C

Table Chapter 5

Table C.1: Per-seed performance results for all inflow and turbulence-intensity configurations.

Config.	TI	Seed	Dim.	MAE	RMSE	Val. MAE	Time [min]
global-global	no-ti	1	3	0.6718	0.8919	0.7490	11.04
global-global	global-ti	1	4	0.6660	0.9096	0.6935	14.49
global-global	local-ti	1	4	0.6551	0.8991	0.6886	13.61
local-global	no-ti	1	3	0.6746	0.9080	0.7486	9.00
local-global	global-ti	1	4	0.6566	0.8972	0.6888	13.63
local-global	local-ti	1	4	0.6593	0.8919	0.6910	10.86
global-local	no-ti	1	3	0.6661	0.8882	0.7469	7.29
global-local	global-ti	1	4	0.6592	0.9024	0.6875	10.69
global-local	local-ti	1	4	0.6473	0.8817	0.6885	11.62
local-local	no-ti	1	3	0.6613	0.8829	0.7393	7.95
local-local	global-ti	1	4	0.6582	0.8881	0.6852	11.52
local-local	local-ti	1	4	0.6528	0.8780	0.6925	15.09
global-global	no-ti	2	3	0.6687	0.8835	0.7522	16.86
global-global	global-ti	2	4	0.6757	0.9090	0.6943	21.32
global-global	local-ti	2	4	0.6560	0.8777	0.6971	26.00
local-global	no-ti	2	3	0.6805	0.9006	0.7629	10.16
local-global	global-ti	2	4	0.6516	0.8747	0.6906	9.17
local-global	local-ti	2	4	0.6524	0.8783	0.6963	8.13
global-local	no-ti	2	3	0.6811	0.9054	0.7637	5.73
global-local	global-ti	2	4	0.6479	0.8694	0.6918	7.31
global-local	local-ti	2	4	0.6501	0.8719	0.6986	6.23
local-local	no-ti	2	3	0.6753	0.8986	0.7628	5.91
local-local	global-ti	2	4	0.6861	0.9154	0.6962	10.15
local-local	local-ti	2	4	0.6518	0.8746	0.6947	6.63
global-global	no-ti	3	3	0.6966	0.9256	0.7756	7.67
global-global	global-ti	3	4	0.6644	0.8929	0.7006	10.32
global-global	local-ti	3	4	0.6478	0.8633	0.7077	10.01
local-global	no-ti	3	3	0.6771	0.9019	0.7610	12.99
local-global	global-ti	3	4	0.6399	0.8614	0.6820	13.93
local-global	local-ti	3	4	0.6601	0.8860	0.7037	10.50
global-local	no-ti	3	3	0.6928	0.9202	0.7733	5.78
global-local	global-ti	3	4	0.6428	0.8593	0.6975	9.78
global-local	local-ti	3	4	0.6572	0.8825	0.7015	9.65

Continued on next page

Config.	TI	Seed	Dim.	MAE	RMSE	Val. MAE	Time [min]
local-local	no-ti	3	3	0.6610	0.8864	0.7525	11.32
local-local	global-ti	3	4	0.6593	0.8968	0.6963	8.07
local-local	local-ti	3	4	0.6760	0.9035	0.7108	10.17

D

Overview of Selected Wind Farms and Turbine Counts

Table D.1: Wind farms included in the Chapter 6 cross-farm generalisation experiment.

Wind farm	Role	Number of turbines
hornsea2_east	Training	50
future_002	Training	57
doggerbankB_east	Training	58
cluster_040_west	Training	59
future_054	Training	62
cluster_017	Training	68
hkn	Training	72
eastanglia_hub1north	Training	75
borssele_3_4	Training	79
eastanglia_hub2	Training	86
fryslan	Training	89
noordhinder_zuid	Training	93
borssele_1_2	Training	94
doggerbankSE	Training	100
hknw	Training	101
eastanglia_1	Training	102
doggerbankC	Training	113
doggerbankA	Training	113
hkzw	Training	115
future_212	Training	128
eastanglia_hub3	Training	132
sofia	Training	132
cluster_033	Training	136
hkw	Training	141
hkz	Training	144
future_052	Test	68

E

The use of AI tools

AI-based language tools were used to support the writing and presentation of this thesis. In particular, ChatGPT was employed to improve the clarity, coherence, and academic tone of the text by reformulating sentences and suggesting alternative phrasings. An example prompt used for this purpose is: *"Can you rewrite this paragraph in more formal academic English while preserving its original meaning?"*

In addition, ChatGPT was used as a supportive aid to clarify technical terminology encountered in the literature. All analytical decisions, interpretations, and conclusions presented in this thesis remain the sole responsibility of the author.

ChatGPT was also used to support programming-related tasks, such as explaining Python code, assisting with debugging, generating code snippets, and suggesting implementation approaches. These interactions were used to support the development of the Python code used throughout the analysis.

Finally, ChatGPT was used to assist with LaTeX-related tasks, including formatting references and resolving compilation errors in Overleaf.