A Graph-based Wake Model for Robust Power Prediction in Heterogeneous Wind Farms

Master's Thesis of

B.Sc. Niklas Lutze

At the KIT Department of Informatics
Institute for Automation and Applied Informatics (IAI)

First examiner:	TT-Prof. Dr. Benjamin Schäfer
Second examiner:	Prof. Dr. Veit Hagenmeyer
First advisor:	M.Sc. Sebastian Pütz
Second advisor:	M.Sc. Martin Sadric

Matriculation Number: 2295392
November 17, 2025 – May 18, 2026

Karlsruhe Institute of Technology (KIT)
Department of Informatics
P.O. Box 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself. I have not used any other than the aids that I have mentioned. I have marked all parts of the thesis that I have included from referenced literature, either in their original wording or paraphrasing their contents. I have followed the by-laws to implement scientific integrity at KIT.

Karlsruhe, April 30, 2026

[redacted].

(B.Sc. Niklas Lutze)

Abstract

As the global transition toward renewable energy accelerates, optimizing the operational efficiency of wind farms is critical. Aerodynamic wake interactions between turbines present an operational challenge by causing downstream wind velocity deficits and subsequent power losses. While machine learning offers promising surrogate models for these dynamics, many existing approaches are evaluated on homogeneous turbines and limited atmospheric diversity. Additionally, these models typically provide point estimates that overlook the inherent predictive uncertainties. This thesis addresses these limitations by training a Graph Transformer (GT) architecture on a novel pipeline of real-world heterogeneous turbine records and diverse weather profiles to capture complex, farm-wide aerodynamic interactions. Furthermore, a post-hoc heteroscedastic variance estimator is implemented to provide input-dependent uncertainty quantification. Experimental results demonstrate that the GT outperforms turbine-level MLP baselines, achieving a predictive accuracy of 0.019 in MAE in normalized power compared to the MLP's MAE of around 0.060. The GT's R^2 for normalized wake loss of 0.841 shows that it can explain a large portion of wake-induced power deficits. The probabilistic evaluation demonstrates that the heteroscedastic GT provides a superior balance of calibration and sharpness, identifying sensitive operational regimes and complex wake states. Finally, evaluation on a real-world SCADA dataset reveals that while a performance gap exists, simple finetuning allows the GT to partially recover its predictive accuracy to 0.057 in MAE and provide reliable uncertainty estimates. Despite these results, the study identifies limitations regarding spatial extrapolation to unseen layouts due to a reliance on absolute coordinates, as well as boundary artifacts introduced by applying Gaussian assumptions to strictly bounded power data. Nevertheless, this research establishes the GT as a viable surrogate model for robust, uncertainty-aware wind farm control, while the developed dataset generation pipeline provides a versatile foundation for future analyses in complex, heterogeneous environments.

Zusammenfassung

Angesichts der sich beschleunigenden weltweiten Umstellung auf erneuerbare Energien ist die Optimierung der Betriebseffizienz von Windparks von entscheidender Bedeutung. Aerodynamische Nachlaufinteraktionen zwischen Turbinen stellen eine betriebliche Herausforderung dar, da sie zu einem Rückgang der Windgeschwindigkeit stromabwärts und damit zu Leistungsverlusten führen. Zwar bietet maschinelles Lernen vielversprechende Surrogatmodelle für diese Dynamiken, doch werden viele bestehende Ansätze anhand homogener Turbinen und unter begrenzten atmosphärischen Bedingungen evaluiert. Zudem liefern diese Modelle in der Regel Punktschätzungen, die Vorhersageunsicherheiten außer Acht lassen. Diese Arbeit begegnet diesen Einschränkungen durch das Training einer Graph-Transformer-Architektur (GT) auf einer neuartigen Pipeline aus realen, heterogenen Turbinendaten und vielfältigen Wetterprofilen, um komplexe, parkweite aerodynamische Wechselwirkungen zu erfassen. Darüber hinaus wird ein post-hoc-Schätzer für heteroskedastische Varianz implementiert, um eine eingababhängige Quantifizierung der Unsicherheit zu ermöglichen. Experimentelle Ergebnisse zeigen, dass der GT die MLP-Baselines übertrifft und eine Vorhersagegenauigkeit von $0,019$ im MAE der normierten Leistung erreicht, verglichen mit dem MAE des MLP von etwa $0,060$. Der GT-Wert von R^2 für den normierten Nachlaufverlust von $0,841$ zeigt, dass er einen großen Teil der durch den Nachlauf verursachten Leistungsdefizite erklären kann. Die probabilistische Auswertung zeigt, dass der heteroskedastische GT eine überlegene Balance aus Kalibrierung und Schärfe bietet und sensible Betriebsbereiche sowie komplexe Nachlaufzustände identifiziert. Schließlich zeigt die Auswertung eines realen SCADA-Datensatzes, dass zwar eine Leistungslücke besteht, der GT jedoch durch Finetuning seine Vorhersagegenauigkeit auf einen MAE von 0,057 wiederherstellen und zuverlässige Unsicherheitsschätzungen liefern kann. Trotz dieser Ergebnisse identifiziert die Arbeit Einschränkungen hinsichtlich der räumlichen Extrapolation auf unbekannte Layouts aufgrund der Abhängigkeit von absoluten Koordinaten sowie Artefakte, die durch die Anwendung gaußscher Annahmen entstehen. Dennoch etabliert diese Forschung den GT als ein brauchbares Surrogatmodell für eine robuste, unsicherheitsbewusste Windparksteuerung, während die entwickelte Pipeline zur Datensatzgenerierung eine vielseitige Grundlage für zukünftige Analysen in komplexen, heterogenen Umgebungen bietet.

Acknowledgments

Thank you to everyone who has supported me throughout the duration of this thesis, specifically my advisors Sebastian and Martin as well as Laura, for being by my side throughout this journey. I acknowledge support by the state of Baden-Württemberg through bwHPC. Furthermore, I acknowledge the use of generative AI tools, specifically Google Gemini, for writing assistance and editorial refinement, as well as GitHub Copilot for support in code development.

Contents

Glossary

ADM Actuator Disk Model.

AEP Annual Energy Production.

CC Cumulative Curl.

CFD Computational Fluid Dynamics.

CRPS Continuous Ranked Probability Score.

GCH Gauss–Curl Hybrid.

GNN Graph Neural Network.

GT Graph Transformer.

HPC High Performance Cluster.

IEA International Energy Agency.

IQR Interquartile Range.

LES Large Eddy Simulation.

LiDAR Light Detection and Ranging.

MAE Mean Absolute Error.

MaStR Marktstammdatenregister.

ML Machine Learning.

MLP Multi-layer Perceptron.

MPIW Mean Prediction Interval Width.

MSE Mean Squared Error.

NLL Negative Log Likelihood.

PICP Prediction Interval Coverage Probability.

RANS Reynolds-Averaged Navier-Stokes.

RMSE Root Mean Squared Error.

RQ Research Question.

SCADA Supervisory Control and Data Acquisition.

SDPA Scaled Dot-Product Multihead Attention.

Std Standard Deviation.

TI Turbulence Intensity.

WFC Wind Farm Control.

WFFC Wind Farm Flow Control.

WTG Wind Turbine Generator.

1 Introduction

As the global transition toward renewable energy accelerates, wind power has emerged as a cornerstone of the future electricity grid. However, the operational efficiency of wind farms is challenged by aerodynamic interactions between turbines. This chapter introduces the context of Wind Farm Control (WFC) and sets up the research questions centered on the Graph Transformer (GT), uncertainty quantification, and the transferability of synthetic training to real-world data.

1.1 Motivation

The global energy landscape has entered what the International Energy Agency (IEA) defines as the "Age of Electricity", with global electricity demand projected to rise by 40% by 2035 compared to 2024 levels [1]. At the same time, to reduce emissions, the global energy mix is shifting from fossil fuels to renewables. The IEA projects that renewables will surpass coal as the world's primary source of electricity by 2026, driven largely by the scaling of wind and solar technologies [2]. As of mid-2025, total installed global wind capacity reached 1245 GW, satisfying approximately 12% of global electricity demand [3]. Annual installations are currently at around 150 GW and need to increase to 320 GW per year needed to reach the goal of the 2023 United Nations Climate Change Conference to triple renewable energy capacity by 2030 [4]. Specifically in the EU, wind energy is expected to make up for 50% of its electricity mix by 2050 [5]. Overall, this rapid scaling underscores the importance of efficient wind power production.

Modern wind plants typically operate as wind farms, reducing costs, and simplifying operation and maintenance [6]. However, it introduces a number of complexities. A central challenge arises from aerodynamic interactions, where upstream turbines affect downstream ones by influencing air flow [7]. These wake effects are characterized by reduced wind speed and increased turbulence intensity, which can lead to substantial power losses and accelerated structural degradation of downstream units [7]. Therefore, the collective output of a wind farm is often significantly lower than the sum of its turbines' individual capacities [8]. Figure 1.1 visualizes the wind velocity across a wind farm and shows how downstream turbines are inside the wakes of upstream ones, leading to reduced power.

Because of these inter-turbine interactions, it is no longer sufficient to control each wind turbine individually. Instead, the control of all turbines within a farm must be coordinated, which is known as WFC. Wake steering is a specific WFC approach. It involves intentionally

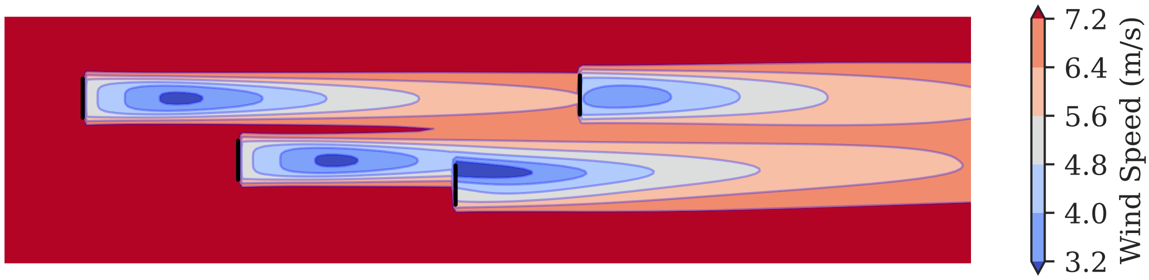


Figure 1.1: Upstream turbines affect downstream units through wake effects and cause power deficits. This contour plot visualizes the flow field of a simulated wind farm.

misaligning upstream turbines relative to the incoming wind to deflect their wakes away from downstream turbines. The effectiveness of wake steering has been repeatedly verified, although its influence is dependent on farm configuration and wind conditions [9]. For example, in a recent field experiment, wake steering yielded an average power gain of 8.3% for wind speeds of 4–10 m/s [10]. In more specific conditions of wind speeds of 6–7 m/s, the average gain was more modest with 4.5% .

To optimize wake steering, it is beneficial to understand the flow dynamics of a wind farm. There are a few options to measure airflow in real-world wind farms ranging from point-based sensors, mounted on the turbine for example, to remote-sensing technologies such as Light Detection and Ranging (LiDAR) [7]. While these capture the real-time state of the system, many optimization strategies rely on predicting how the flow might behave in hypothetical configurations. Wake models address this need by providing a mathematical framework to simulate turbine interactions. While high-fidelity Computational Fluid Dynamics (CFD) simulations yield the most precise results, the corresponding computation time is in the order of days on a High Performance Cluster (HPC) making them unsuitable for real-time control or large-scale optimization tasks [7, 11]. In contrast, low-fidelity analytical models, so called engineering models, are magnitudes faster, offering execution speeds in the orders of seconds [11]. However, the increase in speed comes at the cost of accuracy [12]. Data-driven surrogate models, particularly statistical or Machine Learning (ML) models, represent an attractive alternative, offering a balance between computational costs and accuracy [13]. Instead of being based on manually-defined, physical equations, they learn the relationships between inputs and outputs directly from data. Once trained, these models provide results much faster than high-fidelity models and may even surpass the execution speed of analytical models [14]. In addition, by leveraging training datasets derived from high-fidelity simulations or empirical wind farm observations, these models can capture complex interactions that analytical models might not be capable of [6, 15].

Various approaches have been proposed that apply ML to the wake modeling or power prediction problem [6, 8, 14, 16–28]. Among different ML frameworks, Graph Neural Networks (GNNs) are particularly well-suited for this task [8, 14, 24–27]. Unlike standard ML architectures, which usually require a fixed input size, GNNs represent wind farms as graphs allowing them to generalize across diverse farm layouts. One example is the work by Harrison-Atlas et al. [14]. They developed the Wind Plant Graph Neural Network (WPGNN), providing a 700x speed-up over the FLORIS engineering model. By applying this approach to over 6,800 sites in the USA, they demonstrated that co-optimizing layouts with wake steering can reduce land requirements by an average of 18% and boost annual revenue for large-scale plants by up to US$3.7 million. Expanding beyond power prediction, Santos et al. introduced a layout-agnostic GNN framework for the joint modeling of fatigue loads, rotor-averaged wind speed, and power production [26]. Their study compared various GNN blocks to evaluate generalization across unseen layouts and inflow conditions. More recently, Li et al. proposed a GT surrogate model representing wind farms as fully connected graphs, arguing that global self-attention better captures long-range inter-turbine dependencies than local neighborhood aggregation [28]. Their approach achieved a 99.8% relative power prediction accuracy and, when integrated with genetic algorithms, reduced yaw steering optimization times from over 30 minutes to approximately one minute.

1.2 Problem Statement

While ML models show significant promise in wake modeling, existing research in this domain suffers from three limitations:

Lack of Atmospheric Diversity. First, atmospheric inputs in existing studies are typically constrained to wind speed and turbulence intensity [14, 17, 22, 23, 28]. Furthermore, these approaches sample wind conditions uniformly and independently, ignoring dependencies between the variables. While there are a few approaches that use stochastic dependent sampling and include wind shear, such as [12, 26], wind veer is often overlooked. This thesis further differentiates itself by deriving conditions from real weather forecasts rather than theoretical distributions.

Homogeneity of Turbine and Farm Specifications. A critical gap exists in how models handle turbine heterogeneity. While some approaches limit their scope on a specific wind farm [21, 27, 29], many approaches use a variety of wind farm layouts [14, 26, 28]. However, they are all limited to one type of turbine. In reality, wind farms can consist of multiple turbine models with varying hub heights, rotor diameters, and power curves. None of the reviewed architectures were designed or evaluated for these heterogeneous configurations.

Predictive Uncertainties. Another challenge for the practical deployment of wake steering are uncertainties [30]. The importance of uncertainty in wake modeling has been recognized in the context of wake models. For instance, Zhang et al. utilized Bayesian frameworks to estimate parameter uncertainty in the FLORIS wake model using high-fidelity simulation data [31], while Aerts et al. extended this concept by explicitly separating turbulence-related aleatoric uncertainty, epistemic parameter uncertainty, and systematic model bias using historical operational data [32]. Other research has focused on propagating input and environmental uncertainties, such as wind direction variability and yaw misalignment errors, through wake models to optimize wind farm performance. This is demonstrated by the optimization under uncertainty approach of Quick et al. [30]. However, an uncertainty-aware GNN-model for power prediction has not been explored in this context.

Failure to account for these limitations might lead to unexpected behavior in the field. For ML to be viable for WFC, models must be robust enough to generalize across heterogeneous wind farms while remaining resilient to the uncertainties of the real-world environment.

1.3 Research Questions

This thesis aims to address the limitations mentioned above. It includes the development of a data pipeline that derives atmospheric conditions from real weather forecasts and uses heterogeneous real-world turbine layouts from the Marktstammdatenregister (MaStR) registry. A GT approach based on [28] is applied to this dataset and its behavior in predicting turbine power is analyzed. In addition, a post-hoc heteroscedastic variance estimator inspired by [33] is fitted on the GT model and its ability to estimate predictive uncertainty is evaluated. Lastly, an empirical analysis of the models' generalization behavior on a real dataset is carried out. To summarize, the thesis seeks to answer the following Research Questions (RQs):

- **RQ1:** How does a GT compare to a Multi-layer Perceptron (MLP) baseline when functioning as a wake model for wind farm power estimation on datasets featuring heterogeneous turbines and diverse atmospheric conditions?
- **RQ2:** How does the predictive uncertainty quantification of a heteroscedastic GT compare against homoscedastic GT and MLP baselines?
- **RQ3:** How well do these wake models, pretrained on synthetic data, generalize to real-world Supervisory Control and Data Acquisition (SCADA) data under zero-shot and finetuned settings?

1.4 Thesis Outline

This thesis is organized as follows: Chapter 2 establishes the theoretical foundation, covering wind farm operations, GTs, and uncertainty quantification. Building on this foundation, Chapter 3 details the research methodology, including the data processing pipeline and the proposed model architectures. The empirical findings are presented in Chapter 4, followed by a discussion in Chapter 5 including limitations and future work. Finally, Chapter 6 summarizes the core findings and concludes this thesis.

2 Fundamentals

This chapter establishes the theoretical foundations of the thesis. Section 2.1 begins with an overview of wind farm operation, including turbine control strategies and the motivation for WFC, with a focus on wake steering methodologies. Section 2.2 introduces the architecture and application of GTs, followed by a part addressing uncertainty quantification.

2.1 Wind Farm Operation

This section outlines the principles of wind farm operation. It begins with the fundamentals of wind turbine aerodynamics and physics. Subsequently, wake interactions are introduced to motivate the need for WFC. This is followed by a brief review of wake models and concludes with an overview of wake steering approaches.

2.1.1 Wind Turbine Control

Wind turbines extract energy from the wind. To better understand the underlying physics, it is useful to view the turbine's rotor as a thin disk through which the wind passes. This abstraction is formally known as the Actuator Disk Model (ADM) [7]. Figure 2.1 provides a schematic visualization of the ADM. The following paragraphs give a short introduction to important dynamics under ideal conditions based on the work by Boersma et al. [7].

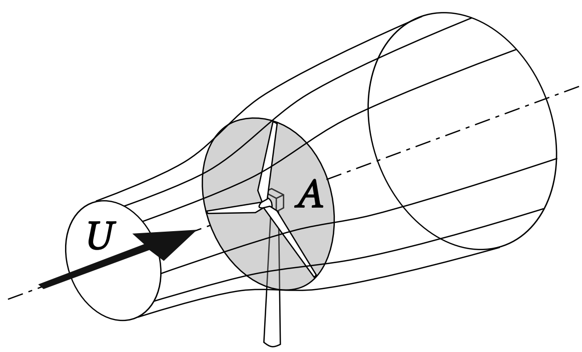


Figure 2.1: Schematic of the ADM. Wind with flow velocity U passes through cross-sectional area A. Adapted from [7].

The available kinetic power at the disk depends on the mass flow rate of air $\frac{\mathrm{d}m}{\mathrm{d}t}$. This rate is a function of the air density ρ [kg/m^3], the cross-sectional area A [m^2], and the flow velocity U [m/s]. If U is assumed to be uniform over the disk, the mass flow rate is governed by

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \rho A U. \tag{2.1}$$

Therefore, the total kinetic power available at the surface A is governed by

$$P_W = \frac{1}{2}\frac{\mathrm{d}m}{\mathrm{d}t}U^2 = \frac{1}{2}\rho A U^3. \tag{2.2}$$

However, not all of this available energy can be extracted by a wind turbine. To bridge the gap between this mathematical abstraction of the rotor plane and actual power generation, a more detailed view on the mechanical architecture of a wind turbine is helpful.

There are multiple types of wind turbines. The most common turbine type is the "upwind horizontal-axis wind turbine" [7]. Figure 2.2 provides a schematic representation of this type. Its main component is the rotor that connects to the generator. The generator among other parts is contained within the nacelle that sits on top of a tower. As wind passes through the rotor plane, the blades capture its momentum and transform it into aerodynamic force. This force turns the rotor, allowing the generator to translate that rotational kinetic energy into electricity.

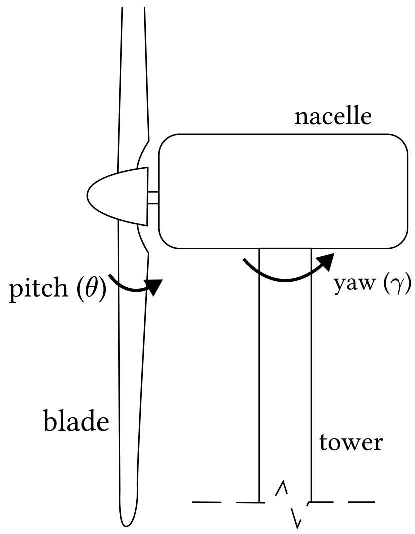


Figure 2.2: Schematic of a horizontal-axis wind turbine illustrating primary structural components and operational control variables. Adapted from [7].

The key control variables for wind turbine control are the generator torque τ_g regulating power conversion, the blade pitch angle θ adjusting aerodynamic responses to wind, and the yaw angle γ orienting the rotor to optimize energy capture [13]. There are additional, less common control variables such as tilting or the use of blade flaps, but they are left out here for simplicity [34].

As wind passes through the rotor, the turbine extracts energy by exerting a resistive force known as thrust F. The thrust magnitude is governed by

$$F = C_T \frac{1}{2} \rho A U^2, \tag{2.3}$$

where C_T is a dimensionless thrust coefficient. Furthermore, building upon Equation 2.2, the actual power P captured by the turbine is expressed as

$$P = C_P \frac{1}{2} \rho A U^3. \tag{2.4}$$

Here, the power coefficient C_P serves as a scaling factor that accounts for the aerodynamic efficiency of the physical rotor. Comparing Equations 2.2 and 2.4, it can be seen that the extracted power P is a portion of the total kinetic power P_W.

Crucially, the extraction of energy slows down the wind passing through the rotor. This phenomenon is quantified by the axial induction factor a, defined as the ratio of the velocity deficit at the rotor relative to the free-stream velocity [7]:

$$a = \frac{U - U_r}{U}, \tag{2.5}$$

where U_r represents the wind velocity at the rotor plane. Rearranging this definition shows that the flow speed at the rotor is reduced to $U_r = (1 - a)U$. The power coefficient C_P, the thrust coefficient C_T and the axial induction factor a are linked. Under zero yaw angle, the following holds:

$$C_T = 4a \cdot (1 - a), \quad C_P = 4a \cdot (1 - a)^2. \tag{2.6}$$

Because previously introduced control variables influence these coefficients, they serve as the primary mechanism for modulating not only the individual turbine's power extraction but also the momentum deficit propagated to the downstream flow field.

As illustrated before, while wind power scales cubically with wind speed, the turbine captures only a portion of that power. This relationship is captured by the turbine's power curve, an example of which is shown in Figure 2.3. This curve also demonstrates how turbine operation is categorized into three specific control regions based on wind speed [7]. In Region I, where conditions fall below the cut-in threshold, the turbine remains inactive as the wind energy is insufficient to overcome mechanical friction. Region II spans from the turbine's cut-in speed to the rated speed. In this zone, the control variables are adjusted to maximize efficiency. Region III covers the range between rated and cut-out speeds, where the focus shifts to regulating rotor speed to ensure safe operation. Finally, at wind speeds exceeding the cut-out limit, the turbine is shut down to prevent structural damage [13].

2.1.2 Wake Physics

As a wind turbine extracts kinetic energy from the wind, it induces a significant modification of the airflow in the downstream region [7]. This "shadow" is called the wake of the turbine [13].

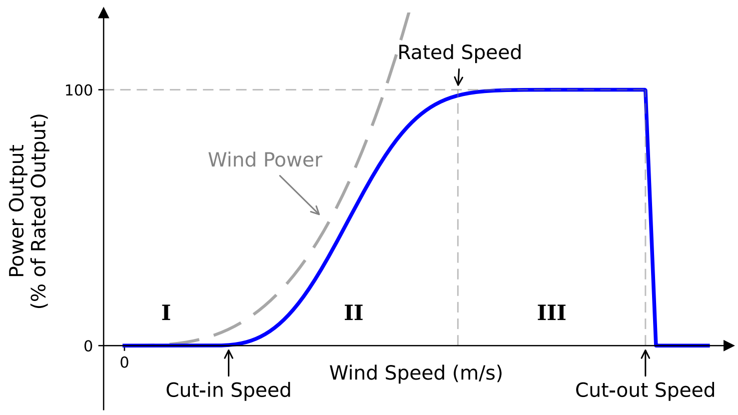


Figure 2.3: While the wind power scales cubically with wind speed, the turbine can only extract a portion of the available power. This figure visualizes a characteristic power curve illustrating the different operational regions (I, II, III) of a wind turbine. Adapted from [7].

It is characterized by a velocity deficit and increased turbulence. As the airflow interacts with the turbine, its characteristics are typically categorized into three regions [13]. The induction region is located on the upwind side of the rotor. In this region, the flow decelerates before reaching the turbine. The region between the turbine and approximately 2-4 rotor diameters downstream is called the near-wake region. The flow behavior in this zone is highly influenced by the specific turbine geometry and blade aerodynamics, resulting in complex, non-homogeneous flow patterns. Further downstream, in the so-called far-wake region, the influence of the specific rotor architecture diminishes. In this region, the wake behavior is primarily governed by atmospheric inflow conditions, thrust coefficients, and turbulent mixing.

Through the wakes, upwind turbines directly influence the performance of those located downwind. Therefore, wake interactions represent a critical challenge in wind farm design and control [34]. This physical coupling necessitates a shift from individual turbine optimization to coordinated WFC.

2.1.3 Wind Farm Control

WFC refers to the coordinated management of all turbines within a farm. It aims to account for inter-turbine interactions to improve the overall “figure of merit” of the entire wind farm, such as total power output or structural longevity [34]. At the industrial level, these control strategies are typically implemented through wind farm controllers, also called SCADA systems, which facilitate the communication and data processing required for farm-wide optimization [35]. As noted by Meyers et al. [34], WFC is a broad term that encompasses all plant-level operations, including maintenance scheduling or the sequential start-up and shut-down of turbines, which are typically independent of aerodynamic coupling. Within this broader context, Meyers et al. use the more specific term Wind Farm Flow Control (WFFC) to describe the subset of control strategies that explicitly target inter-turbine flow interactions. However, the majority of existing literature uses the term WFC to describe these same phenomena. Consequently, in the following sections, both terms are used interchangeably to refer to the aerodynamic coordination of the wind farm.

The primary economic driver for WFC is the reduction of the Levelized Cost of Energy, a metric that balances power production against the total lifecycle expenses [13]. While increasing Annual Energy Production (AEP) is the most common goal, objectives can also include the reduction of structural fatigue loads, providing ancillary services to the electric grid, or minimizing environmental impacts such as noise emissions [9, 34].

The outlook for these technologies is optimistic. Expert elicitations from 2019 indicate that 84% of participants believe WFC technology will be broadly adopted in the future, with approximately 60% of respondents suggesting that even a modest gain of 1.0% in AEP is sufficient to justify the commercial implementation of these systems [9].

The most common technologies to achieve these objectives are axial induction control and yaw control [34]. Axial induction control adjusts the axial induction factor by modifying the blades' pitch angles or the generator's torque [34]. Yaw control includes adjusting the yaw angle, and thus misaligning the rotor plane with the incoming wind [13].

2.1.4 Wake Modeling

Most WFC approaches rely on wake models. As noted by Frutuoso et al., these models are typically categorized into low-, medium-, and high-fidelity frameworks based on the trade-off between computational complexity and physical accuracy [13].

At its most fundamental level, the time-varying, three-dimensional behavior of airflow is characterized by the Navier-Stokes equations, a system for which no general analytical solution exists [7]. Instead, CFD can be applied, most commonly using Large Eddy Simulation (LES). These high-fidelity models offer accurate results but come with a high computational cost. Examples of this type of wake model are the Simulator for Wind Farm Applications (SOWFA) [36] and PALM [37]. To mitigate the high computational costs, mid-fidelity models aim to speed up computations by simplifying the underlying physics, for example by using the Reynolds-Averaged Navier-Stokes (RANS) equations [7]. Examples for mid-fidelity models are the Wind Farm Simulator (WFSim) [38] and FAST.Farm [39].

At the low-fidelity end of the spectrum are analytical models, which are also referred to as engineering models. The two main libraries to work with these models are FLORIS [40] and PyWake [41]. One of the earliest approaches is the Jensen model [42]. Using a top-hat shape, it estimates the velocity deficit in the far-wake region as a constant value across the wake's width. This results in an abrupt jump to free-stream velocity at the wake's discrete boundary. This model has largely been replaced in modern research by Gaussian wake models, such as the one proposed by Bastankhah and Porté-Agel [43].

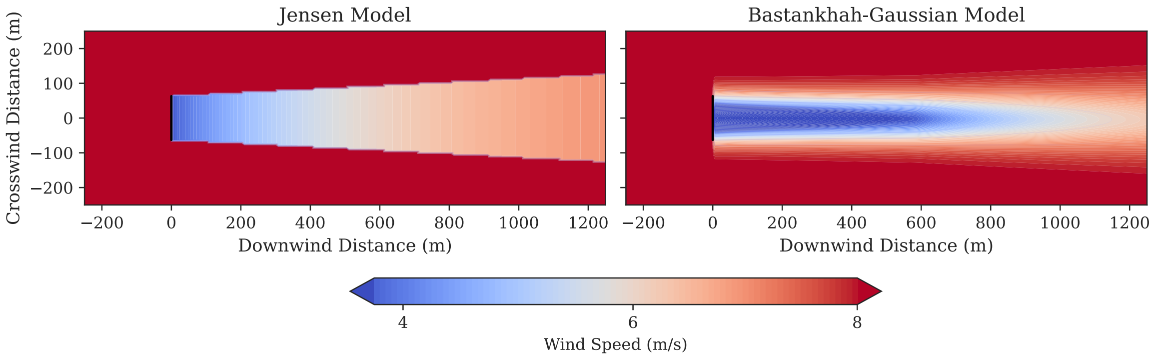


Figure 2.4: Engineering wake models have advanced in accuracy. Comparison of wake velocity profiles for the Jensen and Bastankhah-Gaussian wake models.

These models assume a Gaussian distribution in both the lateral and vertical directions. Figure 2.4 displays how the Jensen Model differs from a Gaussian model. In wind farms with multiple turbines, wakes interact with each other, for example, they overlap [13]. To account for the cumulative effect of multiple wakes, researchers typically employ quadratic superposition methods [13, 28]. A steered wake can also redirect the wakes of downstream turbines, even if they are not yawed. This is known as "Secondary steering" [44]. More recent models thus address these secondary effects, such as the Gauss–Curl Hybrid (GCH) model [45] and its extension, the Cumulative Curl (CC) model [46].

Modern engineering models take in a number of parameters. Apart from the characterization of the examined wind farm, the description of the flow field is critical. For instance, the FLORIS framework incorporates wind speed and direction at a reference height, as well as wind veer and shear [40]. Wind shear represents the change in horizontal wind speed with height and is fundamentally described by the power law [12]:

$$u(z) = u_{\text{ref}} \left(\frac{z}{z_{\text{ref}}} \right)^{\alpha} \tag{2.7}$$

where $u(z)$ is the wind speed at height z, u_{ref} is the known wind speed at a reference height z_{ref}, and α is the wind shear exponent, often referred to just as shear. Complementary to this, wind veer accounts for the directional shift of the wind across the vertical profile. Another critical variable for wake models is Turbulence Intensity (TI). TI is defined as the ratio of the standard deviation of wind speed to the mean wind speed [12]. Higher TI leads to faster wake recovery and thus reducing the wake's length [21, 43]. Wake modeling remains a highly active area of ongoing research as investigators seek to better capture the complex three-dimensional flow phenomena without sacrificing computational speed.

2.1.5 Wake Steering

A widely adopted WFC approach is wake steering via yaw offsets [34]. Figure 2.5 provides a schematic of the idea. By intentionally misaligning a turbine with the incoming wind, the resulting wake is redirected and its shape is altered [34]. While research predominantly focuses on horizontal adjustments, vertical steering by rotor tilting is an emerging area of study [47]. Although the primary objective of wake steering is to maximize aggregate power production [28], it also significantly influences structural loading across the wind farm [34]. Ultimately, the goal of wake steering is to identify the optimal yaw angles relative to site-specific constraints and prevailing atmospheric conditions for a given objective. Wake steering approaches vary widely across several dimensions, including their underlying modeling frameworks, control strategies, optimization algorithms, and validation fidelity.

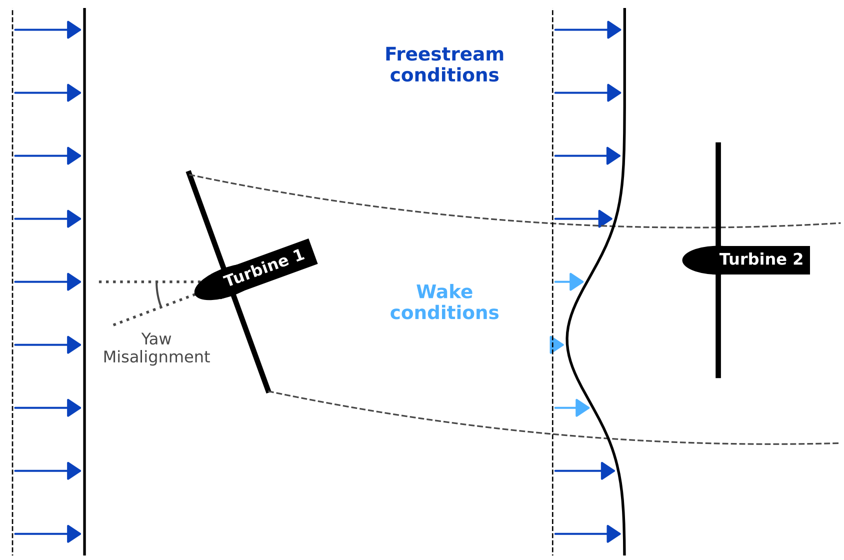


Figure 2.5: Schematic of wake steering via yaw misalignment, illustrating the lateral deflection of the wake trajectory relative to the inflow direction. Adapted from [13].

Meyers et al. categorize these diverse approaches into two primary frameworks [34]: quasi-static and dynamic methods. Quasi-static approaches adapt set-points to meteorological variations, i.e., wind direction, at a relatively low pace. Dynamic approaches aim to include physical details that happen on a smaller time scale, such as wind gusts traveling through the farm.

The foundation of any wake steering strategy is the modeling approach, specifically the mathematical representation of wake aerodynamics. These representations can be broadly distinguished between white-box (physics-based) and black-box (data-driven) models [48]. White-box models rely on analytical or empirical governing equations and differ primarily in fidelity. These include the introduced models in Section 2.1.4. Alternatively, black-box or data-driven models, often referred to as surrogate models, approximate the input-output relationship without explicit physics equations [48]. These include ML architectures ranging from MLP models [49] to decision-tree ensembles like XGBoost [27]. Within data-driven modeling, a critical distinction is drawn based on the model's ability to generalize to unseen wind farm topologies. Layout-specific approaches fit a model to a single wind farm, often requiring retraining if the farm layout changes. To address this limitation, emerging layout-agnostic approaches, particularly GNNs, represent turbines as nodes and wakes as edges, allowing the model to generalize to unseen layouts without retraining [8].

Wake steering approaches also differ significantly in their control strategy [13, 34]. Traditional open-loop control relies on pre-calculated lookup tables based on static conditions, whereas closed-loop control utilizes real-time feedback from sensors to adjust yaw dynamically. Distinct from these are model-free approaches, which do not rely on a wake model at all. Instead, they learn the optimal policy through direct interaction with the environment, typically employing Reinforcement Learning agents to converge on optimal settings iteratively [50].

Regardless of the model used, an optimization algorithm is required to search the solution space for optimal yaw angles. There are many different optimizers available, which can broadly be divided into gradient-based and gradient-free categories. For example, Harrison-Atlas et al. apply the sequential least squares quadratic programming algorithm, a gradient-based optimizer, computing gradients using backpropagation through their neural network [14]. In contrast, two examples of gradient-free optimizers are [28] and [51]. [28] uses a Genetic Algorithm whereas [51] uses the particle swarm optimization algorithm.

Finally, papers mention different validation schemes. Numerical validation is the most common, where authors run simulations using models of varying fidelities [12]. Beyond simulation, experimental validation involves testing turbine models in wind tunnels [52]. The highest level of validation is found in field tests, where approaches are deployed on operating commercial wind farms to confirm power gains under real-world atmospheric conditions [10].

This thesis focuses on the wake model component by implementing surrogate models. They are trained on data generated by an analytical model. Because this underlying engineering model operates under steady-state assumptions, the resulting surrogates are categorized as quasi-static. The specific machine learning architecture employed in this research is the GT, which is introduced in the following section.

2.2 Graph Transformers

In contrast to other domains, such as images, wind farms are arranged in irregular spatial layouts. They do not fit into a strict grid-like structure [24]. Graphs offer a way to model such irregular data. Following Battaglia et al. [53], a graph can be defined as a 3-tuple $G = (u, V, E)$. u denotes global attributes, such as air density or free-stream wind speed. $V = \{\mathbf{v}_i\}_{i=1:N^v}$ is the set of nodes with cardinality N^v. Each node i is described by attributes $\mathbf{v}_i$. When modeling a wind farm, these nodes represent the turbines and are described by attributes such as rotor diameter and hub height. Lastly, $E = \{(\mathbf{e}_k, r_k, s_k)\}_{k=1:N^e}$ is the set of edges with cardinality N^e. Each edge k has a sender node with index s_k and a receiver node with index r_k, and is represented by attributes $\mathbf{e}_k$. In the context of wind farms, an edge attribute might be the distance between two turbines.

In the framework by Battaglia et al., Graph Networks are functions that operate on graphs [53]. Usually, they are implemented with neural networks and therefore are most often referred to as GNNs. At the core of a GNN is a block that takes in a graph representation, performs some computations, and outputs an updated graph representation. Typically, a GNN employs a series of such blocks.

Over the last decades, a vast array of different GNN blocks were proposed. In original formulations, each block collected features from neighboring nodes and aggregated them using unweighted summations [54]. Veličković et al. extended this idea by using attention over a node's neighbors leading to Graph Attention Networks (GAT) [55]. This approach allows neighbors to be weighted differently when updating a node's state.

A problem of traditional GNN approaches is their limitation by local neighborhood, which can lead to over-smooting or over-squashing due to long range dependencies [56]. A recently emerging GNN architecture that address these issues are GTs. Similarly to GAT, the core mechanism of GTs is self-attention [56]. However, GTs allow each node to attend to every other node. In its most basic formulation, the node feature matrix $\mathbf{X}$ is first projected into query, key and value space:

$$\begin{aligned} \mathbf{Q} &= \mathbf{X}\mathbf{W}_Q + \mathbf{b}_Q \\ \mathbf{K} &= \mathbf{X}\mathbf{W}_K + \mathbf{b}_K \\ \mathbf{V} &= \mathbf{X}\mathbf{W}_V + \mathbf{b}_V \end{aligned} \tag{2.8}$$

The attention mechanism is then defined as

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V},$$

where d denotes the feature dimensions of $\mathbf{Q}$ and $\mathbf{K}$. An output projection transforms the results back into feature space

$$\mathbf{X}' = \mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})\mathbf{W}_o + \mathbf{b}_o. \tag{2.9}$$

Usually, multiple attention heads are employed in parallel to allow the model to attend to information from different representation subspaces [28]. Each head i computes its own query, key, and value matrices $(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)$ to produce an independent attention output. The results are concatenated before being passed through the output projection.

By allowing each node to attend to every other node, the graph is effectively seen as a fully-connected structure, stripping away any topological information and its inductive bias. To re-introduce this structural awareness, GTs can inject structural encodings and positional encodings into the node representations. These encodings allow the attention mechanism to distinguish between topologically close neighbors and distant nodes.

The GT approach of this thesis is based on the work by Li et al. [28]. They propose a framework for wind farm power prediction and yaw steering optimisation using GTs. By encoding wind farms as fully connected graphs , the model bypasses the limitations of predefined graph connectivities and over-smoothing often found in traditional GNNs. The self-attention mechanism is designed to autonomously identify aerodynamic interactions and wake effects between turbines. To achieve this, the authors utilized synthetic training data generated by an adapted version of the Plant Layout Generator (PLayGen) and the simulator PyWake. Trained on an "enhanced" dataset generated via genetic algorithms and the industrial-standard simulator PyWake, the surrogate model achieves a relative prediction accuracy of 99.8% on previously unseen layouts.

2.3 Uncertainty Quantification

ML can be seen as the process of extracting a model from data [57]. The goal is to find a model that fits the data well. This is typically done by minimizing the empirical risk, which is the average loss over the dataset [58]. While minimizing empirical risk yields a model that performs well on average, it does not guarantee correctness for individual samples. A model is "never provable correct" and thus any derived prediction is subject to uncertainty [57]. This uncertainty is also called predictive uncertainty [59]. To quantify predictive uncertainty, it is helpful to identify where it is coming from. Literature often distinguishes two main types of uncertainties: Aleatoric and epistemic uncertainty [57]. Aleatoric uncertainty, also referred to as statistical or data uncertainty, is the "variability in the outcome of an experiment which is due to inherently random effects" [57]. It is not a property of the model but is inherent to the data-generating process, thus it is irreducible [60]. Examples are noisy or erroneous data [59]. Epistemic uncertainty, also known as systematic or knowledge uncertainty, arises from insufficient information about the ideal model [57]. Epistemic uncertainty can be reduced by collecting more information. One type of epistemic uncertainty is model uncertainty [59]. It describes the uncertainty that is related to the choice of the model architecture and training process. Distributional Uncertainty is another type of epistemic uncertainty [61]. It is caused by incorrect assumptions about the data-generating process, e.g. a mismatch between the training and test set.

There are various approaches to quantify predictive uncertainty. One approach is deep heteroscedastic regression [33]. In this setting, there is a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, where $\mathbf{x}_i$ are inputs, regressors, covariates or independent variables and y_i is the output, regressand, response variable, or dependent variable. A model f_θ is fit on the dataset by optimizing its model parameters θ. In a deterministic regression setting, this model only outputs a point-prediction $f(\mathbf{x}) = y$. The approach of deep heteroskedastic regression is to use a neural network based model to parameterize a probability distribution. Most commonly, a Gaussian distribution is used. The model then outputs a mean and variance prediction, both dependent on the input. Jordahn et al [33] refer to such networks as mean-variance networks. Traditionally, mean-variance networks are fitted on a training dataset using the Negative Log Likelihood (NLL) as a loss function. Thereby, both the mean and variance predictions contribute to the gradients during learning.

Jordahn et al. identify four flaws of existing deep heteroscedastic regression approaches [33]. First, the training is prone to optimization issues. Due to the NLL, mean-variance networks often lead to inflated variances. Second, optimizing a neural network solely to estimate the mean function can induce last-layer representation collapse. This phenomenon occurs when the model's feature extractor discards input dimensions that do not contribute to minimizing the Mean Squared Error (MSE). Consequently, the resulting latent representations become overly compressed, potentially failing to span the necessary subspace required to characterize the data's heteroscedasticity. Third, they argue that overfitting of the residual variance is an overlooked problem. Lastly, they criticize the practicality of previous methods. As an example, they state that a practical uncertainty-aware approach should retain its mean prediction compared to a deterministic approach.

Jordahn et al. address these issues by proposing learning variance estimator post-hoc on a hold-out dataset using the embeddings of a trained and frozen mean estimator. The variance estimator has the form

$$\sigma_\phi^2(x^*) = \mathrm{sp}\left(\sum_{l \in L_\sigma} W_l^T z^l(x^*)\right), \tag{2.10}$$

where z^l are the mean estimator's latent representations, W_l are the parameters, and sp is the softplus function. This thesis' uncertainty quantification approach is based on the work by Jordahn et al.. It was selected for its simple to implement and fast post-hoc procedure that presumably does not require much hyperparameter tweaking and retains the mean predictions. The adjusted method is explained further in Section 3.3.

3 Methodology

This chapter presents the methodology developed for this thesis. It begins in Section 3.1 with the description of the synthetic data pipeline designed to simulate realistic heterogeneous wind farm scenarios. Section 3.2 introduces the deterministic modeling approach, comparing an MLP baseline with a GT architecture. This is followed by Section 3.3, which extends the deterministic models into a probabilistic framework by implementing both homoscedastic and post-hoc heteroscedastic variance estimation techniques. Finally, Section 3.4 outlines the procedures used to evaluate the models' generalization performance when transitioned from synthetic environments to a real-world SCADA dataset. The core tools and libraries used are scikit-learn [62], pandas [63], NumPy [64], SciPy [65] and PyTorch [66]. The computations were mostly done on the bwUniCluster utilizing its CPU nodes with 96-core *AMD EPYC 9454* and GPU nodes with *NVIDIA A100s* and *H100s*.

3.1 Synthetic Data Pipeline

To address RQ1, a heterogeneous dataset is required that contains diverse turbine configurations and a broad range of atmospheric conditions. To generate data that is physically plausible and statistically representative required making specific design choices. In developing this pipeline, the following five core challenges were addressed:

- **Spatial Configuration:** Establishing a method for collecting realistic wind farm layouts.
- **Turbine Heterogeneity:** Determining the logic for assigning diverse turbine models and specifications across the layouts.
- **Atmospheric Conditions:** Defining the multidimensional space of atmospheric conditions, including wind speed distributions for example.
- **Operational Parameters:** Setting turbine-specific yaw angles to simulate both standard and wake-steering scenarios.
- **Ground Truth Computation:** Applying numerical models to derive the reference performances.

An overview of the architecture of the resulting generation pipeline is illustrated in Figure 3.1. The following subsections detail the implementation of each step, justifying the selected methods and discussing the trade-offs made during the design process.

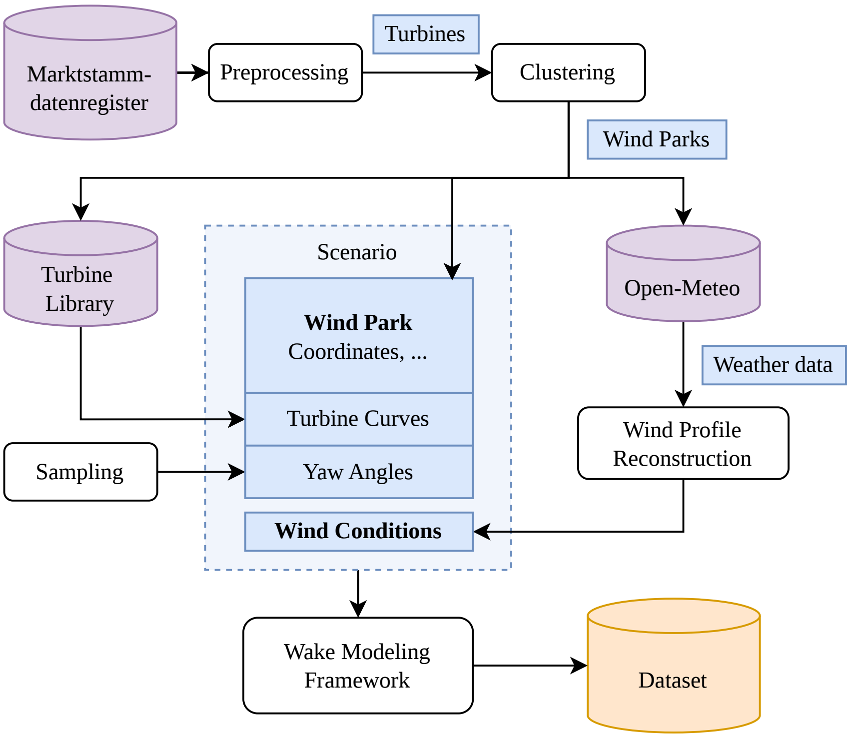


Figure 3.1: Simulation scenarios are constructed by combining real-world clustered wind farm data (MaStR [67]), matched turbine characteristics (Wind Turbine Power Curve Archive [68]), historical weather profiles (Open-Meteo [69]), and sampled yaw angles. These scenarios are subsequently evaluated using a wake modeling framework to produce the final dataset.

3.1.1 Spatial Configuration and Turbine Heterogeneity

Most existing literature focuses on single wind farms or relies on synthetically generated layouts to produce training data, typically restricted to a single turbine model [14, 18]. In contrast, this thesis requires training data featuring multiple turbine types within the same wind farm. While existing synthetic generation tools, e.g. NREL's layout generator [14], could be extended via heuristics to distribute various models, such an approach would require arbitrary assumptions regarding model density and spatial placement. To avoid such design choices, this thesis adopts a data-driven approach utilizing real-world records from the MaStR [67].

The MaStR is a comprehensive online registry maintaining master data for actors in the German energy market, including detailed records of wind turbines [67]. Records describing 32,881 wind turbines were extracted to form the basis of the dataset. To address inherent data inconsistencies, a preprocessing pipeline was implemented. First, a spatial bounding box was applied to filter out erroneous entries located outside Germany's borders. Second, to handle missing information, a Random Forest model was employed to impute 211 missing hub heights based on rotor diameter and rated power. This ensured data completeness without discarding otherwise valid records. Third, the dataset was spatially deduplicated to remove corrupted or overlapping records, ensuring that the final data reflected actual turbine placements accurately. At this step, 2, 684 records were dropped. Finally, an outlier turbine with a hub height of 300 m was removed. Following the preprocessing and cleaning steps, the final dataset comprised 30, 188 entries. The relevant features retained for the training phase are summarized in Table 3.1. While the raw dataset included manufacturer and model information, these fields lacked normalization, for instance, multiple inconsistent entries for the same entity, rendering it unsuitable for direct use. This information was therefore discarded and the assignment of turbine models was instead performed using the alternative approach described further below.

Table 3.1: Summary statistics of the attributes of the filtered MaStR turbine dataset.

Metric	Latitude [°N]	Longitude [°E]	Hub height [m]	Rotor diameter [m]	Rated power [kW]
Min	47.45	5.23	43.00	50.04	560.00
Max	54.90	15.00	203.00	180.00	8000.00

Following data cleaning and imputation, individual turbines were grouped into distinct wind farms to reconstruct realistic farm layouts. This was achieved using a spatial clustering approach based on the DBSCAN algorithm [70]. First, the distance between each pair of turbines was computed using the haversine formula to account for the spherical curvature of the Earth. Then, rather than employing a static distance metric, the clustering used a dynamic, turbine-specific threshold to account for different-sized turbines. This ensured that smaller turbines need to be closer together to be considered part of the same farm, whereas larger turbines are allowed to be further apart. The threshold distance for any given pair was defined as the average of their respective rotor diameters multiplied by a predefined wake factor. A factor of 10 was chosen, meaning turbines had to be separated by at least 10 rotor diameters to be considered part of different wind farms. This value was selected empirically to balance the resulting clustering structure: smaller values led to a fragmentation into many small farms, while larger values caused distinct farms to merge into overly large clusters. All isolated turbines as well as clusters with only two turbines were dropped, ensuring the final dataset consisted solely of multi-turbine layouts suitable for farm-level modeling.

The remaining dataset comprised 28, 252 turbines clustered into 2652 wind farms. Figure 3.2 displays a cutout of this dataset. Many wind farms were quite small, with a median size of 7. 10% of the farms consisted of more than 22 turbines. A total of 75 had more than or equal to 40 turbines. Based on a tolerance of 5 m, 1, 595 farms had at least two turbines with different rotor diameters, 872 at least three, and 504 at least four different diameters. This confirms the heterogeneity of the dataset.

After the turbines were successfully clustered into distinct wind farms, the final preparation step involved assigning specific aerodynamic models to each physical turbine. This assignment is critical to provide the necessary performance characteristics, in particular

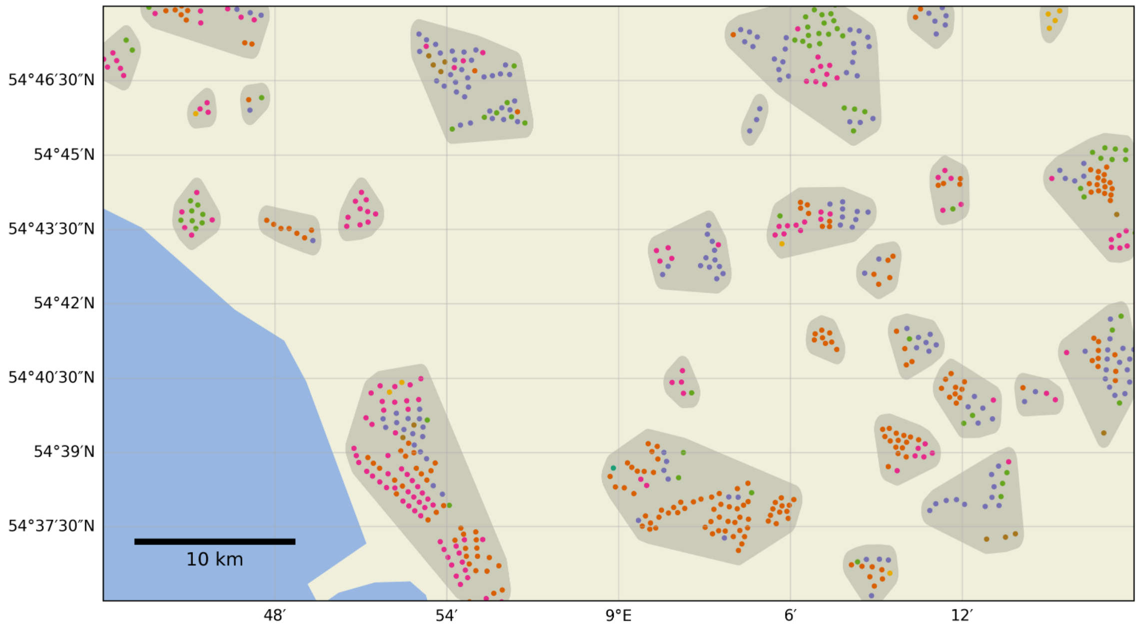


Figure 3.2: Cutout of the synthetic dataset featuring heterogeneous wind farms of varying sizes. The dots indicate individual wind turbines and the shaded regions denote their respective assigned wind farms. The color indicates different turbine types with varied rated power. The map was created with cartopy [71] using map tiles by Natural Earth.

power curves and thrust curves, required for subsequent wake simulations. First, the library *NREL Wind Turbine Power Curve Archive* [68] of wind turbine models was loaded and formatted for compatibility with the FLORIS simulation framework [40]. With the reference library established, each physical turbine from the MaStR dataset was paired with a corresponding aerodynamic model. This assignment was executed by finding the closest match based on rated power, specifically by minimizing the absolute difference between the capacity of the physical turbine and the rated power of the available models in the reference library.

Finally, to facilitate spatial analysis and aerodynamic wake modeling, the physical distances between turbines within each wind farm had to be quantified in meters. Because the raw metadata provided locations in geodetic coordinates (latitude and longitude), which are unsuitable for direct Euclidean distance calculations due to the Earth's curvature, the coordinates for each wind farm were projected onto a local two-dimensional Cartesian coordinate system (x, y). The *pyproj* library was used for this computation [72]. This spatial transformation was performed independently for each farm to minimize projection distortion, thereby preserving the structural geometry of the layouts and ensuring accurate intra-farm distance calculations for subsequent modeling.

3.1.2 Atmospheric Data

To create plausible scenarios, realistic environmental conditions had to be collected. Many existing approaches sample atmospheric variables independently from uniform distributions, e.g. wind speeds between 8 and 15 m/s and TIs between 5% and 15% [28]. More sophisticated frameworks utilize probabilistic modeling, such as Weibull distributions for wind speed or truncated normal distributions for wind shear [12]. In contrast, this thesis adopts an empirical approach instead by acquiring real wind profiles. This meteorological data was sourced via the Open-Meteo Historical Forecast API [69], leveraging the ICON-D2 model [73].

For each previously clustered wind farm, the geographical centroid was calculated to serve as the reference coordinate for atmospheric sampling. To ensure a diverse representation of weather conditions, random hourly timestamps were sampled between January 1, 2023, and December 31, 2025. The queried variables included air temperature (2 m), maximum wind gusts (10 m), and mean wind vectors (speed and direction) at heights of 10 m, 80 m, 120 m, and 180 m. A total of $59,600$ weather conditions were sampled. To transform the raw API outputs into the parameters required for subsequent wake modeling, a processing pipeline was implemented. In Figure 3.1, this step is denoted as Wind Profile Reconstruction. It should be noted that while this pipeline is designed to generate a diverse and representative set of semi-realistic conditions, it relies on empirical approximations rather than strict physical conservation laws. Figure 3.5 shows the distributions of wind speed, TI, shear and veer across the weather samples.

Wind Speed and Direction The wind speed and direction at 120 m elevation were selected as references for further processing. These are also the values that were passed to the wake modeling framework later on. Figure 3.3 visualizes the distribution of the wind speed. As can be seen, the wind speed closely follows a Weibull distribution with a shape $k \approx 2.0$ and scale $\lambda \approx 7.5$. While the shape is similar to the one used by Duthé et al. [12], the scale is much lower, resulting in lower wind speeds. To restrict the simulations to realistic operational states, the data for the actual dataset was filtered to include only samples with mean wind speeds between 3 m/s and 25 m/s, corresponding to typical cut-in and cut-out thresholds.

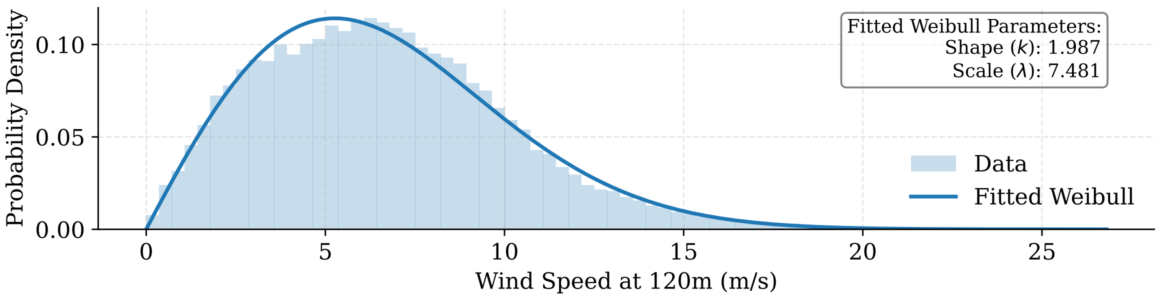


Figure 3.3: Wind speeds collected from weather forecasts follow a Weibull distribution matching distribution assumptions applied in other approaches but with lower scale in comparison [12].

Wind Shear and Wind Veer To estimate the shear coefficient α across the multi-height wind profile (80 m, 120 m, and 180 m), the shear relationship (cf. Equation 2.7) was linearized using a logarithmic transformation:

$$\ln(u) = \alpha \cdot \ln(z) + C$$

The shear exponent α was then computed for each timestamp via least squares regression in the log-log space. To mitigate noise at low operating speeds, the shear value was defaulted to 0.2 for profiles where any wind speed fell below 4 m/s. Finally, the calculated exponents were clipped to a plausible range of [-1.0, 2.0].

To calculate veer, the raw wind directions at the evaluated heights were first unwrapped to prevent discontinuities at the 360-degree boundary. The veer was then determined by calculating the slope of the wind direction relative to height, yielding a gradient in degrees per meter. Similar to the shear calculation, veer was overridden to zero during low wind speed conditions (below 4 m/s) to prevent erratic directional shifts from distorting the dataset.

Figure 3.5 includes a visualization of the joint distribution of veer and shear in the dataset. It reveals a primary concentration of data points at low shear values where veer remains negligible. As the shear coefficient increases beyond 0.20, a positive correlation emerges, characterized by increased variance in veer values.

Figure 3.4 further visualizes the diurnal variations of wind shear and veer, revealing a distinct cyclic pattern throughout the 24-hour period. Both parameters exhibit their lowest values and minimum variance during the midday hours (approximately 10:00 to 16:00), when the wind profile is most uniform. Conversely, both shear and veer increase significantly during the nocturnal and early morning periods (20:00 to 06:00), leading to the conclusion that the hour of the day serves as a proxy for both parameters.

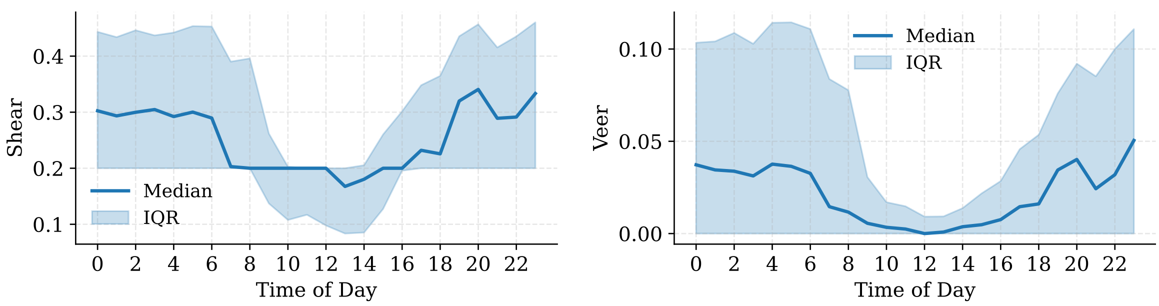


Figure 3.4: Diurnal variation of the wind shear exponent (left) and wind veer (right) over a 24-hour cycle. Both parameters exhibit a distinct cyclic pattern, reaching minimum values and variance during midday, and maximum values at night.

Turbulence Intensity As noted earlier, TI is defined as the ratio of the standard deviation of wind speed to the mean wind speed [12]. Because the standard deviation was not directly available from the raw data it had to be approximated utilizing the available wind gust $u_{\text{gust},10m}$ and mean wind speed $u_{\text{mean},10m}$ data at 10 m. First, the standard deviation of the wind speed at 10 m σ_{10m} was estimated as

$$\sigma_{10m} = \frac{\max(u_{\text{gust},10m} - u_{\text{mean},10m}, 0)}{k},$$

where $k = 3.0$ acts as an empirical scaling factor. An empirical height decay factor of 0.75 was applied to estimate the standard deviation at a nominal hub height of 120 m (σ_{120m}). The final TI at 120 m was computed by dividing σ_{120m} by the 120 m mean wind speed, with the result strictly bounded between 0.02 and 0.40 to ensure stability in the subsequent wake simulations. The empirical factors were adjusted such that the resulting TI values predominantly fall between 3% and 25%, which roughly matches the samples used by Duthé et al. [12]. Figure 3.5 displays the joint distribution of TI and wind speed. The joint distribution shows a non-linear relationship where TI variability is highest at low wind speeds and converges toward a stable, lower baseline as wind speed increases.

3.1.3 Operational Parameters

Following the processing of the atmospheric data, discrete simulation scenarios were generated by pairing specific wind farm layouts with corresponding meteorological conditions. Because large wind farms containing many turbines are inherently underrepresented in the dataset, a proportional sampling strategy was implemented to prevent smaller layouts from dominating the training data. Specifically, the number of scenarios generated for a given wind farm was scaled dynamically, set to 15 times its total number of turbines. For each of these generated scenarios, a localized atmospheric condition was randomly sampled from the farm's corresponding meteorological records. Furthermore, random yaw angles were assigned to each turbine. These angles were generated via Latin Hypercube Sampling, drawn from a normal distribution with a mean of 0° and a standard deviation of 10.0°. This sampling technique ensured a stratified coverage of the multidimensional parameter space, capturing extreme and baseline yaw configurations.

3.1.4 Ground Truth Computation

To create the ground truth for each scenario the framework FLORIS was used as a simulation tool [40]. Specifically, the CC wake model was employed utilizing its default parameter set [46]. This model was selected for its superior ability to capture cumulative wake interactions and secondary steering effects compared to traditional analytical models. It should be noted that the default parameters are primarily calibrated for offshore wind farm conditions, which introduces a degree of approximation when applied to the onshore layouts present in this dataset.

While FLORIS natively supports the parallel execution of multiple simulation scenarios, this built-in functionality is restricted. It does not allow for concurrent variations in atmospheric shear and veer values across different scenarios. To overcome this limitation

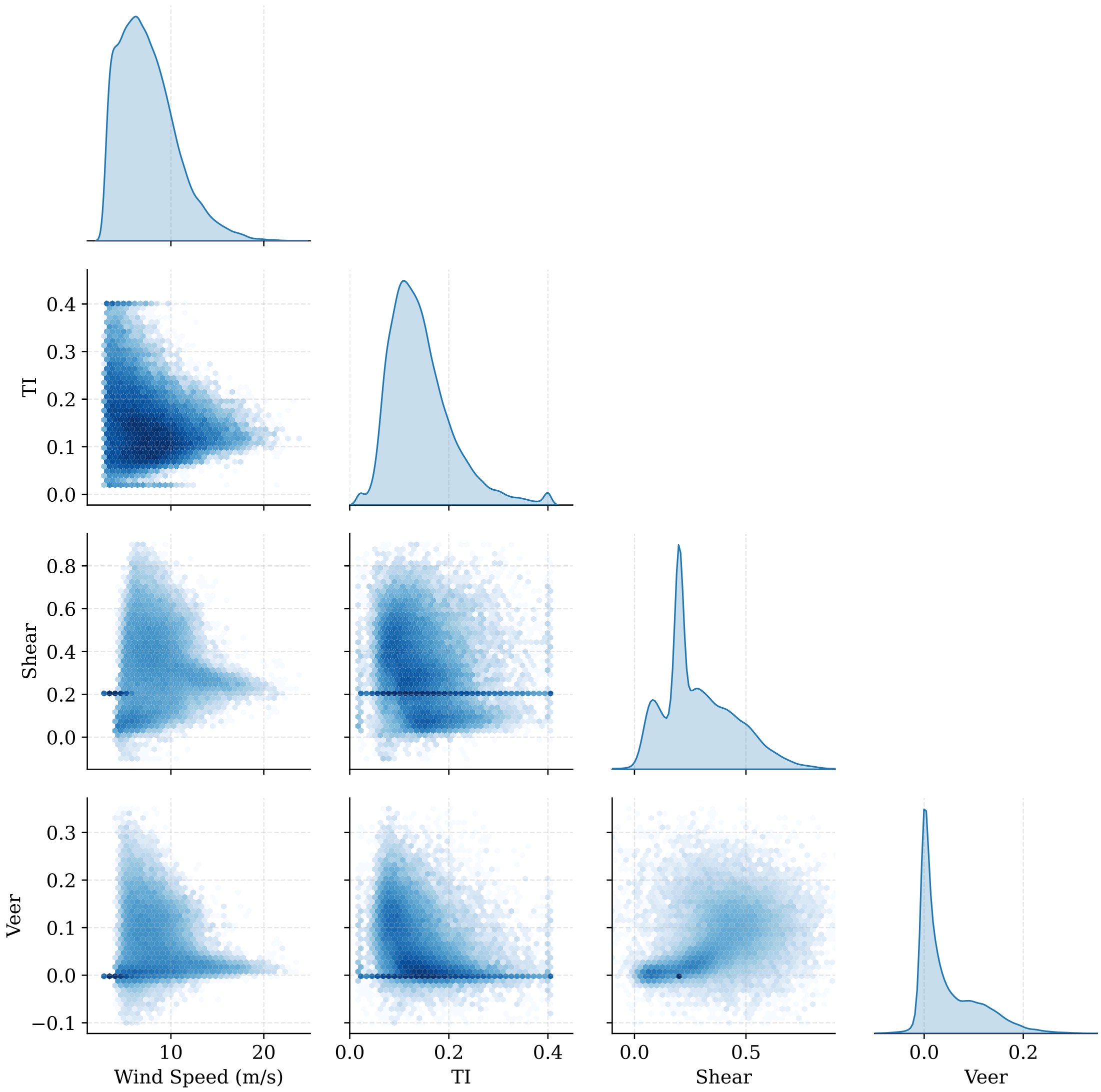


Figure 3.5: Pairplot showing marginal distributions and bivariate joint distributions of wind speed, TI, shear, and wind veer. Although default values are evident, the extensive range of values and varying densities across both the marginal and joint distributions emphasize the wide variety of atmospheric conditions included in the dataset. Inspired by [12].

and maintain computational efficiency across the massive dataset, a custom parallelization strategy was implemented. This approach distributes the independent simulation scenarios across multiple CPU cores, enabling the simultaneous processing of heterogeneous atmospheric conditions.

During the simulations, the absolute power generated by every individual turbine was recorded for each scenario. Additionally, a secondary set of computations was executed with the aerodynamic wake model explicitly disabled. This provided a crucial reference measurement of the theoretical power each turbine would generate in complete isolation, allowing for the quantification of the power deficits directly attributable to aerodynamic wake interactions.

Table 3.2: Dataset stratification. Splitting was performed at the farm level to prevent data leakage between the training, validation, and test sets.

Split	**Farms**	**Turbines**	**Scenarios**
Train	1, 788	17, 782	258, 440
Validation	447	4, 487	63, 654
Test	417	5, 983	68, 675

The dataset was partitioned into training, validation, and test sets using a farm-level splitting strategy to prevent data leakage. The test split was defined to include the real-world reference farm, all farms with more than 64 turbines, and 400 random farms. The remaining data was then split 80/20 between the training and validation sets, respectively. Table 3.2 lists the respective number of farms, turbines and scenarios per split. Figure 3.6 illustrates the spatial dimensions of the wind farms across the three splits, highlighting a noticeably larger extent in the test set due to the inclusion of the largest farms.

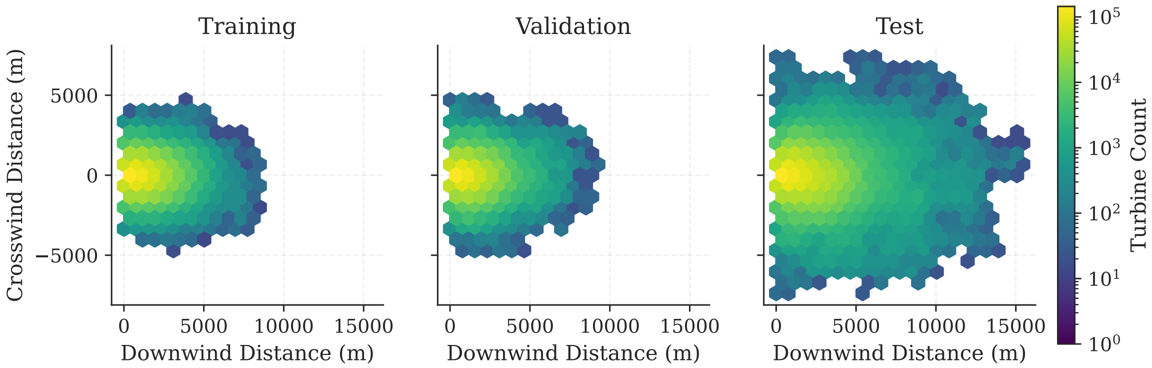


Figure 3.6: The test set covers larger spatial dimensions compared to the training and validation splits because it includes all wind farms containing more than 64 turbines. The hexbin plots display turbine counts on a logarithmic scale relative to downwind and crosswind distances.

3.2 Deterministic Modeling

This section deals with the methodology to examine RQ1 using the previously introduced dataset. It will describe the MLP baseline's and the GT's architecture as well as the corresponding training procedure.

3.2.1 Setup

To account for the diverse turbine models and varying power capacities within the dataset, this approach transforms the task from predicting absolute power to predicting the fraction of rated power, hereafter referred to as normalized power. This normalization allows the model to treat turbines as equivalent units regardless of their individual scales. It also prevents turbines with a higher capacity from dominating the prediction error.

Predicting the normalized power of individual turbines within a wind farm can be seen as a multivariate regression task on a graph $G = (\mathbf{u}, V, E)$. Here, $\mathbf{u}$ captures the global environmental context, while nodes $V = \{\mathbf{v}_i\}$ represent the turbines and their specific features. Adopting the GT approach from [28], the graph is treated as fully connected without explicit edge features. Therefore, the model can be expressed as

$$\mathbf{P}_{\text{true}} = f(\mathbf{u}, V) + \epsilon$$

where $\mathbf{P}_{\text{true}}$ is the vector of normalized turbine powers, f is the mapping function and ϵ are irreducible errors.

Table 3.3: Overview of global and turbine-level features.
*These proxy features were only used for subsequent modeling without explicit shear and veer.

Scope	**Feature**
Global	Wind speed
	Wind direction
	Wind veer
	Wind shear
	Turbulence intensity
	Time of day*
	Day of year*
Turbine-level	Local coordinates
	Hub height
	Rotor diameter
	Thrust curve
	Power curve
	Yaw angle

Table 3.3 provides an overview of both the available global variables as well as turbine-level features. To ensure numerical stability, all inputs underwent a series of domain-specific normalization and transformation steps. Global features define the ambient conditions affecting the entire wind farm. These include the atmospheric parameters, wind speed, wind veer, wind shear, and TI, as well as the temporal context. Wind speed was scaled relative to a 25 m/s reference. The vertical profile and turbulence components (veer, shear, and TI) were standardized using z-score normalization. To capture periodic environmental cycles, the time of day and day of year were transformed into cyclical features using sine and cosine encodings. Turbine-level features characterize the individual technical specifications and operational states of each unit. Both the power curves and the target power variables were normalized by the turbine's rated power, converting absolute Wattage into a dimensionless capacity factor in the range [0,1]. Static design parameters were scaled by reference values: hub heights were divided by 100 m, and rotor diameters by 150 m. Individual yaw angles were normalized relative to a 30° maximum offset. To enforce wind-direction invariance, coordinates were rotated such that the x-axis aligns with the downwind direction and the y-axis aligns with the crosswind direction. These rotated coordinates were then scaled by 1000 m and translated so that the most upwind turbine is anchored at x=0, with the farm centered along the y-axis.

3.2.2 MLP Baseline Model

The baseline architecture is implemented as an MLP that processes each turbine independently. The global variables $\mathbf{u}$ are concatenated to the turbine-level features $\mathbf{v}_i$ of each node. The input to the model is denoted by X_{in}. The model is a deep feed-forward neural network utilizing GELU activations, with both the network depth and hidden layer dimensionality treated as configurable hyperparameters. To ensure the outputs remain within a physically plausible range, the final layer employs a Sigmoid activation.

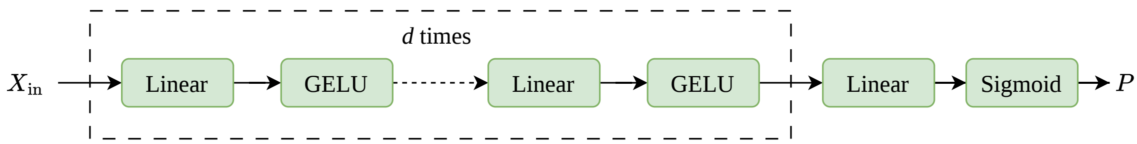


Figure 3.7: Schematic of the MLP architecture. The model consists of d repeated blocks of Linear and GELU activation layers, followed by a final Linear layer with a Sigmoid activation to ensure outputs remain within physically plausible bounds.

The architecture is evaluated through two variants, defined by their feature selection. The first variant, denoted by MLP-Spatial, uses all of the atmospheric and turbine-level features, including coordinates but excluding temporal encodings. The coordinate-blind variant, denoted by MLP-NonSpatial, is identical to the other configuration, but with the coordinate features zeroed out. This variant serves to quantify the model's performance when it does not have access to spatial coordinates.

3.2.3 Graph Transformer

The architecture of the GT is based on the work by Li et al. [28]. Unlike the MLP models, which treat turbines as isolated nodes, the GT utilizes a self-attention mechanism to be able to learn interactions between all turbines in a farm, allowing the network to learn the downstream effects of wake propagation. Similar to the MLP, the global variables $\mathbf{u}$ are concatenated to the turbine-level features $\mathbf{v}_i$ of each node, and the input to the model is denoted by X_{in}. The architecture follows an encoder-processor-decoder structure.

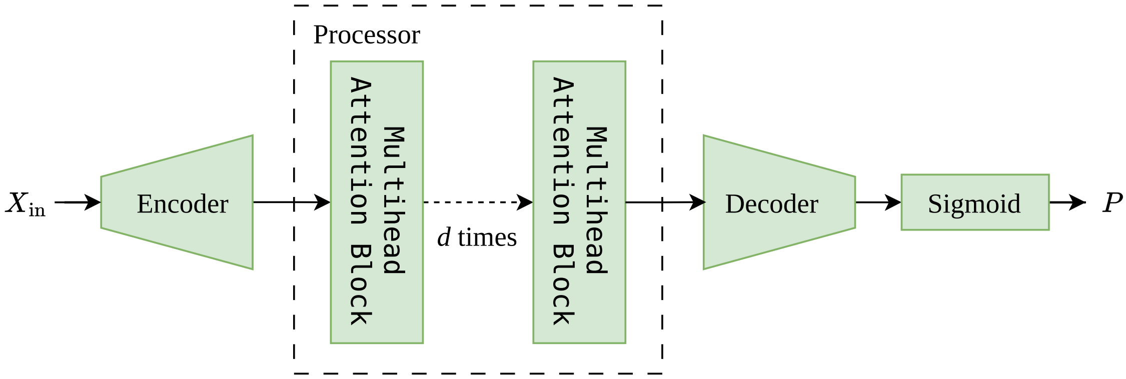


Figure 3.8: Architecture of the GT. The model utilizes an encoder-processor-decoder structure where the encoder maps heterogeneous inputs to a embedding space, the processor employs d repeated multihead attention blocks to model inter-turbine wake interactions, and the decoder produces the normalized turbine-level power estimations P via a Sigmoid activation.

The encoder prepares turbine-level and global features and maps them into a predefined embedding dimension. It works as follows: First, the turbine-specific thrust and power curves are processed by independent curve encoders, comprising two 1D-convolutional layers interspersed with GELU activations. To handle variable curve lengths, an adaptive average pooling layer is applied. These curve embeddings are subsequently concatenated with the remaining turbine-level and global features before being fed into a node encoder. This module, consisting of a pre-activation layer normalization, two linear layers, and a GELU activation, maps the aggregated features into the model's primary embedding dimension.

The core of the model is a sequence of d multihead attention blocks. Each block consists of two primary sublayers: a Scaled Dot-Product Multihead Attention (SDPA) mechanism and a Feed-Forward Network (FFN). Each sublayer is wrapped in a residual connection to preserve gradient flow. The SDPA layer enables the model to learn dependencies between turbines. Given the input node embeddings, the block first applies a layer normalization followed by multihead attention. By treating the wind farm as a fully connected graph, the attention mechanism allows each turbine to aggregate information from all other turbines. The FFN layer operates on each node independently. It consists of two linear transformations with a GELU activation in between.

The final node representations are passed to a node decoder, which reduces the embedding dimension back to a single scalar per turbine. Similar to the MLP architecture, a Sigmoid activation is applied to map the output to a dimensionless capacity factor.

3.2.4 Experimental Setup and Evaluation

The models were trained by minimizing the MSE between the predicted and ground-truth normalized power values. All architectures used the AdamW optimizer with a learning-rate schedule consisting of a linear warmup phase followed by cosine annealing. Early stopping was applied based on the validation loss to reduce overfitting.

Rather than relying on computationally expensive hyperparameter optimization, a manual grid search was performed. Hyperparameters were selected through iterative trials monitoring the results on the validation set. This included testing different model depths, hidden dimensions, batch sizes, learning rates, and number of attention heads, among other parameters. For the MLP-Spatial, the chosen configuration used 3 layers, a hidden dimension of 128, a batch size of 256, a peak learning rate of 3×10^{-3}, a 5-epoch warmup phase, and a maximum of 50 training epochs. The MLP-NonSpatial used the same settings, with coordinate inputs disabled. For the GT, the selected configuration used 4 layers, a node embedding dimension of 256, 4 attention heads, a batch size of 256, a peak learning rate of 10^{-3}, a 10-epoch warmup phase, and a maximum of 200 training epochs.

To ensure the statistical significance of the results, every architecture was trained and evaluated across five different random seeds. To assess model performance, three standard regression metrics are computed: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the Coefficient of Determination (R^2). While the MSE serves as the primary loss function during training, the RMSE and MAE provide more interpretable measures of the average prediction error in terms of normalized power units. The R^2 score is utilized to quantify the proportion of variance in the power output captured by the model relative to a baseline mean predictor. In addition to these power metrics, the models' ability to predict the wake loss is evaluated. The wake loss represents the reduction in power production caused by the aerodynamic interference of upstream turbines. For each turbine, the normalized wake loss WL is calculated as the fraction of power lost relative to its free-stream power P_{free}, which is the power the turbine would produce if it were operating in isolation under the same ambient wind conditions:

$$\text{Normalized WL} = \frac{p_{\text{free}} - p}{p_{\text{free}}} \tag{3.1}$$

In this expression, P represents either the predicted or observed power output. By calculating the same metrics for these wake loss values, the model's effectiveness in capturing wake interactions can be evaluated.

3.3 Probabilistic Modeling

This section deals with the methodology for RQ2. It will introduce the probabilistic, uncertainty-aware baseline as well as the post-hoc heteroscedastic variance estimator. The section will conclude with the experimental setup, including the training procedure.

3.3.1 Setup

While the initial task is framed as a deterministic mapping, the idea of RQ2 is to quantify the uncertainty of a prediction. Therefore, the model output for a turbine is no longer a single scalar value, but a probability density function. As a simplification, a Gaussian distribution is used. Therefore, the model parametrizes the mean and variance of a Gaussian.

While the previously introduced models are prone to structural uncertainty, the underlying data is deterministic because it is generated using a deterministic wake modeling framework. Consequently, since the GT utilizes a complete description of each sample, no aleatoric uncertainty is present essentially. To intentionally introduce aleatoric uncertainty for this second task, and because they are unavailable in the real-world SCADA dataset, wind shear and wind veer were omitted. Instead, "time of day" and "day of year" were used as temporal proxies.

3.3.2 Probabilistic Baseline

A homoscedastic approach using pretrained models served as a baseline. In this configuration, the variance σ^2 is assumed to be constant across all wind farm scenarios and turbines, representing the global expected error of the model. This constant variance is estimated post-training by calculating the MSE on a calibration set:

$$\sigma^2_{\text{base}} = \frac{1}{N_{\text{val}}} \sum_i (p_{\text{true},i} - \hat{p}_i)^2,$$

where N_{val} is the total number of samples in the calibration set, $p_{\text{true},i}$ is the true ground-truth value for the i-th turbine in the validation set, and $\hat{p}_i$ is the corresponding value predicted by the model. This baseline captures the "average" difficulty of the task but fails to account for the fact that certain conditions are inherently harder to predict than others.

3.3.3 Post-hoc Heteroscedastic Variance Estimator

To capture the input-dependent nature of prediction errors, this thesis adapts the heteroscedastic post-hoc framework proposed by Jordahn et al. [33] that was introduced in Section 2.3. This approach is integrated into the GT architecture as visualized in Figure 3.9.

$d + 1$ latent representations, denoted as $Z = \{z_0, z_1, \ldots, z_d\}$, are extracted from the model, where $z_0, \ldots, z_{d-1}$ correspond to the inputs of each transformer attention block and z_d represents the final output of the transformer stack. Each latent tensor is passed through a dedicated linear layer to yield a scalar projection, and the final output is computed as the average across these individual projections. Rather than predicting the variance directly as in [33], the model predicts the log-variance, $s = \log \sigma^2$, which led to a more stable training. To prevent numerical instability during training, the resulting log-variance is clamped to a range of $[-12, 10]$.

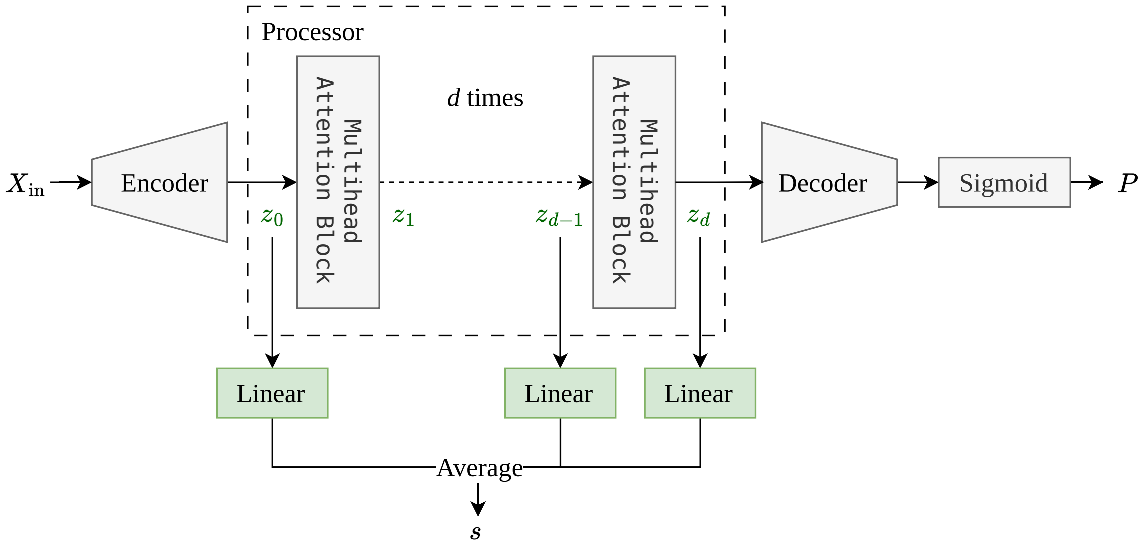


Figure 3.9: Architecture of the post-hoc heteroscedastic variance estimator integrating multi-layer latent representations mapping them to a turbine-level log-variance estimate.

The optimization is performed using the AdamW optimizer, coupled with a learning rate scheduler that combines a linear warmup with a cosine annealing decay. During this process, the parameters of the mean-predictor are frozen, preventing any degradation of the original deterministic prediction performance while the variance head is trained to minimize the Gaussian NLL.

3.3.4 Experimental Setup and Evaluation

To establish the probabilistic frameworks, both the MLPs and the GT were first retrained as mean predictors with temporal encodings enabled and wind veer and shear removed. While both models provided the foundation for the probabilistic baseline, the post-hoc heteroscedastic variance estimator was applied only to the GT. While the post-hoc heteroscedastic approach could have also been applied to the MLP models, this was left out due to time constraints. The same architecture settings were retained for this stage, and the experiments were repeated across five random seeds to assess robustness.

The validation split served as the calibration set for both uncertainty quantification strategies. For the homoscedastic baselines, the predictive variance was derived directly from the validation MSE of the pretrained mean predictor. For the post-hoc heteroscedastic estimator, the variance heads were optimized on the validation split while the mean predictor remained frozen. This calibration stage used AdamW with a peak learning rate of 10^{-5}, a batch size of 256, a 5-epoch linear warmup, and cosine annealing for the remaining epochs. The calibration runs were carried out for 50 epochs and did not use early stopping.

To evaluate the quality of the predictive uncertainty estimates, the model performance is assessed using four main metrics: NLL, Continuous Ranked Probability Score (CRPS), Prediction Interval Coverage Probability (PICP), and Mean Prediction Interval Width (MPIW). The NLL provides a fundamental measure of how well the predicted probability density aligns with the observed data, and is complemented by the CRPS. The CRPS generalizes the MAE to probabilistic forecasts by quantifying the discrepancy between the predicted cumulative distribution function and the empirical distribution of the true observation, thereby robustly penalizing both overconfidence and underconfidence [74].

To further decompose the evaluation into calibration and sharpness, the interval-specific metrics PICP and MPIW are utilized. The PICP quantifies the empirical reliability of the uncertainty estimates by measuring the proportion of true target values that successfully fall within a specified nominal prediction interval, such as a 95% confidence bound [75]. A well-calibrated model will yield a PICP that closely matches this target coverage probability. However, achieving high coverage is trivial if the predicted intervals are excessively wide. Therefore, the MPIW is introduced to measure the sharpness or precision of the predictions by calculating the average width of these intervals across the dataset [75]. A highly performant model must strike an optimal balance between these two competing objectives: maintaining a PICP close to the nominal level to guarantee reliability, while simultaneously minimizing the MPIW to ensure the uncertainty bounds remain tight and practically informative.

3.4 Generalization to Real Data

RQ3 deals with generalization to a real dataset. The following sections will describe how this dataset was preprocessed and how the experiment was designed.

3.4.1 The SCADA Dataset

The SCADA dataset describes twelve turbines of a windfarm in northern Germany. It covers a range of two years and has a frequency of 10 minutes. Because the data comes from raw data exports, it required a number of processing steps to bring the data into a usable format.

The first step was to combine the exports of all turbines into a single dataset. This also included the normalization of column names. The resulting table included 893, 376 rows with the following columns: Wind Turbine Generator (WTG) (identification of the wind turbine), timestamp, power, wind speed, wind direction, and nacelle position. The second step aimed at gathering more information about each turbine. It involved querying the MaStR to collect coordinates, rated power, hub height, rotor diameter and the model of each turbine. Using additional metadata from the original dataset, the WTGs of the dataset could be mapped to the turbines in the MaStR. Figure 3.10 shows the layout of the wind farm. The farm features two turbine types with differing hub heights, rotor diameters and rated powers. Additionally, this step included fetching power and thrust curves from wind-turbine-models.com for both turbine models [76].

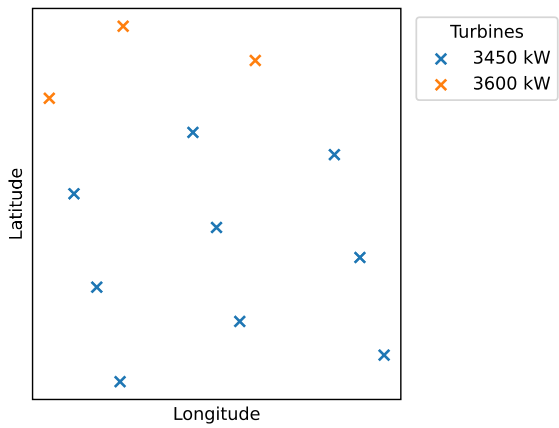


Figure 3.10: The wind farm under examination is heterogeneous featuring two different turbine types.

The third step dealt with data cleaning. SCADA data is intrinsically prone to sensor errors, missing values, and operational anomalies such as maintenance shutdowns or grid curtailment. To ensure the integrity of subsequent modeling steps, the raw data underwent a cleaning process, visualized in Figure 3.11 for one of the turbines.

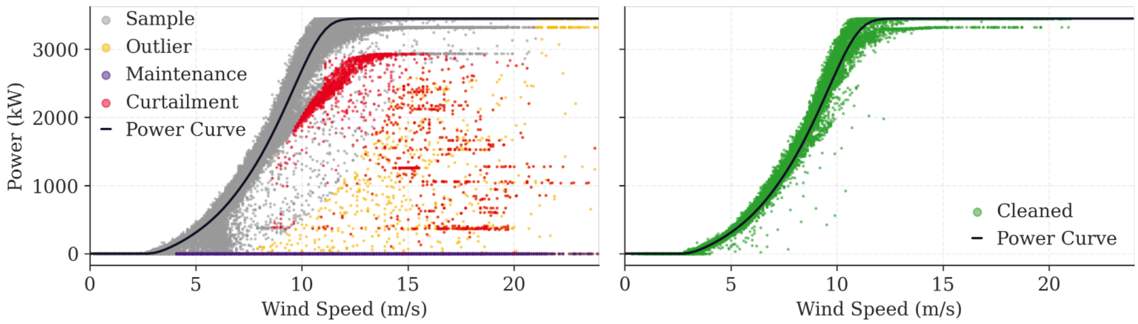


Figure 3.11: Data cleaning isolates normal turbine operation from anomalous readings. The left panel visualizes the raw dataset before filtering, categorizing observations into baseline samples (grey) and anomalies: general outliers (yellow), maintenance periods with zero power output (purple), and curtailment phases (red). The right panel demonstrates the final, cleaned dataset (green), showing that once irregular observations are removed from all turbines, the data roughly conforms to the theoretical wind turbine power curve (black line).

Imputation and Clipping Negative power outputs were clipped to 0 W. Three turbines were missing data for the first year. These timestamps were flagged as incomplete. For one turbine lacking wind measurements entirely, missing speeds were imputed using the farm mean at corresponding timestamps. While this provides a practical approximation, it should be noted that this method simplifies the aerodynamics of the farm by not explicitly accounting for wake effects.

Maintenance Filtering To filter out observations for which the turbine was assumed to be under maintenance, observations with zero power despite the wind speed surpassing a threshold were flagged as abnormal. 4 m/s was selected for this threshold. These samples are highlighted in purple in Figure 3.11.

Curtailment Filtering Curtailment was identified by evaluating the ratio between the observed active power and the theoretically possible power, given the corresponding wind speed and power curve. A data point was flagged as curtailed if the turbine was operating under high wind conditions, the power ratio fell below a predefined threshold, and this ratio remained relatively stable across a specified time window. The exact thresholds were empirically tuned for each individual turbine based on an analysis of the flagged data points. These samples are highlighted in red in Figure 3.11.

Outlier and Yaw Filtering Even after curtailment filtering, transient anomalies can remain. A boundary envelope was constructed around the theoretical power curve to filter out latent outliers. The envelope allowed for a wind speed measurement tolerance of ±3 m/s and a power variance tolerance of 10%. Samples falling outside the following bounds were classified as outliers and removed. Additionally, observations with extreme yaw angles, defined as $|\gamma| > 45$, were discarded. The outliers are highlighted in yellow in Figure 3.11.

Synchronization Finally, to ensure cross-turbine comparability for spatial modeling, any timestamp lacking valid, cleaned readings for all 12 turbines concurrently was dropped. The remaining samples for the corresponding turbine are highlighted in green in Figure 3.11.

The sequential filtering methodology ensured a reliable final dataset, though it necessitated a substantial reduction in total observations. The final synchronized dataset contained 12,934 unique 10-minute timestamps, yielding 155,208 total turbine observations.

Following data cleaning, each timestamp contained valid data points across all twelve turbines, resulting in twelve distinct wind direction and wind speed measurements per interval. However, the subsequent modeling approach requires free-stream conditions, specifically, a unified wind direction and wind speed representing the undisturbed flow entering the wind farm. Prior to approximating these free-stream conditions, it was necessary to align the individual wind direction sensors, as several exhibited significant systematic offsets over the recording period. To correct this, a single turbine was designated as a reference. To ensure the resulting offsets of the remaining turbines were kept relatively low, the turbine exhibiting the maximum mean cosine-transformed correlation with the rest of the fleet was selected for this role. For every other turbine, the circular mean offset relative to this reference was calculated. Each turbine's respective mean offset was then subtracted from its individual measurements to ensure uniform alignment across the wind farm. Once all sensors were consistently aligned, the free-stream conditions were approximated by computing the arithmetic mean of the wind speeds and the circular mean of the aligned wind directions across all operational turbines for each timestamp. Additionally, a TI of 0.15 was selected as a representative value for the onshore site. It is important to note that this is an approximation. Because the average includes wake-affected downstream turbines, this method provides a generalized farm-level reference rather than a purely undisturbed free-stream measurement.

In the final processing step, the dataset was partitioned into training, validation and testing sets. Because the SCADA measurements constitute time-series data, the observations were strictly ordered chronologically to prevent temporal data leakage during validation and evaluation. Consequently, the first 20% of the temporally sorted data was allocated to the training set, the next 40% were assigned to the validation set, while the remaining 40% was reserved as the holdout test set to accurately assess the model's ability to generalize to unseen, future conditions.

3.4.2 Experimental Setup and Evaluation

To evaluate generalization, the models from the previous section were considered in two settings: a zero-shot setting (applied directly without additional training) and a finetuned setting. For finetuning, the pretrained baseline and GT mean predictors were trained on the SCADA-based train split for up to 200 epochs using a learning rate of 5×10^{-5}. Calibration was then performed on the SCADA-based validation split. For the homoscedastic models, the predictive variance was estimated from the MSE on this calibration set. For the heteroscedastic model, the pretrained variance heads were calibrated on the same validation split while keeping the mean predictor fixed, using a learning rate of 10^{-5} for 200 epochs (batch size 64). These experiments were repeated across the same five random seeds used in the previous stages to ensure statistical consistency.

The assessment follows a dual-metric approach. First, the predictive performance of the mean power output is quantified as before, using the RMSE, MAE, and the Coefficient of Determination (R^2). Second, the quality of the uncertainty estimates is evaluated through NLL, CRPS and calibration metrics, including the PICP and MPIW at 50%, 90%, and 95% confidence levels. This comprehensive evaluation aims to determine whether the uncertainty heads learned from synthetic physics-based simulations remain robust when confronted with real-world operational data.

4 Results

This chapter presents the experimental results of the thesis. It begins with Section 4.1, which evaluates the initial deterministic performance of the MLP and GT models. Section 4.2 then compares homoscedastic and heteroscedastic approaches for uncertainty quantification and calibration. Finally, Section 4.3 assesses the robustness of the developed models by examining their generalization capabilities when applied to a real-world SCADA dataset.

4.1 Deterministic Modeling

Overall performance Table 4.1 summarizes the predictive performance of the GT and the MLP baselines across the normalized power metrics. The GT consistently outperforms both MLP variants across the board. It achieves the highest accuracy with an median R^2 of 0.980 and an median RMSE of 0.049. Notably, the MLP-Spatial demonstrates higher predictive power than the MLP-NonSpatial variant, reaching a median R^2 of 0.919 compared to the latter's 0.902. The spread across the five runs with different seeds remains minimal for all metrics, indicating that the observed performance gains are consistent and not the result of outlier runs or stochastic initialization.

Table 4.1: The GT consistently outperforms the MLP variants in predictive accuracy. Results represent the median and [min–max] values across five runs with different random seeds for the normalized power metrics. Arrows indicate the direction of better performance and the best value in each column is marked in bold.

Model Type	**RMSE** ↓	**MAE** ↓	R^2 ↑
Normalized Power			
MLP-NonSpatial	0.109 [0.109 – 0.110]	0.067 [0.067 – 0.067]	0.902 [0.900 – 0.902]
MLP-Spatial	0.099 [0.096 – 0.101]	0.059 [0.058 – 0.060]	0.919 [0.917 – 0.924]
Graph Transformer	**0.049** [0.046 – 0.050]	**0.019** [0.019 – 0.020]	**0.980** [0.980 – 0.983]

Operational Sensitivity A model's predictive accuracy is intrinsically linked to the nonlinear dynamics of power generation. One example of this is the nonlinear relationship between wind speed and generated power. Figure 4.1 characterizes model performance by visualizing the MAE over the input wind speed. It shows the mean MAE over 1 m/s windows across the three model architectures, with the shaded regions representing the

Interquartile Range (IQR). This visualization reveals a direct correlation between the aerodynamic operational regions of the turbines and the resulting predictive accuracy. While the error profiles across all models follow a similar non-linear trajectory, their magnitudes and variances differ significantly. For low wind speeds, the MAE is minimal for all models. Both MLP models exhibit an MAE of approximately 0.01, whereas the GT error starts notably lower, below 0.005. As wind speed increases, the error curves rise to a distinct peak. The GT maintains a relatively tight IQR and peaks at roughly 0.05, while the MLP-Spatial reaches nearly 0.13. The MLP-NonSpatial shows the highest deviation, peaking at almost 0.15. Beyond 15 m/s, the error curves flatten. This trend persists until the final bin, exceeding 20 m/s, where a secondary increase in error is observed. The visualization reveals an important characteristic of the error distribution of all models. For most wind speed bins, the mean MAE is positioned in the upper region of the interquartile range or even extends above the 75th percentile line. This is most pronounced in the region from 12 to 15 m/s. This positioning indicates a right-skewed distribution where a subset of high-error outliers is pulling the mean upward, despite the majority of predictions remaining more accurate.

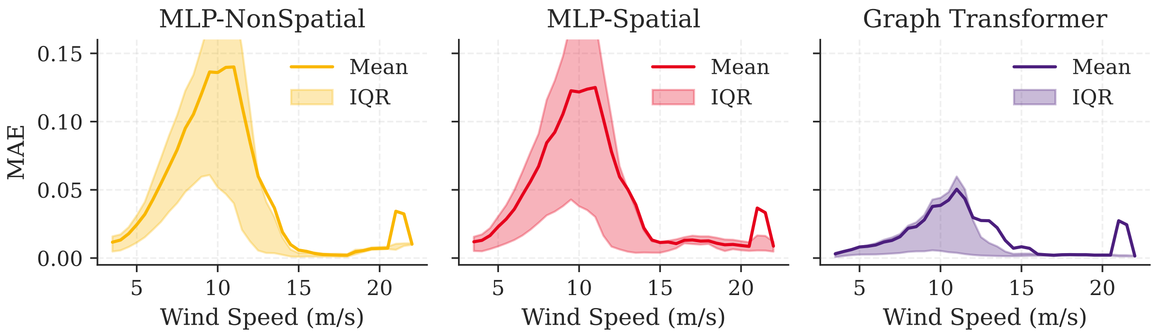


Figure 4.1: MAE depends on the input wind speed, peaking notably around 11 m/s where the MLP models struggle the most. Even in this high-error region, the GT exhibits substantially lower MAE and variance compared to the MLPs. The solid lines represent the mean MAE, and the shaded regions indicate the IQR. Only one run per architecture is shown.

Wake Interactions To give more detailed insights into the models' ability to learn wake interactions, Table 4.2 provides the R^2 metric for the normalized wake loss. As described earlier in Equation 3.1, this metric is derived from the same model outputs using precomputed no-wake predictions as a reference. The performance gaps between the models widen significantly for the wake loss metrics. The GT maintains a robust predictive capability with a median R^2 of 0.841. By comparison, the MLP variants struggle significantly: the MLP-Spatial achieves an R^2 of only 0.208, while the MLP without coordinates yields a near-zero R^2 of 0.016.

Beyond aggregate metrics, Figure 4.2 plots the MAE as a function of the ground truth wake loss, providing insight into how models handle increasing levels of aerodynamic interference. The GT exhibits a relatively robust error profile, with the mean MAE remaining below 0.06 across the entire range of wake loss values. While there is a slight upward trend

Table 4.2: The GT shows robust predictive capability for normalized wake loss, whereas the MLP variants' metrics are poor. Results represent the median and [min–max] of the R^2 metric across five runs with different random seeds. The arrow indicates the direction of better performance and the best value is marked in bold

Model Type	$R^2 \uparrow$
Normalized Wake Loss	
MLP-NonSpatial	0.016 [0.009 – 0.055]
MLP-Spatial	0.208 [0.193 – 0.278]
Graph Transformer	**0.841** [0.787 – 0.855]

as wake loss increases, peaking at approximately 0.05 near the 0.75 normalized wake loss mark, the IQR remains tight. This indicates that the GT provides not only high average accuracy but also consistent predictions in high-interference scenarios.

Conversely, both MLP variants show a substantial deterioration in performance and stability as the wake loss magnitude increases. The error curves for both variants follow a sharp non-linear trajectory, peaking at the same 0.75 mark as the GT, but at significantly higher magnitudes: approximately 0.15 for the MLP-Spatial and 0.18 for the MLP-NonSpatial. Notably, the visualization reveals a dramatic expansion of the IQR for the MLPs in these high-wake regions. For MLP-NonSpatial, the 75th percentile of the error reaches as high as 0.25.

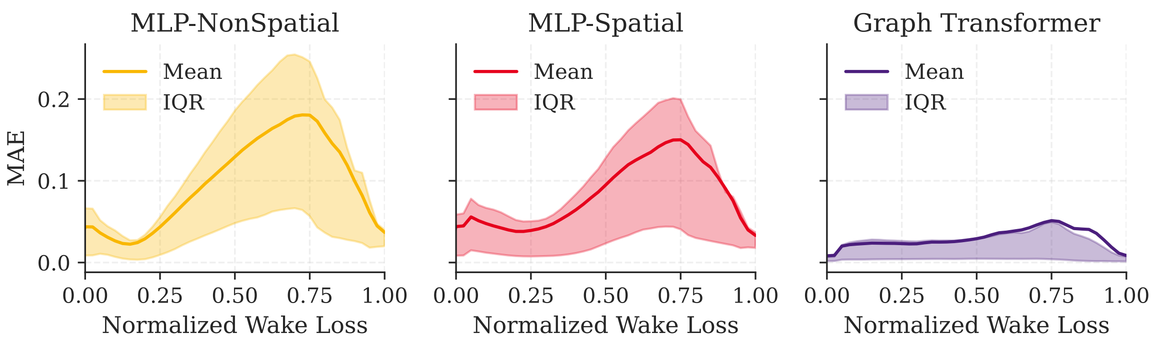


Figure 4.2: While the MLPs show a sharp degradation in performance for samples with high wake losses, the GT demonstrates superior stability and lower overall error. The solid lines represent the mean MAE and the shaded regions indicate the IQR. Only one run per architecture is shown.

As wake losses are highly dependent on a turbine's spatial position within a wind farm, analyzing the spatial distribution of errors provides deeper insights into the models' ability to generalize across layouts. Figure 4.3 illustrates the wake loss MAE across downwind and crosswind coordinates. Notably, the test set includes larger wind farms than those in the training and validation sets, which were limited to crosswind distances of ±5000 m and downwind distances up to 9000 m, see Figure 3.6 as reference.

Within this central training region, the GT significantly outperforms both MLPs, maintaining an MAE near zero (indicated by the dark blue regions). The MLP-Spatial shows relatively uniform error except for the first row (0 m downwind) where its error is closer to zero. The MAE of the MLP-NonSpatial starts around 0.2 and slightly increases with growing downwind distance.

Outside the training bounds, the models diverge sharply. The MLP-Spatial exhibits failure at high crosswind distances (> ±4500 m), with MAE frequently exceeding 0.5. While the GT also shows increased error in out-of-distribution regions, it maintains significantly better stability at high crosswind distances compared to the MLP-Spatial variant. Interestingly, the MLP-NonSpatial shows the most uniform error distribution. While it never achieves the high accuracy of the GT in the center, its lack of spatial coordinates prevents it from overshooting as severely as the MLP-Spatial in the outer crosswind regions.

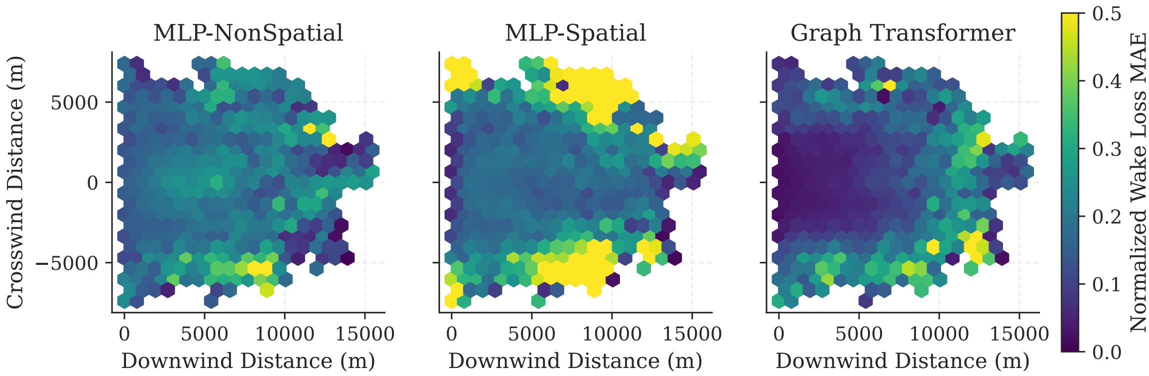


Figure 4.3: While the GT achieves near-zero error within the spatial bounds of the training data, in the outer regions representing larger layouts, the GT's error increases notably. These hexbin plots visualize the spatial distribution of wake loss MAE for one run per architecture.

Turbine Heterogeneity Table 4.3 presents a statistical summary of the MAE achieved by the three models, broken down across the 17 distinct turbine types present in the test set. For each architecture, a single run is analyzed. The MLP-NonSpatial yields the highest overall error, recording a mean MAE of 0.064 and a Standard Deviation (Std) of 0.014 across the varying turbine types. The inclusion of spatial coordinates in the MLP-Spatial model improves these metrics slightly, reducing the mean MAE to 0.058 and the Std to 0.013. Both MLP variants exhibit a wide distribution of errors depending on the specific turbine model being evaluated, with maximum MAE values reaching 0.090 and 0.081, respectively. By comparison, the GT demonstrates a substantially lower average error, achieving a mean MAE of 0.019. Furthermore, the variance in predictive accuracy across the 17 turbine models is notably narrower for this architecture (0.006), and even its worst-case performance (Max MAE of 0.032) outperforms the best-case performance of the MLP-Spatial (Min MAE of 0.027).

Atmospheric Condition Diversity To further evaluate the models under diverse atmospheric conditions, Figure 4.4 illustrates predictive performance as a joint function of wind

Table 4.3: The GT's spread of MAE across the 17 different turbine types in test set is notably lower compared to the MLPs'. The best value in each column is marked in bold. Metrics only reflect one run per architecture.

Model	Mean MAE	Min MAE	Max MAE	Std MAE
MLP-NonSpatial	0.064	0.030	0.090	0.014
MLP-Spatial	0.058	0.027	0.081	0.013
Graph Transformer	**0.019**	**0.010**	**0.032**	**0.006**

shear and wind veer. The hexbin plots reveal that the MLP models are more sensitive to these atmospheric parameters. Both the MLP-NonSpatial and MLP-Spatial variants exhibit substantial error fluctuations, with MAE frequently exceeding 0.10 in regions of moderate to high shear and veer. By comparison, the GT demonstrates improved robustness, maintaining a consistently low and uniform MAE of approximately 0.01 to 0.02 across the entire evaluated spectrum.

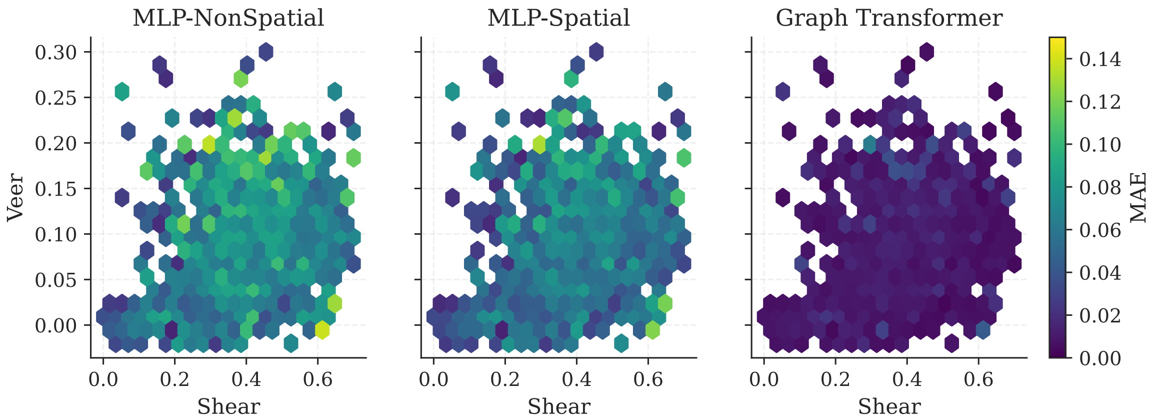


Figure 4.4: The GT's MAE varies less across varying wind shear and veer conditions compared to the MLPs'. Only one run per architecture is visualized.

4.2 Probabilistic Modeling

The evaluation compares three specific model configurations. The homoscedastic versions consists of a corresponding pretrained model where the variance is a fixed global constant. The heteroscedastic GT employs the same GT architecture but incorporates the post-hoc variance estimator to predict input-dependent variance.

Table 4.4: GTs outperform MLPs: Median and [min–max] predictive performance metrics across five random seeds. Arrows indicate the direction of better performance and the best value in each column is marked in bold.

Model Type	RMSE ↓	MAE ↓	R^2 ↑
Hom. MLP-Spatial	0.107 [0.105 – 0.107]	0.064 [0.064 – 0.064]	0.906 [0.906 – 0.910]
Hom. & Het. GT	**0.068** [0.067 – 0.069]	**0.036** [0.035 – 0.037]	**0.962** [0.961 – 0.963]

The predictive performance of the models within the probabilistic framework is summarized in Table 4.4. The reported values represent the median and min/max values calculated across five random seeds for the RMSE, MAE, and R^2. Because the GT configurations use the same mean-predictor, they achieve identical mean predictive metrics with an RMSE of 0.068, an MAE of 0.036, and an R^2 of 0.962. The homoscedastic MLP-Spatial yields an RMSE of 0.107, an MAE of 0.064, and an R^2 of 0.906. The spreads across the runs are relatively small.

Table 4.5: Heteroscedastic GT achieves superior distributional modeling: Evaluation of NLL and CRPS. The table presents the median and [min–max] values across the five runs with different seeds. Arrows indicate the direction of better performance and the best value in each column is marked in bold.

Model Type	NLL ↓	CRPS ↓
Hom. MLP-Spatial	−0.798 [−0.821 – −0.792]	0.053 [0.052 – 0.053]
Hom. GT	−1.134 [−1.158 – −1.103]	0.031 [0.030 – 0.032]
Het. GT	**−1.872** [−1.919 – −1.756]	**0.027** [0.027 – 0.028]

Tables 4.5 and 4.6 present an evaluation of uncertainty quantification and calibration. Performance is assessed using the NLL, CRPS, PICP, and MPIW. The heteroscedastic GT demonstrates superior performance, achieving the lowest NLL of -1.872 and a CRPS of 0.027. This performance outperforms both the homoscedastic GT (NLL -1.134, CRPS 0.031) and the homoscedastic MLP-Spatial (NLL -0.798, CRPS 0.053).

Regarding interval coverage (PICP), the models exhibit varying degrees of calibration across thresholds. At the 95% confidence level, the heteroscedastic GT is the most well-calibrated with a PICP of 0.939, while the homoscedastic GT and the homoscedastic MLP-Spatial appear more overconfident at 0.899 and 0.915, respectively. At the 50% level, all models appear somewhat underconfident, as they capture more data than expected,

Table 4.6: Heterogeneous GT achieves the best calibration without loosing much sharpness. The table presents the median PICP and MPIW across three quantiles. The best coverage is marked in bold. Interval widths can only be interpreted in combination with coverage and are thus not marked.

Model Type	**PICP (50/90/95)**	**MPIW (50/90/95)**
Hom. MLP-NonSpatial	0.659 / 0.886 / 0.919	0.135 / 0.328 / 0.391
Hom. MLP-Spatial	0.677 / 0.885 / 0.915	0.124 / 0.303 / 0.362
Hom. GT	0.692 / 0.871 / 0.899	0.066 / 0.161 / 0.192
Het. GT	**0.597** / **0.899** / **0.939**	0.070 / 0.170 / 0.202

with the homoscedastic GT being the most conservative at 0.692. The advantage of the GT models is revealed when considering interval sharpness (MPIW). While the homoscedastic MLP-Spatial achieves its coverage by producing wider intervals (0.362 at the 95% level), the homoscedastic GT provides the sharpest predictions overall, yielding the narrowest median intervals across all thresholds, specifically 0.066, 0.161, and 0.192. The heteroscedastic GT maintains a balance between these metrics, offering sharper intervals than the baseline (0.202 at the 95% level) while maintaining superior calibration.

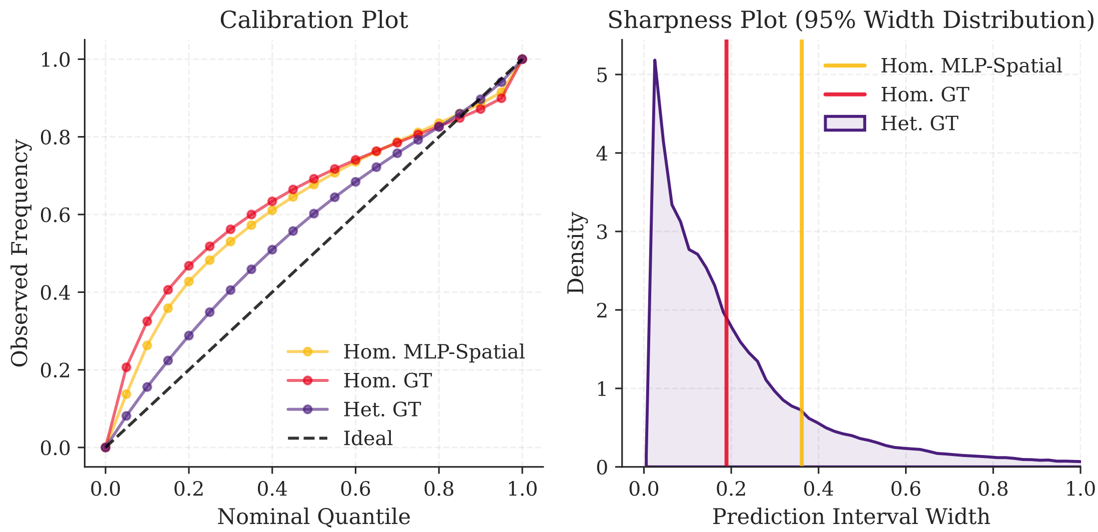
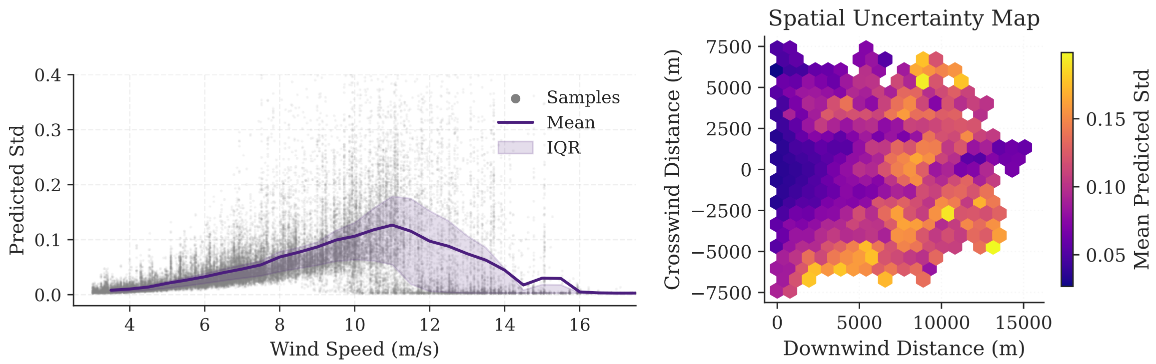


Figure 4.5: Calibration (left) and sharpness (right) plots comparing the calibration and 95% prediction interval widths for the homoscedastic MLP-Spatial, homoscedastic GT, and heteroscedastic GT. All models are underconfident. Homoscedastic models produce constant intervals and the heteroscedastic model outputs intervals of varying widths. Only one run per architecture is shown.

Figure 4.5 provides a detailed assessment of the model's uncertainty quantification through calibration (left) and sharpness (right). In the reliability plot, the relationship between the nominal quantiles and the observed frequency is shown across the three model architectures. Ideal calibration is represented by the dashed diagonal line, where the observed frequency perfectly matches the nominal quantile. All three models exhibit a curve above this ideal line for the majority of the quantile range. This indicates that the models are generally underconfident. The observed frequency of data points falling within the predicted intervals is higher than the nominal quantiles suggest, meaning the uncertainty bounds are wider and more conservative than necessary. Notably, the heteroscedastic GT (purple line) tracks closer to the diagonal than the homoscedastic variants, demonstrating that while it remains slightly underconfident, it achieves a more accurate calibration of its predictive uncertainty.

This calibration behavior is further contextualized by the sharpness plot, which displays the distribution of the 95% prediction interval widths. The homoscedastic MLP-Spatial and homoscedastic GT appear as vertical spikes, confirming a constant variance for all predictions regardless of the input conditions. In contrast, the heteroscedastic GT produces a broad distribution with a sharp peak much closer to zero. This indicates that the model provides narrow intervals for the majority of predictions while selectively increasing uncertainty for more difficult cases, as evidenced by the long tail of the distribution. Consequently, the heteroscedastic GT achieves a superior balance: it remains the most well-calibrated architecture while simultaneously providing more informative and sharp uncertainty bounds than the non-adaptive models.



(a) Uncertainty by wind speed.

(b) Spatial map of uncertainty.

Figure 4.6: The heteroscedastic GT demonstrates an intuitive understanding of its own error margins. (a) Predicted uncertainty peaks in the operational region around 11 m/s. (b) The spatial uncertainty map reveals that the model is more confident for turbines with low downwind distances. Only one run is shown.

To understand the internal logic of the uncertainty estimates, Figure 4.6 visualizes the predicted Std as a function of both operational and spatial parameters. Figure 4.6a shows that the model's uncertainty is intrinsically tied to the wind speed. The predicted uncertainty remains minimal at low wind speeds but follows a non-linear trajectory, peaking at approximately 11 m/s with a mean Std of roughly 0.12. The spatial distribution of this uncertainty, illustrated in Figure 4.6b, reveals a clear gradient of confidence along the flow direction. The model is more confident for turbines with low downwind distances. As the downwind distance increases, the predicted *std* rises accordingly.

The applied approaches parametrize Gaussian distributions, and thus assume residuals that follow a Gaussian. However, this assumption is not confirmed by the data. Table

4.7 presents the distributional metrics, specifically skewness and kurtosis, for the three model architectures segmented across three distinct power bins based on P_{true}, which is the ground truth normalized power of a turbine. For the lowest power bin ($P_{true} \leq 0.1$), the homoscedastic baseline shows a skewness of -4.12 and an excess kurtosis of 36.28. In this same bin, both the homoscedastic GT and heteroscedastic GT models report identical values, with a skewness of -4.92 and a significantly higher excess kurtosis of 53.44. In the intermediate bin ($0.1 < P_{true} < 0.9$), skewness values for all models are closer to zero, ranging from -1.19 to -1.23. Excess kurtosis in this middle regime is lower than in the edge bins, with the homoscedastic baseline recording 3.21, while both GT-informed models show a higher excess kurtosis of 8.52. Within the highest power bin ($0.9 \leq P_{true} \leq 1.0$), the homoscedastic baseline records a skewness of 2.45 and an excess kurtosis of 7.98. The homoscedastic GT and heteroscedastic GT models again show identical results in this high-power regime, each exhibiting a skewness of 3.78 and an excess kurtosis of 17.20. Across all three power bins, the significant deviations from a skewness of 0 and an excess kurtosis of 0 indicate that the residuals are non-Gaussian for all model variants.

Table 4.7: Non-Gaussian residual distributions across all power regimes: Skewness and excess kurtosis for MLP-Spatial and GT model architectures. Only one run is examined respectively.

	$P_{\text{true}} \leq 0.1$		$0.1 < P_{\text{true}} < 0.9$		$0.9 \leq P_{\text{true}} \leq 1.0$	
Model Type	**Skew.**	**Kurt.**	**Skew.**	**Kurt.**	**Skew.**	**Kurt.**
Hom. MLP-Spatial	-4.12	36.28	-1.23	3.21	2.45	7.98
Hom. & Het. GT	-4.92	53.44	-1.19	8.52	3.78	17.20

4.3 Generalization to Real Data

Table 4.8 shows the predictive performance metrics for the generalization to real-world SCADA data. The tables report the median and min/max values across the five random seeds. In the zero-shot setting, both GT variants maintain identical mean predictions, yielding an RMSE of 0.109, an MAE of 0.066, and an R^2 of 0.886. The homoscedastic MLP-NonSpatial exhibits much lower zero-shot performance, with an R^2 of only 0.541. The MLP-Spatial variant resulted in even worse predictions, yielding an R^2 of 0.131. Notably, the spread for both MLP variants is relatively high, whereas the GT's spread is much lower. After finetuning, all models show improvement, with the GT architectures achieving a superior RMSE of 0.092 and an R^2 of 0.917, compared to the MLP-NonSpatial's R^2 of 0.903. Furthermore, the spreads have decreased significantly.

Table 4.8: Poor zero-shot MLP performance: Predictive performance metrics on real-world SCADA data across RMSE, MAE, and R^2. Arrows indicate the direction of better performance and the best value in each column is marked in bold.

Model Type	**RMSE** $\downarrow$	**MAE** $\downarrow$	R^2 $\uparrow$
Zero-shot			
Hom. MLP-Spatial	0.300 [0.220 – 0.394]	0.240 [0.143 – 0.320]	0.131 [−0.505 – 0.533]
Hom. MLP-NonSpatial	0.218 [0.149 – 0.281]	0.141 [0.099 – 0.192]	0.541 [0.237 – 0.784]
Hom. & Het. GT	**0.109** [0.103 – 0.117]	**0.066** [0.062 – 0.071]	**0.886** [0.868 – 0.898]
Finetuned			
Hom. MLP-Spatial	0.113 [0.110 – 0.121]	0.073 [0.072 – 0.081]	0.877 [0.858 – 0.882]
Hom. MLP-NonSpatial	0.100 [0.097 – 0.123]	0.065 [0.062 – 0.078]	0.903 [0.855 – 0.909]
Hom. & Het. GT	**0.092** [0.090 – 0.096]	**0.057** [0.056 – 0.060]	**0.917** [0.911 – 0.921]

Table 4.9 and Table 4.10 provide the uncertainty and calibration metrics on the real-world dataset. In the zero-shot setting, the homoscedastic GT achieves the lowest CRPS of 0.055, while the heteroscedastic GT exhibits a very high NLL of 14.041 despite a competitive CRPS of 0.057. Notably, the spread across the runs is relatively high, particularly for the NLL. Regarding zero-shot interval coverage (PICP), all models underestimate uncertainty, with the heteroscedastic GT showing the lowest coverage of 0.475 at the 95% level. It also produces the sharpest intervals, with an MPIW of 0.090 at the 95% level, compared to 0.192 for the homoscedastic GT and 0.362 for the MLP-Spatial.

Once finetuned, the heteroscedastic GT demonstrates the best overall calibration and sharpness balance, achieving the lowest NLL of −1.370 and a well-calibrated 95% PICP of 0.951 with the narrowest MPIW of 0.314.

Figure 4.7 presents the reliability and sharpness evaluations for one finetuned run for each model architecture. The left panel, a reliability plot, assesses calibration against an ideal dashed diagonal line. The heteroscedastic GT (purple) closely tracks the ideal line, indicating superior calibration compared to the other variants. In contrast, both the

Table 4.9: Finetuning resolves zero-shot calibration failure, specifically for the heteroscedastic variant: Uncertainty quantification and calibration metrics on real-world SCADA data. Arrows indicate the direction of better performance and the best value in each column is marked in bold.

Model Type	**NLL** ↓	**CRPS** ↓
Zero-shot		
Hom. MLP-Spatial	3.810 [1.382 - 7.602]	0.198 [0.121 - 0.277]
Hom. MLP-NonSpatial	1.008 [−0.265 - 2.497]	0.120 [0.078 - 0.162]
Hom. GT	**0.423** [0.154 - 0.749]	**0.055** [0.052 - 0.060]
Het. GT	14.041 [8.741 - 17.533]	0.057 [0.052 - 0.062]
Finetuned		
Hom. MLP-Spatial	−0.750 [−0.769 - −0.688]	0.060 [0.059 - 0.064]
Hom. MLP-NonSpatial	−0.810 [−0.832 - −0.668]	0.055 [0.054 - 0.065]
Hom. GT	−0.954 [−0.978 - −0.926]	0.048 [0.047 - 0.050]
Het. GT	**−1.370** [−1.406 - −1.301]	**0.042** [0.041 - 0.043]

homoscedastic MLP-Spatial (yellow) and the homoscedastic GT (red) are underconfident, as their curves remain considerably above the diagonal across most nominal quantiles. The right panel displays the sharpness plot, illustrating the distribution of the 95% prediction interval widths. As characteristic of homoscedastic models, the homoscedastic GT (red) and homoscedastic MLP-Spatial (yellow) output constant interval widths, represented by singular vertical lines at approximately 0.4 and 0.5, respectively. Conversely, the heteroscedastic GT produces a dynamic distribution of widths featuring a prominent peak near 0.05, demonstrating its ability to adaptively generate much narrower, sharper intervals based on the input data.

Table 4.10: In the zero-shot setting, all models are miscalibrated producing overconfident predictions. After finetuning, all models are much better calibrated, while producing much wider intervals on average.

Model Type	PICP (50/90/95)	MPIW (50/90/95)
Zero-shot		
Hom. MLP-Spatial	0.227 / 0.415 / 0.461	0.124 / 0.303 / 0.362
Hom. MLP-NonSpatial	0.521 / 0.665 / 0.701	0.135 / 0.328 / 0.391
Hom. GT	0.494 / 0.717 / 0.760	0.066 / 0.161 / 0.192
Het. GT	0.205 / 0.416 / 0.475	0.031 / 0.075 / 0.090
Finetuned		
Hom. MLP-Spatial	0.695 / 0.918 / 0.949	0.170 / 0.415 / 0.494
Hom. MLP-NonSpatial	0.761 / 0.950 / 0.969	0.181 / 0.441 / 0.526
Hom. GT	0.704 / 0.916 / 0.944	0.134 / 0.328 / 0.390
Het. GT	0.589 / 0.918 / 0.951	0.108 / 0.264 / 0.314

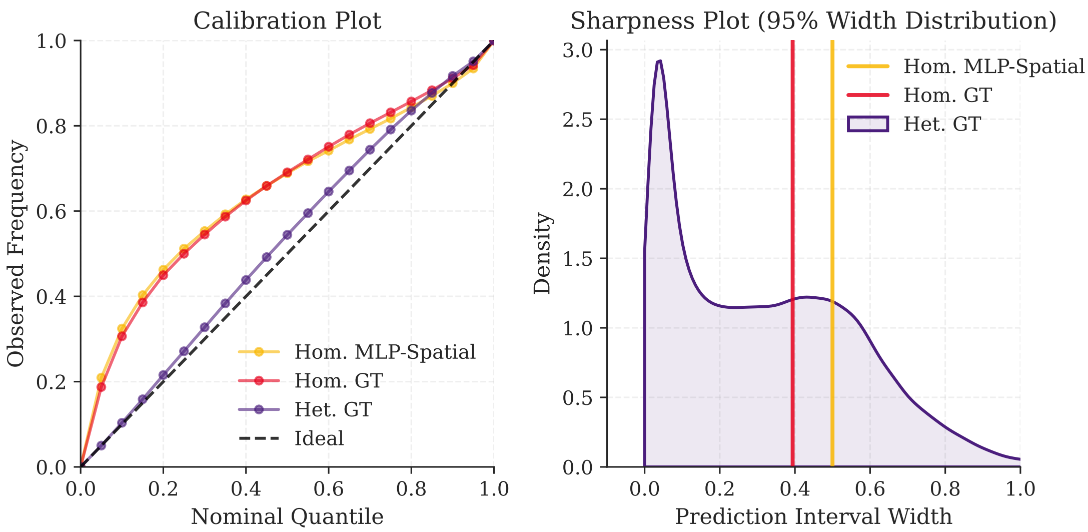


Figure 4.7: The heteroscedastic GT model provides sharper predictions by dynamically adapting its uncertainty estimates: Evaluating model calibration via a calibration plot (left) and prediction precision via a sharpness plot (right). Only one run per architecture is shown.

5 Discussion

This chapter synthesizes findings across the three RQs, progressing from deterministic accuracy and uncertainty modeling to real-world generalization. The discussion highlights the practical implications while addressing current limitations and future research paths.

5.1 Deterministic Modeling

This section discusses findings relevant to RQ1, how a GT compares to an MLP baseline on a diverse dataset. It is important to acknowledge that this comparison does not imply architectural equivalence. Rather, the MLPs serve as a fundamental performance benchmark. While the MLPs operate at a localized, turbine-specific level, the GT integrates information from across the entire wind farm. Initial comparisons of global metrics reveal a clear and expected trend: the GT outperforms both MLP models. Still, the MLP variants already achieve competitive metrics for normalized power with values for R^2 around 0.90. The following subsections characterize the variations in predictive accuracy and identify the specific scenarios where the performance advantage of the GT is most pronounced.

5.1.1 Comparative Analysis of Model Capabilities

Operational Sensitivity The decent performance of MLP-NonSpatial shows that a major portion of power variance in this dataset can be explained purely by global atmospheric conditions and individual turbine operational parameters. One contributing factor are the inherent characteristics of the wind turbine power curve. At free-stream wind speeds below the cut-in threshold, a turbine's power stays at zero regardless of any interaction dynamics. Conversely, at free-stream wind speeds sufficiently above the rated speed, power output stays at rated capacity regardless of velocity deficits. This also provides an explanation for the low MAE of the MLPs for wind speeds at the low end and those above 15 m/s, as Figure 4.1 shows. For these operational states, the models do not need to understand wake interactions to produce accurate predictions.

For wind speeds between these extremes, the MAEs of all three models increases, as can be seen in Figure 4.1. One reason is that, in this interval, the power curve is most sensitive to fluctuations. A small error in estimating wind conditions results in a much larger power error here than it would at the low or high end of wind speed, where the power curve is flat. Because the wind conditions at a turbine are influenced by a complex interplay of all free-stream characteristics and the turbine's hub height, as well as all potential upstream turbines' properties due to inter-turbine interactions, it is hard to identify what exactly the models struggle with in this wind interval.

Wake Interactions While it is difficult to definitively isolate the exact causes of the high MAEs in the high-sensitivity wind speed regions, the performance disparities between models in Figure 4.1 most likely stem from their differing abilities to represent wake interactions. These differences are further detailed by the wake loss metrics in Table 4.2. The MLP-NonSpatial is architecturally incapable of accounting for wake effects, as it lacks spatial context. The near-zero R^2 for the normalized wake loss, confirms that the model fails to capture turbine interactions entirely. Given that spatial coordinates are the sole distinction between the two MLP variants, the observed reduction in error can be attributed to the inclusion of spatial data. While still functioning on a turbine-by-turbine basis, the model might have learned spatial heuristics, such as downwind positions generally correlating with higher velocity deficits although this needs to be analyzed in more detail in the future. The difference between the MLP variance becomes also evident in Figure 4.3. Within the central area represented in the training data, the MLP-NonSpatial exhibits a wake loss MAE that increases alongside downwind distance. This trend highlights its architectural inability to correlate wake interactions with spatial positioning. Conversely, the coordinate-aware MLP-Spatial effectively minimizes error at the front turbine row, suggesting it has identified that upwind positions cannot affected by wake losses. The reduced error across the central area is another hint that this model has learned a general correlation between spatial coordinates and wake-induced deficits. The GT's R^2 of 0.841 for normalized wake loss and its much lower error within the central training area in Figure 4.3 shows its superior ability to explain power losses due to wake interactions. It suggests that the model can capture inter-turbine dependencies that the localized MLPs cannot represent. This difference is further underscored by the analysis of the MAE over normalized wake loss. The fact that the GT maintains an MAE below 0.06 even during periods of high wake loss demonstrates its ability to accurately predict power output under conditions of high aerodynamic interference.

The spatial distribution of errors in Figure 4.3 provides another insight into the wake modeling capabilities of each architecture, particularly regarding their ability to generalize to unseen farm geometries. The models' generalization performance outside the training distribution reveals the limitations of these learned spatial patterns. Interestingly, the MLP-NonSpatial demonstrates the most stable error profile across the entire spatial domain. While it lacks the precision of the other models in the central area, its ignorance of spatial coordinates makes it less susceptible to extrapolation artifacts that occur in the predictions of the coordinate-aware architectures. In contrast, the MLP-Spatial's error increases sharply outside the training region. This suggests that the model's reliance on absolute coordinates led to poor extrapolation when faced with larger farms. Although the GT exhibits less increased errors in these outer regions in comparison to the MLP-Spatial, they indicate a major deterioration when compared to the training distribution. This pattern questions the model's generalization ability to larger wind farms not seen during training.

A plausible explanation is that the GT over-relies on absolute coordinate embeddings, hindering its ability to abstract the relative aerodynamic dynamics between turbines.

Robustness to Atmospheric Conditions The evaluation of the models under varying atmospheric conditions reveals a limitation of localized, turbine-specific architectures. As observed in the results for wind shear, the MLPs exhibit higher sensitivity and error fluctuations. Because these models lack farm-wide spatial context, they seem to struggle to accurately map free-stream conditions to effective hub-height wind speeds when the vertical wind profile becomes complex. The GT appears relatively robust under most shear and veer conditions. Through its attention mechanisms, it seems to internalize complex atmospheric influences on wake behavior. However, a more detailed ablation study would be required to understand how exactly the model uses the shear and veer values for its predictions.

Turbine Heterogeneity The heterogeneous nature of the test dataset, comprising 17 distinct turbine models, further poses a challenge. Differing rotor diameters, hub heights, and distinct power and thrust curves alter both a turbine's individual power generation and its wake generation characteristics. As the results show, the localized MLPs struggle to accurately use these differing physical profiles. This is quantitatively supported by their high variance in accuracy across the 17 turbine types. Theoretically, the GT's attention mechanism should be able to learn to differentiate between turbine models, by embedding specific turbine characteristics as node features within the graph. It can map complex, asymmetrical aerodynamic interactions, such as the wake expansion from a large-rotor turbine impacting a smaller downstream turbine operating at a different hub height. To precisely analyze the transformer's ability to learn cross-type wake interactions would require a detailed analysis of the attention scores or additional experiments, both of which are beyond the scope of this thesis. A proxy of how well a model understands different turbine models is the error consistency across them. The fact that the GT's error variance across turbine models is much lower than the MLPs' ones, suggests that it captures and applies the distinct physical characteristics of the diverse turbine models. Rather than defaulting to a generalized operational profile that fails on less common turbine types, this stability indicates that the GT effectively leverages its node embeddings to adapt to the specific aerodynamic and power generation behaviors of each individual turbine. The model's ability to handle different turbine models enables its usage beyond homogeneous wind farms.

5.1.2 Limitations and Future Work

While the GT demonstrates substantial improvements over localized MLPs, several limitations remain regarding both its architecture and the synthetic data pipeline used for its training. Primarily, the current GT architecture is relatively simple, omitting edge attributes and advanced structural encodings in favor of raw spatial coordinates. As indicated by the model's deteriorating performance outside the training distribution, this reliance on absolute coordinates likely prevents the model from fully abstracting relative inter-turbine dynamics. Future work should investigate the integration of relative distance vectors as edge features to enhance spatial generalization to unseen and larger wind farm layouts. While this was already studied for GNNs [12, 14], it has not been applied to GTs, and not for datasets with different hub heights. Furthermore, the current methodology for embedding distinct turbine characteristics, specifically the power and thrust curves, was implemented naively. Subsequent research should explore alternative representation learning for these curves to verify the network's understanding of cross-model wake interactions.

While the evaluations in this study offer preliminary indications regarding the sources of predictive error, such as wake extrapolation artifacts, these explanations remain inferential. A more detailed explainability analysis is necessary to reveal the precise physical patterns the model leverages or misses. Future studies should prioritize interpreting the GT's attention mechanism to uncover exactly how cross-type wake interactions and complex atmospheric conditions are weighted internally. Such insights will not only validate the model's physical reasoning but also guide targeted architectural refinements.

Furthermore, the ground truth was computed using the CC model in FLORIS, utilizing default parameters that are primarily tuned for offshore wind farms [46]. Wind farm aerodynamics, specifically wake expansion and recovery rates, differ based on surface roughness, which can be different in onshore environments. Consequently, the trained GT may implicitly assume offshore wake recovery profiles. To transition this architecture into a truly universal surrogate model, future work should vary the internal FLORIS parameters during data generation. These parameters could then be provided as explicit conditioning inputs to the ML model, allowing the network to dynamically adjust its internal wake representation based on the specific environmental context of a given farm.

5.2 Probabilistic Modeling

In addressing the RQ2, how well the post-hoc heteroscedastic variance estimation can quantify predictive uncertainty, the results demonstrate a clear distinction in model capabilities. While removing wind veer and shear features slightly reduced point-prediction accuracy, the GT configurations continue to outperform the MLP baseline, which is consistent with the previous section. More relevant to RQ2, the heteroscedastic GT's lower NLL and CRPS indicate a combination of sharper and better calibrated predictions compared to the homoscedastic GT and MLP. To fully contextualize these improvements, a more detailed analysis of calibration and sharpness is conducted in the following.

5.2.1 Comparative Analysis of Model Capabilities

Calibration and Sharpness The evaluation of interval metrics exposes a fundamental limitation of homoscedastic architectures: the constraint of a single, global variance forces a severe trade-off between calibration and sharpness. Because the homoscedastic MLP lacks the spatial context to resolve inter-turbine aerodynamic interactions, it is compelled to output conservative, wide prediction intervals (higher MPIW) to account for its large unexplained variance. In contrast, the homoscedastic GT leverages its superior mean-predictions to reduce this unexplained variance, yielding the sharpest intervals overall. However, due to the homoscedastic formulation, it produces the same narrow uncertainty bound to every prediction. This rigidity causes the homoscedastic GT to be overconfident at the tails, resulting in the poorest 95% coverage (PICP). The heteroscedastic GT resolves this problem by producing a heteroscedastic distribution of interval widths. This is defined by a high density of precise intervals for standard conditions and a long-tailed distribution of increased widths for complex atmospheric states. Consequently, it achieves a superior distributional balance, tracking closest to the ideal reliability diagonal and providing the most accurate 95% coverage while maintaining informative, sharp bounds.

Adaptive Uncertainty The superior calibration of the heteroscedastic GT is driven by its ability to effectively map predictive variance to the underlying physics of the wind farm. As illustrated in Figure 4.6, the model's uncertainty estimates are not arbitrary. They mirror aerodynamic sensitivity. Operationally, the predicted standard deviation peaks around 11 m/s (Figure 4.6a). This wind speed corresponds to the steep, cubic region of a typical turbine's power curve, where even marginal fluctuations in local wind velocity translate into massive variations in power output. The model identifies this region as sensitive and widens its uncertainty bounds accordingly. Spatially, the uncertainty map (Figure 4.6b) reveals a clear gradient along the downstream direction. The model expresses high confidence for front-row turbines, where the inflow conditions are stable and uniform, whereas its confidence decreases further downwind, where wind behavior is more chaotic due to wake interactions.

Boundary Effects and Non-Gaussian Behavior Beyond spatial and operational dependencies, the residual analysis exposes a fundamental mismatch between the chosen probabilistic framework and the physical reality of wind power generation. As shown by the high skewness and excess kurtosis in Table 4.7, all architectures exhibit non-Gaussian error distributions, particularly at operational extremes. These boundary effects are a consequence of predicting normalized power, which is strictly bounded within the unit interval [0,1]. When a turbine operates at its rated capacity, any prediction error is inherently constrained to be one-sided, as a turbine cannot physically generate more than its maximum rating. Conversely, at zero production, errors are similarly constrained, as negative generation is physically impossible. While SCADA data may contain marginal negative values due to internal consumption, these are negligible in this context. Consequently, as predictions approach these thresholds, the error distributions become truncated and asymmetrical. This contradicts the assumptions of the Gaussian NLL objective, which presupposes symmetric, infinite support. By forcing a Gaussian distribution onto strictly bounded data, the models erroneously allocate probability mass to physically impossible states, e.g., predicting a likelihood of power output being negative or larger than one. This mismatch limits the calibration ceiling and likely contributes to the observed underconfidence in reliability metrics.

5.2.2 Limitations and Future Work

While the approach of developing a post-hoc heteroscedastic variance estimator showed superior results over the homoscedastic variants, it represents only one methodology within the broader field of uncertainty quantification. The post-hoc estimator was chosen for its architectural simplicity and computational efficiency, effectively decoupling the training of the mean predictor from the variance estimation. However, future research should comprehensively benchmark this approach against other prominent methodologies, such as Deep Ensembles, Monte Carlo Dropout, or full Bayesian Neural Networks. Comparing these frameworks would clarify whether the post-hoc approach provides the optimal trade-off between calibration accuracy, interval sharpness, and the computational overhead required for real-time WFC.

A second critical limitation lies in the current distributional assumptions. As identified in the residual analysis, the adherence to a Gaussian NLL objective introduces significant boundary artifacts, forcing a symmetric, infinite distribution onto bounded, non-Gaussian data. To mitigate these structural frictions, future iterations of the probabilistic model must explore alternative formulations. Parametric approaches could replace the Gaussian assumption with a Beta distribution, which is naturally constrained to the $[0, 1]$ interval of normalized power. Alternatively, distribution-free methods, such as Quantile Regression, could be employed to directly predict specific confidence intervals without imposing any predefined statistical shape, potentially yielding superior calibration at the operational extremes.

Furthermore, the current heteroscedastic formulation predicts a single variance value representing total predictive uncertainty. For advanced control applications, it is beneficial to disentangle this total variance into its constituent parts: aleatoric uncertainty and epistemic uncertainty. Isolating epistemic uncertainty would provide a direct mechanism for out-of-distribution detection, explicitly warning a controller when the model is operating beyond its learned spatial or environmental bounds.

Ultimately, resolving these limitations is crucial for realizing the primary operational advantage of probabilistic modeling: robust optimization for wake steering. Deterministic surrogate models provide rigid point estimates, forcing optimization algorithms to blindly trust predictions. This risks net power losses if a calculated yaw configuration fails to produce the expected downstream wake deflection. By contrast, a calibrated, uncertainty-aware GT empowers risk-averse control strategies. By accurately quantifying its confidence, identifying low uncertainty at the free-stream rows and high uncertainty deep within compounded wakes, the model allows a controller to optimize aggressively where predictability is high, while adopting conservative baseline strategies in chaotic regimes, thereby bounding the downside risk of wake steering.

5.3 Generalization to Real Data

In addressing RQ3, the evaluation on empirical SCADA data reveals a substantial performance gap driven by the domain shift between the synthetic training environment and real-world operational conditions. When deployed in a zero-shot setting, all model architectures exhibit a notable deterioration in performance, characterized by degraded predictive accuracy and severe miscalibration in uncertainty quantification. However, the results indicate that this gap is not an unrecoverable architectural failure, but rather a calibrational offset. Because the GT models pre-learned robust spatial and aerodynamic priors from the diverse synthetic dataset, they provide an effective foundation for domain adaptation. As detailed in the following analysis, this allows for rapid and effective recovery of both accuracy and distributional calibration through finetuning.

5.3.1 Evaluating Domain Adaptation

The Zero-Shot Domain Shift The initial zero-shot evaluation serves to isolate the models' capacity for physical law extrapolation when applied to unseen, real-world scenarios. As evidenced by the results in Table 4.8, the homoscedastic MLPs exhibit a substantial deterioration in predictive performance. This indicates that the MLPs overfit to the specific spatial and operational distributions of the synthetic training set, thereby lacking the generalization capabilities required for real-world wind farm data. In contrast, the GT architecture demonstrate superior zero-shot robustness compared to the MLP variants. It remains unclear why exactly the GT was more robust to the SCADA data. One hypothesis would be that the GT's global aggregation is more invariant to distribution shift than the local MLP features.

However, this robustness in mean prediction does not extend to the heteroscedastic uncertainty quantification. In the zero-shot setting, the heteroscedastic GT exhibits a pronounced loss of calibration, characterized by a substantial inflation of the NLL and a significant degradation in 95% PICP. When faced with unmodeled real-world stochasticity, such as sensor inaccuracies, or complex dynamics absent from the synthetic data, the heteroscedastic estimator fails to recognize its own epistemic error. It incorrectly processes the unseen SCADA inputs as highly predictable states, outputting overly confident prediction bounds (MPIW). Unexpectedly, the conservative uncertainty bounds of the homoscedastic GT prove to be the safer zero-shot alternative, achieving the lowest CRPS.

Recovery and Adaptation via Finetuning The finetuning phase mitigated the initial domain shift impact, allowing the models to recalibrate to the specific profile of the SCADA dataset. In terms of point-prediction accuracy, the homoscedastic MLP demonstrated a notable recovery. Nevertheless, the GT architecture retained its performance advantage, consistently achieving superior error metrics. The most significant effects of domain adaptation were observed in the probabilistic performance. As illustrated in the sharpness analysis (Figure 4.7), the transition from synthetic to real-world data necessitated a substantial expansion of the predictive intervals. For the homoscedastic models, this recalibration resulted in static interval widths nearly double those observed in the synthetic case. This shift represents the models' global acknowledgement of the increased predictive uncertainty. Whether this was caused by aleatoric uncertainty, such as sensor noise and unmodeled dynamics, or epistemic uncertainty is unclear. Nevertheless, by adopting a single variance estimate, these models remain structurally limited by persistent underconfidence as shown in the reliability plot. In contrast, the finetuned heteroscedastic GT provides much sharper intervals for many samples, while estimating higher uncertainty for others. The superior NLL and CRPS indicate that the model has identified specific regimes where the uncertainty is greater.

5.3.2 Limitations and Future Work

A critical phenomenon observed across the experiments was the significant performance degradation resulting from the domain shift between synthetic and empirical environments. While the naive finetuning approach employed in this thesis facilitated a substantial recovery in predictive accuracy and uncertainty, several fundamental questions regarding model robustness remain. Future research should prioritize a systematic decomposition of this domain shift to isolate the specific contributors, such as sensor bias or complex terrain effects, that drive the zero-shot deterioration. Identifying these factors would enable the development of more resilient architectures or targeted data-augmentation strategies during pre-training. Furthermore, the integration of explicit out-of-distribution detection is essential to prevent overconfident extrapolation in novel environments. Parallel to this, investigating few-shot adaptation techniques would be highly valuable to determine the minimal volume of local SCADA data required for recalibration. Finally, as wind farm dynamics evolve over decadal lifespans due to hardware degradation or environmental shifts, the exploration of continual learning frameworks will be necessary to ensure the surrogate model remains accurate and relevant throughout the turbine lifecycle.

Furthermore, while this study provided an initial evaluation of the SCADA dataset and the resulting model behavior, several facets of the domain require deeper investigation. A primary objective for future research should be the empirical estimation of wake losses within the SCADA data to serve as a benchmark for model performance. Regarding internal model dynamics, a more rigorous interpretability analysis is essential. For instance, examining the GT's attention patterns and conducting a comprehensive sensitivity analysis would clarify how the network weights various operational and spatial features. Such insights are critical to ensuring the model relies on physical aerodynamic reasoning rather than spurious statistical correlations.

Ultimately, future efforts should prioritize the enhancement of predictive accuracy and uncertainty quantification by addressing current informational limitations in the feature set. A primary disadvantage of the existing approach is the anonymity of the turbine entities. While they are characterized by general physical attributes, they lack a unique identifier. In empirical settings, turbine models may exhibit distinct performance signatures due to localized terrain effects or mechanical wear. Introducing unique node-level identifiers could allow the GT to learn these latent, turbine-specific behaviors more easily. Furthermore, the current methodology relies on implicit wind direction by rotating the input spatial coordinates, a technique that assumes rotational invariance across the wind farm. While this is sufficient for synthetic environments with uniform inflow, real-world atmospheric conditions and local topographical influences most probably vary extensively by wind direction. Treating wind direction as an explicit feature could lead to better performance.

Finally, from a practical deployment perspective, the current preprocessing pipeline involves heavy filtering, where many timestamps were discarded due to assumed maintenance or curtailment. While this ensures high accuracy on the remaining samples, it introduces a bias that limits the model's utility in real-time farm management. A truly robust surrogate must be capable of handling data of a rawer nature, including dynamic graph topologies where turbines may go offline or report anomalous sensor freezes.

6 Conclusion

This thesis focused on the development and evaluation of ML-based surrogate models for wind farm wake modeling. To assess these models, a comprehensive pipeline was developed to generate heterogeneous samples that encompass diverse atmospheric conditions, heterogeneous turbine models, and realistic wind farm layouts derived from the MaStR [67] and Open-Meteo meteorological data [69]. The primary model architecture under investigation was a GT that leverages self-attention mechanisms to utilize information from across the entire wind farm to estimate each individual turbine's power output. Its performance was systematically compared against turbine-level MLP baselines. The research examined model performance across three distinct levels of complexity, beginning with an evaluation on the synthetic dataset. Subsequently, the models' ability to estimate predictive uncertainty was evaluated by training a heteroscedastic variance estimator post-hoc and comparing it against homoscedastic formulations. Finally, the models were evaluated on a real-world SCADA dataset in both zero-shot and finetuning settings.

Addressing the first research question, the comparative analysis demonstrates that the GT architecture consistently outperforms the MLP variants in predictive accuracy for diverse wind conditions and in heterogeneous environments. It achieves a normalized power MAE of 0.019 compared to 0.059 for the MLP-Spatial baseline. However, these metrics have to be interpreted carefully. As the $R^2 = 0.902$ of the MLP-NonSpatial shows, a large amount of variance can be explained without considering wake interactions. Metrics on the normalized wake loss give a more nuanced view. Here, the MLP-NonSpatial only achieves an R^2 of 0.016 and is significantly outperformed by the GT with 0.841. Overall, by integrating spatial context, the GT can capture the underlying dynamics of aerodynamic wake interactions better and proves more robust to atmospheric complexity and heterogeneous turbines within the evaluated scenarios compared to the MLP variants. However, the GT's advantage is not unconditional. A major constraint is its reliance on absolute spatial coordinates, which hinders its capability to extrapolate to larger, unseen wind farm layouts. Future iterations of the GT architecture should investigate the integration of relative distance vectors as edge features to enhance spatial generalization.

Regarding the quantification of predictive uncertainty, the evaluation revealed that moving beyond homoscedastic assumptions is critical for accurate modeling. The post-hoc heteroscedastic variance estimator identifies the steep region of the power curve around 11 m/s as the highest-uncertainty operational regime, and assigns increasing uncertainty to turbines further downstream. As a result, it achieves a superior balance of interval sharpness and distributional calibration compared to both the homoscedastic GT and MLP baselines. A further caveat applies to all probabilistic models evaluated here. The residual analysis reveals strongly non-Gaussian error distributions, particularly at the ends of the normalized power range. This poses a fundamental limitation to the probabilistic models that assume Gaussian residuals. Subsequent research should prioritize replacing the Gaussian assumption with alternative, non-Gaussian formulations to resolve these discrepancies. Finally, disentangling the predicted variance into explicit aleatoric and epistemic components would enable active out-of-distribution detection, allowing control algorithms to optimize aggressively in predictable regimes while adopting conservative strategies in chaotic environments.

To answer the third research question, applying models trained purely on synthetic data to empirical, real-world conditions offers a nuanced answer. On predictive accuracy, the GT proved more robust achieving $R^2 = 0.886$ compared to 0.541 for the MLP-NonSpatial and 0.131 for the MLP-Spatial. The uncertainty behavior in the zero-shot setting is completely different, however. The heteroscedastic GT produced a NLL of over 14 indicating a severe deterioration. Rather than inflating its uncertainty in response, it confidently assigns narrow intervals to predictions that are in fact poor, because the input features do not trigger the patterns it associates with high uncertainty. Following a brief finetuning phase, the GT adapted to the real-world conditions, partially recovering its predictive accuracy to $R^2 = 0.917$ and uncertainty estimates to a NLL of -1.370 with well-calibrated interval coverage. Whether this performance is sufficient for operational wake steering deployment depends on the specific control objective and risk tolerance, and calls for dedicated future investigation.

This thesis takes a step toward reliable, uncertainty-aware wind farm surrogate modeling by demonstrating that a GT trained on simulated data can capture aerodynamic interactions across heterogeneous turbine types and diverse atmospheric conditions. The architecture's ability to partially recover predictive accuracy and calibration through fine-tuning on real SCADA data confirms that pre-training on synthetic environments is a viable strategy. Provided its current limitations are addressed, the GT could serve as a probabilistic surrogate model for wake steering optimization under uncertainty, translating the aerodynamic complexity of heterogeneous wind farms into actionable, uncertainty-aware optimization decisions that meaningfully increase energy yield.

Bibliography

1. International Energy Agency. *World Energy Outlook 2025* (IEA, Paris, 2025). https://www.iea.org/reports/world-energy-outlook-2025.
2. International Energy Agency. *Electricity Mid-Year Update 2025* (IEA, July 30, 2025). https://www.iea.org/reports/electricity-mid-year-update-2025.
3. World Wind Energy Association. *WWEA Half-Year Report 2025* (WWEA, Bonn, Germany, 2025). https://wwindea.org/HYR2025.
4. Global Wind Energy Council. *Global Wind Report 2025* (GWEC, Brussels, Belgium, Apr. 2025). https://www.gwec.net/reports/globalwindreport.
5. Fraile, D. *et al. Getting Fit for 55 and Set for 2050* (WindEurope). https://windeurope.org/data/products/getting-fit-for-55-and-set-for-2050.
6. Yang, K. *et al.* A Data-Driven Layout Optimization Framework of Large-Scale Wind Farms Based on Machine Learning. *Renewable Energy* 218, 119240. doi:10.1016/j.renene.2023.119240 (Dec. 1, 2023).
7. Boersma, S. *et al. A Tutorial on Control-Oriented Modeling and Control of Wind Farms* in *2017 Am. Control Conf. ACC* (May 2017), 1–18. doi:10.23919/ACC.2017.7962923.
8. Park, J. & Park, J. Physics-Induced Graph Neural Network: An Application to Wind-Farm Power Estimation. *Energy* 187, 115883. doi:10.1016/j.energy.2019.115883 (Nov. 15, 2019).
9. Wingerden, J. van *et al.* Expert Elicitation on Wind Farm Control. *J. Phys.* doi:10.1088/1742-6596/1618/2/022025 (2020).
10. Wang, H. *et al.* Wake Steering in Offshore Wind Farms: Field Experiment and SCADA-driven Simulations for Quantifying Power and Blade Root Loads. *Ocean Engineering* 347, 124001. doi:10.1016/j.oceaneng.2025.124001 (Feb. 15, 2026).
11. Boersma, S. Towards Closed-Loop Dynamical Wind Farm Control: Model Development and Control Applications. doi:10.4233/uuid:48572080-bc51-4ffe-9ba5-676ee9ab5fcc (2019).
12. Duthé, G. *et al.* Flexible Multi-Fidelity Framework for Load Estimation of Wind Farms through Graph Neural Networks and Transfer Learning. *Data-Centric Eng.* 5, e29. doi:10.1017/dce.2024.35 (Jan. 2024).
13. Frutuoso, T. R. L. *et al.* Advancements in Wind Farm Control: Modelling and Multi-Objective Optimization Through Yaw-Based Wake Steering. *Energies* 18, 2247. doi:10.3390/en18092247 (Jan. 2025).

14. Harrison-Atlas, D. *et al.* Artificial Intelligence-Aided Wind Plant Optimization for Nationwide Evaluation of Land Use and Economic Benefits of Wake Steering. *Nat Energy* 9, 735–749. doi:`10.1038/s41560-024-01516-8` (June 2024).

15. Nai-Zhi, G. *et al.* A Data-Driven Analytical Model for Wind Turbine Wakes Using Machine Learning Method. *Energy Conversion and Management* 252, 115130. doi:`10.1016/j.enconman.2021.115130` (Jan. 15, 2022).

16. Yang, S. *et al.* Machine-Learning-Based Wind Farm Optimization through Layout Design and Yaw Control. *Renewable Energy* 224, 120161. doi:`10.1016/j.renene.2024.120161` (Apr. 1, 2024).

17. Kim, M. & Park, S. Data-Driven Approach for Wind Farm Control: Toward an Alternative to FLORIS. *IEEE Access* 12, 13327–13339. doi:`10.1109/ACCESS.2024.3356611` (2024).

18. Yang, S. *et al.* Cooperative Yaw Control of Wind Farm Using a Double-Layer Machine Learning Framework. *Renewable Energy* 193, 519–537. doi:`10.1016/j.renene.2022.04.104` (June 1, 2022).

19. Xu, Z. *et al.* Model-Guided Learning for Wind Farm Power Optimization. *IEEE Trans. Control Syst. Technol.* 32, 428–439. doi:`10.1109/TCST.2023.3315547` (Mar. 2024).

20. Andersson, L. E. & Imsland, L. Real-Time Optimization of Wind Farms Using Modifier Adaptation and Machine Learning. *Wind Energy Sci.* 5, 885–896. doi:`10.5194/wes-5-885-2020` (July 13, 2020).

21. Ti, Z. *et al.* Artificial Neural Networks Based Wake Model for Power Prediction of Wind Farm. *Renew. Energy.* doi:`10.1016/J.RENENE.2021.03.030` (2021).

22. Almeida, J. O. & Rochinha, F. A. A Probabilistic Learning Approach Applied to the Optimization of Wake Steering in Wind Farms. *J. Comput. Inf. Sci. Eng* 23. doi:`10.1115/1.4054501` (June 6, 2022).

23. Anagnostopoulos, S. J. *et al.* Accelerated Wind Farm Yaw and Layout Optimisation with Multi-Fidelity Deep Transfer Learning Wake Models. *Renewable Energy* 218, 119293. doi:`10.1016/j.renene.2023.119293` (Dec. 1, 2023).

24. Bentsen, L. Ø. *et al.* Wind Park Power Prediction: Attention-Based Graph Networks and Deep Learning to Capture Wake Losses. *J. Phys.: Conf. Ser.* 2265, 022035. doi:`10.1088/1742-6596/2265/2/022035` (May 1, 2022).

25. Schøler, J. P. *et al.* Graph Neural Operator for Windfarm Wake Flow. *Wind Energy Sci. Discuss.,* 1–38. doi:`10.5194/wes-2025-261` (Dec. 4, 2025).

26. Santos, F. d. N. *et al.* Multivariate Prediction on Wake-Affected Wind Turbines Using Graph Neural Networks. *J. Phys.: Conf. Ser.* 2647, 112006. doi:`10.1088/1742-6596/2647/11/112006` (June 2024).

27. Hammer, F. *et al.* Graph Machine Learning for Predicting Wake Interaction Losses Based on SCADA Data. *J. Phys.: Conf. Ser.* 2505, 012047. doi:`10.1088/1742-6596/2505/1/012047` (May 2023).

28. Li, S. *et al.* Learning to Optimise Wind Farms with Graph Transformers. *Applied Energy* 359, 122758. doi:10.1016/j.apenergy.2024.122758 (Apr. 1, 2024).

29. Hulsman, P. *et al.* Optimizing Wind Farm Control through Wake Steering Using Surrogate Models Based on High-Fidelity Simulations. *Wind Energy Sci.* 5, 309–329. doi:10.5194/wes-5-309-2020 (Mar. 5, 2020).

30. Quick, J. *et al.* Wake Steering Optimization under Uncertainty. *Wind Energy Sci.* 5, 413–426. doi:10.5194/wes-5-413-2020 (Mar. 30, 2020).

31. Zhang, J. & Zhao, X. Quantification of Parameter Uncertainty in Wind Farm Wake Modeling. *Energy* 196, 117065. doi:10.1016/j.energy.2020.117065 (Apr. 1, 2020).

32. Aerts, F. *et al.* Bayesian Uncertainty Quantification Framework for Wake Model Calibration and Validation with Historical Wind Farm Power Data. *Wind Energy* 26, 786–802. doi:10.1002/we.2841 (2023).

33. Jordahn, M. *et al. Practical Deep Heteroskedastic Regression* arXiv: 2603.01750 [cs]. Pre-published.

34. Meyers, J. *et al.* Wind Farm Flow Control: Prospects and Challenges. *Wind Energy Sci.* 7, 2271–2306. doi:10.5194/wes-7-2271-2022 (Nov. 25, 2022).

35. Manwell, J. F. *et al. Wind Energy Explained: Theory, Design and Application* 706 pp. ISBN: 978-0-470-68628-7. Google Books: roaTx_Of0vAC (John Wiley & Sons, Sept. 14, 2010).

36. Churchfield, M. J. *et al.* A Numerical Study of the Effects of Atmospheric and Wake Turbulence on Wind Turbine Dynamics. *J. Turbul.* 13, 1–32. doi:10.1080/14685248.2012.668191 (2012).

37. Maronga, B. *et al.* The Parallelized Large-Eddy Simulation Model (PALM) Version 4.0 for Atmospheric and Oceanic Flows: Model Formulation, Recent Developments, and Future Perspectives. *Geosci. Model Dev.* 8, 2515–2551. doi:10.5194/gmd-8-2515-2015 (Aug. 13, 2015).

38. Boersma, S. *et al.* A Control-Oriented Dynamic Wind Farm Model: WFSim. *Wind Energy Sci.* 3, 75–95. doi:10.5194/wes-3-75-2018 (Mar. 6, 2018).

39. Jonkman, J. M. *et al.* in *35th Wind Energy Symposium* (American Institute of Aeronautics and Astronautics). doi:10.2514/6.2017-0454.

40. *NatLabRockies/Floris* National Laboratory of the Rockies, Apr. 3, 2026. https://github.com/NatLabRockies/floris.

41. Pedersen, Mads *et al. PyWake 2.5.0: An Open-Source Wind Farm Simulation Tool* DTU Wind, Technical University of Denmark, Feb. 2023. https://gitlab.windenergy.dtu.dk/TOPFARM/PyWake.

42. Jensen, N. *A Note on Wind Generator Interaction* Report 87-550-0971-9 (Risø National Laboratory, Roskilde, 1983).

43. Bastankhah, M. & Porté-Agel, F. A New Analytical Model for Wind-Turbine Wakes. *Renew. Energy.* doi:10.1016/J.RENENE.2014.01.002 (2014).

44. Fleming, P. *et al.* A Simulation Study Demonstrating the Importance of Large-Scale Trailing Vortices in Wake Steering. *Wind Energy Sci.* 3, 243–255. doi:10.5194/wes-3-243-2018 (May 14, 2018).

45. King, J. *et al.* Control-Oriented Model for Secondary Effects of Wake Steering. *Wind Energy Sci.* 6, 701–714. doi:10.5194/wes-6-701-2021 (May 21, 2021).

46. Bay, C. J. *et al.* Addressing Deep Array Effects and Impacts to Wake Steering with the Cumulative-Curl Wake Model. *Wind Energy Sci.* 8, 401–419. doi:10.5194/wes-8-401-2023 (Mar. 24, 2023).

47. Frederik, J. A. *et al.* Comparison of Wind Farm Control Strategies under Realistic Offshore Wind Conditions: Turbine Quantities of Interest. *Wind Energy Sci.* 10, 755–777. doi:10.5194/wes-10-755-2025 (Apr. 28, 2025).

48. Ye, M. *et al.* Overview of Data-Driven Models for Wind Turbine Wake Flows. *J. Marine. Sci. Appl.* 24, 1–20. doi:10.1007/s11804-025-00683-8 (Feb. 1, 2025).

49. Yang, S. *et al.* Smart Cooperative Control Scheme for Large-Scale Wind Farms Based on a Double-Layer Machine Learning Framework. *Energy Conversion and Management* 285, 116949. doi:10.1016/j.enconman.2023.116949 (June 1, 2023).

50. Korb, H. *et al.* Exploring the Application of Reinforcement Learning to Wind Farm Control. *J. Phys.: Conf. Ser.* 1934, 012022. doi:10.1088/1742-6596/1934/1/012022 (May 2021).

51. Gionfra, N. *et al.* Wind Farm Distributed PSO-based Control for Constrained Power Generation Maximization. *Renewable Energy* 133, 103–117. doi:10.1016/j.renene.2018.09.084 (Apr. 1, 2019).

52. Bastankhah, M. & Porté-Agel, F. Wind Farm Power Optimization via Yaw Angle Control: A Wind Tunnel Study. *J. Renew. Sustain. Energy.* doi:10.1063/1.5077038 (2019).

53. Battaglia, P. W. *et al.* *Relational Inductive Biases, Deep Learning, and Graph Networks* arXiv: 1806.01261 [cs]. Pre-published.

54. Gilmer, J. *et al.* *Neural Message Passing for Quantum Chemistry* in *Proc. 34th Int. Conf. Mach. Learn. - Vol. 70* (JMLR.org, Sydney, NSW, Australia, Aug. 6, 2017), 1263–1272. https://dl.acm.org/doi/10.5555/3305381.3305512.

55. Veličković, P. *et al.* Graph Attention Networks. *Int. Conf. Learn. Represent.* https://openreview.net/forum?id=rJXMpikCZ (2018).

56. Müller, L. *et al.* *Attending to Graph Transformers* arXiv: 2302.04181 [cs]. Pre-published.

57. Hüllermeier, E. & Waegeman, W. Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. *Mach Learn* 110, 457–506. doi:10.1007/s10994-021-05946-3 (Mar. 1, 2021).

58. He, W. *et al.* A Survey on Uncertainty Quantification Methods for Deep Learning. *ACM Comput. Surv.* 58, 179:1–179:35. doi:10.1145/3786319 (Feb. 3, 2026).

59. Wang, F. *et al.* *Uncertainty in Graph Neural Networks: A Survey* arXiv: 2403.07185 [cs]. Pre-published.

60. Abdar, M. *et al.* A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges. *Information Fusion* 76, 243–297. doi:10.1016/j.inffus.2021.05.008 (Dec. 1, 2021).

61. Malinin, A. & Gales, M. *Predictive Uncertainty Estimation via Prior Networks* in *Adv. Neural Inf. Process. Syst.* (eds Bengio, S. *et al.*) **31** (Curran Associates, Inc., 2018).

62. Pedregosa, F. *et al.* Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830 (2011).

63. The pandas development team. *Pandas-Dev/Pandas: Pandas* version latest. Zenodo, Feb. 2020. doi:10.5281/zenodo.3509134.

64. Harris, C. R. *et al.* Array Programming with NumPy. *Nature* 585, 357–362. doi:10.1038/s41586-020-2649-2 (Sept. 2020).

65. Virtanen, P. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* 17, 261–272. doi:10.1038/s41592-019-0686-2 (2020).

66. Ansel, J. *et al. PyTorch 2: Faster Machine Learning through Dynamic Python Bytecode Transformation and Graph Compilation* in *Proc. 29th ACM Int. Conf. Archit. Support Program. Lang. Oper. Syst.* **2** (ACM, Apr. 2024). doi:10.1145/3620665.3640366.

67. Bundesnetzagentur. *Marktstammdatenregister* 2026. https://www.marktstammdatenregister.de.

68. *NatLabRockies/Turbine-Models* National Laboratory of the Rockies, Apr. 3, 2026. https://github.com/NatLabRockies/turbine-models.

69. Zippenfenig, P. *Open-Meteo.Com Weather API* 2023. doi:10.5281/zenodo.7970649.

70. Ester, M. *et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise* in *Proc. Second Int. Conf. Knowl. Discov. Data Min.* (AAAI Press, Portland, Oregon, Aug. 2, 1996), 226–231.

71. Met Office. *Cartopy: A Cartographic Python Library with a Matplotlib Interface* manual (Exeter, Devon, 2010–2015). https://cartopy.readthedocs.io.

72. *Pyproj4/Pyproj* pyproj4, Apr. 6, 2026. https://github.com/pyproj4/pyproj.

73. Deutscher Wetterdienst. *Short-Term Forecasting Model ICON-D2* Data set. 2021. http://data.europa.eu/88u/dataset/0d6ec830-6f73-4f3c-8147-e41e43eaf40f.

74. Gneiting, T. & Raftery, A. E. Strictly Proper Scoring Rules, Prediction, and Estimation. *J. Am. Stat. Assoc.* 102, 359–378. doi:10.1198/016214506000001437 (Mar. 1, 2007).

75. Khosravi, A. *et al.* Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances. *IEEE Trans. Neural Netw.* 22, 1341–1356. doi:10.1109/TNN.2011.2162110 (Sept. 2011).

76. Bauer, L. *Welcome to Wind-Turbine-Models.Com* https://en.wind-turbine-models.com/.