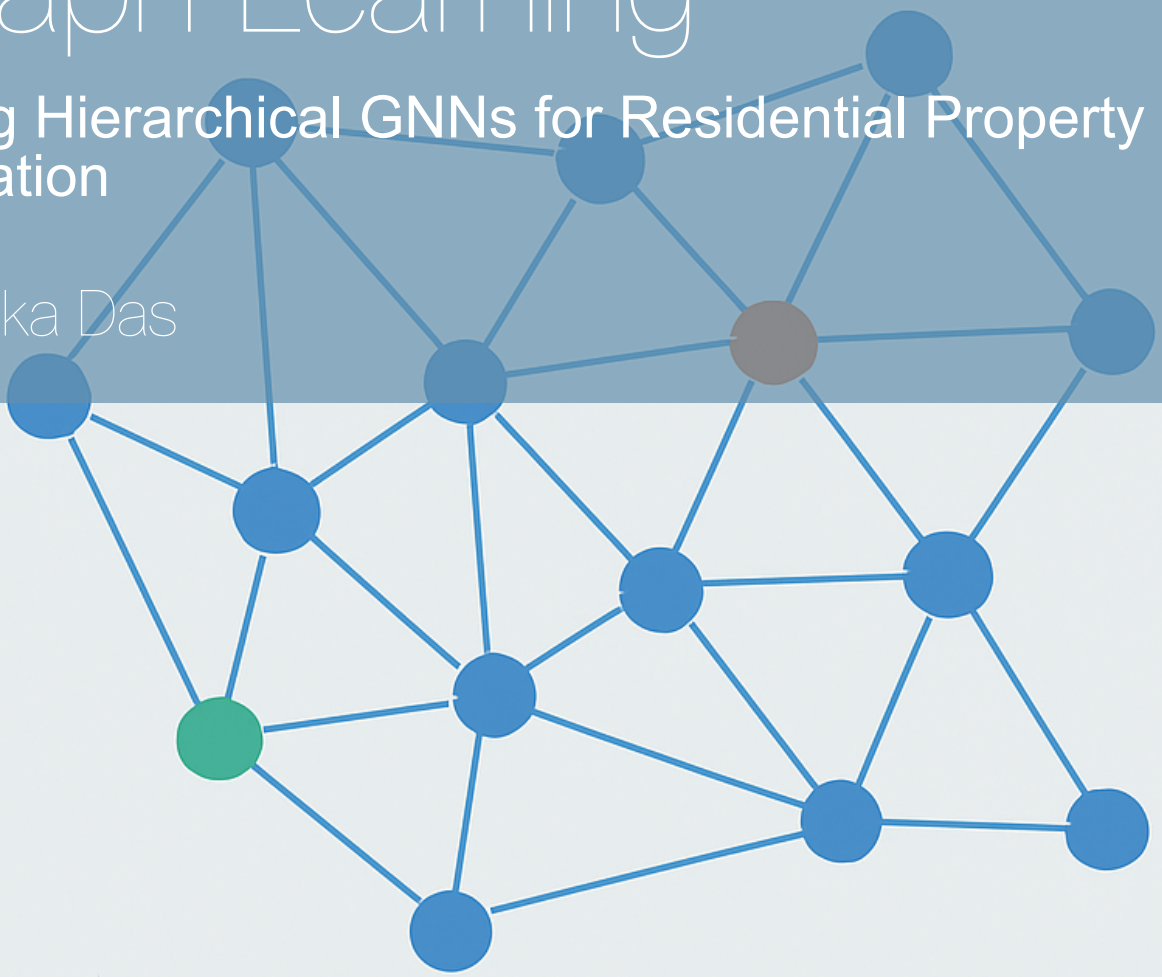
Bidirectional Multi-Scale Graph Learning

Using Hierarchical GNNs for Residential Property Valuation

Aratrika Das



TU Delft

ORTEC FINANCE

Bidirectional Multi-Scale Graph Learning

Using Hierarchical GNNs for Residential Property Valuation

by

Aratrika Das

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on August 28, 2025 at 15:00

Thesis committee:

Chair:	Dr. Elvin Isufi (TU Delft)
Supervisors:	Afrasiab Kadhum (Ortec Finance)
	Julian van Erk (Ortec Finance)
External examiner:	Dr. Kubilay Atasu (TU Delft)
Faculty:	Faculty of Electrical Engineering, Mathematics and Computer Science, TU Delft
Student number:	5298121
Project Duration:	November 2024 - August 2025

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Copyright © Aratrika Das, 2025
All rights reserved.

Preface

This thesis marks the end of my master's studies in Computer Science at TU Delft. This journey has been both one of the most challenging and most rewarding experiences of my life. It has pushed me far beyond my comfort zone and tested my perseverance. This work is not just the outcome of months of research and writing, but also the result of the encouragement, patience, and love I have received along the way. I am deeply grateful to my supervisors, who guided me with wisdom, kindness, and constructive criticism, always encouraging me to think critically while giving me the freedom to explore my own ideas.

I would like to express my deepest appreciation to my supervisor from TU Delft, Dr. Elvin Isufi, for his guidance and academic insight throughout the research process. Your knowledge and passion in this field has been truly inspiring and has also helped nurture my interest in this area. I would also like to extend my thanks to Dr. Kubilay Atasu from TU Delft for serving on my thesis committee and for his time in reviewing my work.

I am also extremely grateful to my supervisors from Ortec Finance, Afrasiab Kadhum and Julian van Erk, for their constant support throughout these past 10 months. Thank you for all the time and industry expertise that you have dedicated towards this project. Without your feedback and guidance this thesis would not be possible. I would also like to express my gratitude to all my colleagues at Ortec Finance. Thank you for for the many inspiring conversations, thoughtful debates, and moments of laughter we shared.

To my family, I owe the deepest thanks. Your unconditional love and belief in me kept me going during moments of doubt and exhaustion. Your support and sacrifices have made me who I am today. To my friends, thank you for reminding me to take breaks, for cheering me on when progress felt slow, and for simply being there when I needed it most. Your friendship and support have been invaluable to me. It has been an incredible journey and I deeply treasure the moments that we have shared and the memories that we have made.

With this thesis, I close an important chapter of my life, while looking ahead with excitement to the next. Whatever challenges the future holds, I know I will carry the lessons and the support I have received here with me.

Aratrika Das
Delft, August 2025

Summary

Residential property valuation is a core component of real estate markets, financial systems, and public policy. Accurate estimation of property values is essential for tasks such as mortgage lending, taxation, urban planning, and assessing housing affordability. However, property valuation remains a difficult challenge because of the complexity of housing markets. Prices are influenced not only by physical attributes such as size, age, and amenities, but also by spatial spillovers, socio-economic conditions, and evolving market dynamics. Traditional econometric models, grounded in hedonic pricing theory, provide interpretable baselines but rest on strong assumptions that rarely hold in practice. Machine learning-based automated valuation models improve predictive accuracy by capturing non-linear feature interactions without making as many assumptions and incorporating diverse data sources. However, they usually treat transactions as independent observations and fail to adequately model spatial or temporal dependencies. More recently, graph-based models have emerged to address spatial autocorrelation, but most rely on static structures or unidirectional hierarchies, which limit their capacity to represent the dynamic and multi-scale character of urban housing systems.

This thesis proposes a Multi-Scale Bidirectional Spatio-Temporal Graph Neural Network (MBSTGNN) for residential property valuation. The model explicitly integrates spatial, temporal, and hierarchical dependencies by representing the housing market as a sequence of dynamic graphs. At the micro level, nodes represent property transactions, while at the macro level, nodes represent neighbourhoods. Bidirectional message passing enables information to flow between these two levels, ensuring that neighbourhood trends are informed by underlying transactions, and conversely, that individual valuations account for broader regional conditions. Temporal dependencies are modelled using a memory mechanism at the macro level, which carries context across discrete time snapshots without relying on recurrent sequence models. Graphs are updated dynamically at each time step to reflect evolving market structure, neighbourhood attributes, and transaction activity. This architecture addresses key gaps in the literature by enabling joint modelling of multi-scale spatial context and temporally coherent dynamics in real estate valuation.

The model is evaluated on a proprietary dataset of housing transactions in Rotterdam, with synthetic data generation employed to ensure privacy preservation. Experimental results demonstrate that MBSTGNN outperforms both a strong tabular baseline (LightGBM), and a macro-level graph based baseline constructed for comparison. In rolling-window forecasting, the model achieves consistently higher predictive accuracy and robustness, particularly in sparse-data regions and among diverse property types. Embedding visualisations further reveal that the model captures domain-consistent patterns: neighbourhoods with similar socio-economic and structural profiles cluster together, even when geographically distant, while temporal embeddings evolve smoothly across years, reflecting gradual market shifts. A qualitative case study of selected Rotterdam neighbourhoods corroborates these findings, showing alignment between the model's learned representations and expert knowledge of urban dynamics. In positioning this work within the broader literature, theoretical comparisons are drawn with state-of-the-art graph-based frameworks such as MugRep, ReGram, and ST-RAP, highlighting how MBSTGNN addresses key limitations in hierarchical and temporal modelling.

In summary, this thesis advances the state of automated property valuation by bridging methodological gaps between econometric, machine learning, and graph-based approaches. By combining predictive performance with structural interpretability, MBSTGNN provides a scalable framework that enhances both academic understanding and practical applications of real estate analytics.

Contents

List of Figures vi

List of Tables viii

1 Introduction 1
- 1.1 Research Questions . . . 3
- 1.2 Thesis Structure . . . 3

2 Background 4
- 2.1 Introduction to Property Valuation . . . 5
- 2.2 Graphs . . . 8
- 2.3 Graphs in Real Estate . . . 10
- 2.4 Graph Neural Networks . . . 11
- 2.5 Advanced Graph Representations and Models . . . 14
- 2.6 Gating Mechanisms . . . 15

3 Literature Review 17
- 3.1 Econometric Approaches to Property Valuation . . . 18
- 3.2 Machine Learning-Based Automated Valuation Models . . . 19
- 3.3 GNNs in Real Estate . . . 20
- 3.4 Hierarchical Graph Learning . . . 21
- 3.5 Research Gaps . . . 22

4 Methodology 23
- 4.1 Problem Formulation . . . 24
- 4.2 Overview . . . 24
- 4.3 Spatio-Temporal Graph Construction . . . 25
- 4.4 Proposed Model . . . 26
- 4.5 Training . . . 28
- 4.6 Discussion . . . 30

5 Experiments 33
- 5.1 Data . . . 34
- 5.2 Data Preprocessing . . . 36
- 5.3 Experimental Setup . . . 36
- 5.4 Baselines . . . 37
- 5.5 Ablation Studies . . . 38
- 5.6 Evaluation Metrics . . . 39
- 5.7 Significance Testing . . . 39

6 Results and Discussion 40
- 6.1 Overall Performance Comparison . . . 41
- 6.2 Temporal Generalisation . . . 42
- 6.3 Spatial Analysis . . . 44
- 6.4 Case Study: Embedding Interpretation . . . 46
- 6.5 Discussion . . . 48

7 Conclusion 51
- 7.1 Thesis Summary . . . 52
- 7.2 Key Findings . . . 52
- 7.3 Future Work . . . 53
- 7.4 Disclosure . . . 53

References **58**

A Model Structure **59**
A.1 Macro Encoder Structure. 59
A.2 Micro Encoder Structure 59
A.3 Hyperparameter Tuning 60

B Additional Results **61**
B.1 Residual Distribution 61
B.2 Additional Results: Spatial Analysis 62
B.3 Ablation Study 63
B.4 Online Learning. 63

C Data Description **64**
C.1 Characteristics of the Transaction Datasets. 64
C.2 Description of Transaction Dataset Features 65
C.3 Description of Neighbourhood Dataset Features 65

D Synthetic Data **68**
D.1 Overall Comparison 68
D.2 Temporal Generalisation 69
D.3 Spatial Analysis. 70

List of Figures

1.1 Heatmap of the average price per m^2 of the neighbourhoods of region of Rijnmond for 3 years. These plots highlight the spatio-temporal nature of real estate data. 2

2.1 Comparison of different types of graphs. Each subfigure shows two graph types stacked vertically. 8
2.2 Illustration of two types of dynamic graphs. (a) A *Discrete-Time Dynamic Graph* (DTDG) is represented as a sequence of graph snapshots G^0, G^1, G^2 at fixed time intervals. (b) A *Continuous-Time Dynamic Graph* (CTDG) records a time-ordered stream of structural events, each with a precise timestamp. 10
2.3 Detailed illustration of the message passing process in GNNs. Each neighbour $u_i \in \mathcal{N}(v)$ sends a message $m_{u_i \to v}$ to v, which are aggregated (e.g., sum/mean/max), and then combined with v's current state via a neural update function to yield $\mathbf{h}_v^{(l+1)}$. 12
2.4 Stacked graph convolution operations adapted from [39, Figure 2] 12

3.1 Classification of literature in property valuation and hierarchical graph learning 18
3.2 Architectural comparison of GNN-based real estate valuation models. MugRep uses separate graphs for transactions and communities. ReGram enhances relational weighting among neighbors. ST-RAP incorporates heterogeneous urban context and temporal encoding. 20

4.1 Hierarchical spatio-temporal graph at timestep t. Micro-level (transaction) nodes form fully connected sub-graphs within each neighbourhood, macro-level (neighbourhood) nodes are linked by geographic edges, and dotted cross-level edges connect each transaction to its neighbourhood 25
4.2 Architecture of the proposed MBSTGNN model. The model consists of micro-level and macro-level encoders, implemented with GraphSAGE and GCN layers, respectively. A cross-level interaction module enables bidirectional information flow: bottom-up propagation aggregates transaction patterns into neighbourhood representations, while top-down propagation incorporates neighbourhood context back into transaction embeddings. 29

5.2 Macro-Level Graph Baseline which learns the embedding of the neighbourhood graph and then uses an MLP to estimate the price after combining with the transaction features. . . . 38

6.1 Temporal generalisation performance: MSE and MAPE variation across evaluation windows for each model. Shaded areas show the range between the 5th and the 95th percentile. . . 43
6.2 Fan chart illustrating the predicted versus actual median price per square meter over time, with interquartile range (IQR) shading. The chart compares MBSTGNN, LightGBM, and the macro-level baseline. The dashed black line represents the actual median prices. The MBSTGNN closely follows market trends with tighter uncertainty bands, indicating improved accuracy and consistency, particularly during periods of volatility. 44
6.3 Spatial evaluation of model performance across neighbourhoods in Rotterdam. Each row shows three spatial heatmaps for a model, revealing spatial variation in predictive accuracy and data coverage. 45
6.4 Map of Rotterdam with the neighbourhoods selected for the case study highlighted 46
6.5 Visualisation of projected UMAP Embeddings for Spatial and Temporal Analysis 47

A.1 Comparison of performance of Variations of the Macro Encoder 59
A.2 Comparison of performance of Variations of the Micro Encoder 59
A.3 Comparison of performance of MBSTGNN when using different hyperparameters 60

B.1 Residual Distribution across different windows 61

B.2 Comparison of performance of Ablated Variants of MBSTGNN through time 63
B.3 Comparison of performance of Variations of the Cross-Level Interaction Module 63
B.4 Comparison of the Performance of Online and Offline Learning of MBSTGNN 63

C.1 Correlation matrix and distribution of prices of both the transaction dataset 64

D.1 Residual Distribution across all windows of models trained on the synthetic dataset 68
D.2 Comparison of the Performance of LightGBM and MBSTGNN trained on the synthetic dataset 69
D.3 Fan chart illustrating the predicted versus actual median price per square meter over time, with interquartile range (IQR) shading. The chart compares MBSTGNN,and LightGBM. The dashed black line represents the actual median prices. 69
D.4 Spatial evaluation of model performance across neighbourhoods in Rotterdam. From left to right: (a) Mean Squared Error (MSE), (b) Mean Absolute Percentage Error (MAPE), and (c) number of transactions. 70

List of Tables

2.1 Comparison of Classical Valuation Methods and Automated Valuation Models (AVMs) . . . 6
2.2 Comparison of Spatial GNN Architectures . . . 13

4.1 Summary of node and edge types in the hierarchical graph. . . . 26

5.1 Overview of transaction-level features. . . . 35
5.2 Overview of neighbourhood feature categories and corresponding attributes. . . . 35
5.3 Training hyperparameters and experimental settings . . . 37

6.1 Comparison of average error metrics over the final rolling windows for each model. Values represent mean $\pm$ standard deviation of MSE and MAPE over the last N prediction windows (5-year window size). The best values per row are **in bold and underlined**. . . . 41
6.2 Model performance by property type over the last 5 prediction windows. Each subtable reports performance of the three models in terms of (a) MSE and (b) MAPE. Transaction count is included to contextualize results. The best value in each row is **in bold and underlined**. 42

B.1 Top 15 neighborhoods with the highest MAPE for MBSTGNN and LightGBM. Common neighborhoods are shaded. . . . 62
B.2 Top 15 neighborhoods with the lowest nonzero MAPE for MBSTGNN and LightGBM. Common neighborhoods are shaded. . . . 62

C.1 Overview of transaction dataset features . . . 65
C.2 Overview of Neighbourhood Dataset Features . . . 65

D.1 Comparison of average error metrics over the final rolling windows for each model. Values represent mean $\pm$ standard deviation of MSE and MAPE over the last N prediction windows (5-year window size). The best values per row are **in bold and underlined**. . . . 68

1

Introduction

This chapter introduces the motivation, background, and research objectives of the thesis. It begins by outlining the importance and complexity of residential property valuation and reviews the limitations of both traditional appraisal methods and current machine learning-based approaches. The chapter then motivates the use of graph neural networks as a promising framework for modelling the spatial and temporal dependencies inherent in real estate data. It also outlines the main contributions of this thesis. In response to the gaps present in the field, the chapter presents the research questions in Section 1.1 that guide this work and introduces the proposed hierarchical spatio-temporal graph neural network model. It concludes by providing an outline of the remaining chapters in Section 1.2.

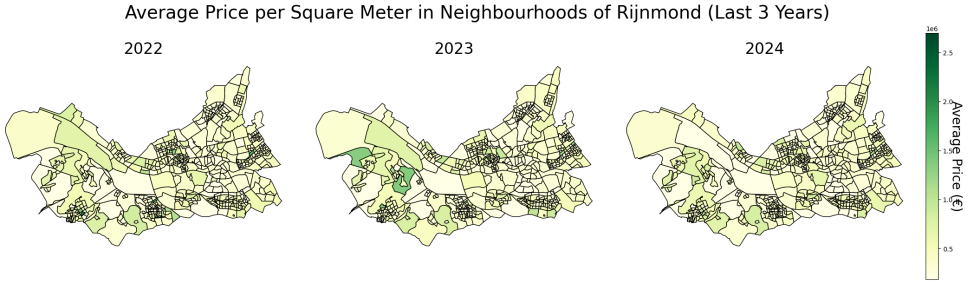


Figure 1.1: Heatmap of the average price per m^2 of the neighbourhoods of region of Rijnmond for 3 years. These plots highlight the spatio-temporal nature of real estate data.

Residential real estate constitutes a major component of both individual financial portfolios and national economic systems. Accurate valuation of residential properties is essential for enabling informed decisions across various domains, including property transactions, mortgage lending, taxation, and urban planning. Misvaluation can lead to significant consequences: overestimated prices reduce liquidity and delay sales, while underestimates lead to financial losses for sellers and tax revenue gaps for local authorities. Moreover, pricing accuracy has direct implications for housing affordability, social equity, and efficient resource allocation.

Despite its critical importance, property valuation remains a challenging task. The price of a home is influenced by a wide range of factors, including physical characteristics (e.g., size, age, amenities), market conditions, spatial context, and socio-economic trends. These factors often interact in nonlinear and location-specific ways. Traditional valuation methods such as the sales comparison, cost, or income approaches rely heavily on expert knowledge and manually selected comparable sales. While widely used in practice, these methods suffer from key limitations: they lag behind market changes, are vulnerable to human bias, and often depend on sparse or outdated transaction data [1].

To overcome these issues, the real estate sector has increasingly adopted automated valuation models (AVMs), which use data-driven algorithms to estimate property values [2]. Early AVMs were based on statistical techniques such as hedonic pricing models. More recently, machine learning methods including decision trees, support vector machines (SVMs), and artificial neural networks (ANNs) have demonstrated superior performance. These models can capture complex, non-linear relationships and scale to large datasets, offering improvements in speed, accuracy, and objectivity. However, most existing AVMs still rely on tabular feature representations and treat transactions as independent samples. As a result, they struggle to model spatial spillovers or time-dependent trends, both of which are fundamental to real estate markets.

Real estate data is inherently spatio-temporal. Property prices are not only determined by what the property is but also where and when the transaction occurs. Prices tend to exhibit strong spatial autocorrelation: Similar homes in close proximity tend to have similar values. Likewise, prices evolve over time based on local supply-demand dynamics, seasonality, and broader economic conditions. Ignoring these dependencies limits the predictive power and generalisability of AVMs.

To address this, recent research has turned to graph-based machine learning, and in particular, graph neural networks (GNNs). GNNs extend deep learning to graph-structured data, enabling the modelling of spatial relationships between entities. In the context of property valuation, nodes can represent transactions or neighbourhoods, while edges encode spatial proximity, demographic similarity, or temporal adjacency. GNNs are well-suited to capturing the complex interdependencies and relational patterns that govern real estate markets. However, current GNN-based approaches often treat the data as static and flat, failing to model the temporal dynamics or hierarchical structure that characterises urban housing systems.

In practice, residential valuation is influenced by both micro-level factors (e.g., individual property transactions) and macro-level context (e.g., neighbourhood characteristics). These two levels are not independent: aggregated neighbourhood trends emerge from individual transactions, and those trends in turn influence future valuations. Despite this, most existing models lack mechanisms for explicitly modelling this bidirectional relationship. Furthermore, temporal changes, such as gentrification, infrastructure projects, or economic shocks, create evolving patterns that must be dynamically incorporated.

This thesis addresses these challenges by proposing a novel multi-scale bidirectional spatio-temporal graph neural network (MBSTGNN) for residential property valuation. The model constructs a dynamic multi-level graph, where nodes represent transactions and neighbourhoods, and temporal edges capture evolving relationships across time. A key innovation of the proposed architecture is its bidirectional learning mechanism, which allows information to flow between micro and macro levels. This enables the model to learn localized patterns while also leveraging broader regional trends, resulting in improved generalization, especially in sparse or volatile areas.

To this end the contributions of this thesis to the field of automated valuation and graph-based spatio-temporal modelling are as follows:

- **Hierarchical Bidirectional Architecture:** We propose a novel hierarchical spatio-temporal GNN that models interactions between micro-level transactions and macro-level neighbourhoods through bidirectional message passing. This enables mutual refinement of local and regional representations, which is absent in existing models.
- **Spatio-Temporal Graph Construction:** We design a dynamic heterogeneous graph structure that encodes spatial proximity, temporal recurrence, and hierarchical organisation of real estate data.
- **Creation of a Public Property Transaction Dataset:** We generate a synthetic property transaction dataset from private data using the Private-PGM framework, ensuring differential privacy guarantees.
- **Spatial Interpretability and Case Study Analysis:** We demonstrate that the learned graph representations capture meaningful geographic and relational patterns. A case study of Rotterdam illustrates the model's ability to reflect domain-consistent spatial trends.

1.1. Research Questions

The central objective of this thesis is to develop and evaluate a graph-based model for residential property valuation that integrates spatial, temporal, and hierarchical dependencies. This objective leads to the following research question:

> **RQ:** *How can spatio-temporal and hierarchical dependencies in real estate data be effectively modelled using graph neural networks to improve automated property valuation?*

This overarching question is addressed through two sub-questions:

> **RQ1:** How does a hierarchical spatio-temporal graph neural network compare to graph-based and non-graph-based baselines in accurately predicting residential property prices across varying property types, regions, and time periods?

Accurate and robust valuation across diverse contexts is critical for the practical adoption of AVMs. This question examines the predictive benefits of the proposed approach under varying spatial, temporal, and market conditions.

> **RQ2:** To what extent does the proposed hierarchical graph structure capture meaningful spatial dependencies and domain-consistent neighbourhood relationships in real estate markets?

Beyond accuracy, valuation models must capture interpretable spatial relationships to be trusted and actionable in practice. Existing approaches rarely assess whether learned graph representations align with real-world geographic and economic patterns. This question evaluates the interpretability and spatial validity of the proposed hierarchical design.

1.2. Thesis Structure

The remainder of this thesis is structured as follows. Chapter 2 provides background context of the problem and introduces the reader to key concepts that this thesis is based on like graphs and graph neural networks. Chapter 3 reviews literature on different AVMS, like Econometric based, ML-based and GNN based AVMs, and techniques used in hierarchical graph learning. Chapter 4 presents the methodology, including the hierarchical graph formulation, model architecture, and training procedure. The experimental setup, description of the data and the baselines is provided in Chapter 5. Chapter 6 presents and discusses the results and analysis, including overall performance comparison, temporal generalisation, spatial behaviour, and case studies.Finally, Chapter 7 concludes with a summary of our findings, limitations, and directions for future work.

2

Background

This chapter outlines the foundational concepts relevant to our work. We begin with an overview of property valuation and its key challenges in Section 2.1, followed by a brief introduction to graph theory in Section 2.2. We then discuss how graphs can model real estate data (Section 2.3), enabling the use of GNNs for property representation learning. Section 2.4 introduces the concept of graph neural networks, and this is followed by some context about advanced graph representations and models in Section 2.5. Finally, we introduce the concept of gating in Section 2.6

2.1. Introduction to Property Valuation

Property valuation is the process of estimating the economic value of real estate and is fundamental to decision making in urban planning, taxation, real estate investment, and financial risk assessment. The value prediction problem involves estimating the market value which depends on the information about the property itself and its surrounding context. Market value is defined as the price that would be agreed upon between a willing buyer and a seller in an open and competitive market, under conditions of full information and no compulsion to act [3]. However, in the real world, we observe transaction prices as proxies of market values. These prices often deviate from their true value due to several structural and behavioural characteristics of the real estate market, such as liquidity constraints, heterogeneity of assets, information asymmetry between buyer and seller, negotiation dynamics, and different bidding processes [4, 5, 6, 7].

In this section, we dive deeper into the problem of property valuation and point out the challenges in this area. In Subsection 2.1.1, we provide a general mathematical view of the problem. We give an overview of the current approaches used for property valuation in Subsection 2.1.2 and elaborate on the existing challenges in this area in Subsection 2.1.3.

2.1.1. Mathematical Formulation of the Property Valuation Task

Here, we aim to provide a mathematical representation of the problem of property valuation. Let:

- $\mathbf{x}_i^{(prop)} \in \mathbb{R}^{d_1}$: vector of property-specific features for property i (e.g., size, age, number of rooms),
- $\mathbf{x}_i^{(neigh)} \in \mathbb{R}^{d_2}$: vector of spatial and neighbourhood features (e.g., proximity to amenities, zoning type),
- $t_i \in \mathbb{N}$: time of the transaction (e.g., month/year),
- $p_i \in \mathbb{R}^+$: observed transaction price,
- $v_i \in \mathbb{R}^+$: latent market value at time t_i.

The objective of property valuation is to learn a predictive function $f : \mathbb{R}^{d_1+d_2+1} \to \mathbb{R}$ such that:

$$\hat{v}_i = f(\mathbf{x}_i^{(prop)}, \mathbf{x}_i^{(neigh)}, t_i). \tag{2.1}$$

However, in practice, we observe only price p_i, which is a noisy realisation of v_i, the estimated market value, i.e.,

$$p_i = v_i + \varepsilon_i, \tag{2.2}$$

where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2(\mathbf{x}_i^{(prop)}, \mathbf{x}_i^{(neigh)}, t_i))$ captures the noise in observed prices.

2.1.2. Approaches to Property Valuation

A wide range of techniques have been developed to estimate the value of the property, ranging from traditional appraisal methodologies to modern machine learning models. These can be broadly categorised into two main families: *classical valuation approaches*, and *automated valuation machines (AVMs)*. Table 2.1 provides the main comparisons between these two approaches.

Classical Approaches

The classical approaches to property valuation do not model the problem as described in 2.1.1. Instead, they are heavily reliant on manual techniques like data analysis and assessments.

The Sales Comparison Approach estimates value by identifying and adjusting comparable past transactions. It is the most commonly used method in residential appraisal, but suffers when there are few reliable comparable transactions or high heterogeneity in housing stock. The Cost Approach estimates value by summing land value with the depreciated replacement cost of structures. It is especially applicable for new or specialised properties. The Income Capitalisation Approach, typically applied to commercial real estate, derives value based on projected net operating income and capitalisation rates or discounted cash flows [8].

Automated Valuation Models

Automated Valuation Models (AVMs) apply econometric, or machine learning techniques to large datasets of historical sales and property attributes to automate the valuation process. In this thesis, we will be focussing on this approach for property valuation for the formulation of the problem.

Table 2.1: Comparison of Classical Valuation Methods and Automated Valuation Models (AVMs)

Aspect	Classical Valuation Methods	Automated Valuation Models (AVMs)
Approach	Expert-driven, rule-based	Data-driven, model-based
Techniques Used	Sales comparison, cost approach, income capitalization	Machine learning, statistical models, hedonic pricing
Data Usage	Limited, manually selected comparables	Large-scale structured datasets
Transparency	High; based on interpretable rules	Variable; may involve black-box models (e.g., ensembles, neural nets)
Handling of Heterogeneity	Limited; struggles with nonlinearities	High; captures complex, nonlinear relationships
Scalability	Low; requires manual effort	High; supports large-scale automated predictions
Accuracy Potential	Depends on appraiser expertise and data quality	Often higher due to pattern learning from large datasets
Regulatory Acceptance	Widely accepted in formal appraisals	Growing use; regulatory standards still evolving

Traditional AVMs, based on automated statistical methods, often rely on hedonic pricing models, which assume property price is a function of observable features (e.g., area, age, room count, proximity to city centre). The seminal work by Rosen [9] formalized this model, which remains widely used for interpreting market dynamics and estimating the value of location-specific amenities. The hedonic model is typically formulated as:

$$p_i = \beta_0 + \sum_{j=1}^{d} \beta_j x_{ij} + \varepsilon_i, \tag{2.3}$$

where p_i is the price of property i, x_{ij} are its features, and β_j are coefficients estimated via regression [10]. Modern machine learning based approaches include tree-based ensembles (e.g., XGBoost, Random Forest), neural networks, and increasingly, graph-based models that incorporate relational or spatial dependencies. These models have been increasingly used to predict property values from large-scale structured datasets [11, 12]. The main model that we will be comparing our approach to is the LightGBM [13] model, a highly optimized implementation of Gradient Boosted Decision Trees(GBDT)[14], which is also used by Ortec Finance for property valuation. Following this we elaborate on the exact mechanisms of this model.

LightGBM

GBDT models operate by sequentially building an ensemble of M decision trees $\{f_m(\cdot)\}_{m=1}^{M}$, where each tree f_m attempts to correct the residual errors of the combined ensemble of previous trees. The overall prediction $\hat{y}_i$ for a sample i is given by:

$$\hat{y}_i = \sum_{m=1}^{M} \eta \, f_m(\mathbf{x}_i), \quad f_m \in \mathcal{F} \tag{2.4}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector for the i-th sample, η is the learning rate controlling shrinkage,and $\mathcal{F}$ is the space of regression trees.

The model is trained to minimise a regularised objective:

$$\mathcal{L} = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) + \sum_{m=1}^{M} \Omega(f_m), \tag{2.5}$$

where ℓ is any differentiable loss function (typically squared error for regression), and $\Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2$ penalises the complexity of each tree (with T being the number of leaves and **w** the leaf weights).

LightGBM improves on traditional GBDT by using: leaf-wise tree growth strategy (as opposed to level-wise), which selects the leaf with the highest potential gain to split. It also uses a histogram-based approximation of continuous features to reduce memory and computation. Gradient-based One-Side Sampling and Exclusive Feature Bundling were integrated for speeding up training with minimal accuracy loss.

2.1.3. Challenges faced by AVMs

Despite significant advances in modelling, property valuation remains a complex and challenging task due to several factors associated with data limitations, economic dynamics, and modelling assumptions.

Data Heterogeneity

Real estate valuation is fundamentally complicated by the intrinsic heterogeneity of individual properties. Unlike standardised financial assets, each real estate asset possesses unique attributes that influence its market value. Properties differ in structure, aesthetics, legal structure, and micro-location features like sunlight, slope or proximity to green space, all of which can significantly affect valuation. These differences are often unobservable, inconsistently recorded, or missing in standard datasets [15, 16, 17]. This data heterogeneity, along with varying data quality, presents major challenges for AVMs, leading to potential bias and reduced model robustness [18, 19].

Spatial Dependence and Non-Stationarity

Location is arguably the most important determinant of property value. The concept of spatial dependency, which is also the basis of the comparable sales method, implies that the value of a property is not independent but influenced by the characteristics and values of nearby properties. This is often formalized through Tobler's First Law of Geography: "everything is related to everything else, but near things are more related than distant things" [20]. The spatial autocorrelation of property prices has been well documented in urban economics and geography [21]. Factors such as neighbourhood safety, school quality, accessibility to public transport, and environmental quality are all inherently spatial and affect how clusters of property values evolve. These relationships are not globally uniform; urban and suburban dynamics vary significantly, which violates the i.i.d. assumptions of classical models [17]. This spatial non-stationarity complicates predictive modelling.

Temporal Dynamics and Market Volatility

Property valuation is also highly sensitive to temporal factors. Real estate markets are dynamic systems, influenced by seasonal patterns, economic cycles, demographic shifts, and local development policies. For instance, housing prices in many urban centres exhibit strong month-over-month and year-over-year trends due to fiscal policy changes, mortgage interest rates, and regional investments [22]. In addition to long-term trends, short-term temporal volatility may arise from factors like temporary construction disruptions, infrastructural upgrades, or pandemic-related effects. These factors imply that a property's value at time t cannot be fully explained without considering its historical context and its neighbourhood's evolution over time. Temporal trends can also vary spatially: some neighbourhoods gentrify rapidly, while others stagnate or decline. Thus, modelling property value dynamics requires approaches that can integrate both temporal dependencies and spatial heterogeneity — a challenge not addressed by classical models, and only partially captured by time-series or panel regression frameworks [23, 24].

Interpretability and Regulatory Compliance

In valuation settings such as tax assessment, lending, or litigation, models must be explainable and legally defensible. While machine learning models can offer superior predictive performance, their lack of interpretability often leads to concerns about fairness, transparency, and accountability [25].

2.2. Graphs

Graphs are fundamental structures used to represent relationships between entities. They are widely applied across various domains, from social networks to transportation systems, as they offer a natural way to model interactions and dependencies.

2.2.1. Basic Definition

A (static) graph, defined as $G = (V, E)$, is a representation of a set of vertices (or nodes) $V = \{v_1, v_2, ..., v_n\}$ and a set of edges $E \subseteq V \times V$ that connect pairs of vertices. An edge connecting any two nodes i and j is defined as $(i, j) \in E$ and represents a relationship or interaction between the two nodes. Nodes that are connected by an edge are called neighbours. Edges can either be directed (i.e., E is an ordered pair of nodes) or undirected (i.e., E is an unordered pair of nodes). The edges may also carry weights to represent the strength or capacity of the interaction between nodes.

A graph can be represented in multiple ways, such as an adjacency matrix A, an edge list, or a set of neighbours for each vertex. The adjacency matrix A is a square matrix where the element A_{ij} represents the presence (or weight) of an edge between nodes v_i and v_j. For undirected graphs, the adjacency matrix is symmetric, i.e., $A_{ij} = A_{ji}$.

Graphs can also be classified based on certain properties. Figure 2.1 represents the following comparisons.

- **Directed vs. Undirected**: A directed graph (digraph) has edges with a direction, while an undirected graph has edges without direction.
- **Weighted vs. Unweighted**: A weighted graph assigns a numerical value to each edge, while an unweighted graph simply indicates the presence or absence of an edge.
- **Connected vs. Disconnected**: A connected graph allows for a path, which is a sequence of edges connecting 2 vertices, between any pair of vertices, while a disconnected graph consists of two or more disconnected components.

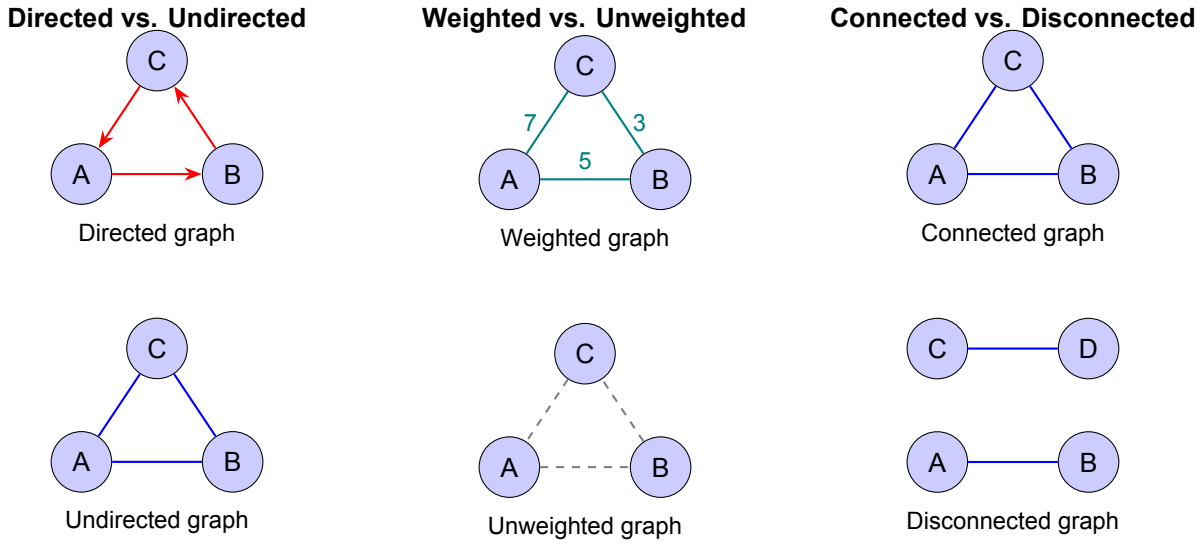


Figure 2.1: Comparison of different types of graphs. Each subfigure shows two graph types stacked vertically.

2.2.2. Graph Signals and GSOs

In many applications, data is naturally associated with the nodes of a graph. This type of data is referred to as a *graph signal*. A graph signal is a function $f : V \to \mathbb{R}^d$ that assigns a value (or a vector of values) to each node in the graph. For example, in a transportation network, a graph signal might represent traffic flow at each intersection; in a sensor network, it might represent temperature or pollution levels at different locations, the stock prices at different times in a financial network, or user behaviours in social networks.

Graph-based methods often exploit the structure of the graph to analyse and process these signals.

A central tool in this process is the *Graph Shift Operator (GSO)*. A GSO is a matrix that defines how information propagates across the graph. Common choices for the GSO include the adjacency matrix A, the degree-normalized adjacency matrix $D^{-1}A$, or the graph Laplacian $L = D - A$, where D is the degree matrix. A degree matrix D of a graph is a diagonal matrix where the entries on the diagonal represent the degree of each node. In the case of a directed graph, the degree of a node is the number of outgoing edges; hence $D_{ii} = \sum_{j=1}^{|V|} A_{ij}$. For a directed graph, the degree of a node is defined as the number of edges that have the node as an endpoint. The adjacency matrix is simple and intuitive but may lead to biased propagation in graphs with nodes of varying degree [26]. The degree-normalized adjacency mitigates this bias by scaling contributions from neighbours, improving stability in learning [27]. The Laplacian, while useful for spectral methods and ensuring symmetry, may suppress signals in sparse graphs and is less intuitive for directional message passing [28, 29].

Applying a GSO to a graph signal allows information at each node to be influenced by the values of its neighbours. This process is analogous to convolution in classical signal processing, where the signal is smoothed or filtered by aggregating values from a local neighbourhood. For example, multiplying the adjacency matrix A with a graph signal **x** produces a new signal A**x**, where each node's value is a weighted sum of the values of its neighbours.

2.2.3. Dynamic Graphs

While most methods in graph learning are based on static graph structures, many real-world networks, such as biological, financial, and social networks, exhibit temporal dynamics where the graph evolves over time. These evolving graphs are referred to as *dynamic graphs*, capturing structural changes such as the addition or deletion of nodes and edges. Dynamic graphs can be broadly categorised based on their temporal representation into two types as shown in Figure 2.2: Discrete-Time Dynamic Graphs (DTDGs) and Continuous-Time Dynamic Graphs (CTDGs) [30, 31].

Discrete-Time Dynamic Graphs (DTDGs) represent temporal evolution as a sequence of graph snapshots sampled at fixed time intervals. Formally, a DTDG is defined as:

$$G = \{G^0, G^1, \ldots, G^T\}, \tag{2.6}$$

where each snapshot $G^t = (V^t, E^t)$ corresponds to the graph structure at discrete time step t, for $0 \leq t \leq T$. This formulation allows static graph learning methods to be applied independently at each time step. However, the primary limitation is the potential loss of temporal granularity between snapshots. As a result, determining the optimal frequency of intervals becomes a trade-off between model accuracy and computational efficiency [31].

Continuous-Time Dynamic Graphs (CTDGs) provide a more fine-grained representation of temporal dynamics by modelling graph evolution as a sequence of timestamped events. A CTDG is typically represented as a tuple (G, O), where G is the initial static graph, and $O = \{o_1, o_2, \ldots, o_N\}$ is a set of temporal observation that describe changes to the graph structure over time.

Each observation $o_i \in O$ is formally represented as a triplet:

$$o_i = (\textit{event}_i, \textit{type}_i, \textit{timestamp}_i), \tag{2.7}$$

where:

- *event*$_i$ refers to the node, edge, or subgraph involved in the event,
- *type*$_i$ specifies the nature of the structural change (e.g., `add_node`, `remove_edge`, `merge_nodes`),
- *timestamp*$_i$ records the exact time at which the event occurred.

To reconstruct the graph state at any given time t, all events with timestamps up to t are applied sequentially to the initial graph G. This continuous representation captures asynchronous and irregular structural changes more precisely than DTDGs [30].

Similar as static graphs, the following can be defined for dynamic graphs.

- **Adjacency Matrix:** At time t, the adjacency matrix $A^t \in \mathbb{R}^{|V^t| \times |V^t|}$ is defined as:

$$A_{ij}^t = \begin{cases} 1 & \text{if } (v_i, v_j) \in E^t, \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

If the graph is weighted, 1 is replaced with the edge weight.

- **Laplacian:**

$$L^t = D^t - A^t, \tag{2.9}$$

where D^t is the degree matrix with entries $D^t_{ii} = \sum_j A^t_{ij}$.

- **Normalised Laplacian:**

$$\mathcal{L}^t = I - (D^t)^{-1/2} A^t (D^t)^{-1/2}. \tag{2.10}$$

- **Node Features:** Node feature matrices at each time step are denoted $X^t \in \mathbb{R}^{|V^t| \times d}$, forming a temporal sequence $\{X^0, X^1, ..., X^T\}$.

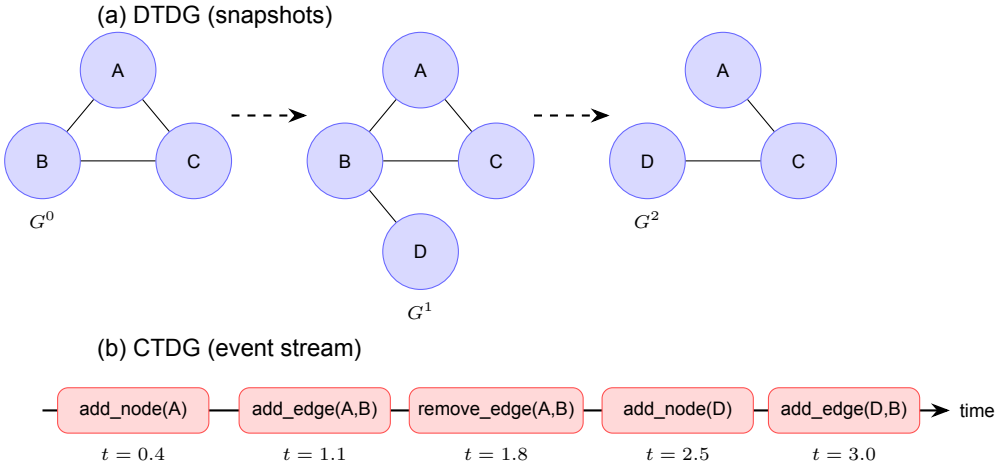


Figure 2.2: Illustration of two types of dynamic graphs. (a) A *Discrete-Time Dynamic Graph* (DTDG) is represented as a sequence of graph snapshots G^0, G^1, G^2 at fixed time intervals. (b) A *Continuous-Time Dynamic Graph* (CTDG) records a time-ordered stream of structural events, each with a precise timestamp.

Spatio-Temporal Graphs

Spatio-Temporal graphs are a specific type of dynamic graphs. The nodes and their spatial connections of these types of graphs do not change with time; however, their importance or weight may vary. At time t, a spatio-temporal graph $G_t = (V, E, A_t, X_t)$ comprises a set of vertices $V = \{v_1, v_2, ..., v_{|V|}\}$ that represent objects that have a certain spatial context. The set of edges E represents the interactions between these vertex pairs. A_t is the (weighted) adjacency matrix of the edges at time t and X_t is the matrix of node features at that time step. Spatio-temporal graphs are used to represent the variation of events or objects with time along with their spatial relationship. They are widely used in areas such as traffic flow prediction and epidemic modelling [32].

2.3. Graphs in Real Estate

Real estate data is inherently relational and spatial, making graphs a natural fit for modelling and learning from such data. Traditional tabular approaches often ignore spatial dependencies and contextual factors, leading to suboptimal performance in tasks like price prediction, and market trend analysis. Graph-based representations offer a way to explicitly capture these dependencies for example by modelling properties, amenities, and geographic entities as nodes, and their spatial or functional relationships as edges [33].

Constructing graphs from real estate data typically involves defining the set of entities (nodes) and the types of relationships (edges) relevant to the problem. Common strategies include:

- **Spatial Proximity Graphs**: One of the most common graph types used in real estate models, where properties (or transactions) are connected based on their physical closeness. This is typically implemented using k-nearest neighbours (k-NN), radius-based thresholds, or Delaunay triangulations. These graphs model local spatial spillovers, such as price diffusion between neighbouring houses[34].

- **Functional or Semantic Graphs**: Nodes may include other urban elements such as schools, public transport stations, or shopping centres, with edges representing access or influence (e.g., within walking distance) greatly expanding semantic context [35]. These graphs support heterogeneous and multi-modal modelling.
- **Temporal Graphs**: When real estate data includes time-series features (e.g., historical prices or transaction timelines), dynamic or temporal graphs can be used, where snapshots or edges evolve over time.
- **Administrative or Hierarchical Graphs:** These graphs layer multiple geographic levels such as homes, postal codes, and municipalities enabling hierarchical context. [36].

Different graph formulations embed specific inductive biases that can help capture spatial spillovers, infrastructure dependencies, or temporal trends, depending on the real estate task, such as price prediction, market segmentation, or liveability analysis.

2.4. Graph Neural Networks

Graph Neural Networks (GNNs) are a powerful class of models designed to operate directly on graph-structured data. Unlike traditional machine learning models, which typically assume Euclidean input structures such as sequences or grids, GNNs are explicitly designed to exploit the relational inductive bias inherent in graphs. GNNs process both graph signals and structure by iteratively aggregating and transforming information from local neighbourhoods, enabling the learning of embeddings that represent nodes, edges, or entire graphs in a way that reflects both feature content and topological context [37].

2.4.1. Overview and Intuition

The core idea of GNNs is that a node's representation should be informed by the representations of its neighbours. In many domains, including social networks, molecular chemistry, and property valuation, this locality-based reasoning is natural: a house's value, for example, is influenced by the area it is located in, and thereby its nearby properties. GNNs formalise this intuition using neural architectures that repeatedly aggregate information from neighbouring nodes.

Each GNN layer updates node features by aggregating and transforming the features of its neighbours. Stacking multiple layers allows the model to capture broader structural patterns in the graph by increasing the receptive field of each node [38].

2.4.2. Message Passing Framework

The message passing paradigm provides a general framework for most GNN architectures. In this framework, the forward pass of a GNN layer consists of two steps: message aggregation and node state update. Figure 2.3 depicts this process.

Formally, for a node $v \in V$ and its neighbourhood $\mathcal{N}(v)$, the update at layer $l+1$ is:

$$\mathbf{h}_v^{(l+1)} = \text{UPDATE}^{(l)}\left(\mathbf{h}_v^{(l)}, \text{AGGREGATE}^{(l)}\left(\left\{\mathbf{h}_u^{(l)} : u \in \mathcal{N}(v)\right\}\right)\right), \tag{2.11}$$

where:

- $\mathbf{h}_v^{(l)} \in \mathbb{R}^d$ is the feature vector of node v at layer l,
- AGGREGATE is a permutation-invariant function (e.g., sum, mean, or max),
- UPDATE typically applies a neural network (often a linear layer with non-linear activation function).

After several message passing layers, a final step known as the readout function is often applied to generate task-specific outputs. This operation aggregates node (or edge) representations into a single vector using techniques such as summation, averaging, or more complex pooling methods.

2.4.3. Graph Convolution as Neighbourhood Aggregation

Graph convolutional layers are a particular instantiation of the message passing framework where aggregation is performed via weighted averaging. For a graph with adjacency matrix A and node feature matrix $X \in \mathbb{R}^{n \times d}$, one common layer formulation is:

$$H = \sigma\left(\hat{A}XW\right), \tag{2.12}$$

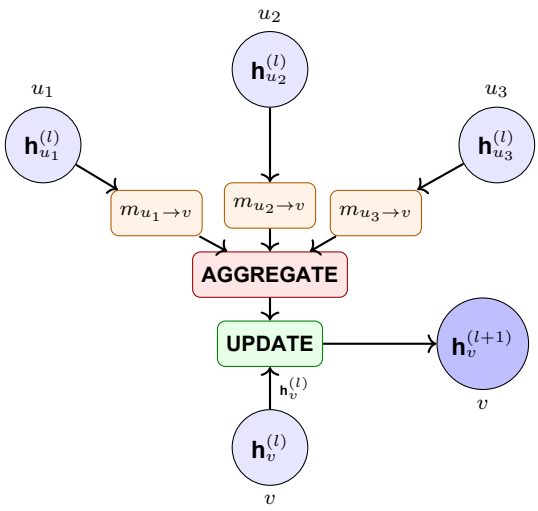


Figure 2.3: Detailed illustration of the message passing process in GNNs. Each neighbour $u_i \in \mathcal{N}(v)$ sends a message $m_{u_i \to v}$ to v, which are aggregated (e.g., sum/mean/max), and then combined with v's current state via a neural update function to yield $\mathbf{h}_v^{(l+1)}$.

where:

- $\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ is the symmetrically normalized adjacency matrix with added self-loops,
- $\tilde{A} = A + I$, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$,
- $W \in \mathbb{R}^{d \times d'}$ is a trainable weight matrix,
- σ is an activation function such as ReLU,
- $H \in \mathbb{R}^{n \times d'}$ is the matrix of output node representations.

This operation mixes features from a node's immediate neighbours (and itself), allowing information to propagate through the graph as layers are stacked, as depicted in Figure 2.4. Graph convolutions form the basis of the **Graph Convolutional Network (GCN)**, which can be considered as a foundational spatial GNN.

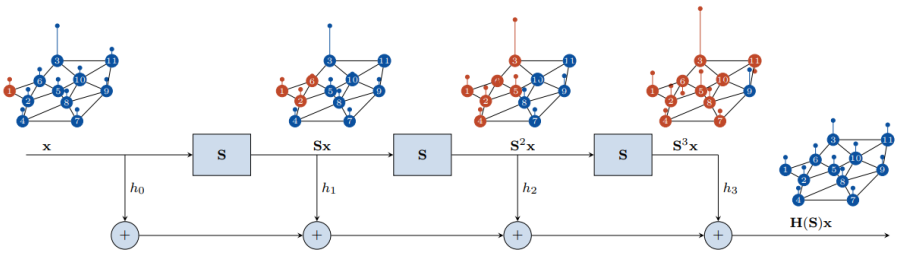


Figure 2.4: Stacked graph convolution operations adapted from [39, Figure 2]

2.4.4. Spatial GNN Architectures

Spatial GNNs define convolutions directly in the node domain based on graph connectivity rather than spectral properties. These models are particularly well-suited for large, dynamic, or inductive learning tasks, and are therefore commonly used in real-world applications.

Two widely used extensions of the spatial convolutions defined in GCNs are as follows:

GraphSAGE [27] generalises neighbourhood aggregation by supporting a variety of aggregation functions. Its layer-wise update is:

$$\mathbf{h}_v^{(l+1)} = \sigma\left(W^{(l)} \cdot \text{AGGREGATE}\left(\{\mathbf{h}_u^{(l)} : u \in \mathcal{N}(v) \cup \{v\}\}\right)\right). \tag{2.13}$$

This approach is efficient and scalable to large graphs and supports inductive learning, but uniform weighting may underperform in graphs with complex neighbour dynamics.

Graph Attention Networks (GATs) [40] assign attention weights to each neighbour, allowing the model to learn the relative importance of neighbours:

$$\mathbf{h}_v^{(l+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \alpha_{vu}^{(l)} W^{(l)} \mathbf{h}_u^{(l)}\right), \tag{2.14}$$

where the attention coefficients $\alpha_{vu}^{(l)}$ are learned using a shared attention mechanism. While this captures asymmetric and context-specific importance among neighbours, it is more computationally expensive due to attention computation.

Table 2.2 summarises the main differences between these highlighted spatial GNNs.

Table 2.2: Comparison of Spatial GNN Architectures

Model	Aggregation Method	Neighbour Weighting	Use Cases and Remarks
GCN [26]	Mean aggregation via normalized adjacency matrix	Fixed weights (based on node degree)	Efficient for transductive tasks on static graphs; less expressive for dynamic or large-scale settings
GraphSAGE [27]	Flexible: mean, LSTM, or pooling-based	Uniform or learned via aggregator function	Scalable and suitable for inductive learning; may underperform in capturing fine-grained neighbour importance
GAT [40]	Weighted sum using attention scores	Learned attention coefficients	Models variable neighbour importance; interpretable but computationally more expensive

2.4.5. GNNs for Regression Tasks

While GNNs have many applications, we will be focussing on node regression tasks where the goal is to predict a continuous value associated with nodes.

For node-level regression, the GNN learns a mapping from graph-structured inputs to continuous outputs:

$$\hat{y}_v = f_\theta(G, X)_v, \tag{2.15}$$

where $\hat{y}_v \in \mathbb{R}$ is the predicted value for node v, such as property price or demand.

The loss function used is typically Mean Squared Error (MSE):

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2. \tag{2.16}$$

In the context of property valuation, GNNs are well-suited to capture the relational dependencies among spatially or semantically connected assets, allowing predictions to leverage information from both the features of a given property and its neighbours.

2.5. Advanced Graph Representations and Models

Graph-structured data in real-world scenarios often exhibit intricate patterns that go beyond the capabilities of standard, flat GNNs. Conventional GNNs typically assume homogeneous node and edge types, operating on a single-level graph structure. However, many practical domains, including social networks, biological systems, and geospatial data like real estate, demand more expressive models that can accommodate hierarchical organization and relational heterogeneity.

To address these complexities, advanced extensions of the GNN paradigm have been proposed. This section introduces two foundational directions in this space: hierarchical graph models, which enable multi-resolution learning over structured data, and heterogeneous graph representations, which incorporate semantic diversity among nodes and edges.

2.5.1. Hierarchical Graph Structures

A hierarchical graph represents data across multiple levels of abstraction or granularity. This structure arises naturally in many domains where entities are embedded within larger organizational structures. For instance, individual properties may belong to neighbourhoods, which aggregate into broader administrative zones. This *multi-level composition* forms a natural hierarchy.

Formally, a hierarchical graph can be modelled as a sequence of graphs:

$$\mathcal{G} = \{G^{(0)}, G^{(1)}, ..., G^{(L)}\}, \tag{2.17}$$

where each graph $G^{(l)} = (V^{(l)}, E^{(l)})$ corresponds to a coarser abstraction than its predecessor. Hierarchies may be constructed based on spatial containment, functional grouping, or topological properties.

Hierarchical Graph Neural Networks (HGNNs) address this by introducing pooling mechanisms that compress the graph structure while retaining essential information [41]. These models operate analogously to convolutional neural networks (CNNs) with pooling layers for downsampling.

Graph pooling involves aggregating nodes or substructures in a lower-level graph to form coarser representations [42]. This is typically done through node selection or clustering based on topological or feature similarity:

$$\mathbf{X}^{(l+1)} = \text{POOL}(\mathbf{X}^{(l)}, A^{(l)}), \tag{2.18}$$

where $\mathbf{X}^{(l)}$ and $A^{(l)}$ are the node features and adjacency matrix at level l,respectively, and `POOL` denotes a general pooling operation that reduces the graph's resolution while preserving key information.

Hierarchical structures are inherently resolution-aware. They enable the model to integrate information at both fine and coarse resolutions, allowing it to recognise patterns that may only be visible at specific levels of granularity. These principles allow hierarchical graphs to serve as powerful abstractions in domains where local and regional interactions jointly determine system behaviour.

2.5.2. Heterogeneous Graph Representations

Unlike homogeneous graphs, which consist of a single node and edge type, *heterogeneous graphs* contain diverse entity types and relational semantics. These structures are particularly relevant in domains such as real estate, where properties, schools, transportation nodes, and amenities all interact in a semantically rich network. A heterogeneous graph is defined as:

$$G = (V, E, \phi, \psi), \tag{2.19}$$

where $\phi : V \to \mathcal{T}_V$ maps nodes to their types and $\psi : E \to \mathcal{T}_E$ maps edges to their relation types. $\mathcal{T}_V$ and $\mathcal{T}_E$ denote the sets of node and edge types, respectively [43].

Modelling such graphs requires more nuanced aggregation schemes that consider types and relations. One common approach is relational message-passing, where the update rule distinguishes messages by edge type $r \in \mathcal{T}_E$:

$$\mathbf{h}_v^{(l+1)} = \sigma\left(\sum_{r \in \mathcal{T}_E} \sum_{u \in \mathcal{N}_r(v)} W_r^{(l)} \mathbf{h}_u^{(l)}\right), \tag{2.20}$$

where $W_r^{(l)}$ is a type-specific transformation matrix and $\mathcal{N}_r(v)$ denotes the neighbours of node v under the relation r [44].

Conceptually, heterogeneous graphs incorporate three key aspects. First, *type semantics* means that each node and edge type contributes unique information, exhibiting different attributes, patterns, and behaviours [45]. Second, *typed neighbourhoods* recognise that a node's neighbours can be grouped according to relationship types, producing multiple relation-specific local contexts. Third, *relational diversity* reflects that different edge types, such as spatial proximity, service coverage, or accessibility, carry distinct meanings and must be handled differently in graph-based computations [44]. Due to this support of multiple semantics and entities, heterogenous graphs allow for a better understanding of complex systems

2.6. Gating Mechanisms

Gating mechanisms are widely used in neural networks to control the flow of information between layers or across time. They introduce adaptive, element-wise weighting that allows a model to decide how much of each input signal to retain or suppress. This is especially important when combining multiple sources of information, such as temporal history, spatial context, or features at different abstraction levels.

2.6.1. Gating in Recurrent Neural Networks

The concept of gating originated in Recurrent Neural Networks (RNNs), where it was developed to address the problem of vanishing gradients in sequence modelling tasks. Two widely used gated RNN architectures are the Long Short-Term Memory (LSTM) network [46] and the Gated Recurrent Unit (GRU) [47].

In both models, gating mechanisms are used to regulate how much of the past hidden state is carried forward and how much new information is incorporated at each time step. These gates are learned functions that control:

- **Retention**: How much of the previous hidden state to keep.
- **Update**: How much new input information to incorporate.
- **Forget (LSTM only)**: How much of the previous state to forget entirely.

For example, the GRU update is computed as:

$$\mathbf{z}_t = \sigma(W_z[\mathbf{x}_t \| \mathbf{h}_{t-1}]) \tag{2.21}$$

$$\mathbf{r}_t = \sigma(W_r[\mathbf{x}_t \| \mathbf{h}_{t-1}]) \tag{2.22}$$

$$\tilde{\mathbf{h}}_t = \tanh(W_h[\mathbf{x}_t \| (\mathbf{r}_t \odot \mathbf{h}_{t-1})]) \tag{2.23}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \tag{2.24}$$

Here, $\mathbf{x}_t$ denotes the input vector at time t, and $\mathbf{h}_{t-1}$ is the hidden state from the previous time step. The operator $\|$ represents vector concatenation, while $\odot$ denotes element-wise multiplication. W_z, W_r, and W_h are learnable weight matrices associated with the update gate, reset gate, and candidate hidden state computation, respectively. The function $\sigma(\cdot)$ is the logistic sigmoid activation, and $\tanh(\cdot)$ is the hyperbolic tangent activation. $\mathbf{z}_t$ is the update gate controlling how much of the previous hidden state is retained, $\mathbf{r}_t$ is the reset gate determining how to combine new input with past information, $\tilde{\mathbf{h}}_t$ is the candidate hidden state, and $\mathbf{h}_t$ is the final hidden state at time t.

2.6.2. Basic Gating Mechanism:

Inspired by these mechanisms, more general gating strategies have been adopted in a variety of deep learning models to combine two information sources—whether across time, spatial regions, feature hierarchies, or modalities. A simplified, generic gating formulation takes two input vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ and computes a fused representation:

$$\mathbf{g} = \sigma(W_g[\mathbf{a} \| \mathbf{b}]) \in [0, 1]^d \tag{2.25}$$

$$\mathbf{h} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{b} \tag{2.26}$$

Here, σ denotes the sigmoid activation function, $[\mathbf{a} \| \mathbf{b}]$ represents the concatenation of vectors $\mathbf{a}$ and $\mathbf{b}$, and $\odot$ indicates element-wise multiplication. The matrix $W_g \in \mathbb{R}^{2d \times d}$ is a learnable weight matrix that parameterises the gating operation. This gating mechanism enables the model to adaptively interpolate between the two input sources at each feature dimension.

2.6.3. Gating in Graph Neural Networks.

In GNNs, gating mechanisms extend naturally to modulate how node embeddings are updated during message passing. Specifically, a node's representation can be updated by combining its current embedding with an aggregated message from its neighbours using a learned gate. The update at layer $l+1$ for node v is computed as:

$$\mathbf{m}_v^{(l)} = \text{AGGREGATE}(\{\mathbf{h}_u^{(l)} : u \in \mathcal{N}(v)\}) \tag{2.27}$$

$$\mathbf{g}_v^{(l)} = \sigma(W_g[\mathbf{h}_v^{(l)} \| \mathbf{m}_v^{(l)}]) \tag{2.28}$$

$$\mathbf{h}_v^{(l+1)} = \mathbf{g}_v^{(l)} \odot \mathbf{h}_v^{(l)} + (1 - \mathbf{g}_v^{(l)}) \odot \mathbf{m}_v^{(l)} \tag{2.29}$$

This formulation enables the model to learn how much of the node's prior state $\mathbf{h}_v^{(l)}$ to retain versus how much of the newly aggregated message $\mathbf{m}_v^{(l)}$ to incorporate. This mechanism has been used in various models to improve flexibility, temporal awareness, and feature selectivity in graph domains. Some common use cases include:

Temporal Gating: When graph snapshots evolve over time, a gate can be used to merge current node embeddings with embeddings from past time steps. This is conceptually similar to GRUs [47] or LSTMs [46], but adapted to graph contexts without recurrent unrolling.

Multi-scale or Hierarchical Fusion: In hierarchical GNNs, node representations at different levels (e.g., fine-grained and coarse-grained) can be combined using gating to preserve both local detail and global context.

Multi-source or Multi-view Fusion: When a node receives messages from different neighbourhoods or subgraphs (e.g., spatial neighbours vs. semantic neighbours), gating enables the model to adaptively weigh each source's importance.

3

Literature Review

In this chapter, we explore the existing literature in the field of automated property valuation techniques and also existing hierarchical graph learning methods. Figure 3.1 provides an overview of the literature review conducted for this thesis. In Section 3.1, we dive into the econometric approaches and in Section 3.2 we give an overview of non-graphical ML based methods. We highlight studies that use graph-based methods for property valuation in Section 3.3. We then move onto a review of hierarchical graph learning methods in Section 3.4. Finally, we conclude the chapter by listing the research gaps identified in this literature review in Section 3.5

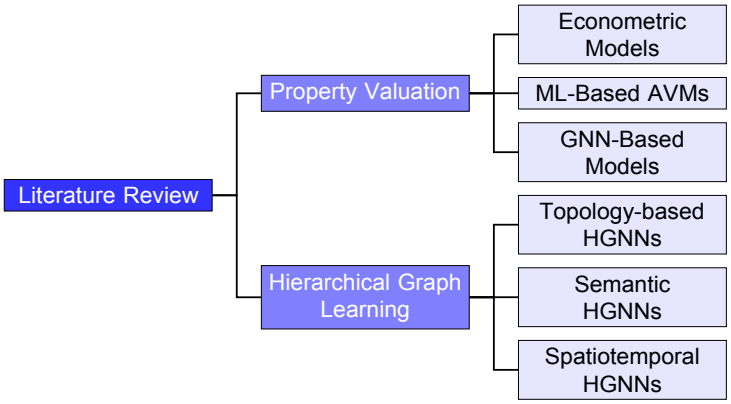


Figure 3.1: Classification of literature in property valuation and hierarchical graph learning

3.1. Econometric Approaches to Property Valuation

Property valuation methods have predominantly relied on econometric models, which are primarily built on hedonic pricing theory. As described in Equation 2.3, these models decompose property prices into the implicit value of their constituent attributes such as size, location, accessibility. [9]. The basic hedonic pricing model rest on the assumption of spatial independence, temporal stationarity, and linearity, which are often violated in real estate data.

To account for spatial autocorrelation in property prices, extensions to the hedonic pricing framework incorporate spatial effects. Liu et al. (2014) proposed a spatial autoregressive (SAR) specification that captures both direct and indirect (spillover) effects, demonstrating that ignoring spatial dependence leads to biased coefficients and underestimated standard errors [48]. Closely related models include the Spatial Lag Model (SLM), where spatial dependence is introduced in the dependent variable, and the Spatial Error Model (SEM), which accounts for spatially correlated error terms. These approaches rely on a spatial weights matrix to define proximity-based relationships between properties. However, the validity of results is sensitive to how this matrix is specified, and mis-specification can be particularly problematic in heterogeneous urban settings or in panel data contexts where spatial and temporal dependence interact.

Temporal variation in property markets has been addressed with models such as the repeat sales method, which analyses price changes for the same property over time to isolate market-wide trends. Introduced by Bailey, Muth, and Nourse [49], this approach assumes that a property's attributes remain constant between sales, allowing the estimation of price indices that reflect pure market movements. Its limitations include restricted applicability where repeat sales data are scarce, vulnerability to sample selection bias, and potential inaccuracy when properties undergo significant alterations between transactions.

More advanced spatio-temporal models have also been developed. Geographically and Temporally Weighted Regression (GTWR) extends Geographically Weighted Regression (GWR) by allowing coefficients to vary across both space and time. Huang, Wu, and Barry [50] show that GTWR can reveal neighbourhood-specific dynamics and seasonal patterns that global models overlook. However, it depends on kernel functions and bandwidth selection, and its performance can degrade in sparse space-time datasets due to instability in local parameter estimation.

The Hierarchical Trend Model (HTM) proposed by Francke and Vos [10] introduces a multi-level structure for estimating local price indices. Trends are modelled at different spatial levels (e.g., region, neighbourhood, street), capturing both broad market movements and localised variation. Temporal trends are incorporated at each level, making HTMs suitable for granular, policy-relevant valuation tasks. While the original formulation assumes linear trends within each level, extensions could accommodate non-linear patterns. These models are statistically rigorous and interpretable, but they can be computationally intensive for large datasets. Furthermore, HTMs treat units at the same level of the hierarchy (e.g., neighbourhoods within a region) as conditionally independent given higher-level trends, meaning they do not explicitly capture cross-neighbourhood dependencies such as spillover effects, shared amenities, or boundary-crossing market influences.

Despite their interpretability and statistical rigour, traditional econometric approaches face several limitations in modern, data-rich valuation contexts. They rely on strong assumptions regarding functional form, spatial and temporal dependence, and often cannot capture complex or evolving relationships across properties or neighbourhoods. Hierarchical and spatio-temporal models improve flexibility but still struggle with overlapping influences, abrupt market changes, and high-dimensional or unstructured data. These constraints motivate the use of machine learning and graph-based methods, which can model non-linear interactions, adaptively learn spatial and temporal dependencies, and integrate diverse data sources.

3.2. Machine Learning-Based Automated Valuation Models

In recent years, ML has emerged as a powerful alternative to traditional econometric models in real estate valuation, driven by its ability to capture non-linear relationships, complex feature interactions, and to integrate diverse data sources such as images, textual descriptions, and spatial attributes. These data-driven techniques are increasingly used to overcome the limitations of parametric models, offering more accurate and adaptive valuation tools.

Algorithms like Random Forests (RF) and Gradient Boosted Decision Trees (GBDTs) have been widely adopted due to their robustness and high accuracy in capturing hierarchical, non-linear relationships between property features and price. Sing et al. showed that boosted tree ensembles offer higher valuation accuracy in Singapore's vertically dense housing market by capturing nonlinear spatial interactions [51]. Jafary et al. applied GBDTs at scale across suburban Melbourne and highlighted their robustness in handling missing spatial features and integrating deep learning components [52]. Chen et al. used GBDTs on multi-source urban data in Shanghai, finding that these models substantially outperformed linear baselines when accounting for complex urban form and structural factors [53]. LightGBM, as described in 2.1.2, has been widely used in the industry for this task [54, 55]. Since this is also a method adopted by Ortec Finance, it will serve as one of our baselines for comparison.

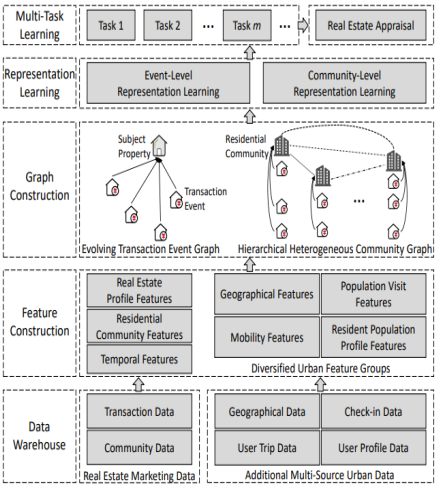
Support Vector Machines (SVMs) have also been explored, primarily for their ability to handle high-dimensional feature spaces via kernel methods. They are suitable for datasets where collinearity among variables or non-linear price dynamics challenge traditional regression approaches. Studies in regions such as Jordan and Morocco have reported competitive performance, particularly in settings with strong local price gradients or limited training data [56, 57].Similarly, Balakrishnan et al. implemented SVMs in a property appraisal system and found that SVMs outperformed neural networks in low-data regions due to its capacity to generalize from limited samples [58]. However, SVMs have seen limited adoption in large-scale, real-world AVMs due to scalability constraints and the availability of more efficient alternatives.

ANNs learn complex mappings between input features, such as property size, age, location, and amenities, and market prices through multi-layered architectures and iterative training. For instance, Ibrahim et al. reviewed ANN-based Computer-Assisted Mass Appraisal (CAMA) systems in the Nigerian real estate sector, highlighting their superior adaptability in handling large, noisy datasets where traditional valuation models underperform [59]. Similarly, Khan et al. compared ANNs with tree-based models and found that deep neural networks achieved higher accuracy in predicting real estate prices, especially in datasets with substantial feature interactions and urban heterogeneity [60]. ANNs have also been combined with hedonic pricing theory in hybrid frameworks. Nor et al. applied a neural network alongside a hedonic model to estimate rental prices in climate-affected areas of Somalia, demonstrating that the ANN component significantly improved prediction accuracy by capturing non-linearities missed by traditional regressions [61].

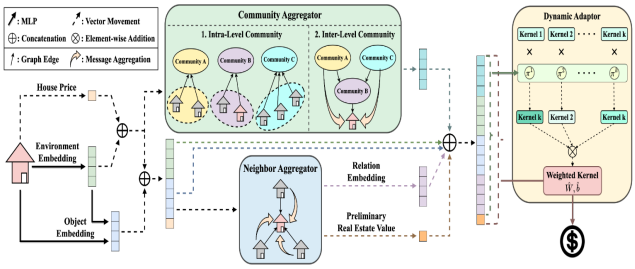
While machine learning models outperform traditional regressions in predictive accuracy due to their flexibility and ability to model complex, non-linear relationships, they are limited by a critical assumption: observations are treated as independent and identically distributed (i.i.d.). This overlooks the relational nature of real estate data, where the value of a property is influenced by neighbouring assets and shared infrastructure. Most ML-based AVMs lack spatial awareness, ignore temporal dynamics, and do not capture relational reasoning. As a result, despite their scalability and accuracy, these models often fail to reflect the true spatial and temporal complexities of real-world housing markets, highlighting the need for graph-based approaches that can model these dependencies explicitly.

3.3. GNNs in Real Estate

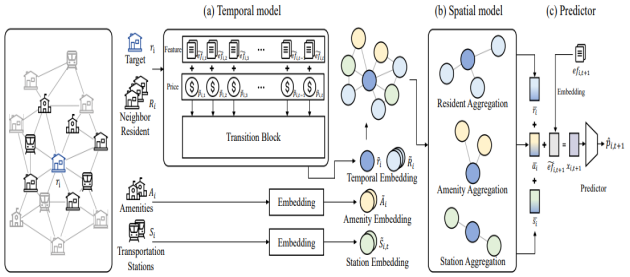
GNNs have recently emerged as a promising tool for real estate valuation, allowing for more sophisticated modelling of spatial dependencies than traditional AVMs. Several frameworks have been proposed, each building upon the limitations of its predecessors. Among these, MugRep, ReGram, and ST-RAP represent significant advancements in leveraging GNNs for property appraisal. Figure 3.2 provides an overview of the respective architectures of these models.



(a) Framework of MugRep adapted from [36, Figure 2]



(b) Framework of ReGram adapted from [62, Figure 2]



(c) Framework of ST-RAP adapted from [35, Figure 1]

Figure 3.2: Architectural comparison of GNN-based real estate valuation models. MugRep uses separate graphs for transactions and communities. ReGram enhances relational weighting among neighbors. ST-RAP incorporates heterogeneous urban context and temporal encoding.

The first application of GNNs in real estate appraisal was introduced by Zhang et al. [36], who proposed MugRep, a Multi-Task Hierarchical Graph Representation Learning framework. This approach sought to address key challenges in property valuation by integrating multi-source urban data, including demographic and mobility patterns, and constructing a multi-level graph-based model. MugRep introduced an evolving transaction event graph that captures spatio-temporal dependencies by linking transactions based on both spatial proximity and temporal recency. To extract meaningful representations from this graph, the model employed GATs at the transaction level. At the community level, graph convolutions were used to capture intra-community interactions, while an inter-community graph was constructed to model correlations between different communities based on demographic, economic, and infrastructural similarities. Additionally, the framework leveraged multi-task learning, where different urban districts were treated as separate tasks to account for region-specific price distributions. Despite its contributions, MugRep exhibited some notable limitations. The model aggregated all features from neighbouring transactions indiscriminately, introducing noise and incorporating irrelevant information. Furthermore, while the framework captured spatial dependencies effectively, it did not explicitly consider the transaction prices of neighbouring properties, which are a crucial factor in real estate valuation. Another shortcoming was its reliance on district-based price segmentation, which ignored correlations between adjacent regions, making it prone to errors in sparsely populated areas.

To address these shortcomings, Li et al. proposed ReGram, a Neighbour Relation Graph Learning Framework [62]. Unlike MugRep, which aggregated all neighbouring node features uniformly, ReGram introduced a more refined approach by learning weighted relationships between transactions, ensuring that only the most relevant neighbours influenced the target valuation. This was achieved through an attention mechanism that assigned different importance scores to neighbouring transactions based on their

relevance. Additionally, ReGram incorporated a preliminary price estimation step, where the valuation of a property was initially determined based on the prices of its neighbours before refining this estimate using additional features. This improvement made the model more sensitive to local price variations. Another key advancement in ReGram was the introduction of a dynamic adaptor, which adjusted the pricing model based on specific property characteristics, making it more flexible than the rigid district-based segmentation used in MugRep. However, while ReGram significantly enhanced spatial modelling, it did not introduce an explicit temporal modelling component. It relied on static statistical measures of past transactions rather than learning dynamic price trends over time. Furthermore, by placing a strong emphasis on neighbour relationships, the model risked overfitting to local transactions, potentially leading to inaccurate predictions in areas with sparse transaction data.

While MugRep and ReGram primarily focused on modelling spatial dependencies, they incorporated temporal information only in a limited statistical form. To more effectively capture the evolving nature of real estate markets, Lee et al. proposed ST-RAP, a Spatio-Temporal Graph Learning Framework [35]. ST-RAP introduced an explicit time-series modelling component by employing GRUs to process transaction histories and learn temporal price dynamics. This allowed the model to capture long-term trends such as seasonal price fluctuations, inflation effects, and broader market cycles. Additionally, ST-RAP extended the spatial modelling capabilities of its predecessors by incorporating a heterogeneous graph structure, which included not only real estate transactions but also local amenities such as schools and hospitals, as well as transportation nodes like subway and bus stations. By integrating these additional entities, ST-RAP provided a more comprehensive understanding of property valuation, recognizing that real estate prices are influenced by accessibility and infrastructure in addition to direct property attributes. The model followed a hierarchical learning approach, first encoding transaction-level historical trends using GRUs, then aggregating spatial information from neighbouring properties, and finally incorporating broader regional factors through heterogeneous graph convolutions.

Although ST-RAP represents a significant improvement over MugRep and ReGram, it is not without its limitations. The combination of GRU-based temporal modelling and heterogeneous graph convolutions introduces substantial computational complexity, making the model expensive to train and deploy on large-scale datasets. Additionally, while the temporal model effectively captures historical trends, it may struggle in areas with sparse transaction data, where meaningful patterns are difficult to extract. Another limitation lies in its static spatial assumptions; although ST-RAP models spatial relationships effectively, the graph structure itself remains largely fixed and does not dynamically update as new transactions occur or as urban infrastructure evolves. This can lead to outdated spatial relationships that fail to reflect real-world changes over time.

3.4. Hierarchical Graph Learning

Hierarchical Graph Neural Networks (HGNNs) provide a principled framework for modelling multi-level relational structures by learning representations across different spatial or conceptual scales. This is particularly important in domains like real estate, where entities, such as individual property transactions, are naturally nested within neighbourhoods, districts, and broader urban systems. Traditional HGNN methods rely on *graph pooling* techniques such as DiffPool [42], Top-k Pooling [41], and SAGPool [63], which learn to cluster or select subsets of nodes to create coarser graphs for higher-level reasoning. However, these approaches are mainly suited for tasks like graph-level classification and do not directly address the needs of node-level prediction in spatially structured, real-world settings, as pooling often discards local structure and assumes that hierarchies emerge solely from graph topology.

More recent works introduce semantic or type-aware hierarchies. For instance, HAN [64] and MAGNN [65] model relational semantics in heterogeneous graphs using meta-path attention and inter-path aggregation. HeCo [66] applies contrastive learning between local and global views to enforce semantic consistency. However, these methods are largely static, flat in their structure, and lack explicit interaction between hierarchy levels, relying instead on embedded abstraction.

Several works begin to address the need for hierarchical temporal or heterogeneous reasoning, though with limitations. HGT [67] and Heterformer [68] model time-aware message passing in heterogeneous graphs but do not incorporate multi-level structure. Similarly, HTGNN [69] models heterogeneous and temporal dynamics across node types, yet lacks bidirectional inter-layer flow or support for asynchronous temporal scales, limiting its capacity to model co-evolving spatial hierarchies such as transactions and

neighbourhoods.

In the context of property valuation, recent works have begun to explore hierarchical and spatio-temporal GNN architectures. ST-RAP [35], for instance, introduces a regional-temporal attention mechanism that captures evolving neighbourhood trends over time. While it incorporates hierarchical signals in a soft manner (e.g., by injecting regional and temporal embeddings), it does not model explicit inter-layer edges or enable structured message passing between transaction-level and regional nodes. MugRep [36] advances this further by employing a multitask learning framework with two separate graphs, one representing transactions and the other for community-level aggregations. However, these graphs operate independently, without shared nodes or edges to enable joint reasoning across levels. Additionally, although HTGNN [69] successfully captures heterogeneous temporal graph dynamics in domains like finance and recommendation systems, it lacks spatial grounding and does not consider multi-resolution hierarchies relevant to urban modelling.

These models reflect broader architectural limitations in the current design of hierarchical GNNs. Most operate with either flat or loosely coupled structures, where hierarchical signals are either abstracted through pooling or injected via soft embeddings, rather than modelled through explicit inter-layer interactions. Information flow is typically unidirectional, aggregated in a bottom-up manner, without allowing for contextual feedback from higher levels. This limits the model's ability to represent top-down influences, which are often critical in domains where macro-level dynamics shape local outcomes. Recent work suggests that bidirectional message passing between levels can enhance representational expressiveness and model fidelity [70, 71], but such designs remain uncommon. Moreover, inter-level connections are rarely instantiated as typed edges within the graph structure, reducing interpretability and constraining relational reasoning across scales. Hierarchies are also typically static, pre-defined either by clustering or topology, which is insufficient for dynamic environments where entities and groupings evolve over time.

3.5. Research Gaps

Despite significant progress in both econometric and machine learning-based AVMs, several critical research gaps persist, especially when it comes to accurately modelling the spatio-temporal and hierarchical nature of real estate markets. This thesis is motivated by the following key limitations identified in existing literature:

1. **Static and Rigid Graph Structures**: Many graph-based models assume a static graph topology, where property relationships (e.g., spatial proximity, infrastructure connectivity) remain fixed over time. This restricts adaptability to dynamic urban processes such as new infrastructure development, zoning changes, or shifts in neighbourhood characteristics. There is a need for models that can dynamically update their graph structure as new transactions or contextual data become available.
2. **Weak Temporal Modelling and Fusion**: Most spatio-temporal GNNs either rely on recurrent models (e.g., GRUs) or fixed temporal embeddings to capture price evolution. These approaches often struggle with non-stationary markets, long-term dependencies, and seasonal fluctuations. Moreover, temporal information is often incorporated late in the pipeline, rather than jointly learned with spatial dependencies. A more integrated and flexible fusion of temporal and spatial signals is necessary.
3. **Lack of Hierarchical Spatial Modelling**: Property values are shaped by multi-scale influences: from micro-level characteristics (e.g., number of rooms) to macro-level dynamics (e.g., neighbourhood safety). Current models often operate at a single level (transaction or regional) or aggregate across levels without structured interaction. Few models support joint message passing across nested spatial hierarchies.

This thesis addresses this need by proposing a novel GNN architecture tailored to the structural and dynamic complexity of real estate valuation.

4

Methodology

This chapter outlines the methodology employed in this study. We begin by formally defining the problem and specifying the predictive objectives (Section 4.1). After giving a high-level overview of our process in Section 4.2, we describe the construction of a hierarchical spatio-temporal graph in Section 4.3. Building on this representation, we introduce the proposed model architecture in Section 4.4. The subsequent section details the training setup. We conclude with a discussion of key architectural design decisions, model limitations, and trade-offs (Section 4.6).

4.1. Problem Formulation

We study the task of residential property valuation, where the goal is to predict the sale price of a property based on both its individual characteristics and the broader spatial and temporal context in which it is situated. Unlike traditional approaches that assume observations to be i.i.d., we treat this as a structured prediction problem defined over a hierarchical and temporally evolving domain.

Real estate data is inherently *non-Euclidean*, shaped by spatial heterogeneity and long-range dependencies that vary over time. Properties are embedded within complex, multi-scale environments where local interactions, neighbourhood-level dynamics, and city-wide trends collectively influence prices. Spatial dependencies arise from geographic proximity, infrastructure, and socio-economic composition, while temporal dynamics reflect both short-term volatility and longer-term market cycles [69, 36].

Conventional hedonic regressions and many machine learning pipelines represent each transaction using a flat feature vector, disregarding spatial adjacency, relational structure, and non-stationary trends. By ignoring these factors, such models often fail to capture spatial spill-overs, temporal variation, and localised market behaviours [72]. The challenge is further compounded by data sparsity and heterogeneity, which limit the effectiveness of high-variance learners when applied without structural priors.

To address this, we frame the valuation task as a hierarchical spatio-temporal learning problem. Each property transaction is associated with a neighbourhood, and both levels evolve over time. This gives rise to a two-tier relational structure: a *micro-level*, capturing interactions between individual transactions, and a *macro-level*, modelling inter-neighbourhood dynamics. These levels are interdependent: regional trends influence local prices, while aggregated transactional activity informs the neighbourhood state.

We propose a graph-based formulation to model this structure. Specifically, we represent each timestep t as a hierarchical graph that supports spatial reasoning across multiple resolutions and tracks temporal evolution. This framework is designed to:

1. Capture property- and region-level interactions using graph representations.
2. Integrate spatial and temporal dependencies within a unified architecture.
3. Enable bidirectional information flow between transactions and neighbourhoods.

Objective

The goal is to predict the sale price of a transaction at time t by leveraging both micro-level features and macro-level context. We define a predictive function f_θ, parametrised by a graph neural network, such that:

$$\hat{p}_i = f_\theta(G_t, i), \tag{4.1}$$

where $\hat{p}_i$ is the predicted price of transaction node i, and G_t is the hierarchical spatio-temporal graph constructed from data up to timestep t. After providing a concise overview of the methodology, the remainder of this chapter details the construction of this graph and the components of the proposed model.

4.2. Overview

This work proposes a hierarchical spatio-temporal graph learning framework for residential property valuation, designed to capture both fine-grained transaction dynamics and broader regional trends as they evolve over time. Rather than treating each sale in isolation, the method models transactions in relation to their spatial and temporal context, reflecting the structured and interdependent nature of real estate markets.

To generate predictions at time t, we construct a hierarchical graph from transaction data in the window $[t-W, t-1]$. W represents the window size. The graph consists of two interconnected layers: a macro-level graph, where nodes represent neighbourhoods with features such as demographic indicators or aggregated statistics, and a micro-level graph, where nodes represent individual transactions grouped by neighbourhood. The micro-level allows the model to account for peer effects and local market signals, while the macro-level provides a broader view of regional behaviour and long-term trends. Cross-level edges link each transaction to its corresponding neighbourhood, enabling the model to connect local activity with regional context.

This hierarchical graph is passed to a structured neural architecture composed of three main modules.

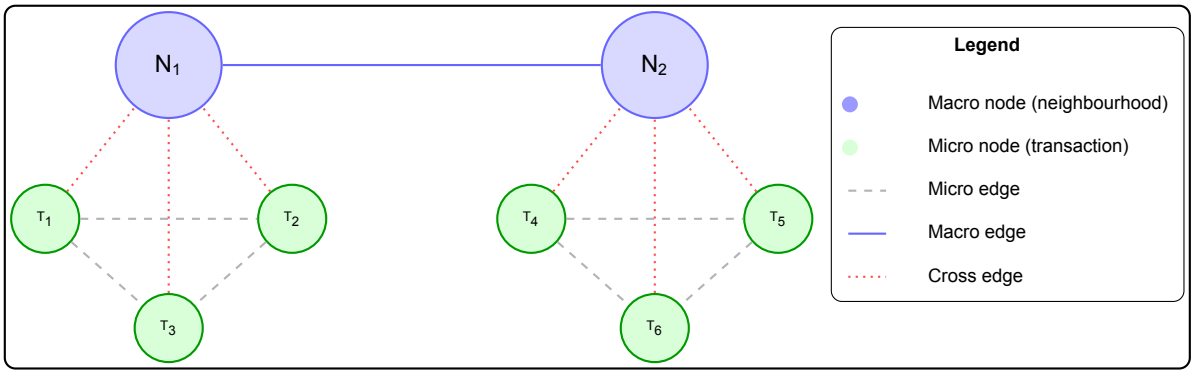


Figure 4.1: Hierarchical spatio-temporal graph at timestep t. Micro-level (transaction) nodes form fully connected sub-graphs within each neighbourhood, macro-level (neighbourhood) nodes are linked by geographic edges, and dotted cross-level edges connect each transaction to its neighbourhood

The micro-level encoder learns transaction embeddings by aggregating information from nearby sales, motivated by the need to capture spatial locality and relational effects. The macro-level encoder models interactions between neighbourhoods and maintains a lightweight memory mechanism to capture temporal dynamics without overcomplicating the architecture. To align the local and global views, a cross-level interaction module performs bidirectional message passing between transactions and neighbourhoods, allowing local patterns to influence regional representations and vice versa.

The resulting transaction embeddings, which integrate spatial, hierarchical, and temporal information, are used to predict log-transformed prices through a regression head. This end-to-end design enables the model to make informed, context-aware predictions while remaining scalable and adaptable to changing market conditions. The following two sections give a detailed description of our proposed methodology.

4.3. Spatio-Temporal Graph Construction

To capture the spatial and temporal dependencies in real estate data, we construct a hierarchical spatio-temporal graph G_t at each discrete time step t. While each G_t is static, the temporal dimension is modelled through sequences of such graphs across time. The term *spatio-temporal* here refers to both the evolving node features and the dynamics of spatial structure over successive graph snapshots.

Each graph G_t consists of three components:

$$G_t = \left(G_t^{(micro)}, G_t^{(macro)}, E_t^{(cross)} \right), \tag{4.2}$$

where:

- $G_t^{(micro)}$: a graph of individual property transactions,
- $G_t^{(macro)}$: a graph of neighbourhood-level nodes,
- $E_t^{(cross)}$: a set of edges linking each transaction to its corresponding neighbourhood.

Figure 4.1 illustrates this hierarchical structure. A detailed breakdown of node and edge types is provided in the subsections below, while Table 4.1 provides a brief overview.

Neighbourhood Graph $G_t^{(macro)}$

The macro-level graph captures regional patterns such as socio-economic indicators, environmental patterns, and infrastructural conditions. It is defined as:

$$G_t^{(macro)} = (V_N, E_N, X_N(year(t))), \tag{4.3}$$

where:

- V_N: nodes representing administrative neighbourhoods,
- E_N: edges encoding spatial adjacency between neighbourhoods,

- $X_N(year(t))$: annually varying feature vectors associated with each neighbourhood including aggregated statistics (e.g., average income, pollution, accessibility). Here, $year(t)$ represents the year of the timestep t.

The macro-level graph captures long-term structural patterns and enables reasoning over broader spatial relationships. Its features update at a coarser temporal resolution (e.g., yearly) than the transaction-level graph. For simplicity, we assume that the definitions of the neighbourhoods and their respective adjacency remain constant throughout time, i.e. there are no new neighbourhoods created or no neighbourhoods merged together. Hence, V_N and E_N are not dependent on time.

Transaction Graph $G_t^{(micro)}$

The micro-level graph models local transaction dynamics at finer granularity. It is defined as:

$$G_t^{(micro)} = (V_{PT}(t), E_{PT}(t), X_{PT}(t)), \tag{4.4}$$

where:

- $V_{PT}(t)$: nodes representing property transactions at or before time t,
- $E_{PT}(t)$: edges connecting transactions within the same neighbourhood
- $X_{PT}(t)$: features describing property characteristics (e.g., size, location, sale date).

Edges $E_{PT}(t)$ form dense sub-graphs within each neighbourhood to reflect local market relationships. The transaction graph evolves more rapidly than the macro graph, capturing month-to-month market activity.

Cross-Level Edges $E_t^{(cross)}$

To enable hierarchical reasoning, we introduce bidirectional edges connecting each transaction node to its containing neighbourhood. These cross-level edges allow the model to learn multi-scale patterns and facilitate interaction across spatial resolutions.

Table 4.1: Summary of node and edge types in the hierarchical graph.

Type	Level	Description	Feature Examples
Node	Micro	Property transaction	Price, size, number of rooms, date, geolocation
Node	Macro	Neighbourhood region	Median income, pollution, transport access, avg. sale price
Edge	Micro–Micro	Links between transactions in same neighbourhood	Spatial proximity, transaction similarity
Edge	Macro–Macro	Links between neighbouring or similar regions	Geographic adjacency
Edge	Cross-Level	Connects a transaction to its neighbourhood	Administrative boundary or spatial inclusion

Temporal Structure

We adopt a discrete-time snapshot approach to model temporal dynamics. The system is represented as a sequence of discrete graphs $\{G_t\}_{t=1}^{T}$, where each G_t reflects the spatial relationships and node attributes relevant at time t. Micro- and macro-level components evolve independently over time.

4.4. Proposed Model

We propose a Multi-Scale Bidirectional Spatio-Temporal Graph Neural Network (MBSTGNN) model. Our model is built on a hierarchical spatio-temporal graph representation $G_t = (G_t^{(micro)}, G_t^{(macro)}, E_t^{(cross)})$ that enables joint reasoning over fine-grained transaction patterns and broader neighbourhood-level dynamics. It consists of three interconnected components: the Micro-Level Encoder, the Macro-Level Encoder, and the Cross-Level Interaction Module. This modular structure supports bidirectional information exchange

between micro- and macro-scales, aligning with the hierarchical structure of urban housing markets. An overview of the full architecture is provided in Figure 4.2.

Micro-Level Encoder

To model transaction-level interactions, we use a GraphSAGE encoder [27] with mean aggregation followed by a linear transformation layer. For each transaction node $i \in V_T(t)$, we compute:

$$\mathbf{h}_i^{(1)} = ReLU\left(W_{micro}^{(1)}[\mathbf{x}_i \| \text{MEAN}(\{\mathbf{x}_j : j \in \mathcal{N}(i)\})]\right), \tag{4.5}$$

$$\mathbf{h}_i^{micro} = ReLU\left(W_{micro}^{(2)}\mathbf{h}_i^{(1)}\right). \tag{4.6}$$

where $ReLU$ is the ReLU activation, $W_{micro}^{(1)}, W_{micro}^{(2)}$ are learnable weight matrices, and $\mathcal{N}(i)$ denotes the neighbours of node i.

This design enables inductive generalisation to unseen transactions by learning an aggregation function over local features, rather than relying on fixed node embeddings. The mean aggregator captures neighbourhood-level peer effects, smoothing over idiosyncratic fluctuations while preserving informative relational signals. This is particularly suited to real estate markets, where recent local sales significantly influence individual property prices.

GraphSAGE also scales efficiently to dynamic graphs, avoiding costly full-graph operations and accommodating the constant arrival of new transactions. This makes it well-suited for large-scale, temporally-evolving datasets.

Macro-Level Encoder

To capture spatial dependencies between neighbourhoods, we apply a two-layer GCN [26]:

$$\mathbf{h}_v^{(1,t)} = \sigma\left(\sum_{u \in \mathcal{N}(v)\cup\{v\}} \frac{1}{\sqrt{|\mathcal{N}(v)||\mathcal{N}(u)|}} W_{macro}^{(1)}\mathbf{x}_u^{(t)}\right), \tag{4.7}$$

$$\mathbf{h}_v^{(t)} = \sigma\left(\sum_{u \in \mathcal{N}(v)\cup\{v\}} \frac{1}{\sqrt{|\mathcal{N}(v)||\mathcal{N}(u)|}} W_{macro}^{(2)}\mathbf{h}_u^{(1,t)}\right). \tag{4.8}$$

where $ReLU$ is the ReLU activation, $W_{macro}^{(1)}, W_{macro}^{(2)}$ are learnable weight matrices, and $|S|$ denotes the cardinality of set S. This spatial encoder propagates and transforms information from neighbouring regions, allowing each neighbourhood embedding to reflect not just its own state, but also the broader spatial context around it.

To account for temporal dynamics, we incorporate a lightweight gated update mechanism inspired by GRUs [73]. Specifically, after computing the macro-level embedding $\mathbf{h}_v^{(t)}$ via message passing at time t, we apply a gated fusion with the previous state $\tilde{\mathbf{h}}_v^{(t-1)}$:

$$\tilde{\mathbf{h}}_v^{(t)} = \alpha_t \odot \mathbf{h}_v^{(t)} + (1 - \alpha_t) \odot \tilde{\mathbf{h}}_v^{(t-1)}, \tag{4.9}$$

$$\alpha_t = \sigma\left(W_\alpha \mathbf{h}_v^{(t)} + U_\alpha \tilde{\mathbf{h}}_v^{(t-1)}\right), \tag{4.10}$$

where $\alpha_t \in [0,1]^d$ is a learnable gating vector, and W_α, U_α are learnable parameter matrices. At $t = 0$, the value of $\tilde{\mathbf{h}}_v^{(0)}$ is $\mathbf{h}_v^{(0)}$. This gated fusion allows the model to integrate new spatial information while retaining memory of historical states. This allows for a smooth temporal evolution of neighbourhood embeddings and mitigates abrupt changes in representation.

Cross-Level Interaction Module

To couple micro-scale transactions and macro-scale neighbourhoods, we introduce cross-level edges and a message-passing mechanism that allow bidirectional interaction between both components. Let $\mathbf{H}^{\text{micro}} \in \mathbb{R}^{N_{PT} \times d_{micro}}$ be the transaction embeddings from the micro encoder, $\mathbf{H}^{\text{macro}} \in \mathbb{R}^{N_N \times d_{macro}}$ the neighbourhood embeddings from the macro encoder, and $\boldsymbol{\pi} \in \{1, \ldots, N_N\}^{N_{PT}}$ the mapping from transactions to their neighbourhoods. The module operates in three stages:

Projection to a shared latent space: Separate linear layers map the micro and macro embeddings into a common hidden space of dimension d_h:

$$\tilde{\mathbf{H}}^{\text{micro}} = W_{pt}\mathbf{H}^{\text{micro}}, \quad \tilde{\mathbf{H}}^{\text{macro}} = W_n\mathbf{H}^{\text{macro}}. \tag{4.11}$$

Here, W_{pt} and W_n are linear maps.

Bottom-up Flow: For each neighbourhood $v \in \{1, \ldots, N_N\}$ we aggregate the projected transactions that belong to it:

$$\bar{\mathbf{h}}_v^{\text{micro}} = \frac{1}{|\mathcal{T}(v)|} \sum_{i \in \mathcal{T}(v)} \tilde{\mathbf{h}}_i^{\text{micro}}, \tag{4.12}$$

where $\mathcal{T}(v) = \{i : \pi_i = v\}$ and π_i is the neighbourhood index of transaction i. A gated fusion then combines the bottom-up signal with the macro embedding:

$$\mathbf{z}_v = \sigma(W_f \bar{\mathbf{h}}_v^{\text{micro}} + U_f \tilde{\mathbf{h}}_v^{\text{macro}}), \tag{4.13}$$

$$\hat{\mathbf{h}}_v^{\text{macro}} = \mathbf{z}_v \odot \tilde{\mathbf{h}}_v^{\text{macro}} + (1 - \mathbf{z}_v) \odot \bar{\mathbf{h}}_v^{\text{micro}}, \tag{4.14}$$

where W_f, U_f are learnable parameters and σ is the sigmoid function. Because the aggregation is an average, occasional extreme sales exert limited influence. Only a consistent upward or downward local pattern will shift the regional state.

Top-down injection: The updated macro embedding $\hat{\mathbf{h}}_v^{\text{macro}}$ is injected back to its transactions using a learnable gate:

$$\boldsymbol{\gamma}_i = \sigma(W_\gamma \tilde{\mathbf{h}}_i^{\text{micro}} + U_\gamma \hat{\mathbf{h}}_{\pi_i}^{\text{macro}}), \tag{4.15}$$

$$\hat{\mathbf{h}}_i^{\text{micro}} = \boldsymbol{\gamma}_i \odot \tilde{\mathbf{h}}_i^{\text{micro}} + (1 - \boldsymbol{\gamma}_i) \odot \hat{\mathbf{h}}_{\pi_i}^{\text{macro}}, \tag{4.16}$$

where W_γ, U_γ are learnable parameters and σ is the sigmoid function. The final outputs are the updated transaction embeddings $\hat{\mathbf{H}}^{\text{micro}}$ and macro embeddings $\hat{\mathbf{H}}^{\text{macro}}$, allowing the two scales to co-evolve over time.

This bidirectional flow allows the model to contextualise micro-level representations using global trends while letting local patterns influence macro states. Unlike prior works such as MugRep [36] or HiSTGNN [71], which support only unidirectional flows, our approach facilitates richer cross-scale dependency modelling.

Readout

The final transaction embedding $\hat{\mathbf{h}}_i^{micro}$ is passed through a regression head (MLP) to predict the price:

$$\hat{y}_i = \text{MLP}(\hat{\mathbf{h}}_i^{micro}). \tag{4.17}$$

This hierarchical model balances local transaction features with evolving regional trends and enables dynamic valuation predictions in a scalable, generalisable framework.

4.5. Training

In this section, we describe our training setup and training objective used for all our models.

4.5.1. Training Objective

We train all models in a supervised regression setting to predict the logarithm of the transaction price $y_i \in \mathbb{R}$ for each property i. Using the log-transformed price stabilises variance and removes heteroscedasticity in the residual errors, supports proportional error modelling, and ensures fairness across different price scales. For each transaction i, the model outputs a prediction $\hat{y}_i$, and learning proceeds by minimizing the Mean Squared Error (MSE) loss:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{|PT_t|} \sum_{i \in PT_t} (\hat{y}_i - y_i)^2, \tag{4.18}$$

where PT_t is the set of property transactions in prediction timestep t.

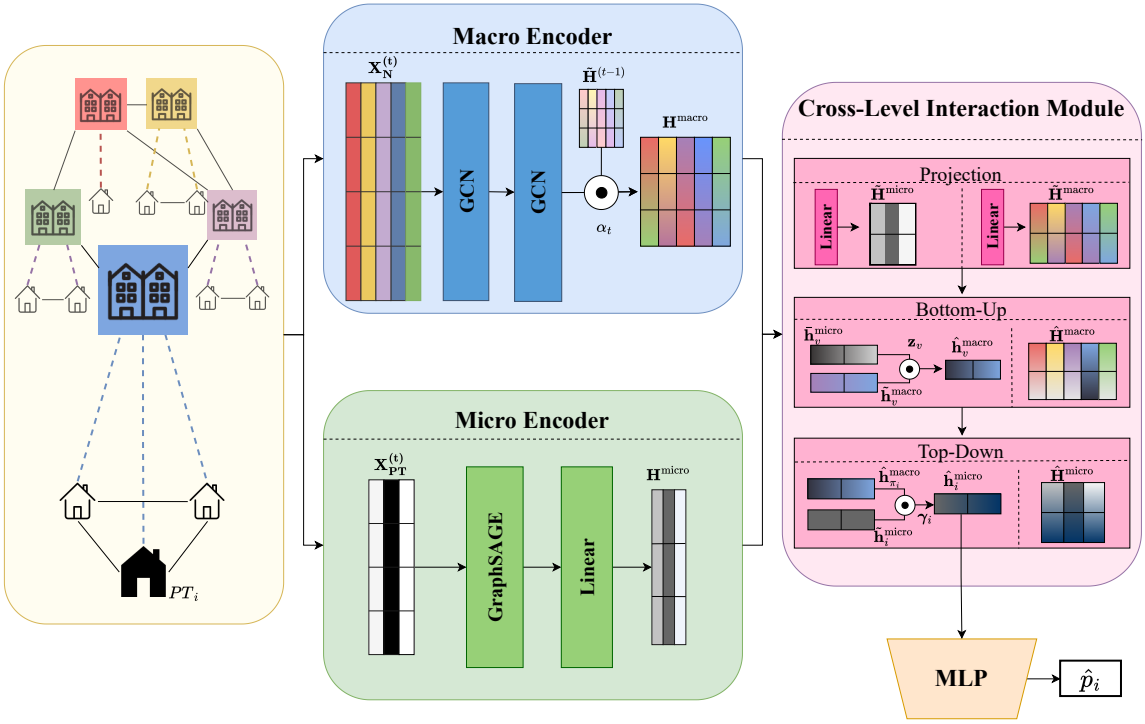


Figure 4.2: Architecture of the proposed MBSTGNN model. The model consists of micro-level and macro-level encoders, implemented with GraphSAGE and GCN layers, respectively. A cross-level interaction module enables bidirectional information flow: bottom-up propagation aggregates transaction patterns into neighbourhood representations, while top-down propagation incorporates neighbourhood context back into transaction embeddings.

4.5.2. Training Procedure

We adopt a rolling window training procedure to align the learning process with the temporal structure of the data and simulate realistic deployment conditions. The model is trained in sequential monthly windows, allowing it to incorporate only historical information available up to each test period. This mimics how data is streamed and made available to AVMs in the industry, for example in the case of Ortec Finance, where they obtain a batch of transactions occurring in the past month from their proprietary data provider, a third-party company that specialises in sourcing and aggregating such information. To scale training and enable localised gradient updates, we adopt subgraph-based mini-batching, which allows the model to focus on relevant spatio-temporal neighbourhoods while preserving the inductive bias of the graph structure.

Rolling Window Framework

Let W be the window size in months. For each evaluation month t, the model is trained on the preceding W months, $[t-W, t-1]$, and evaluated on t. The first $W-1$ months, $[t-W, t-2]$, form the training set, and the final month, $t-1$, serves as the validation set for early stopping. The validation set is never used for parameter updates, ensuring an unbiased performance estimate. Model parameters from the best validation epoch are carried forward to the next window via warm-start initialisation, enabling faster convergence and adaptation to temporal drift. This process repeats until all months are evaluated.

Subgraph-Based Mini Batching

For each training batch within a window, we construct a localised subgraph to improve efficiency and enhance the model's ability to exploit spatial and temporal structure. Specifically, we implement a

neighbourhood-temporal subgraphing strategy that includes:

1. **Neighbourhood filtering**: All transactions associated with neighbourhoods present in the current batch are retained. This ensures that spatial dependencies between geographically related nodes are preserved.
2. **Temporal filtering**: Only transactions with timestamps earlier than or equal to the last timestamp in the current batch are included. This prevents information leakage from future periods.

The resulting sub-graph retains only the nodes and edges relevant to the current batch and supports relational message passing within both the transaction and neighbourhood domains. Additionally, bipartite edges are constructed to represent the relationship between individual transactions and their corresponding neighbourhoods. This enables scalable training on large heterogeneous graphs by operating on smaller, localised portions of the full graph.

Training Loop Per Window

Within each window, we train for a fixed number of epochs using mini-batches of transaction nodes. For each batch we first extract the subgraph for the current batch. After that, the model performs a forward pass over the subgraph. We then compute the MSE loss over batch nodes only and update parameters using gradient descent. Algorithm 1 provides pseudo-code for this training loop.

We apply early stopping based on validation loss to prevent overfitting. During an initial warm-up phase (i.e., first few windows), early stopping is disabled to ensure sufficient training. Model parameters are carried forward between windows via warm-start initialisation, enabling the model to retain useful temporal knowledge and converge faster in subsequent iterations.

Algorithm 1: Rolling Window Training with Subgraph-Based Mini-Batching

Input: Model M, transaction data D, neighbourhood features F, graph builder G, window size W, batch size B, number of epochs E
Output: Trained model and predictions over evaluation months

Initialize model M_0 with random weights
Set evaluation month $t \leftarrow t_{\text{min}} + W$ ▷ First W months reserved for initial training window
while $t \leq t_{max}$ **do**
  Define training period $[t - W, t - 2]$, validation period $t - 1$, and test period t
  Extract $D_{\text{train}}, D_{\text{val}}, D_{\text{test}}$ from D
  Build graph $H \leftarrow G(D_{\text{train}} \cup D_{\text{val}} \cup D_{\text{test}}, F)$
  $I_{\text{train}}, I_{\text{val}}, I_{\text{test}} \leftarrow$ transaction indices for training, validation, and test sets
  Load weights from previous window into M_t (warm start)
  for $e \leftarrow 1$ **to** E **do**
    foreach *batch* $B_k \subset I_{train}$ *of size* B **do**
      Build subgraph H_k centred on B_k with neighbourhood and temporal filtering
      Predict batch outputs $\hat{y}_k \leftarrow M_t(H_k)$ ▷ Forward pass and optimisation
      Compute loss $\mathcal{L}_k$ on batch nodes only
      Update model weights via gradient descent
    end
    Evaluate M_t on I_{val} using MSE
    Apply early stopping if validation loss does not improve for p consecutive epochs
  end
  Store predictions from best validation model for test month t
  Advance evaluation month: $t \leftarrow t + 1$
end

4.6. Discussion

The proposed hierarchical spatio-temporal graph neural network provides a flexible, scalable, and context-aware framework for residential property valuation. This section reflects on the architectural design rationale and highlights the limitations and trade-offs of the model.

4.6.1. Design Rationale and Architectural Decisions

A key motivation behind the model's structure is the inherently hierarchical nature of real estate systems, where local transactions are embedded within broader regional contexts. To capture this multi-scale interaction, the architecture combines a transaction-level graph (micro graph) with a region-level graph (macro graph), coupled through a cross-level message passing mechanism that allows dynamic, bidirectional flow of information.

At the micro level, GraphSAGE was selected as the backbone encoder due to its inductive learning capabilities and efficiency. Unlike transductive models such as GCNs, GraphSAGE does not require access to the full graph during inference, enabling it to generalise to previously unseen nodes, such as new property transactions. Its neighbourhood sampling and mean aggregation provide a robust and scalable mechanism for learning representations in dynamic environments where the graph structure evolves over time. This is particularly important in real estate, where new transactions occur continuously and system-wide retraining is often impractical.

In contrast, the macro-level graph, which represents neighbourhoods or administrative units, is relatively small and stable. Here, a two-layer GCN is sufficient to capture spatial dependencies between adjacent regions. Deeper architectures were deliberately avoided to prevent over-smoothing, which can cause node embeddings to lose their uniqueness. Temporal evolution in macro-level features is modelled using a simple gated update mechanism instead of full recurrent units like GRUs or LSTMs. This choice reflects a trade-off between temporal expressiveness and computational simplicity. Given that macro indicators such as average income or pollution levels tend to change gradually, the overhead introduced by recurrent modules was not justified by a corresponding improvement in performance.

Attention mechanisms, despite their growing popularity in GNN literature, were excluded from the final architecture. In preliminary experiments, attention layers significantly increased computational cost and led to instability in data-sparse neighbourhoods, without offering consistent gains in predictive accuracy. While attention can improve interpretability by assigning importance weights to nodes or edges, its scalability limitations made it unsuitable for our setting, especially at the micro-level. We refer the reader to Appendix B for the results of these preliminary experiments.

The cross-level interaction module plays a central role in the architecture by enabling coordinated learning between the transaction and regional graphs. Unlike prior hierarchical models that assume a unidirectional flow, typically from region to transaction, our framework allows for mutual influence. For instance, sudden clusters of high-value sales can propagate upward to influence regional embeddings, while changes in macro-level economic or policy conditions can trickle down to affect individual valuations. This co-evolution is implemented through simple linear projections and gated fusion mechanisms, which strike a balance between model complexity and representational capacity.

4.6.2. Limitations and Trade-offs

While the proposed hierarchical spatio-temporal GNN framework is designed to balance scalability, expressiveness, and modularity, several limitations arise from these design choices. These constraints manifest particularly in relation to temporal modelling depth, rare event sensitivity, model interpretability, and deployment scalability. Understanding these trade-offs is critical for assessing the model's practical applicability and guiding future work.

One limitation of the model lies in its simplified treatment of temporal dynamics. By employing a gated update mechanism instead of full recurrent architectures, the model sacrifices temporal expressiveness for efficiency. This choice is justified in our setting, where macro-level indicators evolve gradually and transaction timestamps are irregular. However, in markets with pronounced seasonal trends, policy-driven shocks, or volatile investor behaviour, this may lead to underfitting temporal signals. For example, abrupt regime shifts, such as a sudden interest rate change or zoning policy, may not be adequately captured without more expressive temporal mechanisms that model memory and transitions explicitly. Similarly, regions undergoing rapid gentrification or speculative activity may benefit from deeper sequential models capable of recognising long-term autocorrelation patterns.

Another significant trade-off is the use of mean aggregation in GraphSAGE and in the bottom-up cross-level interaction. This approach ensures stability and resistance to noise, but it inherently smooths over outliers. As a result, atypical yet meaningful transactions, such as a luxury property sale in an otherwise affordable

neighbourhood, may be diluted during aggregation. In emerging or transitioning markets where rare events signal structural change, this can limit the model's responsiveness.

Scalability, while a strength compared to other GNNs, still presents trade-offs when viewed relative to non-graph baselines. Models like LightGBM or feedforward neural networks are generally more computationally efficient and easier to parallelise across distributed systems. They do not require graph construction or message passing and can scale linearly with the number of samples. By contrast, even an inductive GNN like GraphSAGE incurs overhead from neighbourhood sampling and aggregation, especially in high-density urban graphs. Moreover, models like LightGBM often perform competitively on tabular real estate datasets, particularly when extensive feature engineering is applied. In such cases, the added complexity of graph-based learning may only offer marginal gains, particularly in static or feature-rich environments.

In terms of deployment, the rolling-window training paradigm may constrain industrial usability in fast-paced environments. Real estate platforms, valuation services, or municipal planners may require real-time updates as new transactions occur. Because the model is currently trained offline, it must be retrained periodically to incorporate fresh data. This imposes latency and computational costs that may not be acceptable in operational settings. In contrast, models like LightGBM can be retrained incrementally or replaced cheaply, and even some recent GNN variants are beginning to explore continual or streaming learning capabilities. Introducing online or incremental learning mechanisms into the current framework is a promising direction to improve responsiveness and maintain predictive performance over time. We conduct preliminary experiments with regard to this and present our findings in Appendix B.4

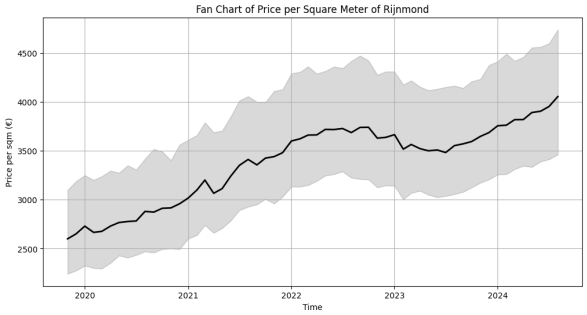
Explainability presents a nuanced trade-off. Traditional models such as LightGBM are often favoured for their global feature importance metrics and straightforward visualisation tools, making them quite interpretable in static tabular settings. However, they lack explicit modelling of spatial relationships and cannot attribute prediction outcomes to neighbouring entities in a principled way. In contrast, the proposed graph-based model, though not fully interpretable in a classical sense, offers a degree of relational explainability that is particularly valuable in spatial domains like housing. By structuring the input data as a graph, the model can trace how features from neighbouring properties or regional trends influence a given transaction's valuation. For instance, GraphSAGE's aggregation functions encode how nearby transactions contribute to an embedding, and post-hoc analysis can reveal which neighbours had the strongest influence on the prediction. Similarly, the gated fusion mechanism between micro- and macro-levels enables interpretation of how much influence each level had in the final output. This allows domain experts to explore why a property was assigned a particular valuation based on the characteristics of its spatial context, something that tree-based models cannot do natively. However, it does not yet reach the instance-level attention-based interpretability seen in more complex GNNs.

Overall, while the proposed model is well-suited for structured mid-scale real estate tasks that benefit from spatial reasoning, its applicability may be limited in scenarios with highly volatile temporal dynamics, sparse data containing rare but influential events, or deployment settings that require full model transparency and rapid, continual updates. These challenges reflect trade-offs inherent in the current architecture, particularly its emphasis on structural expressiveness and modularity over fine-grained temporal modelling or full interpretability. While not fundamental flaws, they do constrain the model's flexibility and robustness in certain real-world applications, underscoring the need for further refinement through the incorporation of more dynamic temporal components, interpretable aggregation strategies, and adaptive learning frameworks.

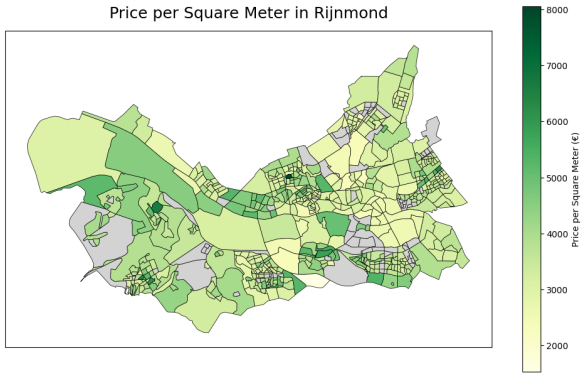
5

Experiments

This chapter presents the experimental framework used to evaluate the proposed model. We start by describing the datasets employed and the preprocessing steps applied to prepare the data in Section 5.1 and Section 5.2. In Section 5.3 we detail the experimental setup, including training configurations and environment details. Next, we outline the baseline methods used for comparison in Section 5.4 and ablation studies conducted to assess the contribution of key components in Section 5.5. Finally, we specify the evaluation metrics utilised to measure model performance and the significance tests used in Section 5.6 and Section 5.7.



(a) Price per square metre of the transactions in Rijnmond throughout time



(b) Price per neighbourhood across Rijnmond

5.1. Data

This study utilises a rich, multi-relational dataset provided by Ortec Finance, encompassing real estate transactions and neighbourhood-level attributes across various regions in the Netherlands spanning from 2014 to 2024. The dataset closely mirrors the data employed in Ortec Finance's AVMs, thereby offering a realistic representation of the challenges encountered in operational real estate valuation. As such, it captures both the detailed property-level heterogeneity and broader contextual patterns typical of real-world real estate data.

For the sake of manageable training times, we perform all experiments using only the data from the COROPPLUS region of Rijnmond. We believe that this subset is diverse and large enough to provide a generalised representation of the performance of the models. In the region of Rijnmond there are 108,998 transactions with the mean price per square metre being around 2,898 euros. The minimum price per square metre is 900 euros while the maximum is 13,095 euros. Figure 5.1a shows the trend of the prices of the properties throughout time. Furthermore there are 625 neighbourhoods in this region. Figure 5.1b shows the distribution of prices across neighbourhoods of this region.

5.1.1. Transaction Data

The transaction dataset is curated from several public and commercial sources. The core records such as transaction date, closing price, and legal address, are obtained from *Kadaster* [1] (the Dutch land registry). These base records are subsequently enriched with (i) *advertisement features* scraped from the *Spotzi*[2] real-estate portal, (ii) *geographic coordinates* from *postcode.nl* [3], and (iii) a multi-level *spatial hierarchy* (province, municipality, COROPPLUS, neighbourhood code, and postal code) supplied by the *Dutch Central Bureau of Statistics (CBS)* [4]. Table 5.1 summarises the main feature categories; the full, column-level specification of the features used in this study is provided in Appendix C.

5.1.2. Synthetic Transaction Data

Since the source transaction data is proprietary, a differentially private synthetic dataset was generated for reproducibility purposes. The raw transaction records were preprocessed by removing outliers, excluding properties built before 1900, and applying logarithmic transformations to living area and plot area. The target variable was defined as the log of price per square metre, de-trended using an exponentially weighted monthly average. All variables were discretised into bins (64 for numerical, or the number of categories otherwise).

Two-way marginals involving the target and selected high-importance variables were computed, with Gaussian noise added to satisfy differential privacy. These noisy marginals were then used to fit a probabilistic graphical model via the Private-PGM framework[74], capturing the joint distribution of the

[1] https://www.kadaster.nl

[2] https://www.spotzi.com

[3] https://www.postcode.nl

[4] https://www.cbs.nl/

Table 5.1: Overview of transaction-level features.

Category	Details	Source
Sale price	Final notarised transaction amount (euros, nominal).	Kadaster
Property attributes	Floor area, number of rooms, construction year, dwelling type, garden presence, energy label, etc.	Spotzi ads
Geolocation	Latitude and longitude .	postcode.nl
Time stamp	Exact transfer date (YYYY-MM-DD).	Kadaster
Spatial hierarchy	Province $\rightarrow$ municipality $\rightarrow$ COROPPLUS region $\rightarrow$ postal code $\rightarrow$ neighbourhood.	CBS

variables. Sampling from this model produced 200,000 synthetic transactions, preserving the statistical structure of the original data while ensuring privacy. Although this captures some of the main trends in the data, the dataset produces is not as complex as the original data. The data can be found on the Github repository[5] of this thesis. The results associated to the experiments conducted on this dataset are documented in Appendix D.

5.1.3. Neighbourhood Data

The macro-level context is captured using neighbourhood-level data primarily sourced from the CBS, and enriched with environmental and liveability indicators sourced from the National Institute for Public Health and the Environment (RIVM)[6]. Each neighbourhood is represented as a macro node in the heterogeneous graph, with attributes encompassing demographic composition, infrastructure access, and socio-economic profiles. In addition, environmental quality indicators such as NO_2, PM2.5, and PM10 concentrations are included to account for air quality differences across regions. Subjective liveability metrics are also incorporated to reflect perceived quality of life. The dataset provides yearly measurements, supporting temporal modelling of neighbourhood dynamics over time.

The neighbourhood feature vector comprises over 150 attributes, which can be broadly categorised as described in Table 5.2. The adjacency structure for the neighbourhood graph is derived using geographical adjacency information provided by CBS, which indicates which neighbourhoods share borders. This adjacency is encoded as an adjacency matrix, where each edge represents a pair of neighbouring spatial units.

Category	Details
Demographics	Total population, gender distribution, age brackets, marital status, household composition, migration background
Socio-economic Indicators	Average income per household and resident, income inequality, benefit recipients by type, employment statistics by industry
Urban/Environmental Characteristics	Population density, building types, occupancy status, property age, green space, pollution levels, traffic noise
Public Infrastructure/Amenities	Distances to hospitals, schools, supermarkets, public transport, cultural facilities, availability within specified radii
Utility Consumption	Average electricity and gas usage by dwelling and tenure type
Land/Spatial Attributes	Surface area composition, postal code concentration, urbanization degree, address density

Table 5.2: Overview of neighbourhood feature categories and corresponding attributes.

[5] https://github.com/AratrikaD/bidirectional-multiscale-graph-learning

[6] https://www.rivm.nl

5.2. Data Preprocessing

Before constructing the spatio-temporal graphs and training the models, several preprocessing steps were applied to ensure consistency, remove noise, and align temporal and spatial features across datasets.

5.2.1. Transaction Data

The data was preprocessed to ensure spatial alignment, temporal consistency, and removal of redundancies. Redundant or duplicated columns and columns related to spatial features higher in granularity than the neighbourhood level such as the province and the municipality code were dropped. The resulting data retained features related to transaction metadata (e.g., date, price), structural attributes (e.g., building type, construction year), and geospatial identifiers (e.g., latitude, longitude, neighbourhood code). Furthermore, data points with missing values for any of the geospatial identifiers were removed. The transactions were chronologically ordered by date to support the rolling window training scheme and the transaction date column (`DATE`) was parsed to extract `YEAR` and `MONTH` fields, which are used for temporal indexing and graph construction. Finally, The transaction price (`SELLING_PRICE`) was log-transformed to obtain a normalized target variable (`LOG_SELLING_PRICE`), improving model stability and interpretability. This cleaned and chronologically sorted dataset serves as the micro-level node input for the hierarchical graph construction and is compatible with the rolling window training strategy described later.

5.2.2. Neighbourhood Data

For each year, the neighbourhood-level dataset includes features such as population density, green space availability, average income, and infrastructure indicators along with their geospatial definitions. These features form the macro-level node attributes used in the hierarchical graph model. An important challenge arises from the fact that neighbourhood definitions (i.e., spatial boundaries and identifiers) are not consistent across years. Neighbourhoods may be merged, split, or redrawn due to administrative changes, resulting in structural inconsistency when comparing data across time. To address this, we selected a fixed reference year (2023) as the canonical neighbourhood layout. The maps for the other years were overlayed on this reference year. Features from earlier years were spatially joined or aggregated to match the reference neighbourhood definitions. Missing values were handled using forward-filling from the closest available previous year to maintain continuity. We constructed spatial adjacency relationships based on the Queen contiguity rule [75], where two neighbourhoods are considered adjacent if they share a border. This process leads to generation of features throughout time for a fixed neighbourhood definition. Hence, the macro-level graphs created do not change structurally.

5.2.3. Feature Scaling and Data Split

To ensure numerical stability and accelerate training convergence, all input features (both micro- and macro-level) were scaled. Scaling was performed separately for each training window:

- The scaler was fit on transaction features from the training period.
- The same scaler was applied to both training and test features for each month.

For each window of size W months, the first $W-2$ months are used for training the model, the transactions in month $W-1$ form the validation set and are used for early stopping. The transactions occurring in the last month of the window are used for testing.

5.3. Experimental Setup

The hyperparameter values that we use for all our experiments with MBSTGNN can be found in Table 5.3. These were found to be the best performing values after performing hyperparameter tuning. More information about the hyperparameter tuning can be found in Appendix A.3.

All experiments were conducted using Microsoft Azure Machine Learning on a dedicated virtual machine with 4 vCPUs, 32 GB of RAM, and running Linux. The environment was managed through AzureML's orchestration tools, ensuring reproducibility and scalability. The software stack includes Python 3.10, PyTorch 2.0, and PyTorch Geometric 2.3.0 for implementing the graph neural network components. The implementation uses key libraries including `pandas`, and `scikit-learn`, along with AzureML packages for experiment tracking and deployment. LightGBM was used as a baseline, implemented via the official lightgbm Python package version 4.0.

Hyperparameter	Value
Optimizer	Adam
Learning Rate	$l = 1e-4$
Weight Decay	$\lambda = 1e-4$
Batch Size	$B = 64$
Temporal Window Size	$W = 61$ months

Table 5.3: Training hyperparameters and experimental settings

5.4. Baselines

To rigorously evaluate the effectiveness of our proposed MBSTGNN model, we compare it against a suite of representative baseline models. These baselines include both traditional machine learning approaches and graph-based architectures, allowing us to assess the contribution of macro-level context and spatial interactions.

5.4.1. LightGBM

LightGBM, as described in Equation 2.4, is a highly efficient gradient-boosted decision tree framework widely used for predictive tasks involving structured tabular data. While it does not explicitly model spatial or relational structure, it can still incorporate spatial context indirectly through input features. In our setting, the model is trained on the concatenation of all available transaction-level attributes and their corresponding neighbourhood-level features (excluding the target variable), allowing it to learn from both local and contextual signals. However, unlike graph-based approaches, LightGBM treats each transaction independently and does not model interactions or dependencies between transactions or regions in a structured manner. As such, it serves as a strong non-relational baseline, helping to assess the added value of explicitly modelling the underlying spatial and relational structure through graph-based learning. In addition to this, it is the ML model used by Ortec Finance. Prior experiments found that it performed better than random forests and neural networks. The performance of LightGBM in terms of speed and accuracy for similar types of data further motivated the choice of the company to adopt this model [13].

5.4.2. Macro-Level Graph Baseline (MacroGNN-MLP)

We initially define a simplified baseline model that relies exclusively on macro-level information. In this model, both the transaction graph $G_t^{(micro)}$ and the cross-level edges $E_t^{(cross)}$ are omitted. This baseline isolates the effect of macro-level spatial context on pricing and serves as a meaningful comparison to evaluate the added value of transaction-level modelling in the form of a graph. Figure 5.2 provides a conceptual diagram of this baseline.

The baseline operates on the neighbourhood-level spatio-temporal graph $G_t^{(macro)} = (V_N, E_N, X_N(t))$. A GCN processes this graph to produce embeddings $\mathbf{h}_v \in \mathbb{R}^d$ for each neighbourhood $v \in V_N$, capturing spatial and contextual information at the macro level. The transaction level information can be thought of as a fully disconnected graph $G_t^{(micro)} = (V_{PT}(t), \emptyset, X_{PT}(t))$

For each transaction i occurring at time t, we:

1. Identify the corresponding neighbourhood node $v(i) \in V_N$,
2. Retrieve its embedding $\mathbf{h}_{v(i)}$,
3. Concatenate it with the transaction's tabular features $\mathbf{x}_i$,
4. Pass the resulting vector $\mathbf{z}_i = [\mathbf{x}_i \| \mathbf{h}_{v(i)}]$ through a feed-forward neural network (MLP).

This yields a prediction $\hat{y}_i \in \mathbb{R}$ for the transaction price:

$$\hat{y}_i = \text{MLP}([\mathbf{x}_i \,\|\, \mathbf{h}_{v(i)}]) \tag{5.1}$$

This model still assumes independence between transactions and lacks fine-grained local interactions among them. It also misses co-evolutionary signals between micro and macro scales. These limitations are addressed in our proposed hierarchical model defined in the following section.

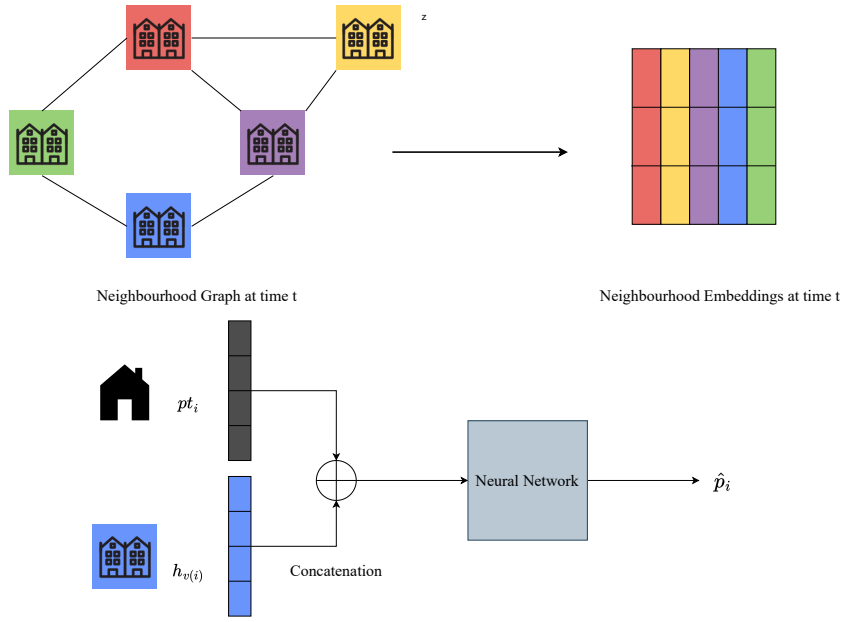


Figure 5.2: Macro-Level Graph Baseline which learns the embedding of the neighbourhood graph and then uses an MLP to estimate the price after combining with the transaction features.

5.5. Ablation Studies

To better understand the contribution of each component in our proposed hierarchical spatio-temporal GNN model, we conduct a series of ablation studies. These experiments isolate key architectural features to evaluate their individual and combined impact on predictive performance. The results associated to these experiments can be found in Appendix B.3 We compare the full model against the following ablated variants:

1. No Bottom-Up Flow: In order to assess the effectiveness of our proposed bidirectional information flow, we convert the cross-level interaction module to support unidirectional flow only. One such way is by removing the bottom-up pooling to see the effect of adding aggregate transaction level information to the macro-level embeddings

2. No Top-Down Injection: We remove the top-down injection to produce another unidirectional variant of the model. this variant helps capture the effect of the injecting neighbourhood-level features to the micro-level.

3. No Temporal Gating Mechanism: In this setting, we remove the temporal gating mechanism in the Macro Encoder along with the historical embedding matrix. This helps measure the effectiveness of the learned gating weights and the incorporation of past feature embeddings.

4. No Macro Enoder: In this model, we remove the Macro Encoder component comprsing of the 2 GCN layers. It is replaced with a linear layer to ensure that the macro-level features are projected to the correct dimensional space. This variant will identify the contribution of the macro-level graph structure and the GCNs in producing neighbourhood feature embeddings.

5. No Micro Encoder: Similar to the above variant, we replace the Micro Encoder with a linear layer. This allows us to assess the effect of the graph structure of the transactions and GraphSage layer constructing the transaction embeddings.

5.6. Evaluation Metrics

To assess the performance of our proposed model and its baselines on the property valuation task, we employ two widely-used regression evaluation metrics: MSE and Mean Absolute Percentage Error (MAPE).The MSE has been defined in Section 4.18. These metrics capture complementary aspects of prediction quality, namely absolute accuracy and relative error, making them suitable for evaluating models in this context.

MAPE quantifies the average magnitude of the prediction error as a percentage of the true value. Unlike MSE, it provides a scale-independent measure of performance, making it easier to interpret in real-world scenarios where properties vary widely in price. It is defined as:

$$\text{MAPE} = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{e^{y_i} - e^{\hat{y}_i}}{e^{y_i}} \right|, \tag{5.2}$$

where $y_i \neq 0$. While MAPE is intuitive and useful for comparative analysis, it can be unstable when true values are close to zero. In our dataset, however, such instances never occur as prices can never be zero.

Using both MSE and MAPE allows for a balanced evaluation of both the absolute deviation of predictions (as captured by MSE) and the relative quality of the estimates (as captured by MAPE). This dual-metric evaluation ensures that our models are not only minimising large deviations but are also maintaining proportional accuracy across varying price ranges.

5.7. Significance Testing

In this study, we employ two significance tests to evaluate model performance: the Diebold-Mariano test and the Wilcoxon signed-rank test.

5.7.1. Diebold-Mariano Test

The Diebold-Mariano (DM) test [76] is used to compare the predictive accuracy of two forecasting models. It tests the null hypothesis that the expected loss differential between the models is zero. Let $e_{1,t}$ and $e_{2,t}$ denote the forecast errors of two models at time t, and $L(\cdot)$ be a loss function, such as the squared error. The loss differential is defined as:

$$d_t = L(e_{1,t}) - L(e_{2,t}) \tag{5.3}$$

The DM test statistic is computed as:

$$DM = \frac{\bar{d}}{\sqrt{\hat{\gamma}_d(0)/T}} \tag{5.4}$$

where $\bar{d} = \frac{1}{T}\sum_{t=1}^{T} d_t$ is the sample mean of the loss differential, $\hat{\gamma}_d(0)$ is the estimated variance of d_t, and T is the number of observations. Under the null hypothesis, DM asymptotically follows a standard normal distribution.

5.7.2. Wilcoxon Signed-Rank Test

The Wilcoxon signed-rank test is a non-parametric test for paired samples, assessing whether their median difference is zero [77]. It is particularly useful when the normality assumption of differences may not hold. For paired observations (x_i, y_i), define the differences:

$$d_i = x_i - y_i \tag{5.5}$$

Rank the absolute differences $|d_i|$ and assign ranks R_i. Let W^+ be the sum of ranks for positive differences and W^- the sum for negative differences. The test statistic is:

$$W = \min(W^+, W^-) \tag{5.6}$$

Under the null hypothesis of zero median difference, W follows a known distribution, which can be approximated by a normal distribution for large samples.

6

Results and Discussion

This section presents a comprehensive analysis of the experimental outcomes. We begin by comparing the overall performance of the proposed model against baseline methods in Section 6.1. Subsequently, we examine temporal generalisation in Section 6.2 and investigate model performance across different spatial regions in Section 6.3. In Section 6.4, we perform a case study to provide insight into the internal representations of MBSTGNN and align our findings with expectations of domain experts. Finally, we conclude with a discussion of the findings and their implications in Section 6.5.

Table 6.1: Comparison of average error metrics over the final rolling windows for each model. Values represent mean ± standard deviation of MSE and MAPE over the last N prediction windows (5-year window size). The best values per row are **in bold and underlined**.

Last N Windows	MacroGNN-MLP	LightGBM	MBSTGNN
	MSE (mean ± SD)		
60 (all windows)	0.0940 ± 0.2452	0.0179 ± 0.0396	**0.0165 ± 0.0387**
36	0.0780 ± 0.1928	0.0188 ± 0.0408	**0.0160 ± 0.0384**
24	0.0555 ± 0.2135	0.0186 ± 0.0410	**0.0157 ± 0.0370**
12	0.0395 ± 0.2742	0.0185 ± 0.0401	**0.0151 ± 0.0358**
5	0.0382 ± 0.0729	0.0198 ± 0.0409	**0.0156 ± 0.0334**
	MAPE (% mean ± SD)		
60 (all windows)	22.12 ± 200.51	10.06 ± 9.72	**9.59 ± 9.55**
36	20.02 ± 109.70	10.47 ± 9.99	**9.53 ± 9.56**
24	17.35 ± 134.20	10.43 ± 10.00	**9.42 ± 9.33**
12	15.68 ± 186.50	10.26 ± 9.52	**9.08 ± 8.82**
5	14.28 ± 12.57	10.63 ± 9.79	**9.29 ± 8.55**

6.1. Overall Performance Comparison

To evaluate the predictive performance of the proposed model, we compare three approaches: the hierarchical spatio-temporal GNN (MBSTGNN), a macro-level graph-based baseline (MacroGNN-MLP), and a strong non-graph machine learning model (LightGBM). Table D.1 presents a summary of results based on the MSE and MAPE across different rolling window horizons. All models are evaluated using a consistent experimental setup in which monthly forecasts are made using a five-year rolling training window. However, the table stratifies results by averaging the error metrics over only the most recent N prediction windows, with $N \in \{60, 36, 24, 12, 5\}$.

It is important to recognize that the three evaluated models differ significantly in how they accumulate learning over time. LightGBM, a decision tree ensemble, adapts quickly to newly available data and typically converges within a small amount of time. In contrast, our proposed hierarchical GNN and the other graph-based baselines are deep learning models with high parameter capacity. Such models generally require more training data and time to reach stable performance [78]. For this reason, early evaluation windows (large N) may not fully reflect the steady-state capabilities of the deep learning models. To address this, results are also stratified across shorter horizons (e.g., N = 5 or 12), which correspond to periods after the models have converged more fully. This segmentation helps distinguish cold-start effects from long-term model competence.

Across all horizons, the proposed MBSTGNN achieves the best overall performance. Its MSE and MAPE are consistently lower than both the macro-level baseline and LightGBM. In the most recent five prediction windows, MBSTGNN achieves an MSE of 0.0156 and MAPE of 9.29%, outperforming LightGBM (MSE: 0.0198, MAPE: 10.63%) and MacroGNN-MLP by a wider margin. These improvements are sustained across all horizons and become more pronounced in recent periods, reflecting the model's ability to exploit increasing temporal context and learn expressive spatial dependencies.

The macro-level baseline (MacroGNN-MLP) performs significantly worse than the other models across all horizons. Its error variance is also much higher, particularly in earlier periods, indicating sensitivity to training dynamics and an inability to capture micro-level transaction effects. Its performance highlights the importance of incorporating fine-grained spatial interactions and bidirectional learning across transaction and neighbourhood levels, the core features of MBSTGNN. While LightGBM performs competitively in earlier horizons due to its fast convergence and strong performance on tabular data, it lacks the capacity to exploit structured spatial information. As a result, its performance stagnates relative to MBSTGNN in later windows.

We also inspected the residual distributions of all models, with Figure B.1 presented in Appendix B. These

Table 6.2: Model performance by property type over the last 5 prediction windows. Each subtable reports performance of the three models in terms of (a) MSE and (b) MAPE. Transaction count is included to contextualize results. The best value in each row is **in bold and underlined**.

(a) MSE by Property Type

Property Type	Count	MacroGNN-MLP	LightGBM	MBSTGNN
Terrace	1310	0.0334 ± 0.0548	0.0179 ± 0.0341	**0.0132 ± 0.0263**
Corner	480	0.0370 ± 0.0586	0.0187 ± 0.0299	**0.0136 ± 0.0278**
Apartment	2437	0.0362 ± 0.0687	0.0193 ± 0.0423	**0.0160 ± 0.0348**
Semi-Detached	297	0.0420 ± 0.0587	0.0203 ± 0.0340	**0.0154 ± 0.0267**
Detached	187	0.1017 ± 0.1819	0.0421 ± 0.0782	**0.0334 ± 0.0641**

(b) MAPE (%) by Property Type

Property Type	Count	MacroGNN-MLP	LightGBM	MBSTGNN
Terrace	1310	13.65 ± 11.46	10.09 ± 9.27	**8.55 ± 7.66**
Corner	480	14.36 ± 11.69	10.62 ± 8.81	**8.83 ± 7.46**
Apartment	2437	13.65 ± 11.23	10.52 ± 9.86	**9.40 ± 8.69**
Semi-detached	297	15.38 ± 10.45	10.66 ± 8.43	**9.28 ± 8.10**
Detached	187	24.99 ± 27.81	15.84 ± 14.39	**14.13 ± 13.19**

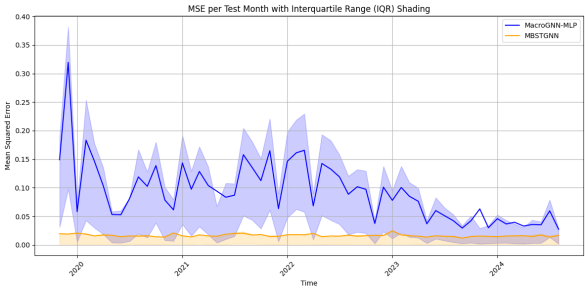
show that MBSTGNN's residuals are more symmetric and concentrated over time than the other models. This suggests that MBSTGNN not only yields lower average error but also generates more calibrated predictions. Additionally, we compare the performance of the models for different property types. Table 6.2 presents the comparative performance of all models across five residential property types over the final five prediction windows. Each subtable reports the mean and standard deviation of error metrics, contextualised by the number of transactions in each category. This allows for a clearer interpretation of both predictive accuracy and consistency under varying data volumes. Across all property types, the Macro-level baseline (MacroGNN-MLP) exhibits the highest mean errors and the widest error dispersions. Its MSE values, for instance, range from 0.0334 (terraced homes) to 0.1017 (detached homes), with standard deviations frequently exceeding the means. Both LightGBM and the proposed MBSTGNN significantly outperform this baseline, indicating the benefits of incorporating richer spatial structure and learned representations.

Notably, MBSTGNN achieves the lowest MSE and MAPE across all five property types. This consistent improvement highlights the model's robustness across both high-volume categories (e.g., apartments and terraced homes) and sparser segments (e.g., detached properties). The gains are particularly evident in detached and semi-detached categories, where higher price variance and limited samples typically challenge model performance. MBSTGNN maintains lower error means and tighter error bands, suggesting more stable predictions even under these more complex conditions.

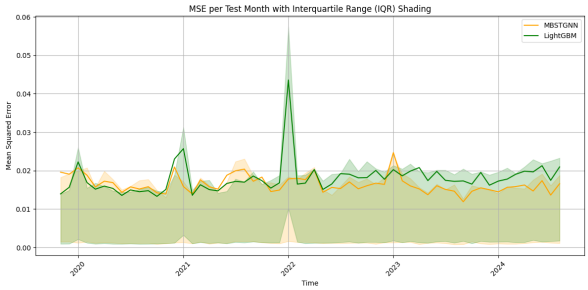
Finally, to statistically assess the significance of the performance differences between models, we employed two complementary tests: the DM test and the Wilcoxon signed-rank test as described in Section 5.7, using the MSE and MAPE across all prediction windows as the unit of comparison. Both tests consistently favoured the proposed MBSTGNN model over LightGBM, with statistically significant differences observed for both MSE and MAPE. In our evaluation, the DM test yielded a test statistic of 2.1951 ($p = 0.0282$) for MSE and 3.3401 ($p = 0.0008$) for MAPE, indicating statistically significant improvements by the proposed model. The Wilcoxon test results showed significance at a higher confidence level, with a test statistic of 460.0 ($p = 0.0022$) for MSE and 394.0 ($p = 0.0004$) for MAPE. Both tests consistently favour the proposed MBSTGNN model over LightGBM, confirming that the observed gains are unlikely due to chance.

6.2. Temporal Generalisation

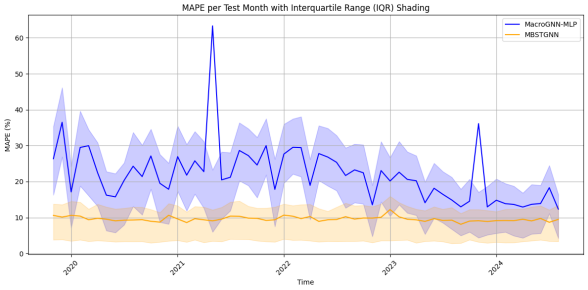
To evaluate the ability of each model to generalise across time, we assess their performance over a sequence of monthly rolling windows. Figure 6.1 presents the pairwise variation in MSE and MAPE across



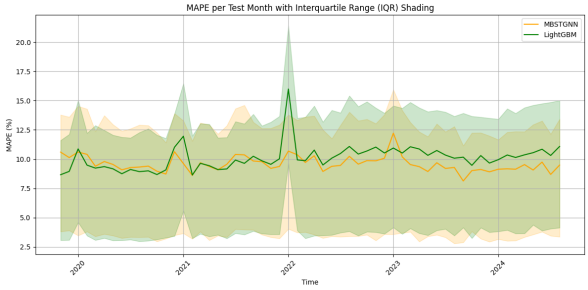
(a) Evolution of MSE of the Proposed Model and Macro-Level Baseline



(b) Evolution of MSE of the Proposed Model and LightGBM Baseline



(c) Evolution of MAPE of the Proposed Model and Macro-Level Baseline



(d) Evolution of MAPE of the Proposed Model and LightGBM Baseline

Figure 6.1: Temporal generalisation performance: MSE and MAPE variation across evaluation windows for each model. Shaded areas show the range between the 5th and the 95th percentile.

these windows, for the proposed hierarchical GNN, the Macro-level baseline and the LightGBM model. Each curve plots the average metric for the test transactions in each window along with the area between the 25th and 75th percentile (i.e. the inter-quartile range), highlighting the variability in performance over time.

In early periods of evaluation, the proposed model exhibits slightly higher error variance due to a cold-start effect, where the model has not yet been exposed to sufficient data to capture meaningful spatio-temporal and hierarchical relationships. This is evident in the MSE plot 6.1b where LightGBM demonstrates slightly more stable performance across windows, with narrower quartile ranges. However, MBSTGNN catches up and eventually surpasses LightGBM as more training data is incorporated, particularly after mid-2021. Both its average error and variability decrease, indicating improved generalisation to temporal shifts in the housing market. The model's ability to leverage both spatial structure and temporal consistency becomes increasingly advantageous in capturing evolving market trends.

MacroGNN-MLP not only shows a higher error than MBSTGNN model, but also a high level of variability. This indicates that the model is unstable and has possibly not converged. Both these graph-based models have been trained for 100 epochs per window. This lack of convergence could be because of the missing information from the transaction-level interactions and inter-graph communication. This model could also require more training steps before convergence as compared to the proposed model.

The fan chart in Figure 6.2 further substantiates these findings by comparing predicted and actual median price per square metre over time. While all models track the broad upward price trajectory, MBSTGNN aligns more closely with the actual price median, especially during price peaks and troughs. This indicates the model's enhanced capacity to adapt to short-term market fluctuations, possibly due to its hierarchical design and bidirectional flow of information between micro-level transactions and macro-level neighbourhood dynamics.

The macro-level baseline consistently underprices transactions by a large amount, showing a lagging response to real market conditions. LightGBM performs better but occasionally deviates significantly, as can be seen by the sharp drops in the curve at the beginning of 2020, 2021, and 2022. MBSTGNN shows

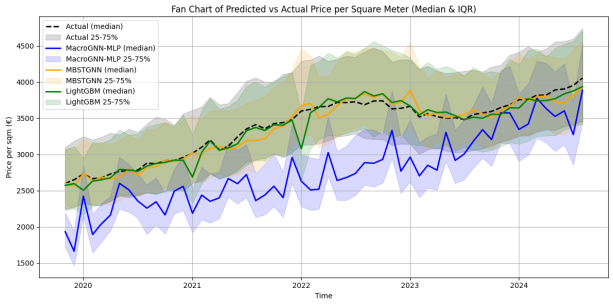


Figure 6.2: Fan chart illustrating the predicted versus actual median price per square meter over time, with interquartile range (IQR) shading. The chart compares MBSTGNN, LightGBM, and the macro-level baseline. The dashed black line represents the actual median prices. The MBSTGNN closely follows market trends with tighter uncertainty bands, indicating improved accuracy and consistency, particularly during periods of volatility.

more robust tracking, with a lower number of shocks in the curve.

6.3. Spatial Analysis

To assess spatial generalisation capabilities, we evaluate how predictive performance varies across neighbourhoods within Rotterdam. Figure 6.3a and Figure 6.3b present neighbourhood-level heatmaps for the MBSTGNN and LightGBM models, the top two performing models, respectively. These visualisations show the spatial distribution of MSE, MAPE, and transaction count averaged over the final twelve prediction windows. The figures provide a high-level view of how each model responds to varying spatial contexts and data densities across the city. To complement these visual diagnostics, Table B.1 lists the 15 neighbourhoods with the highest prediction errors (MAPE) for both models, offering a more granular perspective into areas where each method struggles most.

The heatmaps, in Figure 6.3a and Figure 6.3b, reveal clear differences in how spatial and non-spatial models generalise across the urban landscape. MBSTGNN displays relatively consistent spatial patterns, with smoother gradients in MAPE and MSE across most neighbourhoods. This behaviour is a direct result of the model's ability to propagate information through spatial connections, enabling it to interpolate more effectively in areas with sparse data. It tends to perform well in spatially coherent, residential zones where neighbourhood characteristics are stable and price signals from adjacent areas are informative. In contrast, LightGBM exhibits more fragmented spatial behaviour, with sharper and more erratic spikes in error. These abrupt changes in prediction accuracy often occur in or around areas with irregular land use patterns or limited historical transaction data. Without access to spatial context, LightGBM struggles to generalise beyond the immediate properties observed in the training data.

As shown in Table B.1 in Appendix B, several areas, such as Wibautbuurt and Rhoon-Zuidoost, consistently rank as top error contributors for both models, suggesting they are inherently difficult to model due to anomalous property characteristics or extremely low transaction volumes. This is particularly evident in industrial zones like Spaanse Polder, where a scarcity of residential sales makes any single transaction an outlier. In these extreme cases, while both models struggle, MBSTGNN can occasionally exhibit a higher MAPE, likely because its graph structure aggregates misleading spatial information from adjacent but fundamentally different areas, amplifying the initial error.

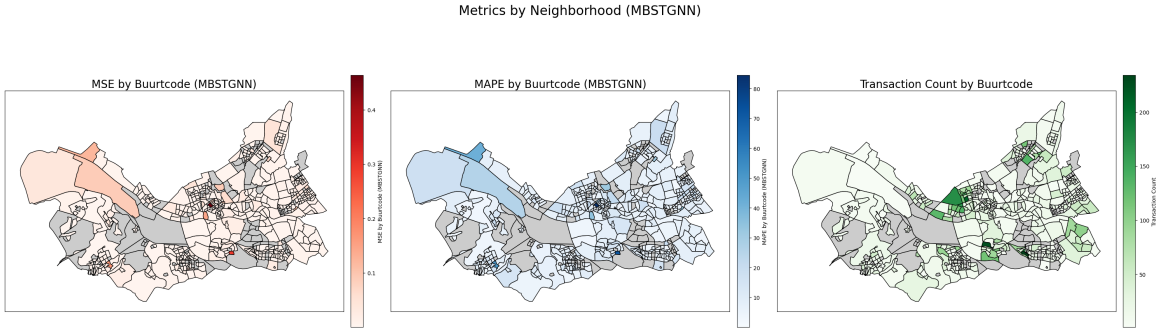
The primary distinction between the models emerges from their handling of spatial context, creating a clear trade-off between local data mastery and spatial generalisation. LightGBM's weakness is most apparent in data-sparse neighbourhoods like Beemster/Purmerhoek (MAPE: 48.4) and Verenambacht (39.5). Lacking a mechanism to leverage spatial relationships, it performs poorly when local data is insufficient. In these exact scenarios, MBSTGNN demonstrates its core strength, using its graph architecture to draw contextual information from better-sampled adjacent areas, thereby improving generalisation. Conversely, this reliance on spatial context becomes a liability for MBSTGNN in geographically unique areas like Strand en Duin (MAPE: 41.8). Here, local factors such as proximity to the sea are the dominant price drivers, and the model's assumption of spatial similarity with inland neighbours is counterproductive. In such cases,

LightGBM's feature-only approach proves advantageous by avoiding this misinformed generalisation.

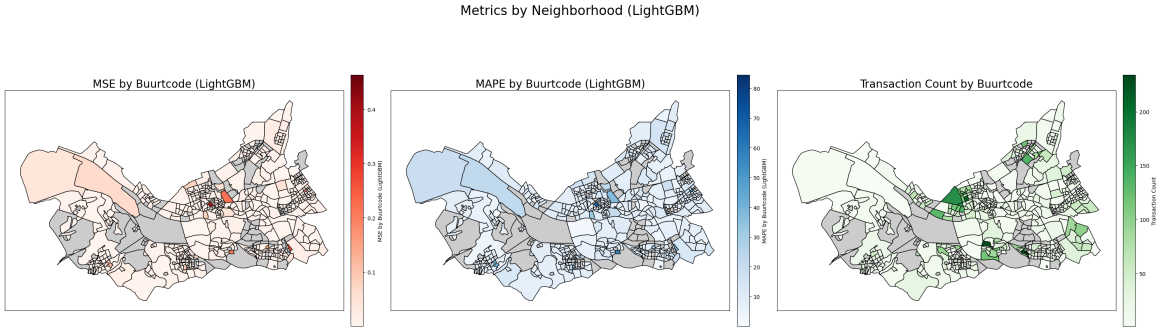
An analysis of the neighbourhoods with the lowest MAPE, shown in Figure B.2, highlights distinct differences in the areas where MBSTGNN and LightGBM achieve their best performance. MBSTGNN achieves its highest accuracy in well-connected residential areas such as Baarsveen, Oosterflank, and Eilandenbuurt, where property prices are spatially consistent and adjacent neighbourhoods provide informative context. These areas often have moderate transaction density and homogeneity in housing characteristics, allowing the graph-based architecture to effectively propagate relevant information through the network. In such environments, the spatial relationships captured by MBSTGNN enhance predictive accuracy, even in cases of data sparsity.

In contrast, LightGBM performs best in compact, internally consistent neighbourhoods like Gebouwenbuurt, Kunstenaarsbuurt-Noord, and Witte Dorp, where ample local data and low feature variability enable the model to learn accurate mappings from features to prices without relying on spatial context. Some areas, such as Tuindorp, Zuidhoek, and Struisenburg, appear in the top-performing lists of both models, suggesting that when both spatial coherence and sufficient transaction data are present, either model can perform well.

These distinct behaviours define the ideal operating environments for each model. The MBSTGNN architecture excels in modelling the interconnectedness of the urban fabric, demonstrating superior performance in contiguous residential areas where property values are strongly correlated with their surroundings. Its strength lies in interpreting the "where" of a property, making it robust across varied and data-sparse landscapes. In contrast, the LightGBM model achieves its lowest error rates in data-dense, homogeneous neighbourhoods where a high volume of transactions allows it to master intricate local pricing patterns from property features alone. Ultimately, the results indicate a complementary relationship: MBSTGNN provides a more robust and spatially aware baseline, while LightGBM excels at exploiting rich local data where spatial context is less critical.



(a) Neighbourhood-level spatial performance of the MBSTGNN model over the last 12 prediction windows. From left to right: (a) Mean Squared Error (MSE), (b) Mean Absolute Percentage Error (MAPE), and (c) number of transactions.



(b) Neighbourhood-level spatial performance of the LightGBM model over the last 12 prediction windows. From left to right: (a) Mean Squared Error (MSE), (b) Mean Absolute Percentage Error (MAPE), and (c) number of transactions.

Figure 6.3: Spatial evaluation of model performance across neighbourhoods in Rotterdam. Each row shows three spatial heatmaps for a model, revealing spatial variation in predictive accuracy and data coverage.

6.4. Case Study: Embedding Interpretation

To complement the quantitative evaluation, this section presents a qualitative case study analysis based on transaction-level embeddings learned by the proposed MBSTGNN model. These embeddings represent the final learned representation of each property transaction, capturing both spatial context and temporal dynamics. By visualising these high-dimensional embeddings using Uniform Manifold Approximation and Projection (UMAP), we aim to examine how the model internally organises transactions based on latent patterns in the data.

The goal of this analysis is twofold. First, it seeks to uncover whether the model's learned representations reflect meaningful spatial and temporal structures consistent with domain knowledge. Second, it investigates whether transactions in socio-economically similar neighbourhoods cluster together, even when not geographically adjacent. Through this case study, we evaluate the interpretability and realism of the MBSTGNN model's internal representations, providing insight into its capacity to capture subtle patterns in the urban housing market.

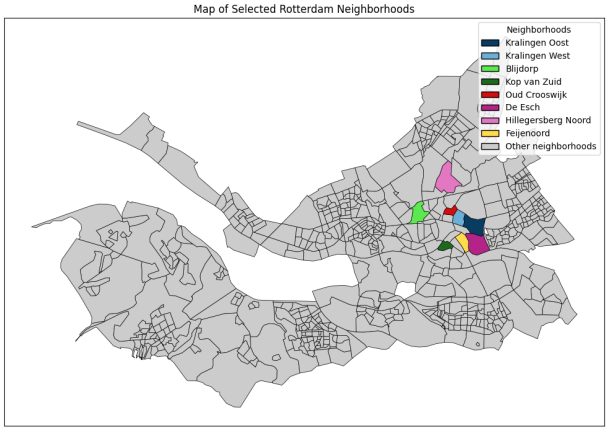


Figure 6.4: Map of Rotterdam with the neighbourhoods selected for the case study highlighted

6.4.1. Neighbourhood Selection

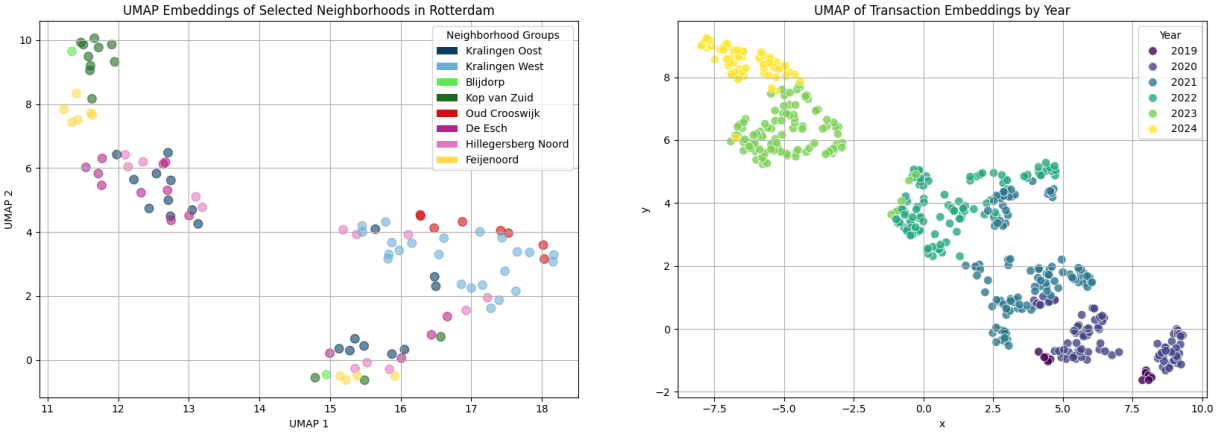
In this case study, we examine a carefully curated subset of nine neighbourhoods that reflect the socio-spatial diversity and historical complexity of Rotterdam. This selection includes both traditionally affluent residential zones and more transitional, lower-income districts with historically lower average property values. This enables a nuanced evaluation of how well the proposed model captures meaningful spatial and economic patterns. The selected neighbourhoods are shown in the map of Rotterdam in Figure 6.4

Kralingen Oost and Hillegersberg Noord represent some of the most prestigious and historically stable areas in Rotterdam. They are characterised by high property values, proximity to parks and lakes, and relatively homogeneous housing stock. These neighbourhoods are known for their strong residential desirability and stable market demand. They also benefit from good connectivity to the city centre, while maintaining a degree of urban insulation and exclusivity. De Esch, a planned development from the late 20th century, is physically isolated by infrastructure and water bodies. Kralingen West, though geographically adjacent to Kralingen Oost, is more mixed in character. It has higher-density housing and a more diverse population.

At the other end of the spectrum are Feijenoord and Oud Crooswijk, historically lower-income neighbourhoods with more heterogeneous housing conditions and complex social dynamics. These areas, especially Feijenoord, have undergone, or are undergoing, various stages of urban renewal, gentrification, and demographic transition. They often feature a mix of older, lower-quality housing stock alongside new developments, leading to fragmented market signals and evolving price structures.

Blijdorp is a centrally located pre-war neighbourhood known for its cohesive architectural style, mid- to upper-income demographics, and strong community identity. Kop van Zuid, in contrast, represents one of the city's most ambitious waterfront regeneration projects, built on former port land and now home to a mix of luxury apartments, cultural institutions, and high-rise developments. Despite its physical separation from more established residential areas, its economic trajectory makes it a compelling candidate for spatial

similarity analysis.



(a) UMAP Embeddings coloured by Neighbourhood

(b) UMAP Embeddings Coloured by Year

Figure 6.5: Visualisation of projected UMAP Embeddings for Spatial and Temporal Analysis

6.4.2. Spatial Analysis of Embeddings

The UMAP projection of the learned transaction embeddings in Figure 6.5a reveals meaningful clustering patterns that align closely with urban structure and socioeconomic similarities among the selected neighbourhoods. By visualising the transactions from the past two years and colouring them by neighbourhood, several distinct groupings emerge that offer insight into how the MBSTGNN model internalises spatial and contextual signals.

A prominent cluster is formed by De Esch, Hillegersberg Noord, and Kralingen Oost, despite their geographical separation. This grouping is consistent with domain expectations: all three are relatively affluent residential areas characterised by low-density housing, stable populations, and high-quality urban environments. The fact that the model embeds transactions from these neighbourhoods closely together suggests that it has learned to associate similar pricing dynamics and structural features across spatially disjoint yet socio-economically aligned contexts.

Another cluster includes Kop van Zuid, Blijdorp, and intermittently, Feijenoord. While this combination may appear unexpected geographically, it reflects recent transformations in urban development and property markets. Kop van Zuid and Blijdorp, although different in their historical trajectories, one being a regenerated post-industrial waterfront and the other a pre-war residential district, now share commonalities in market positioning and housing demand. The occasional overlap with Feijenoord suggests that the model recognises the signs of gentrification that Feijenoord has shown.

A third cluster is formed by Kralingen West and Oud Crooswijk, which are spatially adjacent neighbourhoods. Their proximity in the embedding space is likely influenced by this geographical closeness, given the model's use of spatial edges to propagate information between nearby regions. However, the clustering also reflects shared structural attributes such as mixed housing typologies, moderate transaction volumes, and similar stages of urban development. These areas are characterised by ongoing renewal efforts and transitional socio-economic dynamics, which the model appears to capture by embedding them similarly. This suggests that MB-STGNN learns both from direct spatial relationships and from deeper contextual patterns present across adjacent neighbourhoods.

Overall, the UMAP visualisation supports the claim that the MBSTGNN model effectively captures latent similarities and market patterns that are consistent with expert domain knowledge. It not only respects spatial continuity but also generalises across areas with similar socio-economic characteristics, even when they are physically distant. This qualitative clustering serves as further evidence of the model's ability to learn context-aware, semantically meaningful representations of property transactions.

6.4.3. Temporal Evolution of Embeddings

The UMAP projection of the transaction embeddings, when coloured by year, reveals a clear temporal structuring in the learned representation space. As seen in Figure 6.5b, transactions from the same year tend to form coherent clusters, and these clusters gradually shift across the embedding space from year to year. This smooth progression reflects the model's ability to encode temporal trends in the housing market, capturing how property values, transaction patterns, and neighbourhood dynamics evolve over time.

This behaviour is indicative of temporal continuity in the embedding space: rather than each year's transactions being isolated or erratically placed, the embeddings demonstrate a logical flow. For example, embeddings from 2020 transition smoothly into those from 2021, then 2022. This pattern also validates the model's temporal design. MBSTGNN captures temporal dynamics by training on evolving graph snapshots and incorporating neighbourhood embeddings of the previous timestep. This design allows the model to preserve a sense of temporal continuity and as a result, the learned transaction embeddings carry implicit temporal memory, enabling the model to reflect evolving market trends. This is especially important in the context of real estate, where property values are shaped not just by spatial factors but also by broader temporal dynamics.

Furthermore, this temporal clustering could be leveraged to detect market shifts or policy impacts over time, and even potential signs of gentrification or economic shocks, by observing changes in the position or shape of year-wise clusters. As can be seen, there a few points in 2023 that cluster with the embeddings of 2022, while the remaining form a cluster after visible gap. This could be representative of the change in embeddings due to the downward turn of the average price trends as observed in Figure 6.2. If certain years diverge strongly from previous ones, it may indicate a structural market change worth investigating further.

6.5. Discussion

The results of this study highlight the value of explicitly modelling spatial and temporal structure in residential property valuation. The proposed MBSTGNN architecture consistently outperforms non-graph baselines across multiple evaluation metrics, demonstrating that the inclusion of hierarchical spatial context and time-aware embeddings allows the model to capture underlying market regularities more effectively. In the rest of this section we discuss the strengths of this model and validate our architectural decision with our empirical results in Subsection 6.5.1. We also conceptually compare our model to prior work. The limitations of our model and this study are highlighted in Subsection 6.5.2. Finally, we discuss the real-world relevance of this approach and its implications for the industry in Subsection 6.5.3.

6.5.1. Strengths and Contributions of the Model

The empirical and representational results of this study highlight the key strengths of the MBSTGNN architecture and its advancement over prior approaches. Central to the model's effectiveness is its hierarchical design, which enables bidirectional message passing between micro- and macro-level graphs. This structure allows the model to jointly reason over individual transaction patterns and evolving neighbourhood-level dynamics. Unlike prior models such as MugRep or ReGram, which rely on fixed or unidirectional aggregation, MBSTGNN supports dynamic two-way interaction between spatial scales. This design proves particularly important in real estate markets, where micro-level changes, such as a surge in high-end sales, can influence neighbourhood perception, and where regional developments often cascade down to affect local pricing. The results of the ablation study in Appendix B.3 indicate that this bidirectional information flow performs better unidirectional variants.

Temporally, the model employs a gated memory mechanism at the macro level to retain historical context across snapshots. In contrast to sequence-based models like ST-RAP, which use a GRU at the transaction-level to capture time-dependent effects, MBSTGNN opts for a simpler temporal update mechanism that avoids the added complexity and overfitting risks of full recurrent architectures. This design choice is validated by the UMAP visualisations of transaction embeddings over time, which show smooth transitions across years, suggesting that the model captures structural temporal evolution without requiring dense sequential supervision.

Moreover, the use of dynamic graph snapshots ensures that the model remains responsive to changes in the market. By constructing the graph at each timestep based on the most recent data, the model reflects the evolving spatial and economic conditions of the housing market. This design addresses long-standing

limitations in the literature, such as static topologies or weak temporal integration, and makes the model adaptable to non-stationary urban processes.

Another notable strength is the model's ability to learn spatial embeddings that reflect socio-economic similarity, not just geographic proximity. The emergence of coherent neighbourhood clusters in the learned embedding space demonstrates that the proposed model identifies latent urban structure. In contrast, LightGBM and other tabular baselines treat transactions as independent samples, lacking the relational capacity to uncover such patterns. This limitation becomes especially evident in sparse or emerging neighbourhoods, where relational reasoning provides critical signal in the absence of dense historical data.

Finally, although MBSTGNN is not fully interpretable in the traditional sense, it offers a form of relational explainability through its learned embeddings. These representations enable domain experts to explore patterns of spatial convergence, identify neighbourhoods undergoing transformation, and trace how regional trends shape local valuations. Such insights are difficult to obtain from models like LightGBM, which rely on scalar feature importances and cannot represent structural relationships.

Overall, MBSTGNN demonstrates that combining structured, inter-level graph interactions with lightweight temporal updates can lead to more accurate and context-aware predictions than both traditional machine learning baselines and more complex temporal GNNs. Its modularity, scalability, and spatial awareness make it a compelling approach for modelling dynamic, hierarchical systems like urban housing markets.

6.5.2. Limitations

While MBSTGNN presents several methodological advancements, it also has notable limitations. The most significant is its lack of explicit sequence modelling. While the gated macro memory captures gradual change, it may not detect abrupt regime shifts, such as sudden policy interventions, interest rate changes, or pandemic-driven market disruptions. In such scenarios, models with recurrent units or temporal attention mechanisms may outperform.

The model also shows limited sensitivity to rare events. Its use of mean aggregation in both micro-level encoding and cross-level interaction, while robust to noise, tends to smooth over atypical transactions. This may result in underestimation of high-end properties in transitioning neighbourhoods or speculative hotspots. It also causes the model to fail in anomalous regions with unusual spatial structures, distinctive property types, or highly seasonal dynamics that deviate from dominant urban patterns.

Further, while the hierarchical structure supports multi-scale learning, it still relies on predefined neighbourhood boundaries and feature aggregation at the macro level. This limits resolution in densely built-up areas where intra-neighbourhood variation is high. In such settings, the model may miss fine-grained price drivers that do not align with administrative units.

A further limitation of this study concerns data access and reproducibility. The main experiments and analyses were conducted on a proprietary transaction dataset provided through an industry collaboration. While this partnership enabled access to rich, high-quality records essential for model development and evaluation, it constrains direct reproducibility for the broader research community. To mitigate this limitation, we constructed and released a synthetic dataset using the Private Probabilistic Generative Model (Private-PGM) [74], which preserves the statistical properties of the original transactions while ensuring privacy. Although this synthetic dataset allows independent experimentation and benchmarking, it cannot fully capture all nuances of the proprietary data. Consequently, future work should aim to validate the model on additional public datasets to further strengthen reproducibility and generalisability.

Finally, the model currently lacks proper online or continual learning capabilities. In production environments, this requirement for periodic retraining limits responsiveness and scalability. While the rolling window framework used in this study simulates streaming updates, it does not support continual adaptation and require long retraining times per window. This limitation becomes especially significant when considering the geographic scope of the model. The present study focuses exclusively on the COROP region of Rijnmond, which allowed for controlled experimentation and interpretability. However, scaling the model to a national level, such as the entirety of the Netherlands, would pose serious computational challenges due to the increased volume and structural heterogeneity of transactions. Without online learning or model compression techniques, such large-scale deployment may become impractical. Initial experiments with an online learning paradigm shows a large gap in predictive performance. We refer to Appendix B.4 for these results.

6.5.3. Implications and Application

The findings presented in this work have both methodological and practical implications. From a methodological standpoint, the results support the use of modular hierarchical GNNs for valuation tasks in spatially and temporally structured domains. The approach generalises across neighbourhoods, adapts to temporal shifts, and captures interactions between local and regional patterns. These features are essential in real estate but under-explored in previous models.

Practically, MBSTGNN offers tools for policy design, zoning analysis, and urban planning. The learned embeddings can be used not only to predict prices but also to identify neighbourhood clusters, track signs of gentrification, or flag market anomalies. For instance, sudden shifts in embedding trajectories over time may serve as early indicators of redevelopment or displacement. Similarly, spatial proximity in the learned space could inform housing equity studies or targeted investment strategies.

The model is also scalable and transferable. Because it relies on inductive components and supports subgraph-based training, it can be deployed across different cities or regions with similar data structures. Its modular design allows for future extension, for example, by integrating mobility patterns, natural language property descriptions, or amenity-based graphs.

In general, MBSTGNN contributes a novel and effective approach to real estate valuation that balances prediction accuracy with structural insight, making it valuable not only to model developers but also to urban researchers and decision makers.

7

Conclusion

This chapter concludes our research and this thesis while providing direction for the future. Section 7.1 provides a brief overview of the thesis. Section 7.2 provides an answer to our research questions and presents the key findings of our thesis. Finally, we outline possible future directions in Section 7.3

7.1. Thesis Summary

This thesis presented a hierarchical spatio-temporal graph learning framework for residential property valuation, motivated by the need to better capture the complex spatial and temporal dependencies that shape real estate markets.

In Chapter 1, we introduced the research problem and outlined the limitations of conventional modelling approaches, particularly their inability to account for spatial heterogeneity, local relational effects, and temporal evolution. Chapter 2 introduced key concepts in property valuation, spatial modelling, and graph neural networks. The literature survey in Chapter 3 revealed a gap in existing methods, which either ignore spatial dependencies altogether or rely on fixed and shallow representations of space and time. While recent graph-based approaches attempt to address some of these issues, they often assume static topologies or unidirectional hierarchies, limiting their expressiveness.

In Chapter 4, we proposed a novel model, MBSTGNN, built on a hierarchical spatio-temporal graph structure. Each time step is represented by a two-level graph consisting of transaction nodes and neighbourhood nodes, connected via cross-level edges. The micro-level encoder aggregates information from nearby transactions, the macro-level encoder captures inter-neighbourhood dynamics with temporal memory, and the cross-level module enables bidirectional flow between scales. We also described the rolling window training procedure and subgraph-based mini-batching used to scale learning to large, dynamic graphs.

Chapter 5 described the experimental setup and baseline comparisons. In Chapter 6 we discuss the results of these experiments. Quantitative results showed that MBSTGNN consistently outperformed LightGBM and other graph-based baselines across various metrics, especially in regions with limited training data. The model proved effective in capturing local transaction patterns as well as broader structural changes over time. We analysed the structure of the learned embeddings. Through UMAP visualisation and neighbourhood-level case studies, we showed that the model captures socio-spatial similarities, including clusters of demographically and economically similar regions, even when these are not geographically adjacent. Temporal analysis further revealed that the learned embeddings evolve in a coherent manner, reflecting gradual structural changes in the housing market.

7.2. Key Findings

Through extensive experiments and qualitative case studies in Chapter 6, we examined how the model performs relative to established baselines and whether it captures domain-relevant spatial structure. The findings provide concrete answers to the two central research questions defined in Chapter 1.

> RQ1: How does a hierarchical spatio-temporal graph neural network compare to traditional and graph-based baselines in accurately predicting residential property prices across varying property types, regions, and time periods?

The results demonstrate that MBSTGNN consistently outperforms both traditional models such as LightGBM and graph-based baselines. This advantage is most pronounced in heterogeneous or low-data regions, where relational information provides valuable signal. The model shows strong generalisation across diverse property types and performs robustly over time, with error rates remaining stable even during volatile market periods. Unlike baselines that either ignore structure or rely on fixed graph assumptions, MBSTGNN dynamically constructs context-aware representations that integrate spatial, temporal, and hierarchical information. This design leads to improved predictive accuracy and greater resilience in complex urban environments.

> RQ2: To what extent does the proposed hierarchical graph structure capture meaningful spatial dependencies and domain-consistent neighbourhood relationships in real estate markets?

The case study analysis reveals that the learned transaction embeddings reflect meaningful socio-spatial organisation. Neighbourhoods with similar socio-economic profiles, housing stock, or developmental trajectories are frequently clustered together in the learned embedding space. Furthermore, the temporal embedding trajectories show coherent year-to-year evolution, suggesting that the model preserves market continuity while adapting to gradual structural changes. These findings support the claim that the hierarchical graph structure captures more than spatial proximity.

Overall, this study validates the use of hierarchical spatio-temporal graph learning as a powerful approach for capturing the complex dependencies underlying real estate markets. The architecture not only improves

prediction performance but also enables interpretable, structured insights into urban housing systems.

7.3. Future Work

While this study demonstrates the effectiveness of a hierarchical spatio-temporal graph neural network for residential property valuation in Rotterdam, there remain several promising directions for extending this work. These future avenues span both methodological advancements and broader applications across domains.

One natural direction is to apply the proposed framework beyond the real estate domain. The model's core design, integrating micro-level interactions with evolving macro-level context, is not specific to housing markets and can be adapted to other spatio-temporal problems. For instance, in epidemiology, patient-level interactions (micro) and regional health infrastructure (macro) could be modelled similarly to capture disease spread. In supply chains, shipment-level events and global trade hubs follow a comparable structure. Environmental monitoring, financial risk analysis, and urban mobility forecasting are also domains where hierarchical spatio-temporal dependencies play a key role. Extending the model to these contexts would test its generalisability and position it as a general-purpose framework for structured spatio-temporal learning.

A second line of future work involves augmenting the model with mechanisms to capture sudden external events or policy shifts. The current approach is best suited for smooth temporal evolution, where patterns unfold gradually. However, real-world housing markets are often affected by abrupt shocks such as interest rate changes, new housing regulations, or environmental disruptions. Incorporating exogenous signals into the macro node features, or introducing event-aware modules, could allow the model to respond to these dynamics more effectively. This would improve the model's capacity for counterfactual reasoning and make it more suitable for policy impact analysis.

Third, the lack of online or continual learning is a key limitation in terms of operational deployment. Currently, model updates require retraining on entire windows for new batches of data, which may be computationally prohibitive at larger scales or in fast-moving markets. To address this, future work could explore online GNN variants or hybrid architectures that support incremental updates. Temporal attention mechanisms or memory-based modules could also be integrated to prioritise recent or structurally significant transactions. These enhancements would enable the model to function in real-time settings, such as automated valuation systems used by financial institutions or municipal agencies.

Finally, improving the interpretability and robustness of the model is another important direction. While graph-based models offer a structural advantage, they can be difficult to interpret compared to traditional tabular models. Future extensions could incorporate techniques such as GNNExplainer, attention visualisation, or neighbourhood influence attribution to make the model's decision-making process more transparent. Furthermore, integrating uncertainty quantification, for instance via Bayesian GNNs or ensemble approaches, would allow the model to express confidence in its predictions. This is particularly relevant in high-stakes applications like mortgage lending, taxation, or urban planning, where predictive accuracy must be balanced with accountability.

7.4. Disclosure

We acknowledge the use of AI-assisted tools for refinement of parts of this thesis in accordance with TU Delft's publication guidelines[1]. The cover image was generated by Chat-GPT.

[1] https://www.tudelft.nl/library/actuele-themas/open-publishing/about/policies

References

[1] Nils Kok et al. “Big Data in Real Estate? *From Manual Appraisal to Automated Valuation*”. en. In: *The Journal of Portfolio Management* 43.6 (Sept. 2017), pp. 202–211. DOI: 10.3905/jpm.2017.43.6.202. URL: http://pm-research.com/lookup/doi/10.3905/jpm.2017.43.6.202 (visited on 11/22/2024).

[2] Lviv Polytechnic National University et al. “Application of Machine Learning Methods for Forecasting Real Estate Value”. en. In: *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Systems. Volume I: Machine Learning Workshop*. CoLInS, Apr. 2024. DOI: 10.31110/COLINS/2024-1/007. URL: https://ceur-ws.org/Vol-3664/paper7.pdf (visited on 11/22/2024).

[3] *International valuation standards (IVS): effective 31 January 2022*. en. OCLC: 1443478842. London: International Valuation Standards Council, 2022.

[4] Zhenguo Lin et al. “Illiquidity and Pricing Biases in the Real Estate Market”. In: *Real Estate Economics* 35.3 (2007), pp. 291–330. DOI: 10.1111/j.1540-6229.2007.00191.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6229.2007.00191.x.

[5] Jorge Chica-Olmo et al. “Estimation of Housing Price Variations Using Spatio-Temporal Data”. In: *Sustainability* 11.6 (2019), p. 1551. DOI: 10.3390/su11061551. URL: https://www.mdpi.com/2071-1050/11/6/1551.

[6] Lahiru Neelawala. “Asymmetric Information Between Buyers and Sellers in the Residential Property Market”. PhD thesis. Queensland University of Technology, 2014.

[7] G. Marcato et al. “Asymmetric Patterns of Demand-Supply Mismatch in Real Estate”. In: *Journal of Real Estate Finance and Economics* (2022).

[8] Appraisal Institute. *The Appraisal of Real Estate*. 15th ed. Appraisal Institute, 2020.

[9] Sherwin Rosen. “Hedonic prices and implicit markets: product differentiation in pure competition”. In: *Journal of political economy* 82.1 (1974), pp. 34–55.

[10] Marc K. Francke et al. “The Hierarchical Trend Model for Property Valuation and Local Price Indices”. en. In: *The Journal of Real Estate Finance and Economics* 28.2 (Mar. 2004), pp. 179–208. DOI: 10.1023/B:REAL.0000011153.04496.42. URL: https://doi.org/10.1023/B:REAL.0000011153.04496.42 (visited on 11/22/2024).

[11] B. Glumac et al. “AVMs: Their Role, Advantages and Limitations”. In: *Journal of Property Investment & Finance* (2021).

[12] O. Demirci. *Machine Learning in Property Valuation*. University of Westminster Research Paper. 2021. URL: https://westminsterresearch.westminster.ac.uk/.

[13] Guolin Ke et al. “Lightgbm: A highly efficient gradient boosting decision tree”. In: *Advances in neural information processing systems* 30 (2017).

[14] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232.

[15] John M Clapp. “A semiparametric method for estimating local house price indices”. In: *Real Estate Economics* 32.1 (2004), pp. 127–160. DOI: 10.1111/j.1080-8620.2004.00086.x.

[16] Ho C. Chin et al. “A critical review of literature on the hedonic price model”. In: *International Journal for Housing Science and Its Applications* 27.2 (2003), pp. 145–165.

[17] J. Dubé et al. “A Spatio□Temporal Measure of Spatial Dependence: An Example Using Real Estate Data”. In: *Papers in Regional Science* 92.2 (2013), pp. 395–416.

[18] Elli Pagourtzi et al. “Real estate appraisal: a review of valuation methods”. In: *Journal of Property Investment & Finance* 21.4 (Jan. 2003). Publisher: MCB UP Ltd, pp. 383–401. DOI: 10.1108/14635780310483656. URL: https://doi.org/10.1108/14635780310483656 (visited on 11/22/2024).

[19] Elaine Worzala et al. “An exploratory analysis of the use of artificial neural networks in mass appraisal”. In: *Journal of Real Estate Research* 10.2 (1995), pp. 185–201.

[20] Waldo R. Tobler. “A Computer Movie Simulating Urban Growth in the Detroit Region”. In: *Economic Geography* 46.sup1 (1970), pp. 234–240. DOI: 10.2307/143141. URL: https://www.tandfonline.com/doi/abs/10.2307/143141.

[21] Andrew D. Cliff et al. *Spatial Autocorrelation*. London: Pion, 1973.

[22] Edward L. Glaeser et al. “Housing Supply and Housing Bubbles”. In: *Journal of Urban Economics* 64.2 (2008), pp. 198–217. DOI: 10.1016/j.jue.2008.07.007. URL: https://www.sciencedirect.com/science/article/pii/S0094119008000648.

[23] R. K. Pace et al. “A Method for Spatial–Temporal Forecasting with an Application to Real Estate Prices”. In: *International Journal of Forecasting* 16.2 (2000), pp. 229–246.

[24] R. F. Y. Lin et al. “The Spatial Neural Network Model with Disruptive Technology for Property Appraisal in the Real Estate Industry”. In: *Technological Forecasting and Social Change* 168 (2021), p. 120764.

[25] P. Bidanset et al. “Survey on the Use of AVMs in Government Assessment Offices”. In: *Journal of Property Tax Assessment & Administration* (2022).

[26] Thomas N Kipf et al. “Semi-supervised classification with graph convolutional networks”. In: *International Conference on Learning Representations (ICLR)*. 2017.

[27] Will Hamilton et al. “Inductive representation learning on large graphs”. In: *Advances in neural information processing systems* 30 (2017).

[28] David I Shuman et al. “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains”. In: *IEEE Signal Processing Magazine* 30.3 (2013), pp. 83–98.

[29] Michael M Bronstein et al. “Geometric deep learning: Going beyond Euclidean data”. In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42.

[30] Seyed Mehran Kazemi et al. “Representation Learning for Dynamic Graphs: A Survey”. en. In: (Mar. 2020).

[31] Yanping Zheng et al. *A survey of dynamic graph neural networks*. arXiv:2404.18211 [cs]. Apr. 2024. DOI: 10.48550/arXiv.2404.18211. URL: http://arxiv.org/abs/2404.18211 (visited on 01/15/2025).

[32] Yi Wang. “Advances in spatiotemporal graph neural network prediction research”. In: *International Journal of Digital Earth* 16.1 (Oct. 2023). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/17538947.2023.22 pp. 2034–2066. DOI: 10.1080/17538947.2023.2220610. URL: https://doi.org/10.1080/17538947.2023.2220610 (visited on 01/20/2025).

[33] Faraz Moghimi et al. “Rethinking real estate pricing with transformer graph neural networks (t-gnn)”. In: *2023 International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2023, pp. 1405–1411.

[34] Sarkar Snigdha Sarathi Das et al. “Boosting house price predictions using geo-spatial network embedding”. In: *Data Mining and Knowledge Discovery* 35 (2021), pp. 2221–2250.

[35] H. Lee et al. “ST-RAP: A Spatio-Temporal Framework for Real Estate Appraisal”. In: 2023, pp. 4053–4058. DOI: 10.1145/3583780.3615168.

[36] Weijia Zhang et al. *MugRep A Multi-Task Hierarchical Graph Representation Learning Framework for Real Estate Appraisal*. arXiv:2107.05180. Aug. 2021. URL: http://arxiv.org/abs/2107.05180 (visited on 11/21/2024).

[37] Jie Zhou et al. “Graph neural networks: A review of methods and applications”. In: *AI Open* 1 (Jan. 2020), pp. 57–81. DOI: 10.1016/j.aiopen.2021.01.001. URL: https://www.sciencedirect.com/science/article/pii/S2666651021000012 (visited on 01/22/2025).

[38] Zonghan Wu et al. “A Comprehensive Survey on Graph Neural Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (Jan. 2021). Conference Name: IEEE Transactions on Neural Networks and Learning Systems, pp. 4–24. DOI: 10.1109/TNNLS.2020.2978386. URL: https://ieeexplore.ieee.org/abstract/document/9046288?casa_token=et3PaClwh6IAAAAA:QohrOVBghNYXePOT9gb6C7jyCij9JKp35NO7BkkzmxrH7NXturlSvfseHf27gP5RFX0xy6Ws (visited on 01/22/2025).

[39] Elvin Isufi et al. “Graph filters for signal processing and machine learning on graphs”. In: *IEEE Transactions on Signal Processing* 72 (2024), pp. 4745–4781.

[40] Petar Veličković et al. “Graph attention networks”. In: *arXiv preprint arXiv:1710.10903* (2017).

[41] Hongyang Gao et al. “Graph u-nets”. In: *international conference on machine learning*. PMLR. 2019, pp. 2083–2092.

[42] Zhitao Ying et al. “Hierarchical Graph Representation Learning with Differentiable Pooling”. In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/hash/e77dbaf6759253c7c6d0efc5690369c7-Abstract.html (visited on 01/10/2025).

[43] Chuan Shi et al. “A survey of heterogeneous information network analysis”. In: *IEEE Transactions on Knowledge and Data Engineering* 29.1 (Jan. 2017), pp. 17–37. DOI: 10.1109/TKDE.2016.2598561. URL: https://ieeexplore.ieee.org/document/7536145 (visited on 06/24/2025).

[44] Chuxu Zhang et al. “Heterogeneous Graph Neural Network”. en. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Anchorage AK USA: ACM, July 2019, pp. 793–803. DOI: 10.1145/3292500.3330961. URL: https://dl.acm.org/doi/10.1145/3292500.3330961 (visited on 12/05/2024).

[45] Yizhou Sun et al. “PathSim: Meta path-based top-k similarity search in heterogeneous information networks”. In: *Proceedings of the VLDB Endowment*. Vol. 4. 11. 2011, pp. 992–1003.

[46] Sepp Hochreiter et al. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[47] Junyoung Chung et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling”. In: *arXiv preprint arXiv:1412.3555* (2014).

[48] Yu Liu et al. “Uncovering patterns of inter-urban trip and spatial interaction from social media check-in data”. In: *PloS one* 9.1 (2014), e86026.

[49] Martin J Bailey et al. “A regression method for real estate price index construction”. In: *Journal of the American Statistical Association* 58.304 (1963), pp. 933–942.

[50] Bo Huang et al. “Geographically and temporally weighted regression for modeling spatio-temporal variation in house prices”. In: *International Journal of Geographical Information Science* 24.3 (Mar. 2010). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/13658810802672469, pp. 383–401. DOI: 10.1080/13658810802672469. URL: https://doi.org/10.1080/13658810802672469 (visited on 11/22/2024).

[51] T. F. Sing et al. “Boosted Tree Ensembles for AI-based AVMs”. In: *Journal of Real Estate Finance and Economics* (2022).

[52] Peyman Jafary et al. “Automating property valuation at the macro scale of suburban level: A multi-step method based on spatial imputation techniques, machine learning and deep learning”. In: *Habitat International* 148 (2024), p. 103075.

[53] L Chen et al. “Measuring impacts of urban environmental elements on housing prices based on multisource data—a case study of Shanghai, China”. In: *ISPRS International Journal of Geo-Information* 9.2 (2020), p. 106. DOI: 10.3390/ijgi9020106. URL: https://www.mdpi.com/2220-9964/9/2/106.

[54] Tianqi Li et al. “Constructing a highly accurate price prediction model in real estate investment using LightGBM”. In: *2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR)*. Sept. 2021, pp. 273–276. DOI: 10.1109/MIPR51284.2021.00051. URL: https://ieeexplore.ieee.org/document/9565563/ (visited on 05/08/2025).

[55] Evert Guliker et al. “Spatial Determinants of Real Estate Appraisals in The Netherlands: A Machine Learning Approach”. en. In: *ISPRS International Journal of Geo-Information* 11.2 (Feb. 2022). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 125. DOI: 10.3390/ijgi11020125. URL: https://www.mdpi.com/2220-9964/11/2/125 (visited on 05/10/2025).

[56] M.A. Shbool et al. “Real estate decision-making: precision in price prediction through advanced machine learning algorithms”. In: *International Journal of Housing Markets and Analysis* (2025). DOI: 10.1108/IJHMA-01-2025-0004. URL: https://www.emerald.com/insight/content/doi/10.1108/IJHMA-01-2025-0004/full/html.

[57] A. Ouchlif et al. “Predicting Property Prices with Machine Learning: A Case Study in the Moroccan Real Estate Market”. In: *Handbook of Research on Applications of AI and Machine Learning*. IGI Global, 2025. URL: https://www.igi-global.com/chapter/predicting-property-prices-with-machine-learning/368410.

[58] D. Balakrishnan et al. “Flask-Powered Machine Learning Model for Accurate Real Estate Price Prediction”. In: *2024 Global Conference on Computing, Power and Communication Technologies (GlobConPT)*. IEEE. 2024. DOI: 10.1109/GlobConPT58049.2024.10862259. URL: https://ieeexplore.ieee.org/abstract/document/10862259/.

[59] Isyaku Ibrahim et al. “Application of Computer Assisted Mass Appraisal (CAMA) in Real Estate Profession in Nigeria: A Review of Valuation Modern Techniques”. In: *ResearchGate* (2025). URL: https://www.researchgate.net/publication/389561660_APPLICATION_OF_COMPUTER_ASSISTED_MASS_APPRAISAL_CAMA_IN_REAL_ESTATE_PROFESSION_IN_NIGERIA_A_REVIEW_OF_VALUATION_MODERN_TECHNIQUES.

[60] AU Khan et al. “Comparative Evaluation of Machine Learning and Deep Learning Models for Real Estate Price Prediction”. In: *ResearchGate* (2025). URL: https://www.researchgate.net/publication/389559711_Comparative_Evaluation_of_Machine_Learning_and_Deep_Learning_Models_for_Real_Estate_Price_Prediction.

[61] MI Nor et al. “Impact of Living and Environmental Factors on Rental Prices in Baidoa’s Climate-Affected Real Estate Market: A Hybrid Approach Using Hedonic Regression and Neural Networks”. In: *Frontiers in Sustainable Cities* (2025). DOI: 10.3389/frsc.2025.1511761. URL: https://www.frontiersin.org/journals/sustainable-cities/articles/10.3389/frsc.2025.1511761/full.

[62] Chih-Chia Li et al. *Look Around! A Neighbor Relation Graph Learning Framework for Real Estate Appraisal*. arXiv:2212.12190 version: 1. Dec. 2022. URL: http://arxiv.org/abs/2212.12190 (visited on 11/18/2024).

[63] Junhyun Lee et al. “Self-attention graph pooling”. In: *International conference on machine learning*. pmlr. 2019, pp. 3734–3743.

[64] Xiao Wang et al. “Heterogeneous graph attention network”. In: *The world wide web conference*. 2019, pp. 2022–2032.

[65] Xinyu Fu et al. “Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding”. In: *Proceedings of the web conference 2020*. 2020, pp. 2331–2341.

[66] Xiao Wang et al. “Self-supervised heterogeneous graph neural network with co-contrastive learning”. In: *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 2021, pp. 1726–1736.

[67] Ziniu Hu et al. “Heterogeneous graph transformer”. In: *Proceedings of the web conference 2020*. 2020, pp. 2704–2710.

[68] Bowen Jin et al. “Heterformer: Transformer-based deep node representation learning on heterogeneous text-rich networks”. In: *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*. 2023, pp. 1020–1031.

[69] Yujie Fan et al. *Heterogeneous Temporal Graph Neural Network*. arXiv:2110.13889 [cs]. Oct. 2021. DOI: 10.48550/arXiv.2110.13889. URL: http://arxiv.org/abs/2110.13889 (visited on 05/12/2025).

[70] Zhiqiang Zhong et al. “Hierarchical message-passing graph neural networks”. In: *Data Mining and Knowledge Discovery* 37.1 (2023), pp. 381–408.

[71] Minbo Ma et al. “HiSTGNN: Hierarchical spatio-temporal graph neural network for weather forecasting”. In: *Information Sciences* 648 (Nov. 2023), p. 119580. DOI: 10.1016/j.ins.2023.119580. URL: https://www.sciencedirect.com/science/article/pii/S0020025523011659 (visited on 03/06/2025).

[72] Yu Liu et al. “Uncovering patterns of inter-urban trip and spatial interaction from social media check-in data”. In: *PLoS ONE* 9.1 (Jan. 2014). arXiv:1310.0282 [cs], e86026. DOI: 10.1371/journal.pone.0086026. URL: http://arxiv.org/abs/1310.0282 (visited on 05/12/2025).

[73] Aldo Pareja et al. “Evolvegcn: Evolving graph convolutional networks for dynamic graphs”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 04. 2020, pp. 5363–5370.

[74] Ryan McKenna et al. “Graphical-model based estimation and inference for differential privacy”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 4435–4444.

[75] Luc Anselin. “Spatial econometrics”. In: *Handbook of spatial analysis in the social sciences* (2022), pp. 101–122.

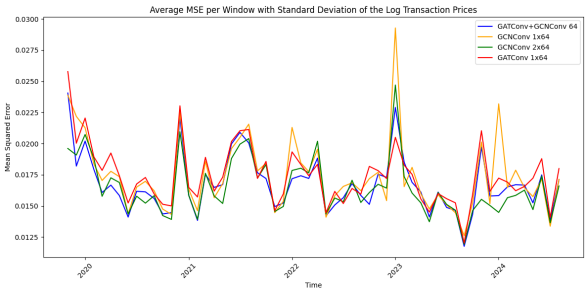
[76] Francis X Diebold et al. “Comparing predictive accuracy”. In: *Journal of Business & economic statistics* 20.1 (2002), pp. 134–144.

[77] Frank Wilcoxon. “Individual comparisons by ranking methods”. In: *Biometrics bulletin* 1.6 (1945), pp. 80–83.

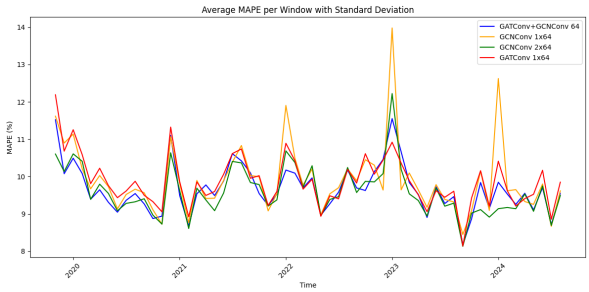
[78] Ian Goodfellow et al. *Deep Learning*. MIT Press, 2016.

A Model Structure

A.1. Macro Encoder Structure



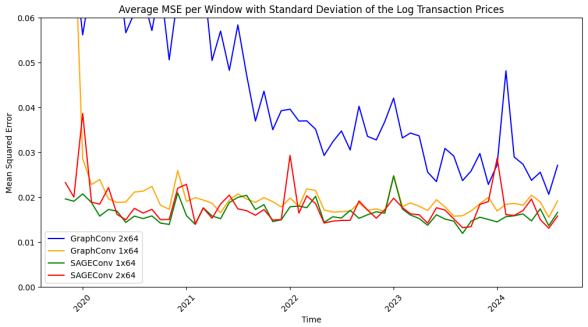
(a) Comparison of the MSE of Variations of the Macro Encoder



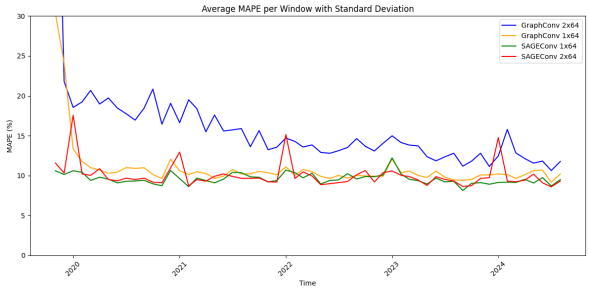
(b) Comparison of the MAPE of Variations of the Macro Encoder

Figure A.1: Comparison of performance of Variations of the Macro Encoder

A.2. Micro Encoder Structure



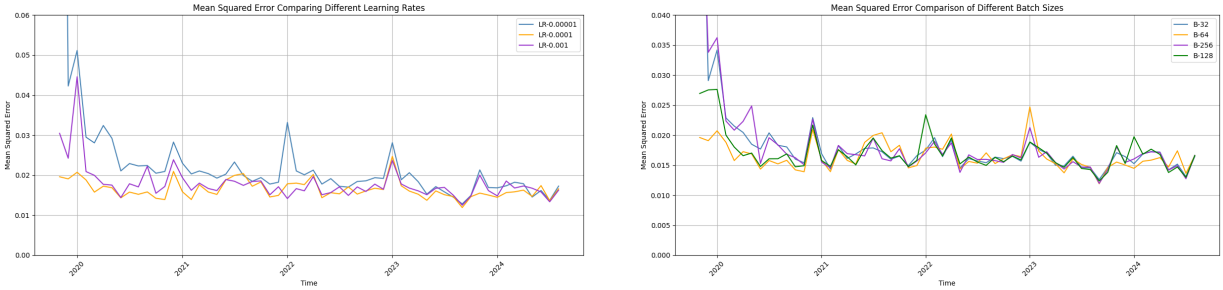
(a) Comparison of the MSE of Variations of the Micro Encoder



(b) Comparison of the MAPE of Variations of the Micro Encoder

Figure A.2: Comparison of performance of Variations of the Micro Encoder

A.3. Hyperparameter Tuning



(a) Comparison of Different Learning Rates

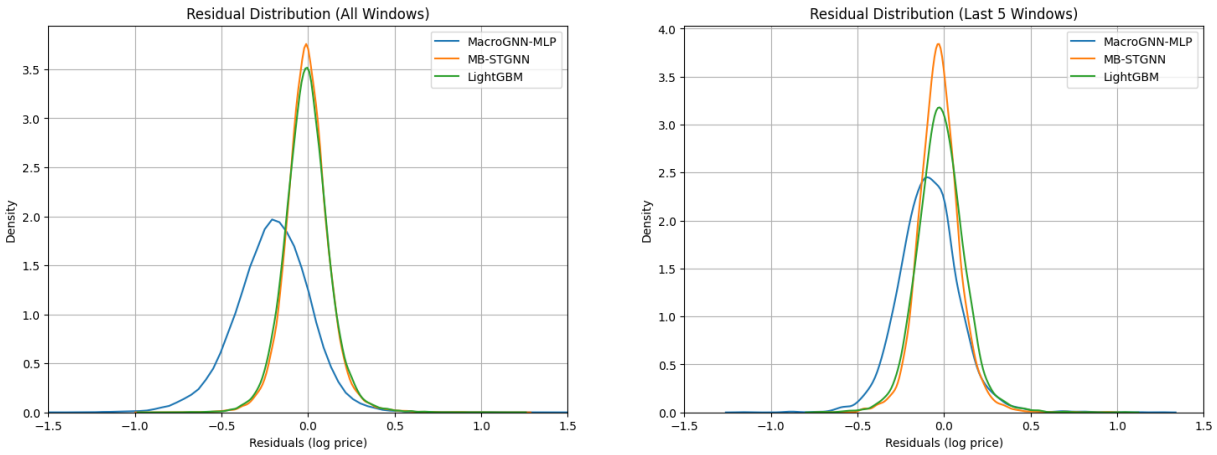
(b) Comparison of Different Batch Sizes

Figure A.3: Comparison of performance of MBSTGNN when using different hyperparameters

B

Additional Results

B.1. Residual Distribution



(a) Residual Distribution across all windows

(b) Residual Distribution across the last 5 windows

Figure B.1: Residual Distribution across different windows

We analyse the distribution of prediction residuals to gain more insight into the error characteristics such as the bias and variance of the models. Figure B.1 presents the distribution of residuals calculated across all windows and across the 5 most recent windows. The residuals are defined as $\hat{y}_i - y_i$, or the difference between the predicted log transaction price and the actual log transaction price.

The MBSTGNN model exhibits the tightest and most symmetric residual distribution, with low variance and a pronounced peak around zero. This suggests strong calibration and high prediction stability, especially in regions well-supported by training data. The LightGBM model shows wider residual spread, indicating slightly higher variance. The macro-only baseline again performs worst, with a noticeably broader and flatter distribution. The model's lack of transaction-level granularity leads to underfitting and poor localization of property-level effects. All the models show a slight bias with a tendency to undervalue the properties. This bias is most pronounced in the Macro-level baseline and least in the MBSTGNN model.

B.2. Additional Results: Spatial Analysis

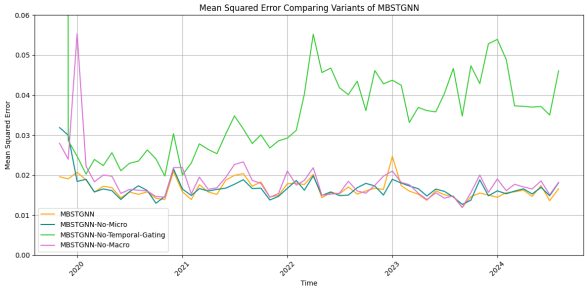
Table B.1: Top 15 neighborhoods with the highest MAPE for MBSTGNN and LightGBM. Common neighborhoods are shaded.

Rank	Neighborhood (MBSTGNN)	MAPE (%)	Neighborhood (LightGBM)	MAPE (%)
1	Wibautbuurt	84.45	Wibautbuurt	67.21
2	Rhoon-Zuidoost	72.43	Rhoon-Zuidoost	55.52
3	Morgen Stont	50.39	Beemster/Purmerhoek	48.45
4	Strand en Duin	41.77	Morgen Stont	40.43
5	Vijfsluizen	34.52	Verenambacht	39.49
6	Zeeheldenbuurt-oost	33.26	Spaanse Polder	37.30
7	Bedrijvenpark Noord-West	31.42	BT Vaanpark 4	33.13
8	Staatsliedenbuurt	31.37	De Baronie	33.07
9	Hoeksekade	30.01	Vijfsluizen	31.70
10	Schieveste	25.86	Bachplein	26.36
11	Krekelveen	24.41	Gaarden	25.61
12	Spaanse Polder	22.09	Nolensbuurt	23.04
13	De Hoek-4	21.83	Bedrijvenpark Noord-West	22.57
14	’s-Gravenpark	21.79	Egel- en Salamanderveen	21.70
15	Nolensbuurt	21.74	Schenkel-Noordoost	21.45

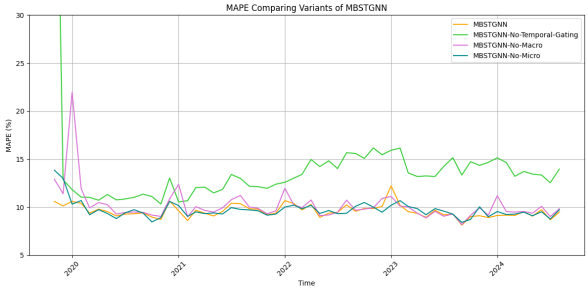
Table B.2: Top 15 neighborhoods with the lowest nonzero MAPE for MBSTGNN and LightGBM. Common neighborhoods are shaded.

Rank	Neighborhood (MBSTGNN)	MAPE (%)	Neighborhood (LightGBM)	MAPE (%)
1	Baarsveen	0.0121	Brielle	0.2077
2	Gouden Griffelbuurt	1.2065	Kikkerveen	0.4109
3	Eilandenbuurt	1.3948	Gebouwenbuurt	0.9279
4	Zuidhoek	1.7246	Zuidhoek	2.5594
5	Vogelbuurt	1.7608	Sterrenkwartier-Oost	1.4325
6	Struisenburg	1.8785	Struisenburg	2.2533
7	Eilandenbuurt	2.0155	Witte Dorp	1.6504
8	Oosterflank	2.0930	Heijplaat	2.1645
9	Heliushaven	2.2984	Stadsdriehoek	2.1976
10	Berkelse Poort	2.4512	Achterzeedijk West	2.2327
11	Tuindorp	2.4539	Tuindorp	2.5819
12	Oranjewijk 2	2.5446	Kapelpolder	2.3205
13	Meerwede Noordoost	2.6305	Zeeheldenbuurt-west	2.5769
14	Rockanje	2.7893	Maaswijk-Noordoost	2.8796
15	Kleine Goote	2.8119	Nieuw-Helvoet	3.0403

B.3. Ablation Study

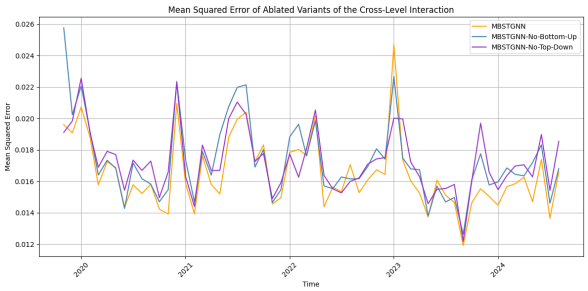


(a) Comparison of MSE for Ablated Variants of MBSTGNN

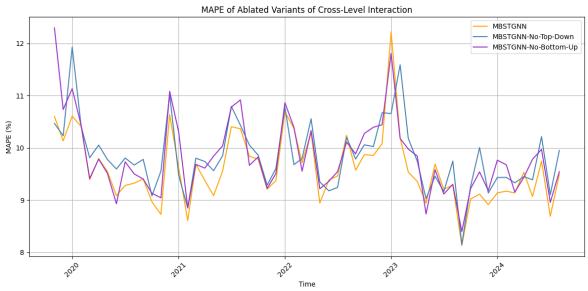


(b) Comparison of MAPE for Ablated Variants of MBSTGNN

Figure B.2: Comparison of performance of Ablated Variants of MBSTGNN through time



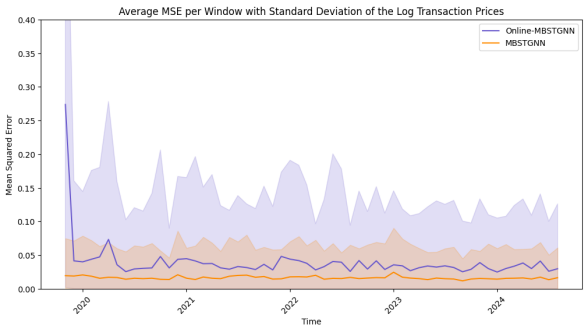
(a) Comparison of the MSE of Variations of the Cross-Level Interaction Module



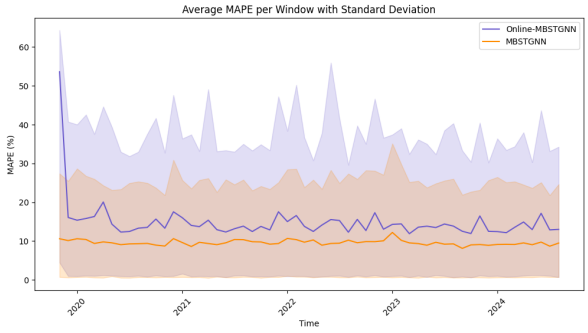
(b) Comparison of the MAPE of Variations of the Cross-Level Interaction Module

Figure B.3: Comparison of performance of Variations of the Cross-Level Interaction Module

B.4. Online Learning



(a) Comparison of the MSE of Online and Offline Learning of MBSTGNN

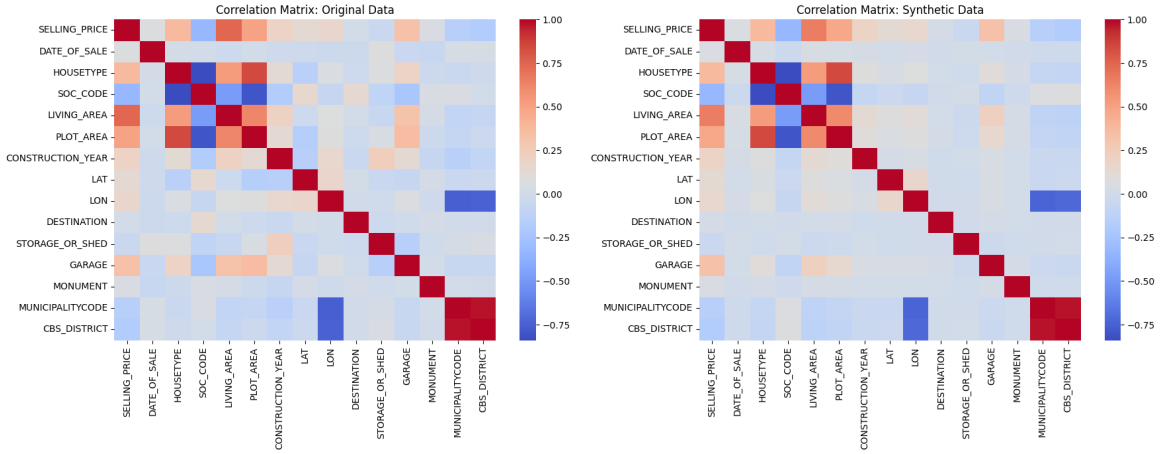


(b) Comparison of the MAPE of Online and Offline Learning of MBSTGNN

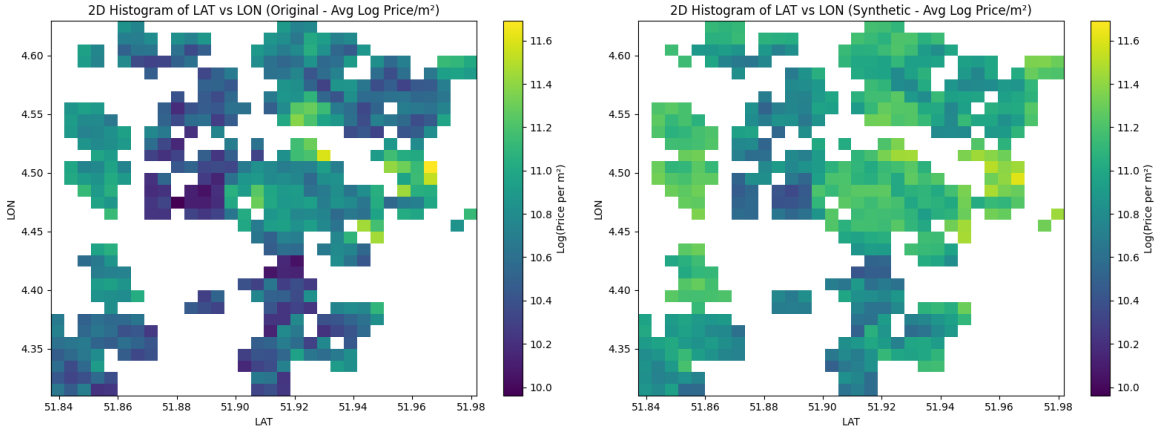
Figure B.4: Comparison of the Performance of Online and Offline Learning of MBSTGNN

C Data Description

C.1. Characteristics of the Transaction Datasets



(a) Correlation Matrix of the original transaction dataset and the synthetic transaction dataset.



(b) Distribution of Price per Square Metre of original transaction and synthetic datasets with respective to longitude and latitude

Figure C.1: Correlation matrix and distribution of prices of both the transaction dataset

C.2. Description of Transaction Dataset Features

Table C.1: Overview of transaction dataset features

Feature Name	Description
TRANSID	Unique identifier for the transaction
DATE	Date of the transaction
SELLING_PRICE	Selling price of the property
PROPERTY_TYPE	Categorical code indicating the type of dwelling
SOC_CODE	Indicates whether the property is social housing (1=yes, 0=no)
LIVING_AREA (m^2)	Total calculated usable living area
PLOT_AREA (m^2)	Land plot area
BUILD_YEAR	Year of construction
NEIGHBOURHOOD_CODE	Code of the neighbourhood in which the property is located
LAT	Geographic latitude coordinate
LON	Geographic longitude coordinate
DESTINATION	Designated land use or zoning category
DUMMY_STORAGE	Indicates the presence of a storage unit (1=yes, 0=no)
DUMMY_GARAGE	Indicates the presence of a garage (1=yes, 0=no)
DUMMY_MONUMENT	Indicates whether the property is a registered monument (1=yes, 0=no)

C.3. Description of Neighbourhood Dataset Features

Table C.2: Overview of Neighbourhood Dataset Features

Category	Feature	Description
Geographic	BUURTCODE	Unique code identifying the neighborhood.
Geographic	YEAR	Year in which the data was collected.
Environmental	LEEFBAAROMETER	Composite livability index score for the area.
Environmental	GROEN	Percentage of green space in the neighborhood.
Environmental	EC	Average household energy consumption.
Environmental	NO2	Annual average nitrogen dioxide concentration ($\mu g/m^3$).
Environmental	PM2_5	Annual average fine particulate matter concentration ($\mu g/m^3$).
Environmental	PM10	Annual average particulate matter concentration ($\mu g/m^3$).
Environmental	GELUIDSHINDERTOTAAL	Index measuring total environmental noise disturbance.
Health	AFSTANDTOTHUISARTSENPRAKTIJK	Distance to the nearest general practitioner practice (km).
Health	HUISARTSENPRAKTIJKBINNEN1KM	General practitioner within 1 km (binary).
Health	HUISARTSENPRAKTIJKBINNEN3KM	General practitioner within 3 km (binary).
Health	HUISARTSENPRAKTIJKBINNEN5KM	General practitioner within 5 km (binary).
Health	AFSTANDTOTHUISARTSENPOST	Distance to the nearest out-of-hours general practice (km).
Health	AFSTANDTOTAPOTHEEK	Distance to the nearest pharmacy (km).
Health	AFSTANDTOTZIEKENHUISINCLBUITENPOLIKLINIEK	Distance to the nearest hospital including outpatient clinics (km).
Health	ZIEKENHUISINCLBUITENPOLIKLINIEKBINNEN5KM	Hospital/outpatient clinic within 5 km (binary).
Health	ZIEKENHUISINCLBUITENPOLIKLINIEKBINNEN10KM	Hospital/outpatient clinic within 10 km (binary).
Health	ZIEKENHUISINCLBUITENPOLIKLINIEKBINNEN20KM	Hospital/outpatient clinic within 20 km (binary).
Health	AFSTANDTOTZIEKENHUISEXCLBUITENPOLIKLINIEK	Distance to the nearest hospital excluding outpatient clinics (km).
Health	ZIEKENHUISEXCLBUITENPOLIKLINIEKBINNEN5KM	Hospital within 5 km (binary).
Health	ZIEKENHUISEXCLBUITENPOLIKLINIEKBINNEN10KM	Hospital within 10 km (binary).
Health	ZIEKENHUISEXCLBUITENPOLIKLINIEKBINNEN20KM	Hospital within 20 km (binary).
Commercial	AFSTANDTOTGROTESUPERMARKT	Distance to the nearest large supermarket (km).
Commercial	GROTESUPERMARKTBINNEN1KM	Large supermarket within 1 km (binary).
Commercial	GROTESUPERMARKTBINNEN3KM	Large supermarket within 3 km (binary).
Commercial	GROTESUPERMARKTBINNEN5KM	Large supermarket within 5 km (binary).
Commercial	AFSTANDTOTOVDAGELLEVENSMIDDELEN	Distance to the nearest daily goods store (km).
Commercial	OVDAGELLEVENSMIDDELENBINNEN1KM	Daily goods store within 1 km (binary).
Commercial	OVDAGELLEVENSMIDDELENBINNEN3KM	Daily goods store within 3 km (binary).
Commercial	OVDAGELLEVENSMIDDELENBINNEN5KM	Daily goods store within 5 km (binary).
Commercial	AFSTANDTOTWARENHUIS	Distance to the nearest department store (km).
Commercial	WARENHUISBINNEN5KM	Department store within 5 km (binary).
Commercial	WARENHUISBINNEN10KM	Department store within 10 km (binary).

Continued on next page

Table C.2 – continued from previous page

Category	Feature	Description
Commercial	WARENHUISBINNEN20KM	Department store within 20 km (binary).
Commercial	AFSTANDTOTCAFEED	Distance to the nearest café (km).
Commercial	CAFEEDBINNEN1KM	Café within 1 km (binary).
Commercial	CAFEEDBINNEN3KM	Café within 3 km (binary).
Commercial	CAFEEDBINNEN5KM	Café within 5 km (binary).
Commercial	AFSTANDTOTCAFETARIAED	Distance to the nearest snack bar (km).
Commercial	CAFETARIAEDBINNEN1KM	Snack bar within 1 km (binary).
Commercial	CAFETARIAEDBINNEN3KM	Snack bar within 3 km (binary).
Commercial	CAFETARIAEDBINNEN5KM	Snack bar within 5 km (binary).
Commercial	AFSTANDTOTRESTAURANT	Distance to the nearest restaurant (km).
Commercial	RESTAURANTBINNEN1KM	Restaurant within 1 km (binary).
Commercial	RESTAURANTBINNEN3KM	Restaurant within 3 km (binary).
Commercial	RESTAURANTBINNEN5KM	Restaurant within 5 km (binary).
Recreational	AFSTANDTOTBIBLIOTHEEK	Distance to library (km)
Recreational	AFSTANDTOTZWEMBAD	Distance to swimming pool (km)
Recreational	AFSTANDTOTKUNSTIJSBAAN	Distance to ice rink (km)
Recreational	AFSTANDTOTMUSEUM	Distance to museum (km)
Recreational	MUSEUMBINNEN5KM	Museum within 5 km (binary)
Recreational	MUSEUMBINNEN10KM	Museum within 10 km (binary)
Recreational	MUSEUMBINNEN20KM	Museum within 20 km (binary)
Recreational	AFSTANDTOTPODIUMKUNSTENTOTAAL	Distance to performing arts venues (km)
Recreational	PODIUMKUNSTENTOTAALBINNEN5KM	Performing arts venues within 5 km (binary)
Recreational	PODIUMKUNSTENTOTAALBINNEN10KM	Performing arts venues within 10 km (binary)
Recreational	PODIUMKUNSTENTOTAALBINNEN20KM	Performing arts venues within 20 km (binary)
Recreational	AFSTANDTOTPOPPODIUM	Distance to pop music venues (km)
Recreational	AFSTANDTOTBIOSCOOP	Distance to cinema (km)
Recreational	BIOSCOOPBINNEN5KM	Cinema within 5 km (binary)
Recreational	BIOSCOOPBINNEN10KM	Cinema within 10 km (binary)
Recreational	BIOSCOOPBINNEN20KM	Cinema within 20 km (binary)
Recreational	AFSTANDTOTSAUNA	Distance to sauna (km)
Recreational	AFSTANDTOTZONNEBANK	Distance to tanning salon (km)
Recreational	AFSTANDTOTATTRACTIE	Distance to attraction (km)
Recreational	ATTRACTIEBINNEN10KM	Attraction within 10 km (binary)
Recreational	ATTRACTIEBINNEN20KM	Attraction within 20 km (binary)
Recreational	ATTRACTIEBINNEN50KM	Attraction within 50 km (binary)
Education	AFSTANDTOTSCHOOLBASIS	Distance to primary school (km)
Education	SCHOOLBASISBINNEN1KM	Primary school within 1 km (binary)
Education	SCHOOLBASISBINNEN3KM	Primary school within 3 km (binary)
Education	SCHOOLBASISBINNEN5KM	Primary school within 5 km (binary)
Education	AFSTANDTOTSCHOOLVOORTGEZET	Distance to secondary school (km)
Education	SCHOOLVOORTGEZETBINNEN3KM	Secondary school within 3 km (binary)
Education	SCHOOLVOORTGEZETBINNEN5KM	Secondary school within 5 km (binary)
Education	SCHOOLVOORTGEZETBINNEN10KM	Secondary school within 10 km (binary)
Education	AFSTANDTOTSCHOOLVMBO	Distance to VMBO school (km)
Education	SCHOOLVMBOBINNEN3KM	VMBO within 3 km (binary)
Education	SCHOOLVMBOBINNEN5KM	VMBO within 5 km (binary)
Education	SCHOOLVMBOBINNEN10KM	VMBO within 10 km (binary)
Education	AFSTANDTOTSCHOOLHAVOVWO	Distance to HAVO/VWO school (km)
Education	SCHOOLHAVOVWOBINNEN3KM	HAVO/VWO within 3 km (binary)
Education	SCHOOLHAVOVWOBINNEN5KM	HAVO/VWO within 5 km (binary)
Education	SCHOOLHAVOVWOBINNEN10KM	HAVO/VWO within 10 km (binary)
Employment	AFSTANDTOTBRANDWEERKAZERNE	Distance to fire station (km)
Employment	BANENTOTAALBINNEN10KM	Total jobs within 10 km
Employment	BANENTOTAALBINNEN20KM	Total jobs within 20 km
Employment	BANENTOTAALBINNEN50KM	Total jobs within 50 km
Employment	BANENAGRARISCHBINNEN10KM	Agricultural jobs within 10 km
Employment	BANENAGRARISCHBINNEN20KM	Agricultural jobs within 20 km
Employment	BANENAGRARISCHBINNEN50KM	Agricultural jobs within 50 km
Employment	BANENNIJVENERGIEBINNEN10KM	Industry and energy jobs within 10 km
Employment	BANENNIJVENERGIEBINNEN20KM	Industry and energy jobs within 20 km
Employment	BANENNIJVENERGIEBINNEN50KM	Industry and energy jobs within 50 km
Employment	BANENCOMMDIENSTBINNEN10KM	Commercial services jobs within 10 km

Continued on next page

Table C.2 – continued from previous page

Category	Feature	Description
Employment	BANENCOMMDIENSTBINNEN20KM	Commercial services jobs within 20 km
Employment	BANENCOMMDIENSTBINNEN50KM	Commercial services jobs within 50 km
Employment	BANENNIETCOMMDIENSTBINNEN10KM	Non-commercial services jobs within 10 km
Employment	BANENNIETCOMMDIENSTBINNEN20KM	Non-commercial services jobs within 20 km
Employment	BANENNIETCOMMDIENSTBINNEN50KM	Non-commercial services jobs within 50 km
Demographics	INWONERSAANTAL	Number of residents
Demographics	AANTALGEBOORTEN	Number of births
Demographics	AANTALOVERLIJDEN	Number of deaths
Demographics	LEEFTIJDSGEMIDDELDE	Average age
Demographics	LEEFTIJDKLASSEN	Population by age group
Demographics	GESLACHTSVERDELING	Population by gender
Housing	AANTALHUIZEN	Number of houses
Housing	WOONOPPERVLAKTE	Residential area (km²)
Housing	WONINGTYPE	Housing type distribution
Housing	HUIZENBINNEN1KM	Houses within 1 km
Housing	HUIZENBINNEN5KM	Houses within 5 km
Housing	HUIZENBINNEN10KM	Houses within 10 km
Utilities	AFSTANDTOTWATERLEIDING	Distance to water pipeline (km)
Utilities	AFSTANDTOTGASLEIDING	Distance to gas pipeline (km)
Utilities	AFSTANDTOTELEKTRICITEITSLEIDING	Distance to electricity grid (km)
Utilities	AFSTANDTOTAFVALVERWERKING	Distance to waste processing facility (km)
Income	GEMIDDELDINKOMEN	Average income per household
Income	INKOMENSPERCENTIEL	Income percentile distribution
Income	AANTALINKOMENLAAG	Number of low-income households
Income	AANTALINKOMENHOOG	Number of high-income households
Safety	GELUIDSHINDERTOTAAL	Total noise nuisance
Safety	AFSTANDTOTBRANDWEERKAZERNE	Distance to fire station (km)
Transportation	AFSTANDTOTOPRITHOOFDVERKEERSWEG	Distance to main road (km)
Transportation	AFSTANDTOTTREINSTATIONSTOTAAL	Distance to nearest train station (km)
Transportation	AFSTANDTOTBELANGRIJKOVERSTAPSTATION	Distance to major transfer station (km)
Transportation	PERSONENAUTOSTOTAAL	Total number of private cars
Transportation	PERSONENAUTOSPERHUISHOUDEN	Cars per household
Transportation	MOTORFIETSEN	Number of motorcycles

D

Synthetic Data

D.1. Overall Comparison

The results of the comparison of MBSTGNN and LighGBM on the synthetic dataset corroborates our previous results. MBSTGNN once again outperforms LightGBM.

Table D.1: Comparison of average error metrics over the final rolling windows for each model. Values represent mean $\pm$ standard deviation of MSE and MAPE over the last N prediction windows (5-year window size). The best values per row are **in bold and underlined**.

Last N Windows	LightGBM	MBSTGNN
MSE (mean $\pm$ SD)		
60 (all windows)	0.0519 $\pm$ 0.1014	**0.0389 $\pm$ 0.0863**
5	0.0198 $\pm$ 0.0409	**0.0156 $\pm$ 0.0334**
MAPE (% mean $\pm$ SD)		
60 (all windows)	17.67 $\pm$ 17.27	**15.07 $\pm$ 15.40**
5	18.63 $\pm$ 16.37	**13.94 $\pm$ 13.55**

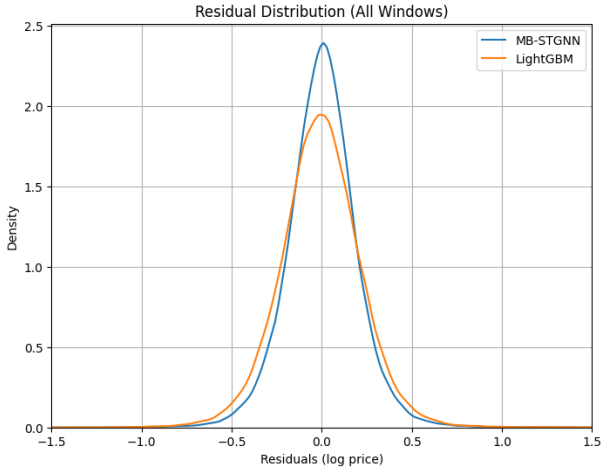
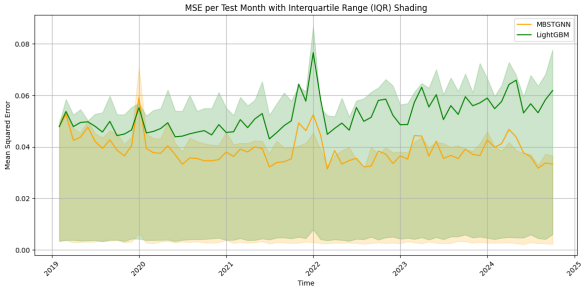
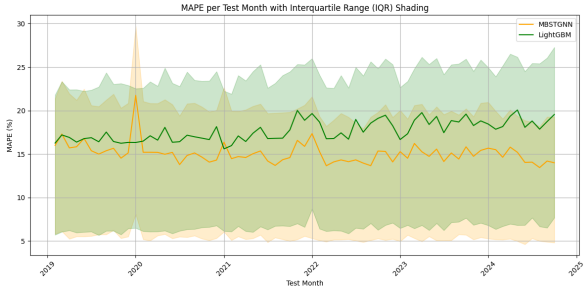


Figure D.1: Residual Distribution across all windows of models trained on the synthetic dataset

D.2. Temporal Generalisation



(a) Evolution of MSE of the Proposed Model and Macro-Level Baseline



(b) Evolution of MAPE of the Proposed Model and LightGBM Baseline

Figure D.2: Comparison of the Performance of LightGBM and MBSTGNN trained on the synthetic dataset

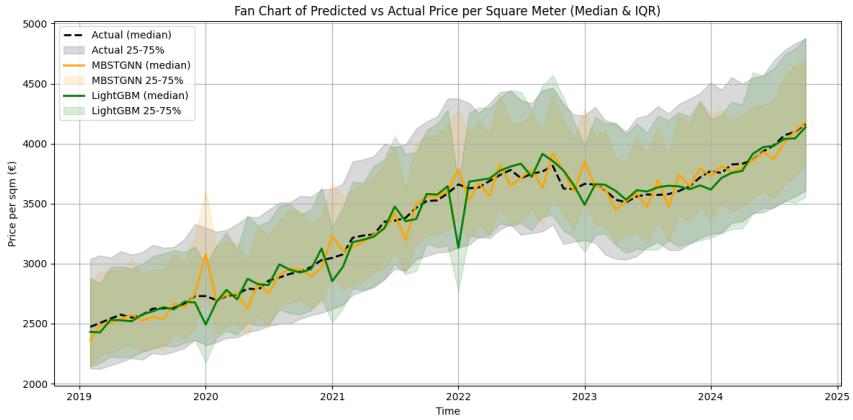
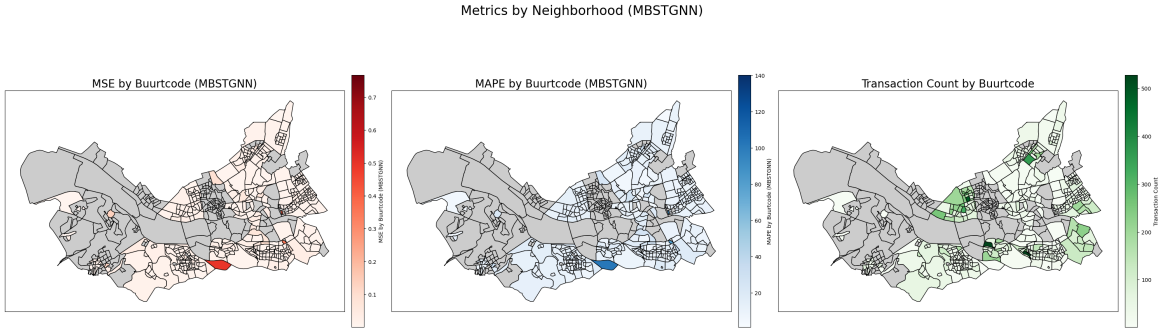
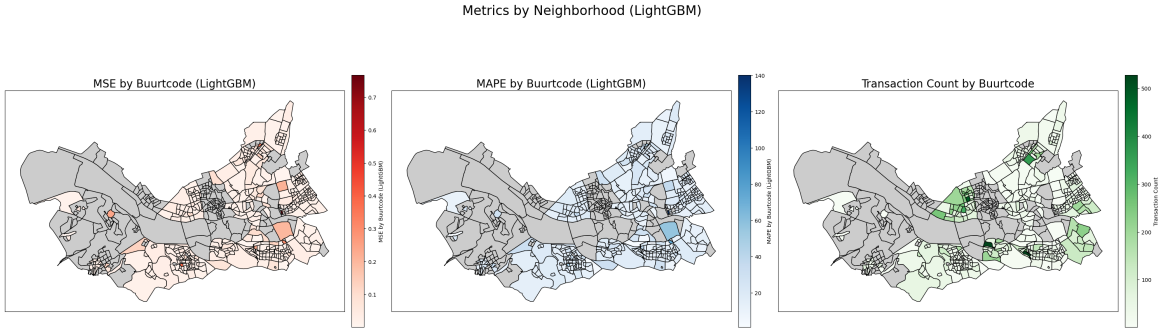


Figure D.3: Fan chart illustrating the predicted versus actual median price per square meter over time, with interquartile range (IQR) shading. The chart compares MBSTGNN,and LightGBM. The dashed black line represents the actual median prices.

D.3. Spatial Analysis



(a) Neighbourhood-level spatial performance of the MBSTGNN model trained on synthetic data over the last 12 prediction windows.



(b) Neighbourhood-level spatial performance of the LightGBM model trained on synthetic data over the last 12 prediction windows.

Figure D.4: Spatial evaluation of model performance across neighbourhoods in Rotterdam. From left to right: (a) Mean Squared Error (MSE), (b) Mean Absolute Percentage Error (MAPE), and (c) number of transactions.