

2024/25

Geomatics

# ST-SimNet

A Spatio-Temporal Graph Neural Network  
for Urban Freight Forecasting

Master thesis

Rafał Marek Tarczyński  
4439899

Supervisors:

MSc Finn Winkelmann TNO

dr.ir. Martijn Meijers TU Delft

dr.ir. Azarakhsh Rafiee TU Delft

MSc Amin Jalilzadeh TU Delft

TNO | TU Delft



MSc thesis in Geomatics

**ST-SimNet: A Spatio-Temporal Graph  
Neural Network for Urban Freight  
Forecasting**

Rafał Marek Tarczyński

June 2025

A thesis submitted to the Delft University of Technology in  
partial fulfilment of the requirements for the degree of Master  
of Science in Geomatics

Rafał Marek Tarczyński: *ST-SimNet: A Spatio-Temporal Graph Neural Network for Urban Freight Forecasting* (2025)

© ⓘ ⊖ This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-nd/4.0/>.

Supervisors: Finn Winkelmann *TNO*  
Martijn Meijers *TU Delft*  
Azarakhsh Rafiee *TU Delft*  
Amin Jalilzadeh *TU Delft*  
Co-reader: Saeed Rahmani *TU Delft*

# Abstract

Urban freight transport is a critical yet complex component of city logistics, shaped not only by transport networks but also by the morphological structure of urban areas. Traditional forecasting models often neglect this spatial heterogeneity, relying primarily on traffic counts or infrastructure topology. This thesis proposes and evaluates ST-SimNet, a Spatio-Temporal Simulation Network designed to enhance freight flow prediction by integrating static urban morphology descriptors with dynamic freight data in a graph neural network framework. Focusing on the city of Amsterdam, the study explores the extent to which detailed urban morphology, including building features, land use, and spatial layout, can improve short-term freight flow forecasts at the road network level. Results demonstrate that incorporating static features significantly reduces error variance, improves peak hour prediction, and enhances node-level stability compared to dynamic-only baselines. Furthermore, analysis reveals that nodes with richer morphology information benefit most, while areas with sparse or noisy static features experience challenges that highlight opportunities for future refinement. The findings offer practical insights for integrating machine learning into digital twin platforms for urban mobility, providing a data-driven, spatially aware layer for freight forecasting in operational city planning systems. Limitations and future directions, including adaptive fusion mechanisms and cross-city generalisation, are discussed. Overall, ST-SimNet advances the integration of urban morphology into spatio-temporal predictive models and demonstrates its practical relevance for modern freight planning in complex urban environments.



# Acknowledgements

This thesis marks the culmination of my second Master's programme at TU Delft, and I would like to express my deepest gratitude to those who have supported and guided me throughout this journey.

First and foremost, I would like to thank TNO for providing the opportunity to work at the intersection of research and practice. The insights and resources offered by my colleagues were invaluable. In particular, I am grateful to Finn Winkelmann for his continuous feedback, technical clarity, and patient guidance during my internship.

I also wish to thank my supervisors at TU Delft: Martijn Meijers, Azarakhsh Rafiee and Amin Jalilzadeh for their critical input, encouragement, and support in shaping the academic direction of this project. Their expertise in geomatics and machine learning greatly enriched this work.

Special thanks are due to Shenglang Du and Nail Ibrahimli for sharing their expertise in machine learning and providing valuable feedback. I am especially grateful to Saeed Rahmani, who not only shared his insights but also provided detailed comments as the co-reader of this thesis.

I would also like to express my deep gratitude to my parents for their unwavering support and belief in me.

This research would not have been possible without the collaboration and generosity of all the people mentioned above.



# Contents

<b>1. Introduction</b>	<b>1</b>
1.1. Aim of the Research	3
1.2. Hypothesis and Rationale	3
1.3. Knowledge Gap	4
1.4. Research Questions	5
1.5. Main Research Question	6
1.6. Research Sub-Questions	6
1.7. Relevance and Significance of the Research Questions	7
1.7.1. Academic Contribution	7
1.7.2. Practical Relevance	8
<b>2. Related work</b>	<b>9</b>
2.1. Spatio-temporal GNNs for Traffic Prediction	9
2.2. Graph Neural Networks for Road Networks	10
2.3. Urban Morphology in Graph-based Urban Analysis	10
2.4. Relational Inductive Biases and Model Generalisability	12
2.5. Integration with Digital Twin Frameworks	12
2.6. Baseline STGCN	13
2.6.1. Training Pipeline and Datasets	13
2.6.2. Performance and Contributions	15
2.6.3. Relevance as a Baseline:	16
2.7. Summary of Contributions of ST-SimNet	16
<b>3. Methodology</b>	<b>19</b>
3.1. Notation and Glossary	19
3.2. Overview	20
3.3. Input Data	20
3.3.1. Dynamic Freight Flow Data	20
3.3.2. Urban Morphology and Road Networks	21
3.3.3. Data Preprocessing	22
3.4. ST-SimNet Architecture	30
3.4.1. Data Preparation and Loading	32
3.4.2. ST-Conv blocks	34
3.4.3. UMD Fusion	36
3.4.4. Output Block	37
3.5. Model Training Setup	38
3.6. Validation and Evaluation	39
<b>4. Results</b>	<b>41</b>
4.1. Experiment Design	41
4.2. Model Introduction	41

Contents

4.3. Baseline: STGCN with Dynamic Input Only . . . . .	42
4.3.1. Model Configuration . . . . .	42
4.3.2. Training Performance . . . . .	42
4.3.3. Prediction Accuracy . . . . .	43
4.3.4. Spatial Heterogeneity of Performance . . . . .	43
4.3.5. Conclusions . . . . .	43
4.4. ST-SimNet: Dynamic + Static Input . . . . .	45
4.4.1. Model Configuration . . . . .	45
4.4.2. Urban Morphology Data (UMD) Features . . . . .	46
4.4.3. Scenario 1: Weekdays Only - Amsterdam West . . . . .	49
4.4.4. Scenario 2: Weekdays + Weekends - Amsterdam West . . . . .	56
4.5. Application of ST-SimNet to a Larger Area . . . . .	60
4.5.1. Training Performance . . . . .	60
4.5.2. Prediction Accuracy . . . . .	62
4.5.3. Spatial Heterogeneity of Performance . . . . .	62
4.5.4. Feature Contribution Analysis . . . . .	62
4.5.5. Visual Inspection of Results in QGIS . . . . .	65
4.5.6. Conclusion . . . . .	66
<b>5. Conclusions and Future Work . . . . .</b>	<b>69</b>
5.1. Discussion . . . . .	69
5.2. Limitations . . . . .	70
5.3. Conclusions . . . . .	71
5.4. Future Work . . . . .	73
<b>A. Reproducibility self-assessment . . . . .</b>	<b>75</b>
A.1. Marks for each of the criteria . . . . .	75
A.2. Personal Reflection . . . . .	75
<b>B. Areas of Interest . . . . .</b>	<b>77</b>
<b>C. MASS-GT parameters . . . . .</b>	<b>79</b>
C.1. Vehicle Types . . . . .	79
C.2. NSTR Goods Classification . . . . .	79
C.3. Logistic Segment Classification . . . . .	80
C.4. Transport Flow Classification . . . . .	80

# List of Figures

2.1. Spatial-temporal correlation is dominated by the road network structure. <b>(a)</b> Traffic sensors distributed in the road network. <b>(b)</b> Dynamic spatial-temporal dependence from time $t - T$ to $t + T'$ . Taken from <a href="#">Xiong et al. [2024]</a> . . . . .	9
2.2. Architecture of the Relational Fusion Network (RFN). Relational Fusion Network (RFN) fuses node-relational and edge-relational features across multiple layers to improve learning on heterophilic road networks. Taken from <a href="#">Jepsen et al. [2019]</a> . . . . .	11
2.3. Conceptual overview of spatial homogeneity in urban road networks. Homogeneity is defined as the similarity of link existence patterns between different parts of the same city (intra-city) or across cities (inter-city). The metric, derived from GNN-based link prediction, correlates with socioeconomic indicators such as GDP and population growth, and reveals urban structural similarities transferable across cities. Taken from <a href="#">Xue et al. [2021]</a> . . . . .	12
2.4. Detailed architecture of the Spatio-Temporal Convolutional Block (Spatio-Temporal Convolutional Block (ST-ConvBlock)) in the original Spatio-Temporal Graph Convolutional Network (STGCN) model [ <a href="#">Yu et al., 2018</a> ]. The block uses a “sandwich” design with two 1D gated temporal convolutions (Gated Linear Unit (GLU)-based) on either side of a spatial graph convolution layer. The first temporal convolution reduces feature dimensionality, acting as a bottleneck to improve computational efficiency. The central graph convolution captures spatial dependencies across the graph defined by the road network. The final temporal convolution restores the dimensionality, enabling the model to reconstruct time-dependent patterns. A residual connection links the block’s input to its final output to facilitate gradient flow and model stability during training. This modular design allows the architecture to be stacked in deeper networks, making it both scalable and interpretable for traffic prediction tasks. Figure taken from <a href="#">Yu et al. [2018]</a> . . . . .	14
3.1. Cropped road network showing nodes and edges within the Amsterdam study area EPSG:28992. . . . .	24
3.2. Graph structure of the road network built from spatial data using <code>networkx.DiGraph</code> EPSG:28992. . . . .	25
3.3. Sparsity pattern of the adjacency matrix. White space indicates absence of direct connection between node pairs. . . . .	26
3.4. Node-level distribution of inherited buildings, EPSG:4326. . . . .	28
3.5. Overview of the Spatio-Temporal Simulation Network (ST-SimNet) architecture. Each coloured region highlights a core module: the blue components represent the two stacked ST-ConvBlocks; the purple section denotes the Fusion Block where static Urban Morphology Descriptors (UMDs) are integrated; and the orange module at the end represents the Gated Temporal Convolution layer used in each ST-Conv Block. . . . .	31

List of Figures

3.6. Graph-structured traffic data. Each $v_t$ indicates a frame of current traffic status at time step $t$ , which is recorded in a graph-structured data matrix. Adapted from [Yu et al., 2018]. . . . .	33
4.1. Training and Validation Loss over Epochs (STGCN) . . . . .	42
4.2. Learning Rate Schedule during Training (STGCN) . . . . .	43
4.3. Scatter Plot: True vs Predicted Flow (STGCN) . . . . .	44
4.4. Histogram of Prediction Errors (STGCN) . . . . .	44
4.5. Per-node Flow Statistics: Predicted vs True (STGCN) . . . . .	45
4.6. Gradient-based saliency scores for static urban morphology features in one representative Area of Interest. Highlighted features contribute most strongly to the model’s node-level predictions. . . . .	48
4.7. Training and Validation Loss over Epochs (ST-SimNet, Weekdays) . . . . .	49
4.8. Learning Rate Schedule during Training (ST-SimNet, Weekdays) . . . . .	50
4.9. Scatter Plot: True vs Predicted Flow (ST-SimNet, Weekdays) . . . . .	51
4.10. Histogram of Prediction Errors (ST-SimNet, Weekdays) . . . . .	52
4.11. Per-node Flow Statistics: Predicted vs True (ST-SimNet, Weekdays) . . . . .	52
4.12. Relationship between Static Feature Richness and MAE (ST-SimNet, Weekdays) . . . . .	53
4.13. Over-estimated node lacking building-derived UMD features. . . . .	53
4.14. Nodes with high flows and under-estimated values . . . . .	54
4.15. Node with a single building-derived UMD source shows near-ideal prediction. . . . .	54
4.16. Training and Validation Loss - ST-SimNet with Weekend Data . . . . .	56
4.17. Learning Rate Schedule - ST-SimNet with Weekend Data . . . . .	57
4.18. Scatter Plot: True vs Predicted Flow (ST-SimNet, Weekdays + Weekends) . . . . .	58
4.19. Histogram of Prediction Errors (ST-SimNet, Weekdays + Weekends) . . . . .	59
4.20. UMD Feature Richness vs Prediction Error (ST-SimNet, Weekends Included) . . . . .	59
4.21. Per-node Flow Statistics: Predicted vs True (ST-SimNet, Weekdays + Weekends) . . . . .	60
4.22. Learning rate schedule during training for the full-area model. . . . .	61
4.23. Training and validation loss (MAE) over 100 epochs. . . . .	61
4.24. Histogram of prediction errors. Most predictions are close to the ground truth. . . . .	62
4.25. Predicted vs. true flow values for all nodes and all time steps. . . . .	63
4.26. Mean absolute error binned by UMD feature norm. Moderate feature richness leads to best performance. . . . .	64
4.27. Per-node prediction statistics. Predicted and true values exhibit strong correspondence. . . . .	64
4.28. Spatial distribution of over-predicted nodes, shown in red. Most are located along major highway infrastructure (A10) . . . . .	65
4.29. Summary of over-predicted nodes and corresponding features. . . . .	66
A.1. Reproducibility criteria to be assessed. . . . .	75
B.1. AOIs for trainings, Amsterdam, EPSG:28992. Amsterdam West (orange) with 817 nodes; Centre of Amsterdam (red) with 10691 nodes. . . . .	77

# List of Tables

3.1. Glossary of Symbols Used in <i>ST-SimNet</i> . . . . .	19
3.2. Selected Features for <i>ST-SimNet</i> Node Enrichment . . . . .	29
4.1. Encoded Values for Categorical UMD Features . . . . .	47
C.1. Vehicle Type Classification . . . . .	79
C.2. NSTR Goods Classification . . . . .	79
C.3. Logistic Segment Classification . . . . .	80
C.4. Transport Flow Classification . . . . .	80



# List of Algorithms

3.1. Cropping road network . . . . .	22
3.2. Directed graph adjacency . . . . .	23
3.3. Building-to-node join . . . . .	27
3.4. PC6 cropping . . . . .	28
3.5. PC6-to-node join . . . . .	29
3.6. Random walk normalised adjacency matrix (out-degree) . . . . .	32
3.7. GLU-based Temporal Convolution . . . . .	34
3.8. Spatial Graph Convolution . . . . .	36
3.9. UMD fusion layer . . . . .	37
3.10. Output Block . . . . .	38



# Acronyms

AOI	Area of Interest	41
BAG	Basisregistratie Adressen en Gebouwen	8
CBS	Centraal Bureau voor de Statistiek (Statistics Netherlands)	3
CNN	Convolutional Neural Network	9
DT	Digital Twin	3
DC	Distribution Centre	1
DCRNN	Diffusion Convolutional Recurrent Neural Network	17
GFAGNN	Gated Fusion Adaptive Graph Neural Network	9
GLU	Gated Linear Unit	xi
GCN	Graph Convolutional Network	10
GNN	Graph Neural Network	2
GNoME	Graph Networks for Materials Exploration	2
GSO	Graph Shift Operator	30
GIS	Geographic Information System	12
MAE	Mean Absolute Error	15
MASS-GT	Multi-Agent Simulation System for Goods Transport	1
MLP	Multi-Layer Perceptron	13
MSE	Mean Squared Error	30
NDVI	Normalised Difference Vegetation Index	46
NTL	Night-Time Lights	46
PC6	Six-digit Dutch postcode (postcode 6)	6
RFN	Relational Fusion Network	xi
RNN	Recurrent Neural Network	20
RMSE	Root Mean Squared Error	15
ST-ConvBlock	Spatio-Temporal Convolutional Block	xi
ST-GNN	Spatio-Temporal Graph Neural Network	3
ST-SimNet	Spatio-Temporal Simulation Network	xi
STGCN	Spatio-Temporal Graph Convolutional Network	xi
TGNN	Temporal Graph Neural Network	4
TNO	Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek (Dutch Organisation for Applied Scientific Research)	20
TT	Transshipment Terminal	1
UMD	Urban Morphology Descriptor	xi
VMA	Verkeersmodel Amsterdam	3
WMAPE	Weighted Mean Absolute Percentage Error	15



# 1. Introduction

The Netherlands, with its dense urban centres and intricate logistical networks, presents a compelling case study for examining the potential of advanced computational methods in optimising freight transportation. Freight transport remains a vital enabler of economic activity, with road freight continuing to dominate modal share despite broader shifts in the logistics sector. According to the TNO report *Decamod: Toolbox voor rekenen aan CO<sub>2</sub>-reductie in transport en logistiek* (2020) [TNO, 2020], road transport is projected to carry a slightly reduced share of total freight—from 75% in 2014 to 73% by 2030. Yet its environmental impact remains disproportionately high: road freight is expected to account for approximately 81% of CO<sub>2</sub> emissions from freight transport by 2030, due to its lower energy efficiency compared to rail and inland shipping and the dominance of short-haul, high-frequency trips within urban areas.

To systematically model and interpret freight flows at the urban scale, this study leverages the classification schema embedded within the Multi-Agent Simulation System for Goods Transport (MASS-GT) simulation tool. These structured classifications form the operational backbone of MASS-GT and are essential for defining vehicle characteristics, freight profiles, logistic modalities, and flow structures. MASS-GT serves as the simulation engine generating the spatio-temporal freight flows that form the dynamic input to the ST-SimNet model. These simulation outputs are aligned to the road network topology, with node-level freight intensities. The following appendices provide more insight into the modal and structural mechanisms of MASS-GT:

- Appendix C.1 describes the ten distinct vehicle categories used in the model, ranging from small trucks and vans to articulated lorries and specialised freight carriers. This classification is crucial for estimating load capacities, network compatibility, and spatial reach, all of which influence routing logic, congestion dynamics, and accessibility constraints.
- Appendix C.2 outlines the NSTR (NST 2007) goods taxonomy, distinguishing between sectors such as agricultural products, machinery, building materials, and chemicals. These commodity classes are key to modelling differentiated freight demand, with each type exhibiting distinct temporal profiles, routing behaviours, and sensitivity to urban form.
- Appendix C.3 further segments the logistics domain based on operational modality, e.g. temperature-controlled logistics, facility services, construction flows, parcel deliveries, and waste collection. These segments not only inform routing constraints and delivery schedules but also correlate with distinct urban morphology features such as zoning type, building density, and land-use intensity.
- Finally, Appendix C.4 details the transport flow typology used to define directional relationships between producers, Distribution Centres (DCs), Transshipment Terminals

## 1. Introduction

(TTs), and end consumers. This classification separates internal urban flows from interregional and external movements and provides essential structure for interpreting freight assignment patterns and link-level flow directions.

By embedding these structured categories into both the simulation and prediction framework, this study ensures consistency between the behavioural assumptions of *MASS-GT* and the learning architecture of *ST-SimNet*. These definitions provide not only domain realism and reproducibility but also serve as interpretable anchors for the integration of spatial features such as building function, land use, and socio-demographic context. Together, they allow the model to learn freight movement as a function of both infrastructure and morphology.

Given that 73% of road freight in the Netherlands is confined to domestic transport, where modal shifts to rail or water are often infeasible, these structured segmentations provide the granularity needed to model nuanced policy interventions. Even marginal improvements in route efficiency or load balancing, when aggregated across segments and vehicle types, can yield significant environmental and operational gains [TNO, 2020].

Many complex relationships can be effectively represented through knowledge graphs, which serve as powerful structures for modifying, enhancing, and generating new graphs [Martin and Reichmann, 2024]. Their versatility has been increasingly leveraged in solving real-world problems, particularly within scientific and industrial domains. For instance, NVIDIA has harnessed the potential of Graph Neural Network (GNN)s to optimise physical structures for additive manufacturing, leading to significant advancements in lattice structure simulation and predictive modelling [Jain et al., 2024]. This approach has demonstrated how graph-based models can streamline design processes, improve material efficiency, and reduce computational costs in complex engineering tasks.

Similarly, researchers from institutions including Google DeepMind developed the Graph Networks for Materials Exploration (GNoME) framework, which utilises GNN to evaluate material stability based on structural and compositional properties. By scaling the training of these networks, the GNoME framework has achieved remarkable generalisation capabilities, enabling the discovery of over 2.2 million stable crystal structures and significantly enhancing the efficiency of materials discovery [Merchant et al., 2023]. These examples underscore the transformative potential of graph-based models in diverse applications, from optimising manufacturing processes to accelerating scientific discoveries.

Although not directly connected to my graduation topic, those novelties present the potential of GNN. My graduation research explores the use GNN with an additional temporal dimension, in combination with the TNO's Digital Twin platform to visualise the results and enhance our understanding of the relationship between city morphology and transportation networks. This study aims to model the impact of certain elements of urban tissue morphology on road freight transportation flow patterns.

This research is centred on the city of Amsterdam and investigates how static UMD can inform short-term freight flow prediction. The study focuses on understanding whether and under what conditions, morphological features such as building dimensions, facade orientation, and land use categories improve model accuracy at the local scale. Rather than treating static data as a default input, the work critically evaluates its contribution relative to dynamic time series alone, revealing when such features help, and when they introduce noise. In doing so, this thesis offers insights into the role of urban form in shaping transport dynamics, with implications for more spatially aware prediction models and their integration into operational digital twins.

Within the context of the MSc Geomatics program at TU Delft, the study embraces skills in spatial data analysis, geospatial datasets, and the application of machine learning techniques in geographical contexts directly corresponds with several core courses in the program. Notably, the course *Machine Learning for the Built Environment* (GEO5017) explores the foundations of machine learning methodologies, *Python Programming for Geomatics* (GEO1005) develops proficiency in programming skills essential for geospatial data processing and analysis. Additionally, *Sensing Technologies* (GEO1001) provides foundational knowledge in data acquisition methods, which is crucial for understanding and implementing various sensing techniques in geospatial research. *Geo Database Management Systems* (GEO1006) provided insights in data storing and handling, which will be crucial for managing big datasets for this project.

These courses collectively equipped me with the skills necessary to manage, analyse, and interpret complex spatial data, which are essential for the successful execution of this research.

## 1.1. Aim of the Research

The aim of this research is to extend and evaluate an existing [STGCN](#) architecture developed by [Yu et al. \[2018\]](#) for the task of short-term urban freight flow forecasting at high spatial resolution. While the original STGCN effectively captures spatio-temporal patterns from dynamic traffic data, it does not incorporate static urban morphology or contextual features. This work introduces a novel fusion block into the architecture, enabling the integration of static descriptors such as building function, land use, and socio-demographic indicators directly into the node representation.

To this end, the study leverages dynamic freight data from TNO's Digital Twin (DT) framework, including outputs from [MASS-GT](#) and [Verkeersmodel Amsterdam \(VMA\)](#) simulations, alongside static features provided by Amin Jalilzadeh and postcode-level statistics [Centraal Bureau voor de Statistiek \(Statistics Netherlands\) \(CBS\)](#). These data are fused at the node level within the newly designed [ST-SimNet](#) model, allowing the network to account for both temporal variation and spatial heterogeneity in the built environment. The ambition is three-fold: (i) to improve predictive accuracy by contextualising freight flow dynamics with urban form, (ii) to enhance model interpretability and transferability across urban contexts, (iii) to enable integration into digital twin frameworks by reducing computational overhead for traffic predictions. The proposed fusion mechanism constitutes the core architectural contribution of this thesis, demonstrating how spatial context can be systematically embedded within Spatio-Temporal Graph Neural Network ([ST-GNN](#))s for freight forecasting.

## 1.2. Hypothesis and Rationale

Freight transportation models predominantly rely on road network structures and traffic data to predict flows. However, urban morphology and socio-demographic factors play a fundamental role in shaping transportation patterns. This research hypothesises that integrating building attributes and socio-demographic data into Graph Neural Network training will enhance the predictive accuracy of freight transportation flow models. By capturing the underlying urban and socio-demographic dynamics that influence freight movements, this

## 1. Introduction

approach has the potential to provide more comprehensive and adaptive predictions to support urban transportation planning and optimization efforts.

**Rationale** Buildings exert a significant influence on freight movement patterns through their diverse functions, densities, and spatial distributions within the urban landscape. For instance, commercial zones typically experience higher levels of freight activity compared to predominantly residential areas, as commercial enterprises often generate specific freight demands and logistical requirements [Cruz-Daraviña and Suescún, 2021]. Similarly, large industrial facilities can be major hubs of freight generation, creating unique demand patterns that shape the overall freight distribution within the city. These effects are important because they are an indirect measure of the amount of freight a building receives or sends.

On the other hand, residential and low-density areas, which generally exhibit lower levels of economic activity, are often less appealing for targeted road infrastructure planning and logistics optimisation efforts. As a result, these regions may experience reduced freight transport intensity, leading to spatial disparities in freight accessibility and connectivity across the urban fabric. By incorporating comprehensive data on building attributes, land use patterns, and socio-demographic factors into the training of Graph Neural Networks, researchers can uncover hidden relationships and patterns within the freight distribution system, particularly in areas where infrastructure development and transport accessibility may be constrained or uneven. This holistic approach can provide valuable insights to support more informed and adaptive urban transportation planning and optimization strategies.

Traditional transportation forecasting models often struggle in data-scarce environments, requiring extensive calibration and domain-specific knowledge. By incorporating additional urban features, this study aims to bridge these gaps and provide a more adaptable, data-driven approach. Graph-based models, particularly Temporal Graph Neural Network (TGNN), are well-suited to capturing dynamic relationships between spatial entities, making them a promising choice for this task.

### 1.3. Knowledge Gap

Despite significant advancements in urban freight transportation modelling, a critical gap persists in the integration of urban morphology - specifically, the spatial configuration and land use patterns of urban areas into these models. Traditional freight transport models often emphasize sensor data, infrastructure capacity, and policy impacts, yet they frequently overlook how the physical layout of a city influences freight movement and demand [Gonzalez and Smith, 2023]. This oversight is particularly pronounced in data-scarce environments, where limited access to detailed transportation and logistics data hampers the development of accurate and responsive models.

Urban morphology plays a pivotal role in shaping transportation dynamics. The arrangement of roads, the distribution of commercial and residential zones, and the density of urban development directly affect freight routes, delivery efficiency, and overall logistics planning. For instance, the phenomenon of logistics sprawl, characterised by the relocation of logistics facilities from inner urban areas to suburban zones, has been observed to increase truck travel distances and associated emissions [Aljohani and Thompson, 2016]. Understanding these spatial nuances is essential for creating models that accurately reflect real-world freight movements.

In parallel, the emergence of machine learning techniques, particularly *ST-GNN*, offers promising avenues for modelling complex spatiotemporal relationships in transportation systems. *ST-GNNs* are adept at capturing both the spatial dependencies inherent in transportation networks and the temporal dynamics of traffic flows [Jiang et al., 2023]. Their application has shown potential in traffic prediction tasks, leveraging the topological structure of road networks to forecast traffic conditions accurately.

Despite these technological advancements, there remains a paucity of research focused on the application of *ST-GNNs* to model the influence of urban morphology on freight transportation. The proposed research aims to bridge this gap by developing an *ST-GNN*-based framework that incorporates urban form characteristics into freight transport models.

Addressing this knowledge gap is crucial for several reasons. Firstly, integrating urban morphology into freight models can lead to more sustainable and efficient logistics operations by aligning transportation planning with the physical realities of urban environments. Secondly, in data-scarce settings, leveraging the structural information of urban form can serve as a proxy for missing data, thereby improving model robustness and applicability. Ultimately, this research endeavours to contribute to the development of holistic urban freight transport models that are both data-efficient and sensitive to the spatial intricacies of urban landscapes.

## 1.4. Research Questions

While some recent models have begun to incorporate both road network structure and urban morphology, many remain limited to either dynamic traffic data or static built-environment features. This fragmentation highlights a persistent gap: infrastructure-based models may effectively describe how freight moves across a city, yet often overlook the spatial drivers - such as land use patterns or logistical demand generators, that underpin those flows. Conversely, morphology-driven models account for land use but often neglect the structural constraints and affordances of the transport network. A prevalent strategy in urban flow modelling involves discretising the city into uniform grid cells, each treated as a node in a spatio-temporal graph [Wang et al., 2024]. While computationally convenient, such grid-based representations often misalign with actual road geometries, junctions, or administrative zones, thereby masking spatial heterogeneity and connectivity nuances. In most spatio-temporal graph neural networks, the graph structure is defined solely by adjacency or distance, leaving out contextual heterogeneity between urban areas. Recent studies suggest that embedding additional context - such as land use, built environment features, or socio-economic indicators, can improve generalisability and predictive performance [Rahmani et al., 2023]. To bridge this methodological divide, this work introduces and evaluates *ST-SimNet*: a spatio-temporal graph neural network architecture designed to fuse dynamic freight data with static urban morphology data. By aligning graph-based temporal forecasting with morphological context, *ST-SimNet* aims to better capture the interplay between urban structure and movement patterns in complex metropolitan environments.

In the *ST-SimNet* framework, the urban environment is modelled as a spatio-temporal graph, where each node represents a fine-grained spatial unit in the city and is enriched with both dynamic and static information. Nodes correspond to locations along the road network and are associated with feature vectors that encode local urban morphology and demographic context. These static features are derived from building-level and block-level datasets, such

## 1. Introduction

as those from the CBS, and aggregated to the PC6 postcode level. Each node's feature vector may include attributes such as the number of dwellings, total residential population, commercial floor area, or industrial building volume - proxies for freight generation and attraction. In effect, the feature vector serves as a morphological fingerprint for each urban area. The edges of the graph are defined by the underlying transportation infrastructure, linking nodes that are physically connected through the road network. This spatial structure preserves actual routing pathways rather than relying on mere spatial proximity or grid-based simplifications. Built on this graph, *ST-SimNet* incorporates temporal learning modules to capture how freight activity evolves at each node and how changes propagate through the network over time. This enables the model to learn both local temporal dynamics and broader spatial interactions - such as peak-hour congestion, delivery clustering, or spillover effects due to upstream bottlenecks.

### 1.5. Main Research Question

The main question guiding this thesis is:

*To what extent can insights into urban morphology, modeled with Spatio-Temporal Graph Neural Networks, enhance the accuracy and adaptability of freight transportation predictions in the Netherlands?*

This question encapsulates the core aim of the research: to develop a forecasting approach that bridges the gap between infrastructure-based modeling and the rich context of urban form. It emphasizes the need to combine two types of data (network connectivity and morphology) in a single predictive model and asks how this can be achieved through a suitable deep learning architecture.

### 1.6. Research Sub-Questions

To address the main research question in a structured way, the following sub-questions are formulated. Each sub-question corresponds to a specific aspect of the methodology and is intended to guide the investigation into how to effectively build and use the *ST-SimNet* model:

1. **Architecture Suitability:** *What are the key components and mechanisms of the *ST-SimNet* architecture required to capture both the spatial dependencies and temporal dynamics of urban freight flows?* This sub-question prompts an examination of the model design (e.g. graph convolution layers, temporal sequence layers, etc.) and seeks to verify that the chosen architecture (or any modifications thereof) is capable of learning the complex patterns in the data. It also implies a comparison or baseline check to ensure that the added complexity of a *ST-GNN* is justified by improved performance or insight, thus testing the architecture's merit against simpler alternatives.
2. **Graph Structure Design:** *How should the graph representing the urban freight system be constructed using the available data (road network and spatial units), and what is the impact of different graph design choices on forecasting performance?* Here I investigate how to define the nodes and edges of the graph. For example, should each node correspond exactly to each junction, a Six-digit Dutch postcode (postcode 6) (PC6) area, or are there reasons

## 1.7. Relevance and Significance of the Research Questions

to cluster or subdivide areas? How are edges defined - only between adjacent areas sharing a border, or between areas that are connected by a major road within a certain distance threshold?

- 3. Feature Integration Strategies:** *What is the most effective way to integrate urban morphological features (e.g. building usage, density, and other CBS-derived statistics) into the ST-SimNet model, and how do different feature aggregation or encoding techniques influence the accuracy of freight flow forecasts?* This question focuses on the input features assigned to nodes (and potentially edges). I explore how building-level or zone-level data should be processed and included in the model. The goal is to determine which features and integration methods provide the most predictive power. This could involve experiments where certain features are included or excluded, or different normalisation and embedding techniques are applied, to see their effect on the model's performance. By answering this, it can be ensured that the rich urban morphology data is leveraged in the best possible way within the GNNs framework.

These sub-questions together break down the overarching problem into manageable research tasks: choosing/designing the model architecture, constructing the spatial graph, and incorporating features. Addressing them collectively answers the main research question, as each is a crucial piece of the puzzle in building an effective ST-GNN for freight forecasting.

## 1.7. Relevance and Significance of the Research Questions

Formulating the research questions in the above manner underscores their relevance to both academic literature and practical applications.

### 1.7.1. Academic Contribution

From an **academic perspective**, this research addresses a key limitation in current transportation modelling literature. While spatio-temporal graph neural networks have become powerful tools for mobility and traffic forecasting, they have largely relied on network topology and historical flow data as inputs [Rahmani et al., 2023]. In doing so, they often neglect the finer-grained aspects of urban morphology - such as land use patterns, the spatial distribution of economic activity, and built environment characteristics, which are known to influence freight flows.

This thesis proposes a unified framework, *ST-SimNet*, that integrates road infrastructure with detailed urban morphology descriptors by embedding static morphology features into the graph's node attributes. This enables the model to learn not only how freight flows propagate through a road network, but also why certain areas generate or attract more freight activity than others. Through empirical evaluation, it is shown that incorporating urban morphology significantly improves predictive accuracy, particularly in spatially heterogeneous regions.

This work contributes to the theoretical understanding of how freight transportation interacts with the built environment and demonstrates that integrating morphological data into GNNs can yield more accurate and interpretable models than those based solely on infrastructure or time series data. Furthermore, by adapting ST-GNNs, previously applied mainly

## 1. Introduction

to passenger traffic or coarse mobility patterns to the freight domain, this research validates their applicability to sparser and more irregular datasets, extending the relevance of deep learning in urban freight modelling. These findings have implications for both transportation research and the machine learning community, offering a blueprint for future GNN architectures that incorporate spatial context beyond topology.

### 1.7.2. Practical Relevance

From a **practical standpoint**, this research supports the development of more effective freight planning strategies, particularly in data-rich environments like the Netherlands. Dutch cities - dense, historic, and increasingly strained by e-commerce - present a complex logistics challenge. At the same time, they offer fine-grained, openly available data (e.g. PC6-level CBS statistics, building registries Basisregistratie Adressen en Gebouwen (BAG), and road networks) that make them ideal testing grounds for data-driven mobility models.

By combining these datasets in *ST-SimNet*, this study demonstrates how freight flow forecasting can move beyond purely network-based approaches. In particular, the results show that static urban features can enhance predictive performance, especially in spatially heterogeneous areas. This has real-world implications: municipalities and logistics providers can use such models to anticipate where delivery volumes may surge, where bottlenecks could arise, and how zoning or infrastructure changes may impact urban freight dynamics. For example, areas with high densities of retail or warehouse activity might require policy adjustments such as timed loading zones or local delivery restrictions. Conversely, areas with weak or noisy predictions may benefit from targeted data collection or improved feature attribution, as highlighted by this study's analysis of error patterns linked to sparse or noisy static inputs.

While the focus here is on the Netherlands, the approach is transferable. As more cities begin to assemble comparable datasets, the lessons from *ST-SimNet* can help inform smarter freight management globally. The model's modular architecture and reliance on widely available input types make it particularly suited for integration into real-time planning systems, such as digital twins or urban logistics platforms. In summary, this work not only contributes to academic modelling approaches but also offers actionable guidance for data-informed urban freight management providing a bridge between machine learning techniques and practical city logistics.

## 2. Related work

### 2.1. Spatio-temporal GNNs for Traffic Prediction

Graph Neural Networks have emerged as powerful tools for modelling relational data in diverse domains [Wu et al., 2021]. In transportation, spatio-temporal GNN models are used to capture both the network structure of roads and temporal dynamics of traffic flows. Early studies represented road networks on grid-like structures and applied Convolutional Neural Network (CNN)s, but this ignored irregular road topology (see Figure 2.1)[Xiong et al., 2024].

Subsequent approaches leveraged graph representations built from road sensor networks, using fixed sensor-defined adjacency matrices for graph convolution. However, fixed connectivity fails to reflect changing traffic patterns over time [Xiong et al., 2024]. To address this, Xiong et al. [2024] proposed the Gated Fusion Adaptive Graph Neural Network (GFAGNN), which integrates adaptive graph convolutions and attention mechanisms to capture dynamically changing spatial dependencies in traffic flow. GFAGNN fuses long-term and short-term features via a gating module, yielding improved accuracy over static-adjacency baselines. While effective on sensor data (e.g. traffic speed from loop detectors), GFAGNN relies on sensor-defined graphs and does not incorporate information about the urban morphology or built environment surrounding those sensors. This limitation motivates the inclusion of vector-formatted urban context in the predictive model.

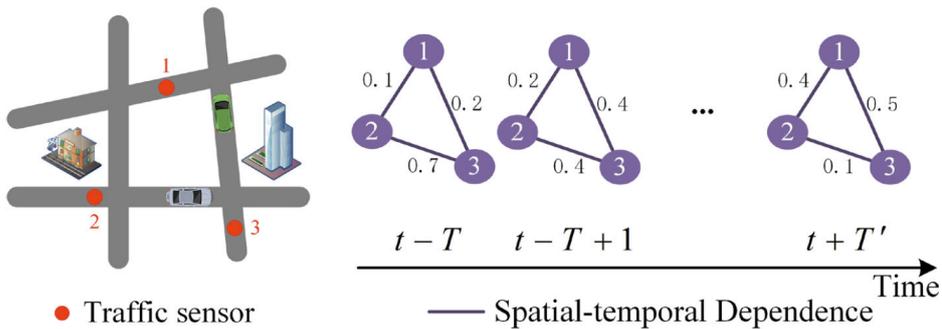


Figure 2.1.: Spatial-temporal correlation is dominated by the road network structure. (a) Traffic sensors distributed in the road network. (b) Dynamic spatial-temporal dependence from time  $t - T$  to  $t + T'$ . Taken from Xiong et al. [2024].

## 2.2. Graph Neural Networks for Road Networks

Beyond traffic time-series forecasting, recent work has focused on the unique characteristics of road network graphs themselves. Standard GNN architectures often assume network homophily (neighbours share similar attributes), an assumption that is weak in road networks which exhibit heterophilic patterns (e.g. a highway connected to local streets) [Jepsen et al., 2019].

[Jepsen et al., 2019] introduced the RFN to tackle these challenges. RFN is an edge-centric GNN that aggregates information from multiple sources: node attributes, edge (road segment) attributes, and “between-edge” relations, to more robustly learn road network representations. This relational approach makes RFN resilient to volatile homophily and sparse connectivity in road graphs. An overview of the RFN architecture is shown in Figure 2.2. Empirically, RFN significantly outperforms conventional Graph Convolutional Network (GCN)s on road-specific tasks like speed limit classification by leveraging the graph’s structural features.

However, RFN is primarily a topological learning framework: it considers static graph properties and does not incorporate temporal dynamics (traffic variation over time). Moreover, RFN’s use of road network attributes is largely internal to the network (e.g., connectivity and road-specific features); it does not model external urban features such as land use, built-form, or other morphological context of the road segments.

In contrast, the proposed ST-SimNet extends the relational learning concept by integrating external urban morphology data with spatio-temporal modelling, enabling the GNN to reason about how the surrounding environment of roads influences freight traffic on them.

## 2.3. Urban Morphology in Graph-based Urban Analysis

The role of urban morphology and context in network modelling has been highlighted by studies outside of pure traffic forecasting. Xue et al. [2021] developed a graph-based method to quantify the spatial homogeneity of urban road networks. Their approach uses GNN embeddings to measure how similar sub-graphs of a city’s road network are to each other, finding that these homogeneity metrics correlate strongly with socioeconomic factors like GDP and population growth. An overview of their concept is illustrated in Figure 2.3, where the similarity between a subnetwork and the global network topology captures the spatial coherence of urban structures. Notably, by transferring their model across 30 different cities, Xue et al. [2021] revealed structural commonalities in road networks globally (e.g. inter-city similarities between networks in Europe and the US).

This underscores the value of incorporating urban topology and external data (such as socioeconomic or land-use indicators) into graph models for understanding cities. However, the work of Xue et al. [2021] remains analytical rather than predictive - it mines structural patterns but does not forecast traffic or freight flows. It also focuses on time-invariant properties (road layout, long-term indicators) and thus lacks a temporal component. In contrast, ST-SimNet builds upon this idea of integrating urban context by not only embedding built environment features, but also modelling their temporal interactions through a spatio-temporal GNN.

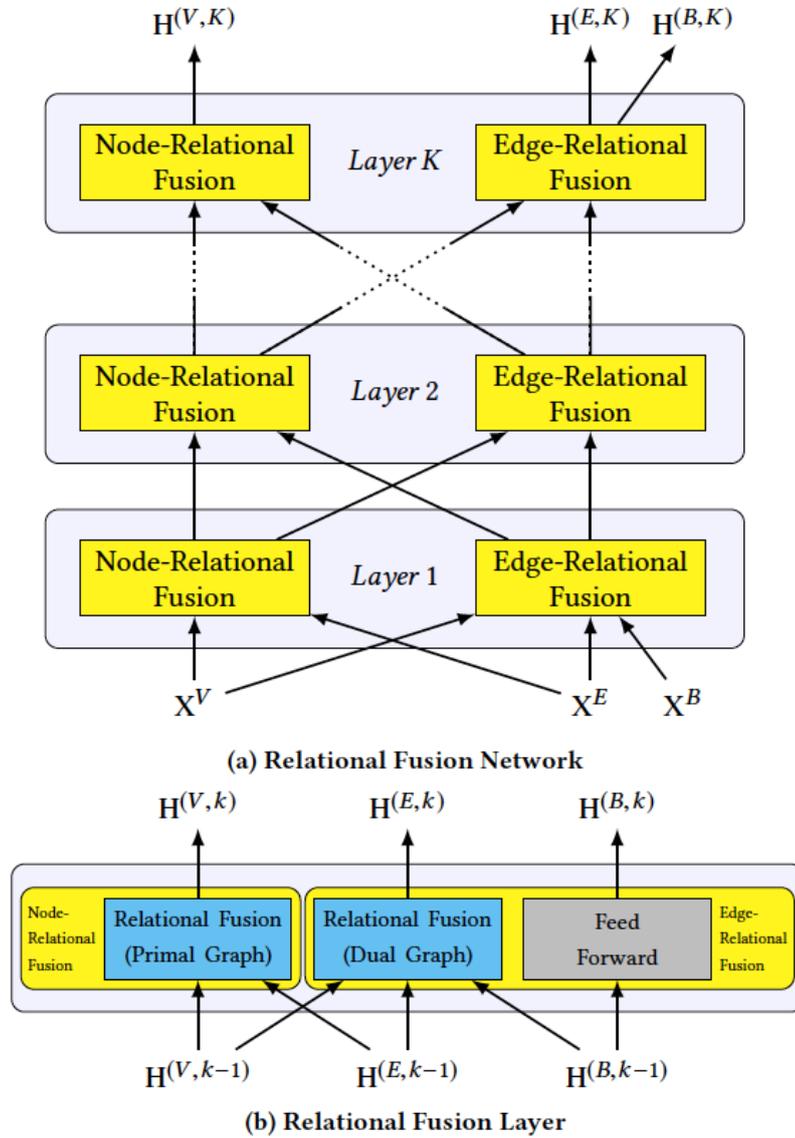


Figure 2.2.: Architecture of the Relational Fusion Network (RFN). RFN fuses node-relational and edge-relational features across multiple layers to improve learning on heterophilic road networks. Taken from Jepsen et al. [2019].

## 2. Related work

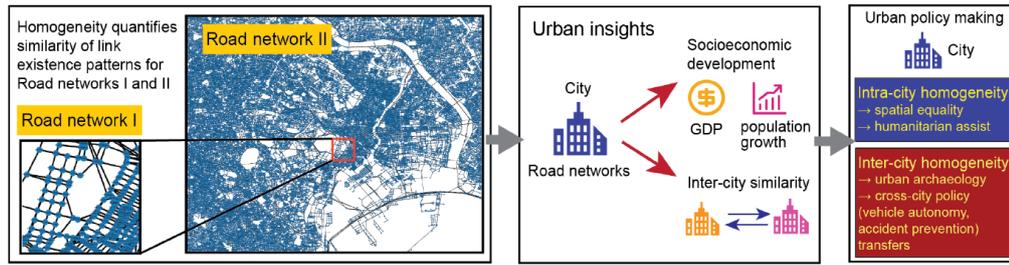


Figure 2.3.: Conceptual overview of spatial homogeneity in urban road networks. Homogeneity is defined as the similarity of link existence patterns between different parts of the same city (intra-city) or across cities (inter-city). The metric, derived from GNN-based link prediction, correlates with socioeconomic indicators such as GDP and population growth, and reveals urban structural similarities transferable across cities. Taken from Xue et al. [2021].

## 2.4. Relational Inductive Biases and Model Generalisability

Across these studies, a common theme is the importance of relational inductive bias — designing models that respect the relational structure of data (roads, connections) to achieve better generalisation [Battaglia et al., 2018]. Battaglia et al. [2018] argue that graph-based architectures inherently encode such inductive biases, enabling combinatorial generalisation in reasoning about entities and their relations. *ST-SimNet* is grounded in this principle: by using the road network graph as the backbone, the model infuses domain knowledge of connectivity and spatial layout. An advantage of GNNs with strong relational inductive bias is their potential to transfer across contexts. Jepsen et al. observed that an RFN trained in one region can be applied to another, thanks to its inductive nature, suggesting the ability to learn traffic dynamics that generalise to unseen road networks [Jepsen et al., 2019]. Similarly, Xue et al. [2021] demonstrated transferring a graph model across cities to compare structural homogeneity.

*ST-SimNet*'s design was conducted with cross-city generalisability in mind. By incorporating vectorized morphology data (which is often available universally, e.g. open street maps and urban Geographic Information System (GIS) layers) along with traffic data, *ST-SimNet* can learn patterns in a data-rich city and apply them to a data-sparse city. This is particularly valuable for freight transportation, where detailed sensor data may be limited in smaller cities. In summary, this approach relies on relational biases (through the graph network) and shared urban features to ensure the model remains robust when deployed in different cities or when only limited local training data are available. This addresses a key limitation of prior spatio-temporal GNN models, which typically require extensive site-specific training data and may not generalise well beyond the original city of training.

## 2.5. Integration with Digital Twin Frameworks

This work is aligned with the growing need for data-driven components in urban DT platforms such as TNO's *Urban Strategy* [TNO, 2023]. These systems simulate city dynamics

to evaluate the impacts of policies and infrastructure changes. In current DT implementations, freight flow is often modelled using traditional techniques like the four-step travel demand model, which rely on assumptions and calibrated parameters rather than learning directly from historical data. *ST-SimNet* offers a complementary approach by serving as a predictive module that learns directly from temporal freight patterns and static urban morphology. Unlike conventional models, it provides fine-grained, node-level forecasts at high temporal resolution, enabling dynamic simulation of freight movement based on observed behaviours.

While this model is not yet integrated into TNO’s DT, it demonstrates the potential to enhance digital twins with short-term forecasting capabilities grounded in machine learning. For instance, the model could be used to estimate how new zoning regulations or warehouse developments affect freight volumes in specific urban areas. This would add a predictive layer that updates continuously as new data becomes available, enabling scenario testing with more realistic behavioural feedback. In this way, *ST-SimNet* bridges the gap between rule-based simulation and adaptive, data-informed urban forecasting.

## 2.6. Baseline STGCN

*STGCN* is a deep learning model that integrates graph-based spatial learning with temporal convolution for traffic forecasting [Yu et al., 2018]. The core architecture is composed of stacked *ST-ConvBlocks*, followed by a fully connected output layer built with a Multi-Layer Perceptron (MLP). In the original design, Yu et al. [2018] used two *ST-ConvBlocks* in sequence, although more can be added for increased model capacity. Each *ST-ConvBlock* features a “sandwich” structure: a GCN layer in the middle, flanked by input and output gated temporal convolutions. The *ST-ConvBlock* design further employs a bottleneck architecture reducing the channel dimensionality prior to the GCN and restoring it afterward, thus decreasing the number of trainable parameters while retaining representational power. Layer normalisation is also applied within each block to enhance generalisation [Yu et al., 2018].

### 2.6.1. Training Pipeline and Datasets

**Input-output formulation:** *STGCN* is trained in a supervised sequence-to-sequence fashion on graph-structured time series data. The input to the model is a sliding window of  $M$  past time steps of traffic measurements (e.g. vehicle speeds) across all sensors in the network. This input is represented as a tensor  $V_{t-M+1:t} \in \mathbb{R}^{N \times M}$ , where  $N$  is the number of graph nodes (i.e., traffic sensors) and  $M$  is the number of past time steps.

The model’s objective is to forecast the future values  $\hat{V}_{t+1:t+H} \in \mathbb{R}^{N \times H}$ , representing the traffic state over the next  $H$  time steps. While  $H = 1$  is often used for single-step forecasting, the model can be extended to multi-step prediction. In the original work by Yu et al. [2018], the authors used  $M = 12$  (representing one hour of history using 5-minute intervals) and predicted up to  $H = 9$  time steps (i.e., 45 minutes into the future), with evaluations reported at 15, 30, and 45-minute horizons.

After passing through two stacked *ST-ConvBlocks*, the learned spatio-temporal features are processed by an additional 1D temporal convolution layer (without gating) to align them

## 2. Related work

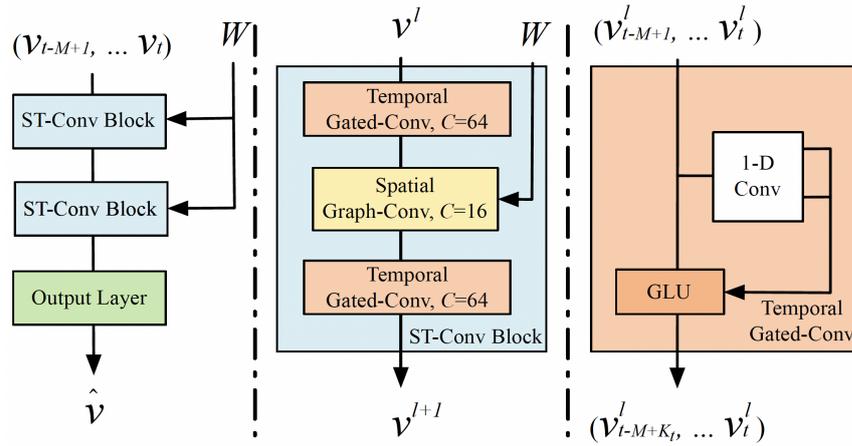


Figure 2.4.: Detailed architecture of the Spatio-Temporal Convolutional Block (ST-ConvBlock) in the original STGCN model [Yu et al., 2018]. The block uses a “sandwich” design with two 1D gated temporal convolutions (GLU-based) on either side of a spatial graph convolution layer. The first temporal convolution reduces feature dimensionality, acting as a bottleneck to improve computational efficiency. The central graph convolution captures spatial dependencies across the graph defined by the road network. The final temporal convolution restores the dimensionality, enabling the model to reconstruct time-dependent patterns. A residual connection links the block’s input to its final output to facilitate gradient flow and model stability during training. This modular design allows the architecture to be stacked in deeper networks, making it both scalable and interpretable for traffic prediction tasks. Figure taken from Yu et al. [2018].

with the desired forecasting horizon. Finally, a fully connected layer produces the output traffic values for each node.

The entire model is trained end-to-end by minimising the L2 loss (Mean Squared Error) between the predicted outputs and the ground truth values:

$$\mathcal{L} = \frac{1}{NH} \sum_{i=1}^N \sum_{j=1}^H (V_{i,t+j} - \hat{V}_{i,t+j})^2$$

Optimisation is performed using stochastic gradient descent techniques, such as the Adam optimiser. Due to its convolutional nature, *STGCN* avoids sequential computation and can be parallelised across time steps, resulting in faster training compared to recurrent models.

**Datasets (METR-LA and PEMS-BAY):** The original *STGCN* paper evaluated model performance on two real-world benchmark datasets: METR-LA and PEMS-BAY. These datasets have since become standard references in traffic forecasting literature.

The **METR-LA** dataset comprises traffic speed readings collected from 207 loop detectors installed on Los Angeles County freeways. The data spans approximately four months, from March to June 2012, and is recorded at 5-minute intervals. The **PEMS-BAY** dataset contains speed measurements from 325 sensors deployed across the San Francisco Bay Area, covering a six-month period in 2017 (January to June), also with 5-minute resolution.

Each dataset is accompanied by a predefined sensor network graph. In this graph:

- **Nodes** represent traffic sensors.
- **Edges** encode spatial relationships based on road proximity. These are weighted using a Gaussian kernel applied to pairwise distances between sensors, yielding an adjacency matrix  $A \in \mathbb{R}^{N \times N}$  where  $A_{ij}$  reflects the strength of spatial interaction between sensor  $i$  and  $j$ .

This adjacency matrix is central to the spatial modelling component of *STGCN*, guiding the message-passing in graph convolution layers.

The model is trained to predict future traffic speeds at each node using sliding windows of historical data. Evaluation is conducted across multiple forecasting horizons (e.g., 15, 30, and 60 minutes ahead), using common regression metrics: Mean Absolute Error (**MAE**), Root Mean Squared Error (**RMSE**), and Weighted Mean Absolute Percentage Error (**WMAPE**). These benchmarks demonstrate the ability of *STGCN* to model complex spatio-temporal dependencies in urban road networks.

## 2.6.2. Performance and Contributions

**Accuracy and Generalisation:** *STGCN* demonstrated strong predictive accuracy and generalisation across multiple traffic forecasting benchmarks, substantially outperforming traditional and deep learning baselines. On the widely used METR-LA and PEMS-BAY datasets, *STGCN* achieved consistently lower prediction errors than classical statistical models such as ARIMA and Support Vector Regression (SVR), as well as surpassing deep learning alternatives like feedforward neural networks and LSTMs across several forecast horizons.

## 2. Related work

In experiments on a Beijing traffic dataset and a subset of PEMS, Yu et al. [2018] reported that *STGCN* reduced the 15-minute Mean Absolute Error (MAE) by approximately 10% compared to a standard LSTM, and even outperformed the graph-enhanced GRU baseline (GCGRU) for both short- and medium-term predictions. The performance gains were particularly notable on larger, more topologically complex networks, such as California’s PeMS sensor system, where the ability to model spatial dependencies via graph convolution enabled more accurate learning of congestion propagation and traffic patterns.

Furthermore, *STGCN* showed enhanced capacity to capture sharp temporal dynamics, such as the onset and dissipation of rush-hour congestion. The model was able to predict the timing of peak transitions more effectively than recurrent models, which often lagged due to their sequential processing limitations. This behaviour underscores *STGCN*’s ability to generalise to dynamic urban conditions rather than overfitting to historical averages. By jointly modelling spatial graph structures and temporal convolution, *STGCN* set a new benchmark for spatio-temporal traffic prediction at the time of its introduction, demonstrating robustness across cities, road network topologies, and temporal scenarios.

### 2.6.3. Relevance as a Baseline:

Since its introduction, the *STGCN* model by Yu et al. [2018] has become a widely accepted baseline for spatio-temporal prediction tasks, particularly in traffic forecasting. Owing to its strong performance, conceptual clarity, and computational efficiency, it is frequently used as a benchmark against which newer methods are evaluated. The influence of *STGCN* has extended beyond its original scope, helping to standardise datasets such as METR-LA and PEMS-BAY as canonical testbeds within the community.

Numerous subsequent models have drawn architectural inspiration from *STGCN*, extending it with mechanisms such as attention layers, dynamic graph construction, or residual learning schemes. Nevertheless, the original *STGCN* retains its value due to the balance it strikes between predictive accuracy and architectural simplicity. Its modular spatio-temporal block design and fully convolutional structure make it both interpretable and efficient, enabling rapid experimentation and adaptation to new domains.

## 2.7. Summary of Contributions of ST-SimNet

*ST-SimNet* advances the field of spatio-temporal graph learning by proposing a novel extension to the established *STGCN* architecture, specifically tailored for freight forecasting in morphologically complex urban environments. While *STGCN* demonstrated that gated temporal convolutions combined with graph convolutions can effectively model traffic dynamics, it was limited to homogeneous, sensor-rich contexts and lacked integration of static urban features.

This thesis introduces architectural innovations, contextual enhancements, and generalisability improvements to address these limitations. The three key contributions are as follows:

1. **Morphological Feature Fusion into *STGCN*:** *ST-SimNet* systematically extends *STGCN* by introducing a learnable fusion block that integrates static urban morphology descriptors with dynamic freight data, adds dropout, weight decay, set of different optimisers (eg. adamw, nadamw, lion), and early stopping. Unlike prior GNN-based

models (e.g., Diffusion Convolutional Recurrent Neural Network (DCRNN), GFAGNN) that rely solely on time-series from traffic sensors, *ST-SimNet* incorporates node-level features such as land use, building footprint, floor area ratio, and demographic statistics. These features are embedded and fused via a gated mechanism during training, enabling the model to assign dynamic importance to static inputs. This approach enables interpretable, spatially-aware learning and moves beyond static concatenation or early fusion methods.

- 2. Deepened Temporal Context with Configurable ST-Conv Blocks:** *SST-SimNet* generalises the sandwich architecture of *STGCN* by introducing a tunable number of stacked *ST-ConvBlocks* and allowing for controlled temporal kernel sizes. This design enables the model to capture longer-term dependencies across multiple peak periods while maintaining the spatial topology of the road network. In contrast to recurrent or pooled architectures, *ST-SimNet* leverages fully convolutional processing with gated activations (e.g., GLU, GTU), enhancing scalability, parallelisation, and training stability. By explicitly controlling the compression depth via  $K_o$ , the model can dynamically adjust its output mechanism to preserve or summarise temporal patterns as required.
- 3. Designed for Cross-City Generalisability and Digital Twin Readiness:** A key architectural goal of *ST-SimNet* is to enable generalisation across cities with heterogeneous spatial morphology and sparse training data. By incorporating relational inductive bias and morphology-aware static features, the model is designed to adapt to new geographies with minimal retraining. Although this thesis focuses on a single urban context, the modular and lightweight architecture is suitable for real-time inference and integration into digital twin frameworks such as TNO's Digital Twin. This sets the foundation for future applications in cross-regional freight forecasting where transferability is critical.

Together, these contributions address three persistent gaps in the literature: (i) the absence of morphological and land-use context in adaptive traffic *GNNs*; (ii) the lack of structured temporal depth in graph-based freight forecasting models; and (iii) the limited generalisation capacity of sensor-dependent models trained on single-city datasets. *ST-SimNet* proposes a modular, interpretable, and operationally deployable architecture that advances the predictive modelling of urban freight systems and sets the stage for morphology-aware integration in digital twin ecosystems.



## 3. Methodology

### 3.1. Notation and Glossary

For readers' convenience, I attach additional glossary of symbols used in *ST-SimNet* (see Table 3.1).

Table 3.1.: Glossary of Symbols Used in *ST-SimNet*

Symbol	Description
$N$	Number of nodes (road junctions) in the graph
$F$	Number of input features per node (dynamic)
$F_s$	Number of static urban morphology features per node
$T$	Total number of time steps in the dataset
$L$	Length of input history window (e.g. 24 = 2 hours)
$H$	Prediction horizon in time steps (e.g. 12 = 1 hour)
$B$	Batch size (number of training samples per batch)
$K_t$	Temporal convolution kernel size (in number of time steps)
$K_s$	Spatial convolution kernel size (number of hops in GCN)
$\mathbf{x}_t$	Dynamic freight flow vector at time $t$ , $\in \mathbb{R}^N$
$x_{i,t}$	Freight flow at node $i$ and time $t$
$\hat{y}_{i,t}$	Predicted freight flow at node $i$ and time $t$
$\mathbf{s}_i$	Static feature vector of node $i$ , $\in \mathbb{R}^{F_s}$
$\mathbf{A}$	Adjacency matrix representing road connectivity
$\tilde{\mathbf{A}}$	Graph Shift Operator (GSO), row-normalised
$\mathbf{W}$	Trainable weight matrix in graph convolution layers
$\mathbf{h}_i$	Intermediate node feature matrix after ST-Conv operations
$\hat{\mathbf{y}}$	Model output vector with predicted flows
$\alpha$	Learnable parameter controlling fusion of static and dynamic features
$z$	Latent feature representation after temporal aggregation
$\mathcal{L}$	Loss function (e.g. Mean Squared Error, MSE)
$\eta$	Learning rate used for optimiser (e.g. AdamW)
$p$	Dropout rate, regularisation strength
$\lambda$	Weight decay coefficient (L2 regularisation)
$\gamma$	Learning rate decay factor for scheduler
step.size	Epoch interval at which to apply learning rate decay

## 3.2. Overview

The proposed methodology uses a *STGCN* approach to model urban freight traffic, leveraging both spatial dependencies (road network structure) and temporal patterns (time-varying flows). Traffic and freight flow data are highly nonlinear and complex, exhibiting strong correlations across both time and space [Al Sahili and Awad, 2023]. Traditional time-series models or *CNN*/Recurrent Neural Network (*RNN*)-based approaches struggle to capture these interdependencies on non-Euclidean road networks [Yu et al., 2018]. I therefore formulate the prediction problem on a graph representation of the city and employ an *ST-GNN* architecture to jointly learn spatial and temporal features. In particular, the model *ST-SimNet* is inspired by the *STGCN* framework [Yu et al., 2018], with modifications to integrate static urban morphology data. By using *ST-GNN*, the model can encode how freight flows propagate through the road network over time, and handle the complex, non-stationary dynamics of urban freight transport [Al Sahili and Awad, 2023]. In summary, this methodology is chosen to exploit the strengths of graph neural networks for learning patterns in network-structured data and improve predictive performance for freight flows.

## 3.3. Input Data

This research employs three categories of data to achieve its forecasting objectives: (i) dynamic freight flow data derived from *DT* outputs of Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek (Dutch Organisation for Applied Scientific Research) (*TNO*), based on simulations from *MASS-GT* and *VMA*, capturing temporal variation in freight vehicle volumes at road network nodes; (ii) static *UMDs*, including building attributes and *PC6*-level socio-demographic indicators from *CBS*, providing spatial context; and (iii) a directed road network graph encoding the connectivity of the urban infrastructure. These inputs are integrated within the *ST-SimNet* architecture, an extension of the *STGCN*, to forecast short-term freight intensity at each node, one hour into the future, at 5-minute intervals. The model aims to minimise predictive error by learning spatio-temporal dependencies in freight flow, enriched by morphology-aware node representations derived from static contextual features.

### 3.3.1. Dynamic Freight Flow Data

The dynamic input data consist of time-series measurements of freight traffic intensity at the road level, obtained from *TNO*'s Digital Twin simulations. In particular, I use output from the Multi-Agent Simulation System for Goods Transport (*MASS-GT*) and the Verkeersmodel Amsterdam (*VMA*) model processed by *TNO*'s *DT*. *MASS-GT* is an agent-based urban freight model that simulates logistic decisions and shipment movements in cities [de Bok and Tavasszy, 2018], producing detailed truck delivery tours and schedules. *VMA* is a small-scale traffic assignment model for the Amsterdam region, with thousands of zones and road links, capable of routing vehicular trips through the road network [Spruijtenburg et al., 2025]. Combined, these two models provide a realistic proxy for urban freight flows: *MASS-GT* generates the freight trips (origins, destinations, timing, vehicle types), and *TNO*'s *DT* assigns these trips to specific road paths, yielding traffic volumes on road segments over

time. The result is a spatio-temporal dataset of freight vehicle counts on road network links for each time interval.

For use in *ST-SimNet*, the model operates directly at the node level, with each node corresponding to a specific point in the road network. The dynamic input  $x_{i,t}$  for node  $i$  at time  $t$  represents the local freight traffic intensity at that node - the number of freight vehicles traversing or stopping near that location within a 5 minutes interval. These values are pre-processed from simulated truck trips and structured as a time series per node with a time resolution of 5 minutes. I denote by  $\mathbf{x}_t \in \mathbb{R}^N$  the vector of freight intensities across all  $N$  nodes at time  $t$ . To stabilise training and handle the strong variability in flow magnitudes (e.g. peak vs off-peak hours), the input series are transformed using a  $\log(1+x)$  transformation followed by Min-Max normalisation to the  $[0,1]$  range. The resulting temporal resolution is fixed at  $\Delta t = 5$  mins, and the full sequence  $\{\mathbf{x}_t\}_{t=1}^T$  forms the dynamic input to the model during training and evaluation.

The dataset covers two temporal configurations, both with a resolution of  $\Delta t = 5$  minutes. The weekday-only scenario is based on one average working day, resulting in 289 time steps per node ( $24 \text{ hours} \times 12 \text{ intervals/hour} + 1 \text{ initial step}$ ). To create additional weekday data, this single-day sequence was modulated using small perturbations to generate synthetic but realistic temporal variability across five working days. The weekend dataset is entirely synthetic, designed to reflect plausible low-activity patterns and behavioural shifts typical of Saturday and Sunday logistics, followed by the inclusion of the first half of Monday to capture ramp-up effects. This extended configuration totals 2,448 time steps per node.

For the full Amsterdam network of  $N = 10,691$  nodes, this yields a dynamic input tensor of shape  $[T, N] = [289, 10,691]$  for the weekday-only model, and  $[2,448, 10,691]$  for the mixed scenario. Each time step records the estimated freight vehicle count at every node, derived from the *MASS-GT* and *VMA* model outputs. This setup enables the evaluation of model robustness under both consistent weekday conditions and more heterogeneous temporal scenarios involving weekend variations.

### 3.3.2. Urban Morphology and Road Networks

This section describes the static urban morphology features and road network graph construction.

The building dataset is characterized by a set of static attributes capturing the urban morphology of each building in every PC6 area. These features are derived from building-level data and include attributes such as the predominant land-use function, the total building floor area (size), and occupancy indicators - precise features included are discussed in section 3.3.3. The land-use function feature encodes the type of activities in the zone (for example, residential, commercial, industrial, or mixed-use), which correlates with freight demand generation (industrial or retail zones tend to send/receive more goods than purely residential areas). The building size feature (e.g. total floor space or number of units) provides a measure of the capacity or scale of activity in the zone. Occupancy data (such as the number of businesses or households, or occupancy rates) reflect the intensity of use of the buildings. These building-level attributes are aggregated to nearest nodes by summing or averaging across all buildings in the zone. The result is a feature vector  $\mathbf{s}_i$  for each node  $i$  representing the static built environment characteristics of that zone. These urban morphology features serve as time-invariant inputs that can inform the model about each location's freight generation potential and usage context.

### 3. Methodology

In parallel, the city's road network is used to define the graph structure connecting the nodes. A directed graph is constructed  $G = (V, E)$  where each node  $v_i \in V$  corresponds to a road junction, and edges  $e_{ij} \in E$  are established between two nodes  $i$  and  $j$  if there is a direct road connection between them. In practice, this means that if any road segment in the underlying street network links node  $i$  to node  $j$ , then an edge is added  $(i, j)$  in the graph. The adjacency relationships are determined using the spatial road network data. This approach yields a graph that mirrors the actual connectivity of the city: nodes that are adjacent or well-connected by roads become neighbors in the graph. Let  $\mathbf{A} \in \mathbb{R}^{N \times N}$  denote the adjacency matrix of the graph (with  $A_{ij} = 1$  if nodes  $i$  and  $j$  are connected, otherwise 0). The graph construction based on road network topology ensures that spatial information is explicitly encoded for the GNN. In total, this study area graph consists of  $N$  nodes and a set of edges reflecting the road network links between those zones. This static graph  $G$  is used throughout training to perform graph convolutions over the freight flow data.

#### 3.3.3. Data Preprocessing

**Data Cropping** To reduce processing overhead and limit the analysis to the metropolitan area of Amsterdam, a bounding box is first applied to crop the network. The resulting subset includes only those nodes and edges that fall within this spatial boundary. For efficient storage and integration with GIS tools, the cropped network was saved as GeoPackage (.gpkg) format.

This spatially filtered network forms the foundation for all subsequent operations. Figure 3.1 illustrates the result of this cropping step, showing the road network constrained to the relevant spatial extent (big simulation scenario).

---

**Algorithm 3.1:** Crop a road network to a bounding box and save as .gpkg

---

**Input:** Road network dataset with spatial coordinates; bounding box coordinates file (CSV)

**Output:** Cropped road network saved as GeoPackage (.gpkg)

- 1 Load the bounding box coordinates from CSV file;
  - 2 Extract columns representing X and Y coordinates;
  - 3 Create a polygon from the coordinate pairs to define the bounding box;
  - 4 Convert the polygon into a GeoDataFrame with the appropriate CRS (e.g. EPSG:28992);
  - 5 Load the original road network as a GeoDataFrame;
  - 6 Perform spatial intersection between the road network and the bounding box polygon;
  - 7 Store only features within the bounding box;
  - 8 Save the cropped network to disk using the GeoPackage format;
- 

**Generation of Adjacency Matrix** Following the spatial cropping of the road network, a directed graph structure was instantiated using the `networkx.DiGraph` class (see Figure 3.2). Nodes and edges, as derived from the GeoPackage files - cropped data, were added to this graph to reflect the true topological structure of the urban road network. The directed nature

of the graph allows the model to account for one-way streets and directional traffic flows, which are common in urban settings.

From the constructed directed graph, a sparse adjacency matrix was generated to encode node-to-node connectivity. The use of a sparse matrix format is particularly advantageous in this context due to the inherent sparsity of urban road networks - most nodes are connected to only a limited subset of others, reflecting the physical layout of streets and intersections. Representing this structure in a dense format would result in substantial memory inefficiencies, as the majority of matrix entries would be zero, and would incur unnecessary computational overhead during graph operations.

---

**Algorithm 3.2:** Build a directed graph and generate its sparse adjacency matrix

---

**Input:** Cropped node and edge shapefiles

**Output:** Sparse adjacency matrix representing the directed road network

- 1 Load node shapefile into a `GeoDataFrame`;
  - 2 Load edge shapefile into a `GeoDataFrame`;
  - 3 Create an empty directed graph  $G$  using `networkx.DiGraph()`;
  - 4 Extract a set of valid node IDs from the node data;
  - 5 **foreach**  $edge(a, b)$  in edge data **do**
  - 6     **if**  $a$  and  $b$  are both in the set of valid node IDs **then**
  - 7         Add a directed edge from  $a$  to  $b$  in  $G$ ;
  - 8 Convert the directed graph  $G$  into a sparse adjacency matrix using `scipy.sparse`;
  - 9 Save the sparse matrix to disk in compressed `.npz` format;
- 

A sparse representation, by contrast, stores only the non-zero elements, enabling more efficient storage and significantly faster linear algebra computations during both model training and inference. This efficiency becomes critical as the network scales to hundreds or thousands of nodes. The sparsity structure of the adjacency matrix is visually illustrated in Figure 3.3, where white space denotes the absence of a direct connection between node pairs. The strong scattered clusters reflect both local connectivity and the modular structure of the urban network.

### 3. Methodology

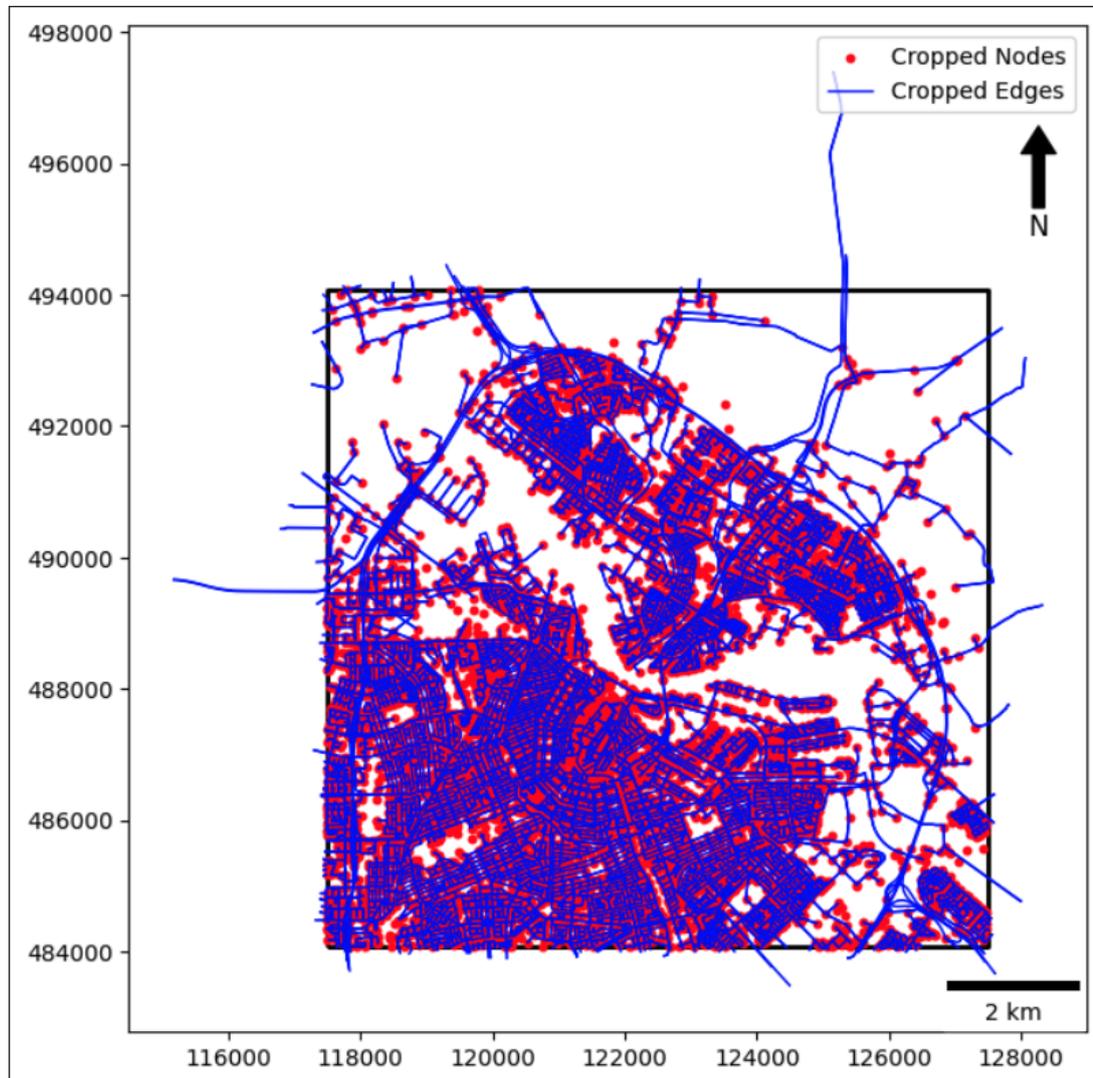


Figure 3.1.: Cropped road network showing nodes and edges within the Amsterdam study area EPSG:28992.



Figure 3.2.: Graph structure of the road network built from spatial data using `networkx.DiGraph` EPSG:28992.

### 3. Methodology

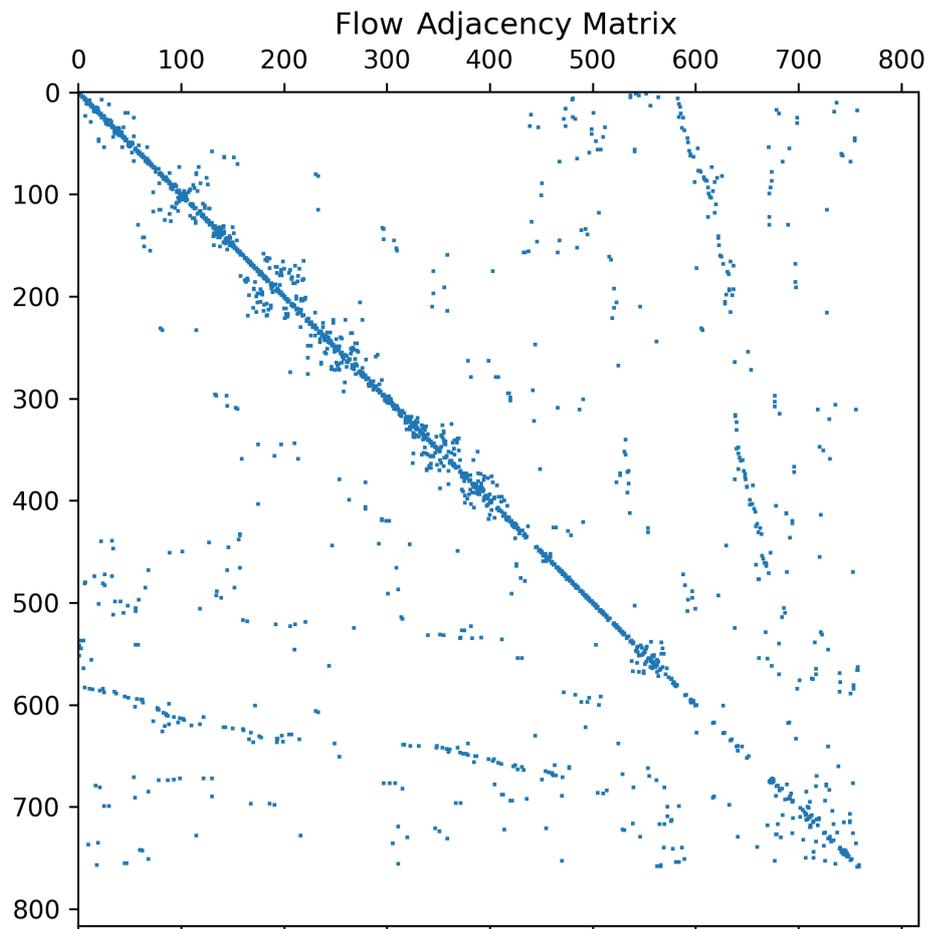


Figure 3.3.: Sparsity pattern of the adjacency matrix. White space indicates absence of direct connection between node pairs.

**Buildings Dataset** To incorporate fine-grained urban morphological context into each node, the building dataset is used as a source of high-resolution descriptors. Each building point includes attributes such as construction year, function, volume, and facade orientation. These attributes are spatially joined to graph nodes using a nearest-neighbour approach: each building is assigned to its closest graph node based on Euclidean distance in EPSG:28992 coordinate space. For each node, a group-wise aggregation is performed on the set of buildings assigned to it. Aggregation strategies include median for numerical temporal attributes (e.g. construction year), sum for quantities (e.g. building volume, number of dwelling units), and mode for categorical features (e.g. building function or energy label).

The result is a node-level morphological profile, effectively transforming building-level micro-data into structured urban descriptors attached to the graph. This transformation is visualised in Figure 3.4, where nodes are colour-coded by the number of buildings they inherit, indicating the spatial variability of morphological coverage across the network. Nodes without any nearby buildings receive a zero vector in this version of *ST-SimNet* and rely solely on dynamic data and static data inherited from PC6 data.

---

**Algorithm 3.3:** Assign building data to nearest graph nodes and compute aggregated morphological descriptors

---

**Input:** Graph nodes (GeoDataFrame), buildings dataset with morphological attributes (CSV)

**Output:** Node-level urban morphology profiles

- 1 Load the building data and convert it into a GeoDataFrame with CRS EPSG:28992;
  - 2 Load the graph node data from shapefile with the same CRS;
  - 3 For each building, find the nearest graph node using spatial join;
  - 4 Assign the building and its attributes to that node;
  - 5 Group buildings by their assigned node;
  - 6 **foreach** *node* **do**
    - 7   Aggregate building attributes:
      - Use median for temporal features (e.g. construction year)
      - Use sum for quantities (e.g. volume, units)
      - Use mode for categorical fields (e.g. function type)
  - 8 Attach the aggregated attributes to the corresponding nodes in the graph;
  - 9 **foreach** *node without buildings* **do**
  - 10   Assign a zero vector as morphological input;
- 

**PC6 Dataset** In addition to fine-grained building-level features, broader contextual indicators are incorporated from the six-digit postcode (PC6) dataset provided by CBS. This dataset includes aggregated socio-demographic attributes such as total population, number of dwellings, proportion of owner-occupied housing, and household composition.

To spatially constrain the dataset to the study area, the national PC6 polygons are first cropped to a predefined bounding box, as shown in Algorithm 3.4. Then, each node in the road network is spatially assigned to a PC6 area using a point-in-polygon operation

### 3. Methodology



Figure 3.4.: Node-level distribution of inherited buildings, EPSG:4326.

(Algorithm 3.5). If a node falls within a PC6 boundary, it inherits the corresponding demographic attributes. These features are appended directly to the node’s attribute vector and are treated in the same way as building-derived features in the model. While building-level data informs the immediate built environment, the PC6 attributes provide higher-level demographic context, offering an additional layer of abstraction over local freight generation potential.

---

**Algorithm 3.4:** Crop the national PC6 polygon dataset to the study area using a bounding box

---

**Input:** Full PC6 polygon dataset, bounding box geometry (GeoPackage)

**Output:** Subset of PC6 polygons cropped to area of interest

---

- 1 Load the bounding box as a GeoDataFrame;
  - 2 Retrieve spatial bounds:  $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$ ;
  - 3 Load the full PC6 dataset using the bounding box as a spatial filter;
  - 4 Optionally save the cropped PC6 polygons to disk as a new GeoPackage;
- 

**Choice of Features** While a wide array of features was available from both the building-level and PC6-level datasets, a subset was selected to ensure relevance, interpretability, and consistency across nodes.

From the building dataset, features were aggregated per node using spatial proximity (nearest neighbour) and summarised via meaningful statistics. The selected features can be found in Table 3.2

---

**Algorithm 3.5:** Assign socio-demographic attributes from PC6 areas to graph nodes using point-in-polygon join

---

**Input:** Graph nodes (GeoDataFrame), cropped PC6 polygons with demographic attributes

**Output:** Node-level demographic profiles from PC6 zones

- 1 Ensure both PC6 polygons and node dataset are in EPSG:28992;
  - 2 Perform point-in-polygon spatial join to assign each node to a PC6 zone;
  - 3 **if** *node lies within a PC6 polygon* **then**
  - 4   └ Copy the socio-demographic attributes from PC6 to the node;
  - 5 Append PC6 attributes to the node's feature vector;
  - 6 **foreach** *node outside any PC6 zone* **do**
  - 7   └ Assign default or missing values;
- 

Table 3.2.: Selected Features for *ST-SimNet* Node Enrichment

Feature Source	Selected Features and Description
Building-Level	<p>bouwjaar (median): Median construction year; reflects building age near each node.</p> <p>verblijfsobjecten, oppervlakteverblijfsobjecten, volume (sum): Indicate built intensity.</p> <p>gem_hoogte, gem_bouwlagen (mean): Capture vertical structure.</p> <p>north_shared_length, north_facade_length (sum): Estimate facade exposure.</p> <p>distance (mean): Mean distance from node to nearby buildings.</p> <p>ndvi_mean_100m, ntl_mean_500m, and their std. dev. (mean): Indicate environmental context.</p> <p>meestvoorkomendelabel, function, building_function, residential_type, non_residential_type (mode): Capture dominant usage types.</p> <p>building_count: Number of buildings near node; proxy for density.</p>
PC6-Level	<p>aantal_inwoners, aantal_woningen: Population and housing counts.</p> <p>percentage_koopwoningen: Home ownership indicator.</p> <p>aantal_part_huishoudens: Private household count.</p>

*Note:* Selected variables reflect structural, functional, and contextual diversity across urban space.

### 3. Methodology

The selection aimed to balance structural descriptors (e.g. volume, facade length), functional indicators (e.g. building use types), and environmental or socio-demographic variables (e.g. NDVI, NTL, population). Features with excessive missing values, high redundancy, or limited variation across the study area were excluded - often the case with PC6 data.

## 3.4. ST-SimNet Architecture

The architecture of *ST-SimNet* is based on the original *STGCN* model proposed by Yu et al. [Yu et al., 2018]. While the foundational structure of spatio-temporal graph convolutional blocks is preserved, the implementation has been substantially reworked and extended to suit the specific needs of freight flow forecasting.

The initial codebase was adapted from an open-source PyTorch implementation by hazdzz<sup>1</sup>, which itself was a reimplementaion of the original TensorFlow version. This PyTorch foundation enabled greater modularity and experimentation. On top of that base, I further developed and adapted the architecture in several key ways:

- Rewritten the data pipeline and training logic for traffic flow prediction (in terms of freight volume) rather than velocity estimation.
- Modified the loss functions and model output structure to align with node-level freight intensity as the prediction target.
- Replaced original activation functions in forward pass of graph convolution with softmax.
- Introduced a dedicated Fusion Block to integrate static urban morphology descriptors (UMDs) via convex combination.
- Implemented flexible graph construction using directed Graph Shift Operators (GSOs), with support for random-walk normalisation and asymmetric edge weights.

The resulting architecture consists of two stacked *ST-ConvBlocks*, each combining temporal convolution layers, graph convolution layers, and dropout for regularisation. After dynamic processing, the temporally aggregated node embeddings are combined with projected static features in a learnable fusion layer. A final MLP maps the fused representation to predicted node-level freight flow at a fixed time horizon. The model is trained end-to-end using gradient descent, minimising either MAE or Mean Squared Error (MSE) loss depending on evaluation goals.

This extended *ST-SimNet* framework enables fine-grained and spatially aware short-term freight prediction and supports integration with additional static or contextual data modalities in future iterations.

Each *ST-ConvBlock* performs a sequence of operations that jointly model temporal dynamics and spatial dependencies. As shown in the centre of Figure 3.5, each block begins with a temporal convolution layer along the input sequence, followed by a graph convolution over the spatial structure of the network using a directed GSO, and concludes with a second temporal convolution. The temporal convolutions are implemented as 1D convolutions with GLUs, which introduce a learnable gating mechanism to filter relevant temporal signals while suppressing noise. Each GLU splits the feature map into candidate and gate components,

---

<sup>1</sup><https://github.com/hazdzz/stgcn>

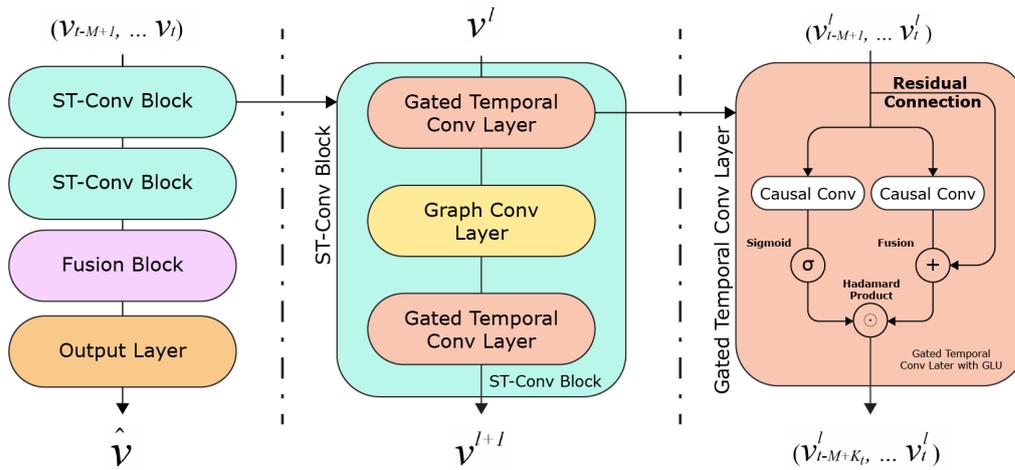


Figure 3.5.: Overview of the *ST-SimNet* architecture. Each coloured region highlights a core module: the blue components represent the two stacked *ST-ConvBlocks*; the purple section denotes the Fusion Block where static *UMDs* are integrated; and the orange module at the end represents the Gated Temporal Convolution layer used in each *ST-Conv Block*.

applying a sigmoid activation to the gate and computing an element-wise product. This mechanism is particularly effective in isolating meaningful local patterns such as delivery peaks or troughs [Dauphin et al., 2017].

Spatial dependencies are captured using a *GCN* defined over the road network topology. The adjacency matrix is directional and normalised using a random-walk scheme to preserve the flow semantics of urban freight. No self-loops are added, and asymmetry is maintained to respect the directionality of road connectivity. These two *ST-ConvBlocks* form the core of the dynamic processing pipeline and are visualised in the blue and purple regions of Figure 3.5.

Following the dynamic encoding, the architecture performs a temporal aggregation step, averaging feature maps across the time dimension. This results in a single latent vector per node that summarises recent freight flow dynamics. At this stage, static information is introduced through a Fusion Block, illustrated in purple in Figure 3.5. Here, static *UMDs* are projected into the latent space via a dedicated two-layer *MLP*. The projected static vector is then combined with the dynamic vector using a convex combination controlled by a learnable scalar fusion coefficient  $\alpha \in [0, 1]$ . This operation produces a unified node representation that encodes both temporal patterns and morphology-aware spatial context.

Following the fusion of dynamic and static representations, the final output layer consists of a fully-connected *MLP* that maps each fused node embedding to a scalar prediction. The result is a vector of length  $N$ , where each value represents the forecasted freight flow at a node in the road network. In this configuration, the model predicts the freight intensity one hour into the future (i.e., 12 time steps ahead at a 5-minute resolution), using the fused spatio-temporal representation as input.

*ST-SimNet* is designed to generalise across heterogeneous spatial contexts, leveraging both temporal dynamics and static urban form. Its modular structure enables transparent inte-

### 3. Methodology

gration of additional data streams, and the fusion mechanism allows for adaptive weighting of static versus dynamic information per node. The architecture is trained using MAE loss, with regularisation applied via dropout and weight decay. All components are optimised jointly using backpropagation.

#### 3.4.1. Data Preparation and Loading

The *ST-SimNet* model relies on prepared spatial-temporal input data and a graph structure reflecting the connectivity of road infrastructure. This section outlines the core preprocessing steps, particularly the loading of dynamic freight flow data and the construction of the Graph Shift Operator (GSO) from a directed road network.

**Adjacency Matrix Loading** The road connectivity is stored in a compressed sparse adjacency matrix `adj.npz`. This matrix is loaded in CSC format to enable efficient matrix operations during training. For freight modeling, the adjacency matrix is *directed*, meaning entries  $A_{ij} = 1$  imply a one-way connection from node  $i$  to node  $j$ . For the Amsterdam case, the number of nodes is  $N = 817$ —for small simulation, and  $N = 10691$ .

**Graph Shift Operator (GSO)** Unlike typical GNN setups that symmetrise the adjacency matrix or inject self-loops, *ST-SimNet* computes a GSO that preserves directionality and avoids self-connections, as described earlier.

The GSO is derived using *random walk normalization based on out-degree*, ensuring the matrix rows sum to one and preserving the interpretation of flow leaving each location. Formally, the selected GSO  $\tilde{\mathbf{A}}$  is computed as:

$$\tilde{\mathbf{A}} = D_{\text{out}}^{-1} \mathbf{A},$$

where  $D_{\text{out}}$  is the diagonal matrix of out-degrees.

---

**Algorithm 3.6:** Compute random walk normalised adjacency matrix using out-degree

---

**Input:** Directed adjacency matrix  $A \in \mathbb{R}^{N \times N}$

**Output:** Random-walk normalised adjacency matrix  $\tilde{\mathbf{A}}$  (out-degree based)

- 1 **if**  $A$  is not sparse **then**
  - 2     └ Convert  $A$  to a sparse CSC matrix;
  - 3 **if**  $A$  format is not CSC **then**
  - 4     └ Convert  $A$  to CSC format;
  - 5 Compute out-degree vector:  $d_{\text{out}} \leftarrow \sum_j A_{ij}$  for each node  $i$ ;
  - 6 Replace zero entries in  $d_{\text{out}}$  with 1 to avoid division by zero;
  - 7 Compute inverse degrees:  $d^{-1} \leftarrow \text{np.power}(d_{\text{out}}, -1)$ ;
  - 8 Form diagonal matrix  $D_{\text{out}}^{-1}$  from  $d^{-1}$ ;
  - 9 Compute GSO:  $\tilde{\mathbf{A}} \leftarrow D_{\text{out}}^{-1} \cdot A$ ;
-

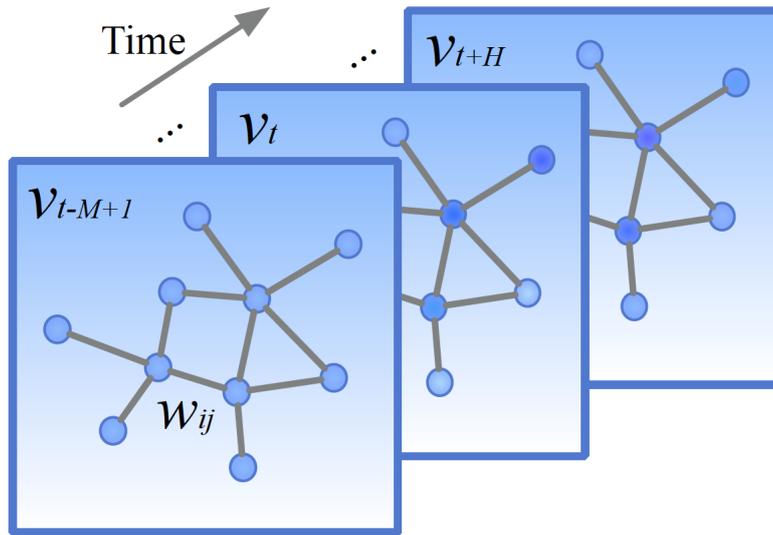


Figure 3.6.: Graph-structured traffic data. Each  $v_t$  indicates a frame of current traffic status at time step  $t$ , which is recorded in a graph-structured data matrix. Adapted from [Yu et al., 2018].

**Time Series Normalisation and Transformation** Dynamic freight flow input is represented as a time series  $x_t \in \mathbb{R}^N$ , where  $x_{i,t}$  is the freight intensity at node  $i$  at time  $t$ . To reduce the impact of large magnitude differences between peak and off-peak hours, each value is first log-transformed using  $\log(1 + x)$ . This transformation compresses large values and helps stabilise the learning process.

Subsequently, the transformed values are scaled to the  $[0, 1]$  range using Min-Max normalisation, but crucially, the scaling parameters (minimum and maximum) are computed using the *training set only*. This avoids data leakage and ensures that the model does not have access to future information during training. The same scaling is then applied to the validation and test data using these fixed training-derived parameters.

Each training sample is constructed as a pair consisting of a sequence of  $L$  historical time steps (e.g. 2 hours of 5-minute intervals,  $L = 24$ ) and a prediction target  $H$  steps ahead (e.g. 1 hour later,  $H = 12$ ). These input-output pairs are extracted using a sliding window across the time series (illustrated in Figure 3.6). After prediction, the inverse transformations (Min-Max inverse and then  $\exp(x) - 1$ ) are applied to the model outputs to recover freight volumes in the original scale.

**Urban Morphology Features (Static)** In addition to temporal freight signals, each node is assigned a static feature vector  $s_i$  drawn from the Urban Morphology Dataset. These vectors encode local building characteristics. The UMD matrix is loaded from a CSV file, sorted by node ID, and preprocessed to remove missing values. It is stored as a tensor of shape  $[N, F_s]$ , where  $F_s$  is the number of static features.

### 3. Methodology

**Output** After preprocessing, the pipeline outputs:

- $x_{\text{train}}, x_{\text{val}}, x_{\text{test}}$ : tensors of shape  $[\text{num\_samples}, 1, L, N]$
- $y_{\text{train}}, y_{\text{val}}, y_{\text{test}}$ : target flows of shape  $[\text{num\_samples}, N]$
- $\text{umd\_tensor}$ : static urban morphology matrix of shape  $[N, F_s]$
- $\text{gso}$ : precomputed graph shift operator  $\tilde{\mathbf{A}} \in \mathbb{R}^{N \times N}$

#### 3.4.2. ST-Conv blocks

Conceptually, an ST-Conv block is structured as a “sandwich” of two temporal convolution layers around a spatial graph convolution layer [Yu et al., 2018]. Each component is described in detail below:

**Temporal Convolution with GLU gating:** The block begins with a 1D convolution along the temporal dimension for each node’s time series. This temporal convolution uses a kernel of length  $K_t$  (e.g.  $K_t = 3$  time steps) and multiple filters to capture recent temporal trends in freight flow. To introduce non-linearity and a gating mechanism, Gated Linear Units (GLUs) is adopted for the temporal convolution layers [Yu et al., 2018] [Huang et al., 2020]. In a GLU layer, the convolution output is split into two halves:  $\mathbf{P}$  and  $\mathbf{Q}$ .  $\mathbf{P}$  represents candidate features and  $\mathbf{Q}$  serves as a gate. A sigmoid activation  $\sigma(\cdot)$  is applied to  $\mathbf{Q}$  to obtain gating values between 0 and 1, and then an element-wise product is taken:  $\mathbf{H}_{\text{temp}} = \mathbf{P} \odot \sigma(\mathbf{Q})$ . In essence, the GLU allows the network to selectively filter the temporal features, letting important patterns pass through while suppressing less relevant signals [Huang et al., 2020]. This gated temporal convolution thus produces a set of new feature maps for each node, encoding short-term freight flow dynamics (e.g. capturing peaks or drops in the recent hours) modulated by the gating mechanism. A residual connection is also used which adds the input (after appropriate dimension match) to the GLU output, ensuring that the original time series information is retained and improving gradient flow [Huang et al., 2020].

---

**Algorithm 3.7:** Temporal convolution with Gated Linear Units (GLU) for learning short-term dynamics

---

**Input:** Input tensor  $\mathbf{X} \in \mathbb{R}^{B \times C_{\text{in}} \times T \times N}$ ; kernel size  $K_t$ ; activation function (GLU or GTU)

**Output:** Temporal features  $\mathbf{H}_{\text{temp}} \in \mathbb{R}^{B \times C_{\text{out}} \times T' \times N}$

- 1 Align dimensions for residual connection;
  - 2  $\mathbf{X}_{\text{resid}} \leftarrow \text{Align}(\mathbf{X})[:, :, K_t - 1 :, :];$
  - 3 Apply causal temporal convolution;
  - 4  $\mathbf{Z} \leftarrow \text{CausalConv2d}(\mathbf{X}, \text{kernel} = (K_t, 1), \text{out\_channels} = 2C_{\text{out}});$
  - 5 Split convolution output;
  - 6  $\mathbf{P} \leftarrow \mathbf{Z}[:, : C_{\text{out}}, :, :];$
  - 7  $\mathbf{Q} \leftarrow \mathbf{Z}[:, C_{\text{out}} :, :, :];$
  - 8 Apply gated linear unit;
  - 9  $\mathbf{H}_{\text{temp}} \leftarrow (\mathbf{P} + \mathbf{X}_{\text{resid}}) \odot \sigma(\mathbf{Q});$
-

**Graph Convolution (Spatial):** The second component in the ST-Conv block is a spatial graph convolution that propagates information across the nodes of the road network graph  $G = (V, E)$ . This operation enables the model to learn spatial dependencies between connected locations, which is essential for capturing the spread of freight activity through the transport network. Unlike classical GCNs that assume undirected graphs and symmetric normalisation (with added self-loops), *ST-SimNet* uses a *random-walk normalised adjacency matrix*  $\tilde{\mathbf{A}}$  constructed from a directed graph, without added self-loops. This design choice is motivated by the inherent directionality of real-world freight movement, where outgoing and incoming connections have distinct implications.

Formally, if  $\mathbf{H}_{\text{in}} \in \mathbb{R}^{N \times F}$  is the matrix of node features output by the previous temporal layer, the graph convolution computes:

$$\mathbf{H}_{\text{spatial}} = \tilde{\mathbf{A}} \mathbf{H}_{\text{in}} \mathbf{W},$$

where  $\tilde{\mathbf{A}} = D^{-1} \mathbf{A}$  is the row-normalised adjacency matrix (with  $D$  the out-degree matrix), and  $\mathbf{W} \in \mathbb{R}^{F \times F'}$  is a learnable weight matrix. No activation function is applied within the graph convolution layer itself; instead, a non-linearity (ReLU) is applied immediately after, as part of the ST-Conv block.

This operation performs a directional aggregation: each node integrates weighted feature information from its immediate in-neighbours, modulated by the transition probabilities encoded in  $\tilde{\mathbf{A}}$ . This enables the model to capture freight flow phenomena such as congestion build-up and upstream-downstream dependencies. The use of a row-normalised GSO helps preserve the scale of incoming signals and supports stable training, while remaining well-suited to directed, non-symmetric transport networks.

To improve computational efficiency and control model complexity, bottleneck strategy is applied to the architecture in this layer: the output feature dimension  $F'$  is set smaller than the input  $F$ , which acts as an implicit regulariser and is consistent with prior spatio-temporal GNN design principles [Yu et al., 2018].

**Second Temporal Convolution:** The spatially convolved features  $\mathbf{H}_{\text{spatial}}$  are then passed through a second temporal convolution layer, again implemented as a Gated Linear Unit (GLU). This layer uses a 1D convolution along the temporal dimension of  $\mathbf{H}_{\text{spatial}}$ , followed by a gating mechanism. It refines temporal trends in the spatially-informed features, producing an output tensor  $\mathbf{H}_{\text{temp2}} = \mathbf{P}' \odot \sigma(\mathbf{Q}')$ , where  $\mathbf{P}'$  and  $\mathbf{Q}'$  are the two halves of the convolution output. As in the first temporal layer, the convolution has kernel size  $K_t$  and no padding, reducing the temporal dimension by  $K_t - 1$ . Over two such layers in each ST-Conv block, the effective temporal reduction becomes  $2(K_t - 1)$ . This design choice aligns with causal learning principles: it ensures that predictions at any timestep depend only on preceding inputs, avoiding leakage of future information. Additionally, the lack of padding induces a temporal compression effect, forcing the model to extract and summarise relevant temporal patterns over progressively shorter windows [Yu et al., 2018]. This second convolution finalises the “temporal-spatial-temporal” structure of the block and prepares the feature maps for the next stage. Notably, the output channel dimension is restored to the original width (e.g. 64) after being bottlenecked in the spatial layer, following the symmetric bottleneck architecture of STGCN [Yu et al., 2018].

**Normalization and Dropout:** To stabilise training and improve generalisation, each ST-Conv block concludes with a layer normalisation and dropout step. Specifically, the output of the second temporal convolution is first transposed and passed through a *LayerNorm* operation,

---

**Algorithm 3.8:** Spatial graph convolution using  $\tilde{\mathbf{A}}$ , with support for Chebyshev polynomial filtering or standard first-order GCN

---

**Input:** Feature tensor  $\mathbf{X} \in \mathbb{R}^{B \times C \times T \times N}$ ; graph shift operator  $\tilde{\mathbf{A}} \in \mathbb{R}^{N \times N}$ ; spatial kernel size  $K_s$ ; convolution type (`cheb_graph_conv` or `graph_conv`)

**Output:** Spatially filtered features  $\mathbf{H}_{\text{out}} \in \mathbb{R}^{B \times C' \times T \times N}$

- 1 Permute input dimensions:  $\mathbf{X} \leftarrow \text{permute}(\mathbf{X}, (0, 2, 3, 1))$ ;
- 2 **if** *convolution type is cheb\_graph\_conv* **then**
- 3     Initialise:  $\mathbf{X}_0 \leftarrow \mathbf{X}$ ;
- 4     Compute first order:  $\mathbf{X}_1 \leftarrow \tilde{\mathbf{A}} \cdot \mathbf{X}_0$ ;
- 5     **for**  $k \leftarrow 2$  **to**  $K_s - 1$  **do**
- 6          $\mathbf{X}_k \leftarrow 2 \cdot \tilde{\mathbf{A}} \cdot \mathbf{X}_{k-1} - \mathbf{X}_{k-2}$ ;
- 7     Stack  $\mathbf{X}_0, \dots, \mathbf{X}_{K_s-1}$  and apply weights:  $\mathbf{H}_{\text{spatial}} \leftarrow \sum_{k=0}^{K_s-1} \mathbf{X}_k \cdot \mathbf{W}_k$ ;
- 8 **else if** *convolution type is graph\_conv* **then**
- 9     Apply GSO:  $\mathbf{H}_1 \leftarrow \tilde{\mathbf{A}} \cdot \mathbf{X}$ ;
- 10    Multiply with weight matrix:  $\mathbf{H}_{\text{spatial}} \leftarrow \mathbf{H}_1 \cdot \mathbf{W}$ ;
- 11 Permute back and apply residual connection;
- 12  $\mathbf{H}_{\text{out}} \leftarrow \text{Align}(\mathbf{X}) + \text{permute}(\mathbf{H}_{\text{spatial}}, (0, 3, 1, 2))$ ;

---

which standardises the activations across feature dimensions for each individual data point (rather than across a batch). This helps mitigate covariate shift, accelerate convergence, and maintain stable gradients, especially in models where the batch size is small [Ioffe and Szegedy, 2015]. Following normalisation, a dropout layer with rate  $p = 0.2$  is applied, randomly setting a fraction of the features to zero during training. This prevents over-fitting by discouraging the model from relying too heavily on specific feature pathways.

*ST-SimNet* model stacks **two** such ST-Conv blocks in series. The output of the first block feeds into the second block (which has the same internal structure of:

TempConv  $\rightarrow$  GraphConv  $\rightarrow$  TempConv, plus normalisation and dropout)

By stacking two blocks, the model can capture longer-range dependencies (both in time and space) than a single block alone. The first block learns low-level temporal patterns and local spatial interactions; the second block can build on those to model more complex patterns (for example, combining information from two-hop neighbors in the graph, or capturing effects with a slightly longer temporal range due to the sequential convolution). After the two ST-Conv blocks, we obtain a final tensor of node features that encapsulates recent temporal information and spatial neighbor influences for each node.

### 3.4.3. UMD Fusion

**Static Feature Integration and Output Layer:** In *ST-SimNet*, static urban morphology data is integrated at the final prediction stage. This was a practical design choice made for implementation feasibility, rather than a result of comparative evaluation with earlier fusion strategies.

Let  $\mathbf{h}_i \in \mathbb{R}^F$  denote the dynamic embedding of node  $i$  after the last ST-Conv block and temporal pooling, and let  $\mathbf{s}_i \in \mathbb{R}^S$  represent the static urban morphology vector (see Sec-

tion 3.3.2). These static features are passed through a separate two-layer Multi-Layer Perceptron (MLP) to obtain a projected vector in the same latent space as  $\mathbf{h}_i$ . The model then computes a learnable fusion:

$$\mathbf{z}_i = (1 - \sigma(\alpha)) \cdot \mathbf{h}_i + \sigma(\alpha) \cdot \text{MLP}_{\text{umd}}(\mathbf{s}_i),$$

where  $\alpha$  is a trainable scalar parameter and  $\sigma(\cdot)$  is the sigmoid function. This fusion allows the model to balance the contribution of static and dynamic signals during prediction.

The resulting vector  $\mathbf{z}_i$  is then passed through a two-layer output MLP with ReLU activation and dropout. This final layer maps each node’s fused representation to the forecasted freight flow. In the current configuration, the model performs  $H = 12$  step-ahead prediction, outputting a single scalar per node after temporal compression.

This late-stage integration allows the dynamic processing layers to focus exclusively on learning temporal and spatial freight dynamics, while static features provide contextual information for correcting and calibrating the final output. Built environment characteristics, such as industrial density or land use, can thus adjust the baseline prediction for each node, similar to fixed effects or covariates in classical time-series models. This fusion mechanism enables the model to remain interpretable while capturing spatial heterogeneity in freight activity.

---

**Algorithm 3.9:** Final-stage fusion of static morphology and dynamic node features using a learnable scalar gate and output MLP

---

**Input:** Dynamic node embeddings  $\mathbf{H} \in \mathbb{R}^{B \times F \times N}$ ; static morphology features  $\mathbf{S} \in \mathbb{R}^{N \times S}$

**Output:** Predicted node-level freight flows  $\hat{\mathbf{Y}} \in \mathbb{R}^{B \times N}$

- 1 Project static features to latent space using MLP:  $\mathbf{U} \leftarrow \text{MLP}_{\text{umd}}(\mathbf{S})^\top$ ;
  - 2 Expand  $\mathbf{U}$  to match batch size:  $\mathbf{U}_{\text{batch}} \leftarrow \mathbf{U}.\text{unsqueeze}(0)$ ;
  - 3 Compute temporal average of ST-Conv output:  $\mathbf{H}_{\text{pooled}} \leftarrow \text{mean}(\mathbf{H}, \text{dim} = 2)$ ;
  - 4 Compute fusion weight:  $\sigma_\alpha \leftarrow \sigma(\alpha)$ ;
  - 5 Fuse static and dynamic features:  $\mathbf{Z} \leftarrow (1 - \sigma_\alpha) \cdot \mathbf{H}_{\text{pooled}} + \sigma_\alpha \cdot \mathbf{U}_{\text{batch}}$ ;
  - 6 Transpose for MLP input:  $\mathbf{Z}_{\text{perm}} \leftarrow \text{permute}(\mathbf{Z}, (0, 2, 1))$ ;
  - 7 Pass through output MLP:  $\mathbf{Z}_1 \leftarrow \text{ReLU}(\text{Dropout}(\text{Linear}_1(\mathbf{Z}_{\text{perm}})))$ ;
  - 8 Compute final prediction:  $\hat{\mathbf{Y}} \leftarrow \text{Softplus}(\text{Linear}_2(\mathbf{Z}_1))$ ;
  - 9 Remove singleton channel:  $\hat{\mathbf{Y}} \leftarrow \hat{\mathbf{Y}}.\text{squeeze}(-1)$ ;
- 

#### 3.4.4. Output Block

The final output layer of *ST-SimNet* is responsible for mapping the fused node representations combining both spatio-temporal and static urban morphology features into forecasted freight flow values. Depending on the temporal compression depth (determined by the number of ST-Conv blocks and convolutional kernel sizes), the model dynamically selects one of two configurations for the output block.

If the compressed temporal length  $K_o$  is greater than one, a dedicated *OutputBlock* module is applied. This block is structured as a fully-connected feed-forward network operating on

### 3. Methodology

the temporal output of the last ST-Conv block. It includes linear transformations, non-linear activations, and dropout, and is designed to consolidate temporal summaries into a single forecasted scalar per node.

When  $K_o = 0$  (i.e. maximum temporal compression has occurred), the output block is bypassed. Instead, a two-layer Multi-Layer Perceptron (MLP) is directly applied to the temporally aggregated and UMD-fused node representations. This MLP comprises a linear transformation ( $\mathbf{W}_1$ ), followed by a ReLU activation, dropout, and a final linear projection ( $\mathbf{W}_2$ ) to obtain a scalar output.

In both configurations, a final softplus activation ensures non-negativity of the predicted flow values:

$$\hat{y}_i = \text{softplus}(\mathbf{z}_i + b),$$

where  $b$  is a learnable bias term. This design ensures both smooth output gradients during training and realistic positive flow predictions.

---

**Algorithm 3.10:** Apply output block if  $K_o > 1$ , else use direct MLP; both end with softplus to ensure non-negative forecasts

---

**Input:** Fused representation  $\mathbf{Z} \in \mathbb{R}^{B \times N \times F}$ ; compression length  $K_o$   
**Output:** Predicted flows  $\hat{\mathbf{Y}} \in \mathbb{R}^{B \times N}$

---

```

1 if  $K_o > 1$  then
2   Apply output block;;
3    $\mathbf{Z}_1 \leftarrow \text{ReLU}(\text{Dropout}(\text{Linear}_1(\mathbf{Z})))$ ;
4    $\hat{\mathbf{Y}} \leftarrow \text{Softplus}(\text{Linear}_2(\mathbf{Z}_1) + b)$ 
5 else
6   Bypass output block and apply 2-layer MLP;;
7    $\mathbf{Z}_1 \leftarrow \text{ReLU}(\text{Dropout}(\text{Linear}_1(\mathbf{Z})))$ ;
8    $\hat{\mathbf{Y}} \leftarrow \text{Softplus}(\text{Linear}_2(\mathbf{Z}_1) + b)$ 

```

---

## 3.5. Model Training Setup

ST-SimNet is trained to predict short-term freight flow dynamics from historical sequences of traffic data. Each training sample consists of a pair  $(X_{\text{hist}}, Y_{\text{target}})$ , where

$$X_{\text{hist}} = \{\mathbf{x}_{t-L+1}, \mathbf{x}_{t-L+2}, \dots, \mathbf{x}_t\},$$

$$Y_{\text{target}} = \mathbf{x}_{t+1}.$$

Here,  $X_{\text{hist}} \in \mathbb{R}^{L \times N}$  is a sequence of past flow values for all  $N$  nodes over a history window of length  $L = 24$  (covering 2 hours at 5-minute intervals), and  $Y_{\text{target}} \in \mathbb{R}^N$  is the node-level flow at the next time step. Training pairs are created via a sliding window over the time series.

The model processes mini-batches of such samples and generates predictions  $\hat{\mathbf{y}}_{t+1} \in \mathbb{R}^N$ , which are compared against the ground-truth  $\mathbf{x}_{t+1}$ . The loss function used is Mean Squared

Error (MSE), defined as:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N (\hat{y}_{i,t+1} - x_{i,t+1})^2.$$

This loss is suitable for continuous traffic count regression and penalises larger errors more strongly. MSE was preferred for training due to its smooth gradient properties and emphasis on outliers.

Training uses the AdamW optimiser, which combines adaptive gradient descent with decoupled weight decay. The learning rate is initially set to  $\eta = 0.0005$ , and the Adam parameters are kept at their defaults ( $\beta_1 = 0.9, \beta_2 = 0.999$ ). To guard against overfitting, a weight decay (L2 penalty) of 0.02 is applied to all learnable parameters.

A step-wise learning rate scheduler reduces  $\eta$  by a factor of 0.9 every 5 epochs (adjustable). If the validation loss does not improve for 10 consecutive epochs, early stopping is triggered. This training setup balances stability and convergence speed, and was found to produce robust results across several configurations.

The model is trained for a fixed number of epochs (50—small simulations, 100—big simulation) or until convergence. An epoch is defined as one full pass through all training samples (using a batch size of 12 windows). To prevent over-fitting, early stopping is applied: training is halted if the validation loss does not improve for 10 consecutive epochs. During training, the dropout (Section 3.4) is active (with  $p = 0.3$ ) to randomize the network and improve generalisation; at inference time, dropout is turned off. Training samples are drawn sequentially from the dataset without shuffling, to preserve reproducibility and stability. Each sample already consists of a temporally ordered window of fixed length, so the model learns from coherent temporal contexts.

All model development was implemented in Python using PyTorch Geometric. Training was performed on an MacBook Pro M1 for the smaller area - Amsterdam West, and on a Windows PC with RTX 4090 for the big application. Training stability was observed to be good: the gated convolutions and residual connections contributed to stable gradients and convergence. I did not observe any exploding or vanishing gradient issues, and the training loss consistently decreased to a low level. Batch normalisation further helped in stabilising training by normalising input distributions to each layer. By the end of training, the model typically achieves a small gap between training and validation error, indicating a good fit without severe over-fitting.

## 3.6. Validation and Evaluation

I evaluate the trained [ST-SimNet](#) on held-out test data to assess its predictive performance. The time series data is split into training, validation, and testing periods. I use a chronological split: e.g. the first 70% of the timeline for training, the next 15% for validation (model tuning), and the final 15% for testing. This ensures that the model is always predicting future data from past data and mimics a real forecasting scenario. The test set consists of sequences  $x_t - L + 1, \dots, x_t$  that the model has never seen, and it generates forecasts  $\hat{x}_t + 1$  which I compare against the true  $x_t + 1$ . No overlapping windows between training and test are used (to avoid information leakage).

### 3. Methodology

**Evaluation Metrics:** To quantify model performance, I report three standard regression metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Weighted Mean Absolute Percentage Error (WMAPE). These are defined as:

$$\text{MAE} = \frac{1}{N_{\text{test}}} \sum_{(i,t) \in \text{test}} |\hat{y}_{i,t} - x_{i,t}|,$$

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{test}}} \sum_{(i,t) \in \text{test}} (\hat{y}_{i,t} - x_{i,t})^2},$$

$$\text{WMAPE} = \frac{\sum_{(i,t) \in \text{test}} |\hat{y}_{i,t} - x_{i,t}|}{\sum_{(i,t) \in \text{test}} x_{i,t}}.$$

MAE gives a direct interpretation in vehicle count units, while RMSE emphasises large deviations due to squaring the errors. WMAPE provides a normalised measure of error, expressing the total absolute error as a fraction of the total actual volume, making it particularly useful when comparing flows across zones of different magnitudes. All metrics are computed across the entire test set.

After training, the epoch with the lowest validation loss is saved as the final model. This model is then used on the test set to produce the metrics above. The training and validation loss curves were analysed to ensure the model did not begin to overfit (divergence between training and validation loss). During experiments, the use of early stopping and regularisation (dropout, weight decay) maintained stability – validation loss typically plateaued or slightly increased after a certain point, at which it stopped training. This indicates the model parameters at the chosen checkpoint are near-optimal for generalisation.

## 4. Results

### 4.1. Experiment Design

This section presents the experimental findings from applying both the baseline *STGCN* and the proposed *ST-SimNet* model. Results are reported across several evaluation scenarios, including different data subsets and spatial extents. In each case, the models were trained for 50 epochs—and 100 in case of 4.5— where I quantified model accuracy and explored spatial and temporal patterns of prediction errors using regression metrics and visualisations. Sections 4.3, 4.4 were trained and validated on a subset data, whereas 4.5 was trained on a larger dataset encompassing the centre of Amsterdam with its peripherals. Both Area of Interest (AOI)s, can be found in appendix B

### 4.2. Model Introduction

The baseline model used in this study is the Spatio-Temporal Graph Convolutional Network (*STGCN*) architecture originally introduced by Yu et al. [2018] and described in detail in Section 2.6. This model combines gated temporal convolutions with spectral graph convolutions using Chebyshev polynomials to learn spatio-temporal dependencies in traffic data. In its standard configuration, *STGCN* was designed to forecast traffic velocity from sensor data on a fixed urban network.

In contrast, the proposed *ST-SimNet* extends the original *STGCN* in multiple directions, both architecturally and in terms of input representation. First, the model is adapted to predict freight flow intensity rather than velocity, requiring changes to the loss formulation, data pipeline, and output layers. Second, *ST-SimNet* introduces a dedicated Fusion Block for incorporating static urban context in the form of Urban Morphology Descriptors (*UMDs*), including building usage, façade orientation, population statistics, and land use indicators. These static features are projected into the latent space via a separate *MLP* and blended with dynamic node embeddings using a learnable convex combination.

Structurally, both models consist of two stacked *ST-ConvBlocks*, with temporal convolutions implemented via *GLUs*, followed by graph convolutions over a directed *GSO*. However, only *ST-SimNet* integrates a fusion mechanism for contextual data. As a result, it is capable of learning not just how freight flows evolve over time, but also how they are shaped by the static configuration of the urban environment.

In the following sections, both models are evaluated on identical datasets and tasks to assess the performance gains resulting from the integration of static spatial context. The experiments cover two spatial extents: a focused subset of the city and a larger area encompassing central Amsterdam and its periphery (see Appendix B). Evaluation metrics include *MAE*, *RMSE*, and *WMAPE*, as well as spatial error distribution analysis.

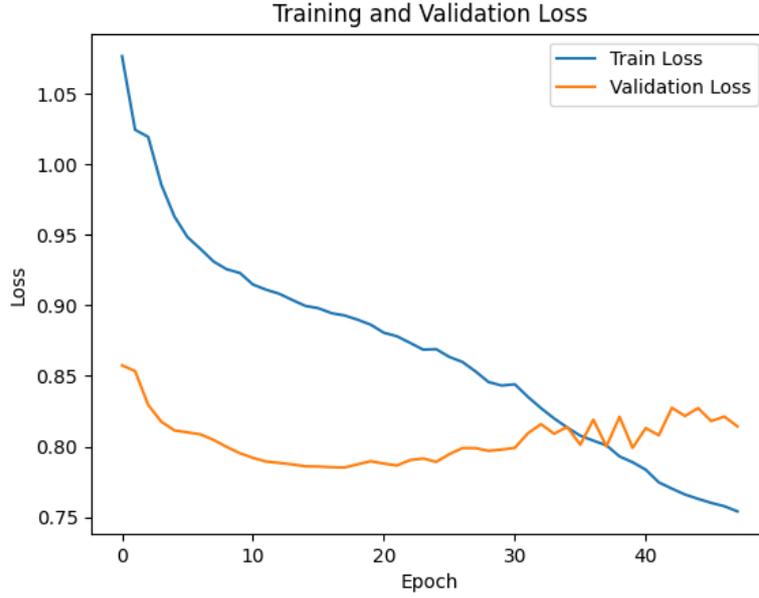


Figure 4.1.: Training and Validation Loss over Epochs (STGCN)

### 4.3. Baseline: STGCN with Dynamic Input Only

#### 4.3.1. Model Configuration

The baseline experiment evaluates the original [STGCN](#) on a freight flow forecasting task across  $N$  spatial zones in the study area. The model is trained exclusively on dynamic input data  $\{\mathbf{x}_t\}_{t=1}^T$ , where  $\mathbf{x}_t \in \mathbb{R}^N$  represents freight intensity per node at time  $t$ , sampled at 5-minute intervals.

The model is trained with a temporal window of  $L = 96$  (past 8 hours) and a prediction horizon of  $H = 6$  (next 0.5 hour). It includes two [ST-ConvBlock](#) with a temporal kernel size  $K_t = 3$  and spatial kernel order  $K_s = 3$ , using GLU (Gated Linear Unit) activations. Dropout is set to 0.5. Optimisation is performed using AdamW with a learning rate  $\eta = 0.0007$ , decaying by  $\gamma = 0.8$  every 5 epochs, L2 weight decay of  $10^{-3}$ , and a batch size of 10.

#### 4.3.2. Training Performance

Training was conducted for up to 50 epochs with early stopping based on validation loss. Figure 4.1 shows the training and validation loss trajectories. The training loss decreases consistently, while the validation loss stabilises early and begins to rise slightly after epoch 30, suggesting mild overfitting.

Figure 4.2 illustrates the learning rate decay. The learning rate is halved every 5 epochs until reaching a final value of approximately 0.00028, allowing for gradual refinement of model weights in later training stages.

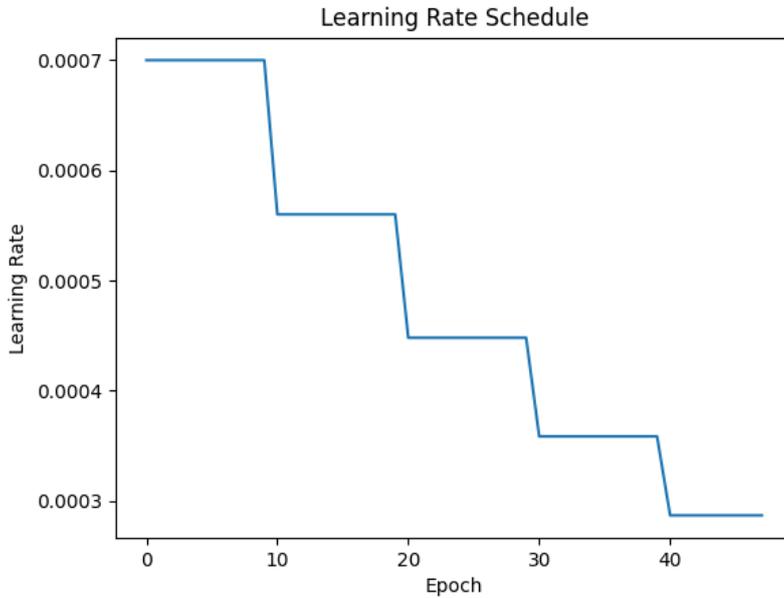


Figure 4.2.: Learning Rate Schedule during Training (STGCN)

### 4.3.3. Prediction Accuracy

Model performance on the test set was evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Weighted Mean Absolute Percentage Error (WMAPE). The scatter plot in Figure 4.3 compares predicted and true flows across all nodes and time steps. Predictions generally follow the ideal line, though variance increases with magnitude - particularly under-predicting at high-flow values.

Figure 4.4 provides the distribution of prediction errors. Most errors are centred around zero within the range of  $[-200, 200]$ , but there are significant long tails on both sides, indicating rare large under- and over-predictions.

### 4.3.4. Spatial Heterogeneity of Performance

The model's behaviour varies substantially across nodes. Figure 4.5 shows the per-node prediction statistics: predicted mean, maximum, and standard deviation of the flows compared to the true mean. While the model captures patterns for low-volume nodes, high-flow nodes exhibit substantial deviation, especially in maximum values, underscoring the challenge of modelling peak freight volumes.

### 4.3.5. Conclusions

The STGCN model provides a solid baseline for short-term freight flow prediction. Training is stable, and the loss curves confirm good convergence. However, the validation perfor-

#### 4. Results

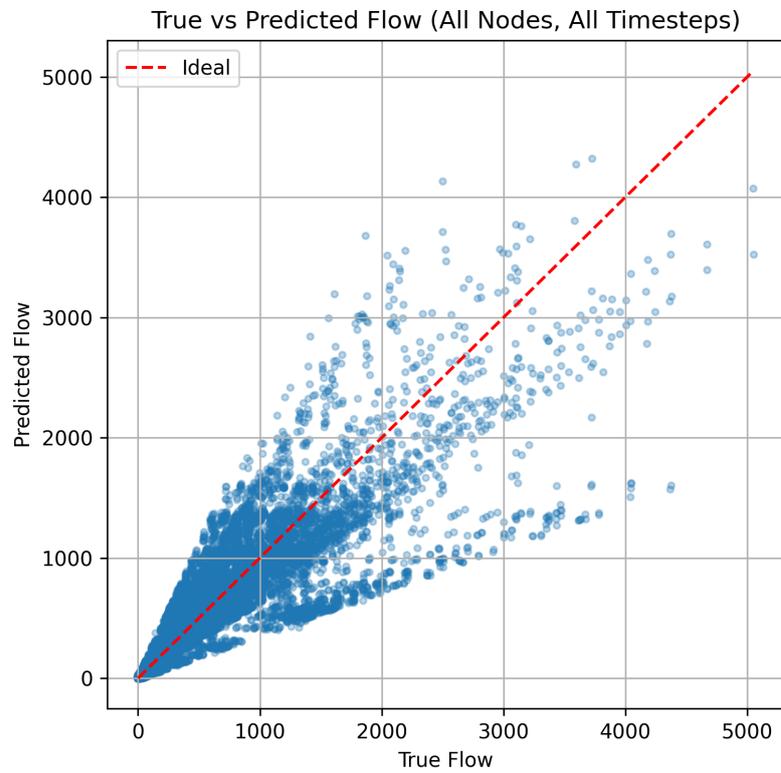


Figure 4.3.: Scatter Plot: True vs Predicted Flow (STGCN)

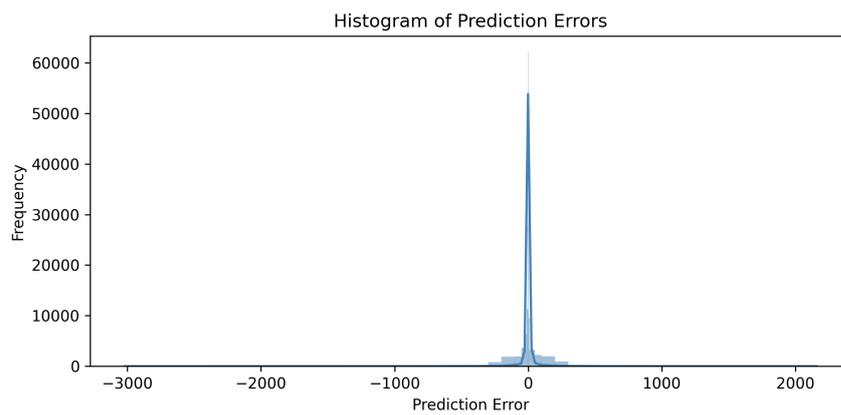


Figure 4.4.: Histogram of Prediction Errors (STGCN)

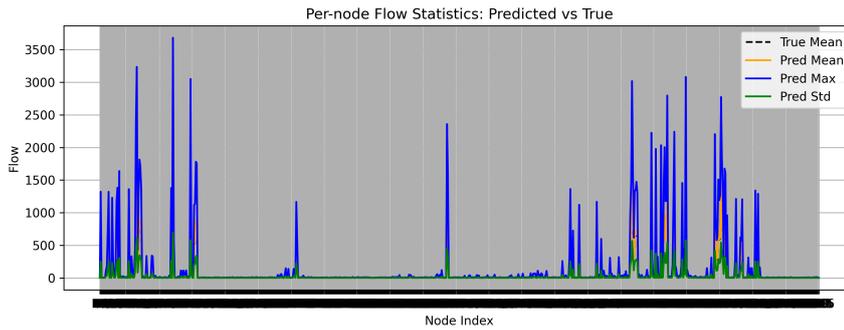


Figure 4.5.: Per-node Flow Statistics: Predicted vs True (STGCN)

mance suggests limited generalisability, with signs of overfitting after around 30 epochs.

A key limitation lies in the model’s inability to predict high-volume nodes accurately. This is likely due to its lack of awareness of static context - such as zoning, land use, or road capacity, which heavily influences freight movement. The original STGCN was developed for predicting traffic speed on highway networks, where value magnitudes are lower and more stable. Applying it to count-based freight flow tasks exposes its difficulty in handling heterogeneous distributions.

Additionally, training remains computationally expensive given the model’s complexity and low batch size. While tuning parameters like  $K_t$ ,  $K_s$ , or adopting alternative graph convolutions may help, these findings point to a deeper need: embedding richer context, such as static features, into the architecture - an idea that motivates the *ST-SimNet* model.

## 4.4. *ST-SimNet: Dynamic + Static Input*

To address the limitations observed in STGCN, particularly its inability to account for heterogeneous node-level characteristics, the proposed *ST-SimNet* model incorporates static features describing the built environment and demographic characteristics of each node. This section presents the configuration and results of *ST-SimNet*, highlighting how static context enriches predictive performance.

### 4.4.1. Model Configuration

The model is trained using a temporal window of  $L = 96$  (past 8 hours) to predict  $H = 6$  steps ahead (next 0.5 hour). It employs two *ST-ConvBlock* with a temporal convolution kernel size  $K_t = 3$  and spatial kernel order  $K_s = 3$ , using the Gated Linear Unit (GLU) as activation. A dropout rate of 0.2 is applied. The model is optimised with AdamW, using a learning rate  $\eta = 0.001$  decaying by  $\gamma = 0.8$  every 3 epochs, L2 weight regularisation of 0.001, and a batch size of 10.

#### 4.4.2. Urban Morphology Data (UMD) Features

ST-SimNet integrates static features derived from two spatial layers: building-level aggregates and PC6-level demographic information.

**Building-Based Features** The building-level features were aggregated per node (NODENR) by computing statistical summaries from all buildings spatially associated with a given node. The following features were included:

- **Construction characteristics:**  
bouwjaar (median), volume, verblijfsobjecten, oppervlakteverblijfsobjecten
- **Physical form:**  
gem\_hoogte, gem\_bouwlagen
- **Facade orientation:**  
shared and total facade lengths per cardinal direction  
(e.g. north\_shared\_length, north\_facade\_length, etc.)
- **Vegetation and illumination:**  
Normalised Difference Vegetation Index (`NDVI`)\_mean\_100m, `NDVI_stddev_100m`,  
Night-Time Lights (`NTL`)\_mean\_500m, `NTL_stddev_500m`
- **Functional classification:**  
meestvoorkomendelabel, function, building\_function, residential\_type,  
non\_residential\_type
- **Other:**  
distance (to nearest node), building\_count

**PC6-Based Features** Additionally, spatial joins were performed with the PC6 layer to include coarse-grained context:

- postcode6,  
aantal\_inwoners, aantal\_woningen, percentage\_koopwoningen,  
aantal\_part\_huishoudens

Together, these features form the static vector  $\mathbf{s}_i \in \mathbb{R}^{F_s}$  for each node  $i$ , concatenated with the learned dynamic embedding prior to the MLP output layer.

The selected features capture a broad spectrum of urban morphological characteristics relevant to freight activity. However, not all features contribute equally. Some categorical variables may introduce noise, especially in zones with highly mixed land use. These include label-dominant columns like `function`, `building_function`, and `non_residential_type`, which were encoded using simple mode selection per node and one-hot encoding (see Table 4.1).

Table 4.1.: Encoded Values for Categorical UMD Features

Feature	Encoded Values
meestvoorkomendelabel (Energy label)	0: A, 1: A+, 2: A++, 3: A+++, 4: A++++, 5: B, 6: C, 7: D, 8: E, 9: F, 10: G, 11: None
function	0: Commercial, 1: Industrial, 2: Residential
building_function	0: non_residential, 1: residential
residential_type	0: Apartment, 1: Corner House, 2: Detached House, 3: None, 4: Terrace or Semi-detached House, 5: Two-and-a-half-story House
non_residential_type	0: Accommodation Function, 1: Education Function, 2: Healthcare Function, 3: Industrial Function, 4: Meeting Function, 5: None, 6: Office Function, 7: Other Use Function, 8: Retail Function, 9: Sport Function

## Feature Importance and Saliency Analysis

To evaluate the contribution of static Urban Morphology Descriptors (UMDs) to the model’s predictions, I computed gradient-based saliency scores after training. These scores reflect the absolute gradient of the output with respect to each input feature, averaged over all nodes and time steps within the selected AOI. This allows us to estimate which static features the model relied on most when generating its forecasts.

The saliency analysis was performed separately for two distinct AOIs in the Amsterdam region. Each AOI represented a unique urban context, including variation in density, land use, and freight dynamics. As a result, the feature importance rankings differed across areas reflecting the fact that *ST-SimNet* adaptively learns which morphological signals matter most in each spatial context. This spatial specificity confirms that the model does not rely on global heuristics but instead exploits locally relevant urban form cues.

For this scenario, the most salient features included:

- `non_residential_type`,
- `south_shared_length`,
- `meestvoorkomendelabel`,
- `bouwjaar`,
- `aantal_inwoners`,
- `NTL_mean_500m`, `building_count`, and `east_shared_length`.

These features predominantly relate to built form intensity, façade orientation, and land use mix, all of which are known to influence last-mile delivery demand and network accessibility. By contrast, features such as `NDVI_mean_100m`, `distance`, and `south_facade_length` showed consistently lower saliency, suggesting limited predictive value in this particular case.

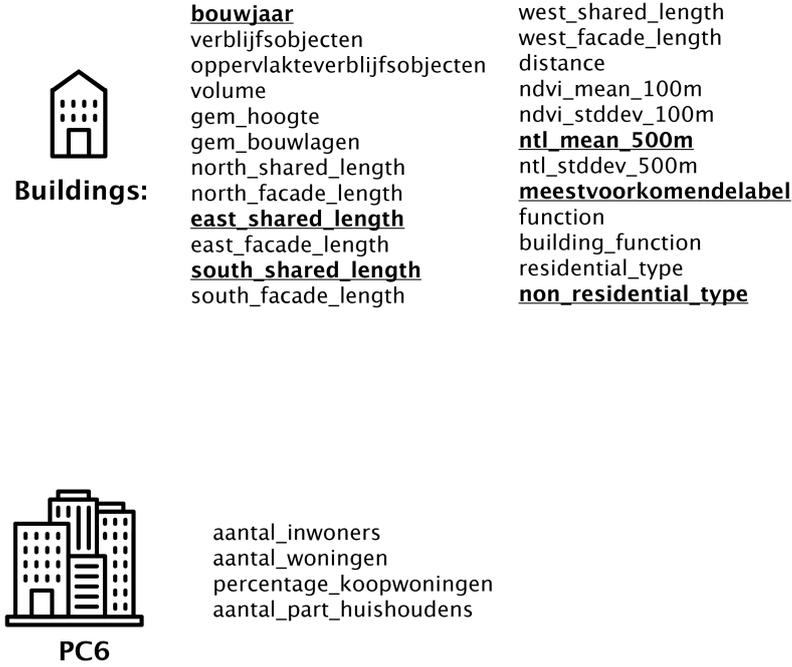


Figure 4.6.: Gradient-based saliency scores for static urban morphology features in one representative Area of Interest. Highlighted features contribute most strongly to the model’s node-level predictions.

Figure 4.6 features chosen from buildings and PC6 datasets. While specific values vary across other AOIs, similar patterns emerge: the model places greater emphasis on features that capture spatial complexity and freight-generating activities. This adaptive saliency behaviour underlines the model’s flexibility and context-awareness, confirming the architectural choice to fuse static descriptors in a spatially meaningful way.

**Weighting Static Information** To control the contribution of static features in the final predictions, the model introduces a learnable scalar parameter  $\alpha \in \mathbb{R}$ , whose value is constrained to the interval  $[0, 1]$  via a sigmoid transformation. The dynamic node embedding  $\mathbf{h}_i$  (obtained from the spatio-temporal blocks) is combined with the transformed static features  $\mathbf{s}_i$  using the following convex fusion:

$$\mathbf{z}_i = (1 - \sigma(\alpha)) \cdot \mathbf{h}_i + \sigma(\alpha) \cdot \text{MLP}_{\text{umd}}(\mathbf{s}_i)$$

where  $\sigma(\cdot)$  denotes the sigmoid function and  $\text{MLP}_{\text{umd}}$  projects static features into the same latent space as  $\mathbf{h}_i$ . This formulation allows the network to learn a smooth and bounded blending between dynamic and static signals. The initial value of  $\alpha$  is set to 0.3, allowing

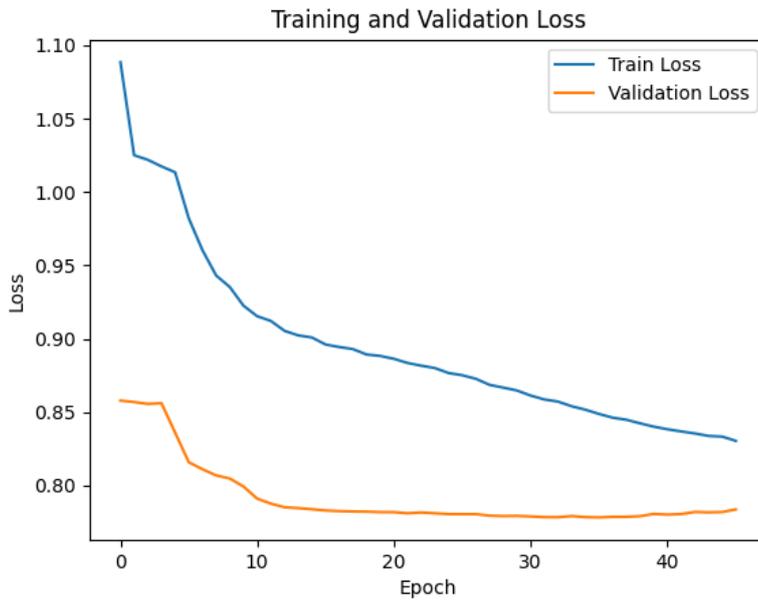


Figure 4.7.: Training and Validation Loss over Epochs (*ST-SimNet*, Weekdays)

the model to place greater initial weight on dynamic features while retaining flexibility to adjust this balance during training.

#### 4.4.3. Scenario 1: Weekdays Only - Amsterdam West

In this scenario, *ST-SimNet* is trained and tested on weekday data only (the same as *STGCN* in the previous experiment). This enables evaluation under more regular and periodic freight flow patterns. The model’s ability to leverage static urban features is assessed through per-node and aggregate performance metrics.

**Training Performance** Figure 4.7 shows the training and validation loss curves over 45 epochs (shorter training, due to early stopping). The validation loss decreases steadily and remains consistently below the training loss, which is slightly inflated due to regularisation effects (e.g. weight decay, dropout rate). This gap is expected and indicates that the model generalises well without over-fitting. The learning rate schedule in Figure 4.8 shows a smooth exponential decay, supporting stable convergence.

**Prediction Accuracy** The scatter plot in Figure 4.9 compares predicted and true freight flows across all nodes and timesteps. Each dot represents a prediction for a single node in a single prediction window, where a window consists of 96 historical time steps used to predict the next 6 steps (i.e., a 30-minute horizon). With 288 time steps per node, this results in 187 such prediction windows per node. Most points fall near the ideal line, indicating strong predictive accuracy. Unlike *STGCN*, *ST-SimNet* better captures high-flow magnitudes

#### 4. Results

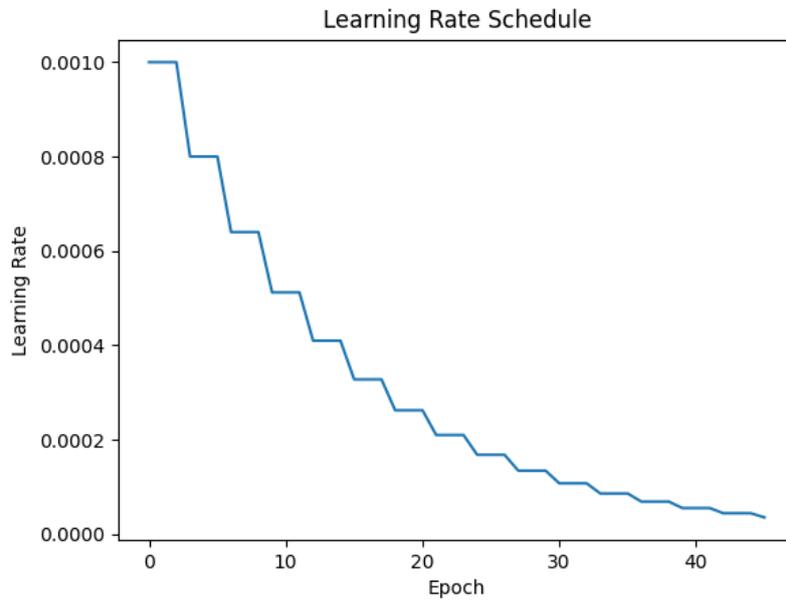


Figure 4.8.: Learning Rate Schedule during Training (*ST-SimNet*, Weekdays)

and shows reduced dispersion for extreme values. Nodes experiencing significant overestimations have been identified as those lacking any UM features from buildings, which is reflected in Figure 4.12.

The histogram in Figure 4.10 further confirms this: the majority of prediction errors are close to zero, with a narrower spread than STGCN, suggesting improved robustness.

**Spatial Heterogeneity of Performance** To examine spatial variability, Figure 4.11 presents per-node flow statistics. *ST-SimNet* captures average flows (dashed black) more accurately across both low and high-traffic nodes, and reduces the peak overestimations seen with STGCN. This improvement stems from incorporating static morphology features, which offer context on building function, type, and spatial configuration.

**Feature Contribution Analysis** Figure 4.12 illustrates how prediction error correlates with the richness of urban morphology data (UMD feature norms). Nodes with richer static feature descriptions exhibit lower average error, confirming that these features aid learning - particularly in more complex or active zones.

**Visual Inspection of Results in QGIS** To qualitatively assess prediction accuracy and spatial model behaviour, several key locations were examined in QGIS. The first striking observation is that nodes without any urban morphology descriptors from nearby buildings tend to suffer from significant overestimation (Figure 4.13). This confirms the importance of context information in guiding the model's understanding of freight activity potential.

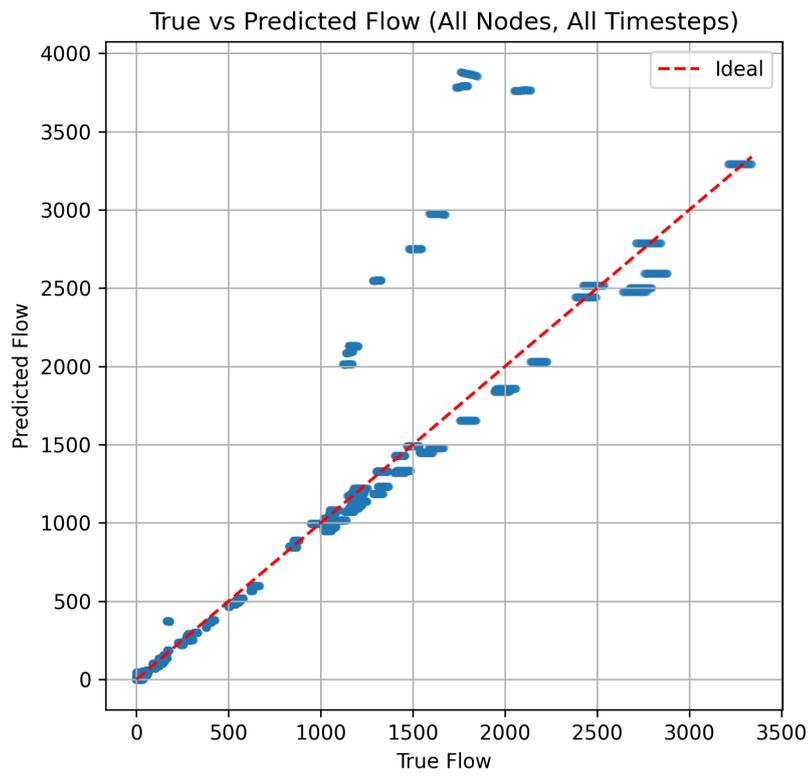


Figure 4.9.: Scatter Plot: True vs Predicted Flow (ST-SimNet, Weekdays)

#### 4. Results

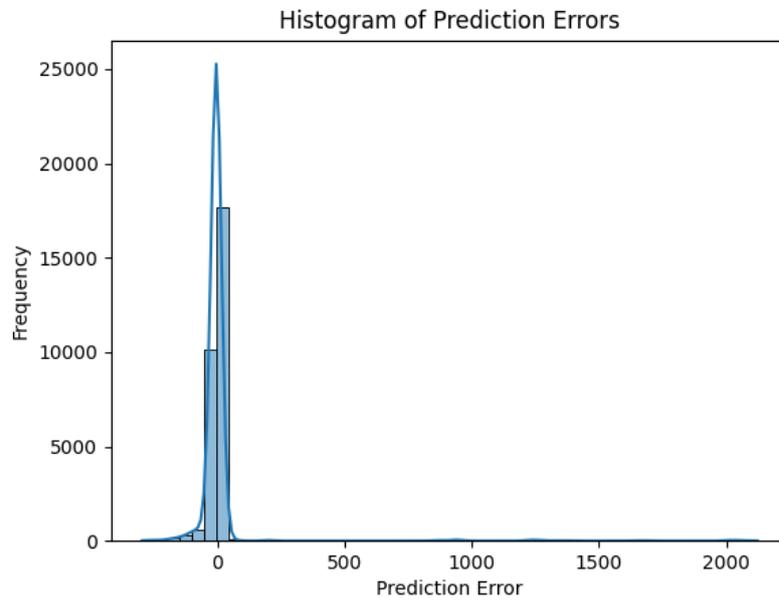


Figure 4.10.: Histogram of Prediction Errors (ST-SimNet, Weekdays)

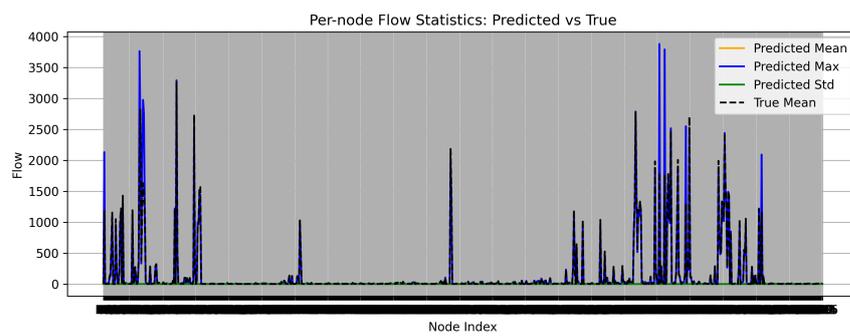


Figure 4.11.: Per-node Flow Statistics: Predicted vs True (ST-SimNet, Weekdays)

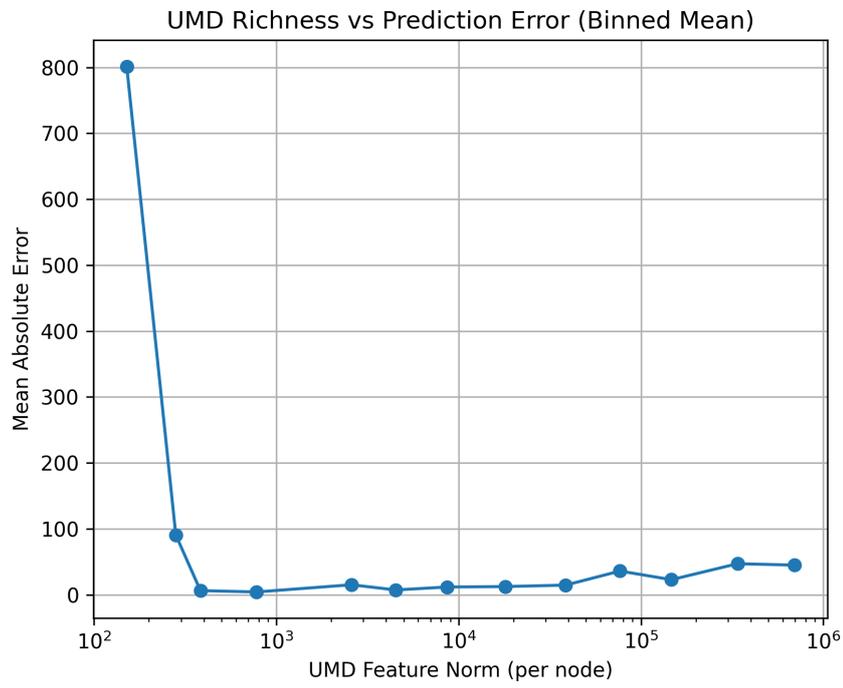


Figure 4.12.: Relationship between Static Feature Richness and MAE (ST-SimNet, Weekdays)



Figure 4.13.: Over-estimated node lacking building-derived UMD features.

#### 4. Results

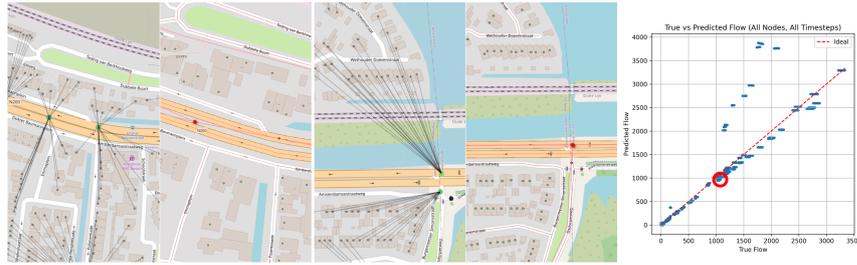


Figure 4.14.: Nodes with high flows and under-estimated values

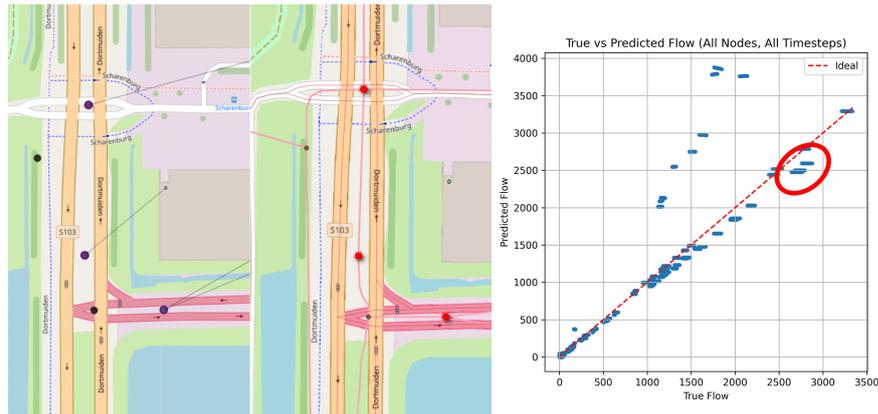


Figure 4.15.: Node with a single building-derived UMD source shows near-ideal prediction.

Conversely, nodes that inherit static features from even a single nearby building show remarkably improved accuracy (Figure 4.15), generally resulting in slightly under-predicted flows but very close to the true values. A number of highway nodes also experience slight but consistent under-prediction, despite inheriting several building features (Figure 4.14). This can likely be attributed to the mismatch between the node’s functional context and its static features—many such nodes, while located on freight-relevant highway segments, inherit morphological descriptors from nearby residential zones. This misalignment may lead the model to underestimate the true freight activity, as it associates the node with lower-flow land use patterns.

In summary, even minimal morphological input from buildings helps the model calibrate better to local conditions. Such visual diagnostics confirm the findings shown earlier in the scatter plots and node-level statistics, and they reinforce the critical role that static data play in enhancing interpretability and robustness of predictions, particularly in morphologically heterogeneous environments.

**Conclusion** By integrating static features, *ST-SimNet* builds a richer understanding of the urban context that shapes freight activity. This is reflected in the overall performance metrics and visualisations. The error histogram (Figure 4.10) shows a sharper peak around zero compared to *STGCN*, indicating fewer extreme prediction errors. Similarly, the scatter plot (Figure 4.9) reveals that predicted values align more closely with the ideal line, especially

for medium-traffic nodes. Nodes with more detailed urban morphology descriptors benefited the most, as further supported by Figure 4.12. However, spatial inspection using QGIS revealed that the model's performance is highly sensitive to the richness and structure of the static input. When no building-level features were available, predictions were consistently overestimated (see Figure 4.13). Interestingly, even a single associated building provided enough contextual signal to drastically improve performance, often bringing the predicted flow close to the true value. In contrast, nodes inheriting features from a very large number of buildings exhibited a tendency toward underestimation, likely due to noisy or diluted feature aggregation (see Figure 4.14).

A major bottleneck stems from the use of a single, global weighting parameter  $\alpha$  for combining static and dynamic signals. While this design is computationally efficient and works well in the majority of cases, it limits the model's adaptability. Nodes with sparse static context cannot rely more heavily on dynamic signals even when needed, and those with dense static features cannot attenuate potential noise. Future versions of *ST-SimNet* could benefit from a node-specific or even feature-specific fusion mechanism to modulate this blend more effectively. Despite these limitations, *ST-SimNet* remains a fast and scalable model with strong generalisation across spatially heterogeneous networks. Its ability to leverage even limited morphological context for meaningful performance gains underlines its potential in urban analytics applications.

#### 4. Results

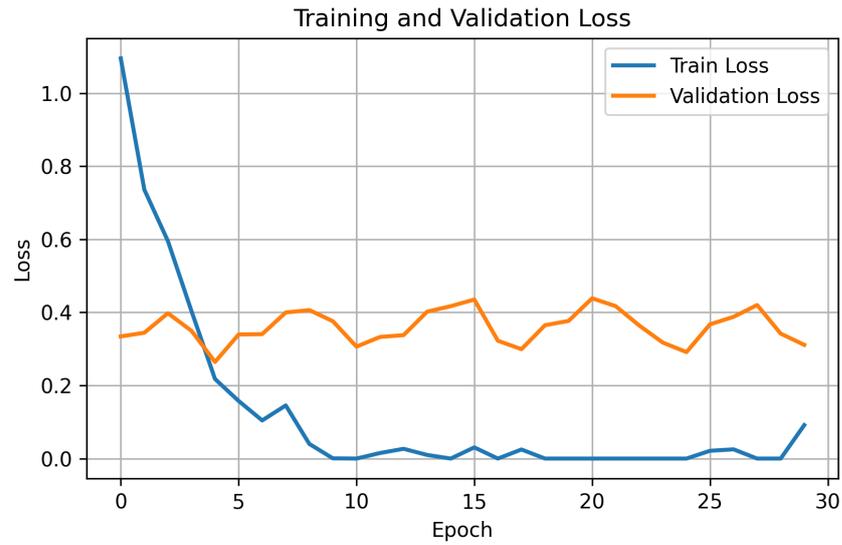


Figure 4.16.: Training and Validation Loss - *ST-SimNet* with Weekend Data

#### 4.4.4. Scenario 2: Weekdays + Weekends - Amsterdam West

This setup introduces higher temporal variability by including weekend data. The goal is to assess the robustness of *ST-SimNet* under noisier and less periodic conditions, and to examine whether static feature integration helps mitigate generalisation challenges posed by irregular patterns.

**Model Configuration** The model is configured with the following hyperparameters:

- Temporal window  $L = 96$  (past 8 hours),
- Prediction horizon  $H = 6$  (next 0.5 hour),
- Temporal convolution kernel size  $K_t = 3$ ,
- Spatial kernel order  $K_s = 3$ ,
- Two *ST-ConvBlock*,
- Activation function: GLU (Gated Linear Unit),
- Dropout rate: 0.3,
- Learning rate  $\eta = 0.0005$ , with decay  $\gamma = 0.8$  every 5 epochs,
- Weight decay (L2 regularisation): 0,001,
- Batch size: 10,
- Optimiser: AdamW.

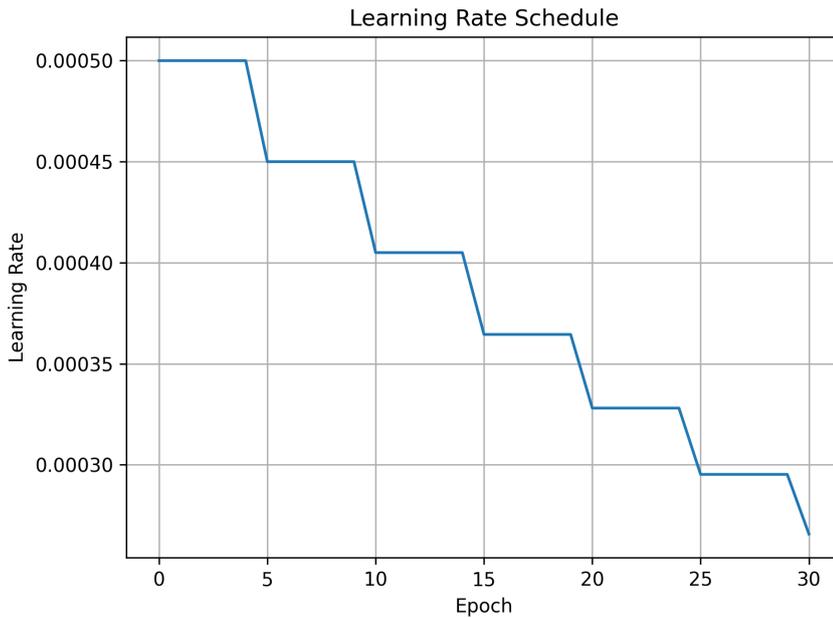


Figure 4.17.: Learning Rate Schedule - *ST-SimNet* with Weekend Data

The inclusion of weekends in the training data introduces disruptions in otherwise predictable weekday traffic flows. These irregularities appear to negatively affect Monday predictions in particular, as the network tries to infer weekday flow patterns using input sequences from the weekend, which are structurally and behaviourally distinct.

Figure 4.16 shows the learning dynamics. While the training loss drops steadily, the validation loss, although jittery, remains relatively flat, indicating early over-fitting and reduced generalisation. The learning rate schedule (Figure 4.17) is followed as intended but does not alleviate the generalisation issue.

**Spatial and Temporal Evaluation** Despite the higher noise, *ST-SimNet* maintains a strong alignment of predicted values with the true values in aggregate terms, as shown in the scatter plot (Figure 4.18) and the error histogram (Figure 4.19). However, both figures reveal higher variance in predictions and more outliers than in the weekday-only scenario. The node-level performance plot (Figure 4.21) indicates that nodes with sparse or noisy UMD vectors suffer more compared to the weekday-only setup.

**Conclusions** The integration of weekend data introduces greater uncertainty, primarily due to the mismatch between training and prediction contexts. When predicting Monday morning flows, the network often draws on input windows dominated by weekend behaviour, reducing predictive accuracy. The histogram (Figure 4.19) shows broader error dispersion, and the scatter plot (Figure 4.18) reveals several high-volume over and underpredictions. The per-node analysis (Figure 4.21) and UMD saliency (Figure 4.20) confirm that some nodes, especially those with either very sparse or very dense UMD descriptors, still benefit from static features.

4. Results

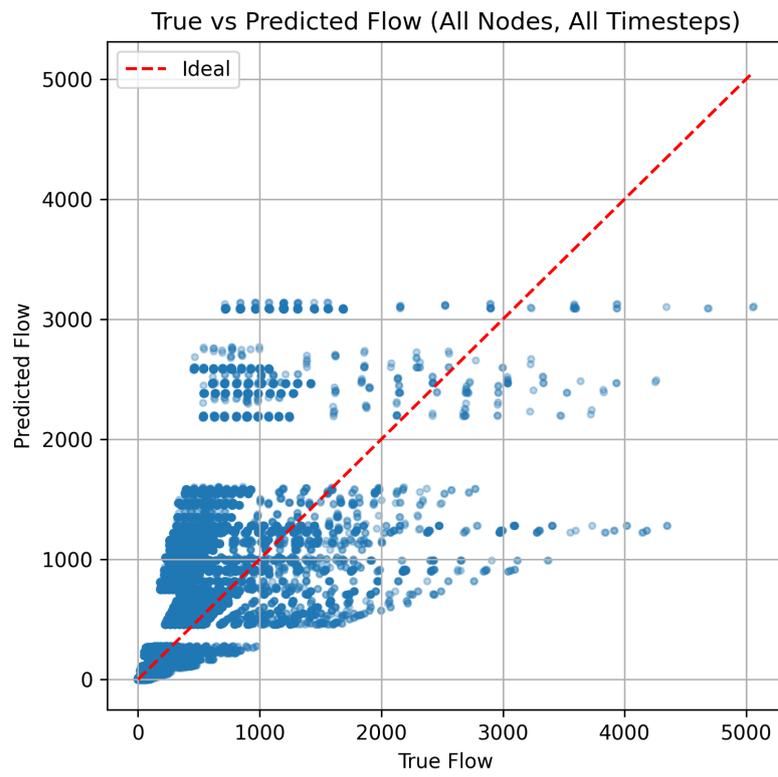


Figure 4.18.: Scatter Plot: True vs Predicted Flow (*ST-SimNet*, Weekdays + Weekends)

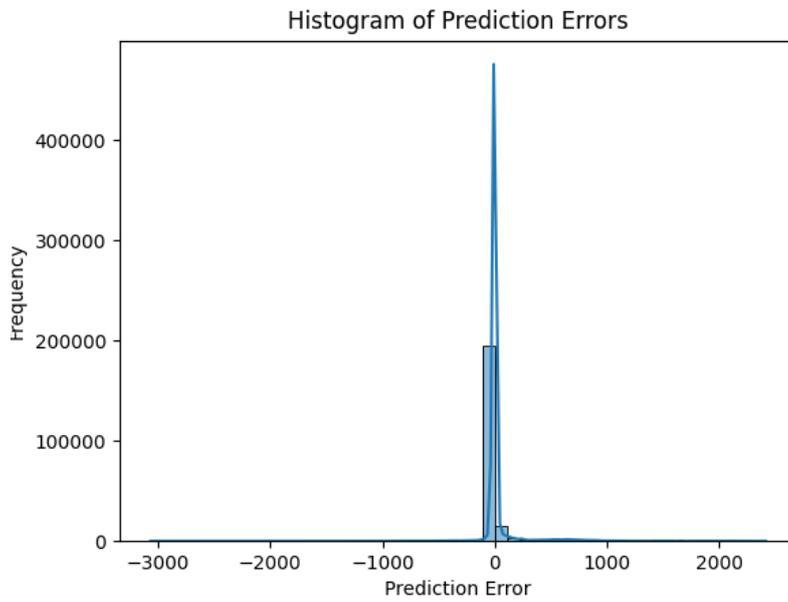


Figure 4.19.: Histogram of Prediction Errors (ST-SimNet, Weekdays + Weekends)

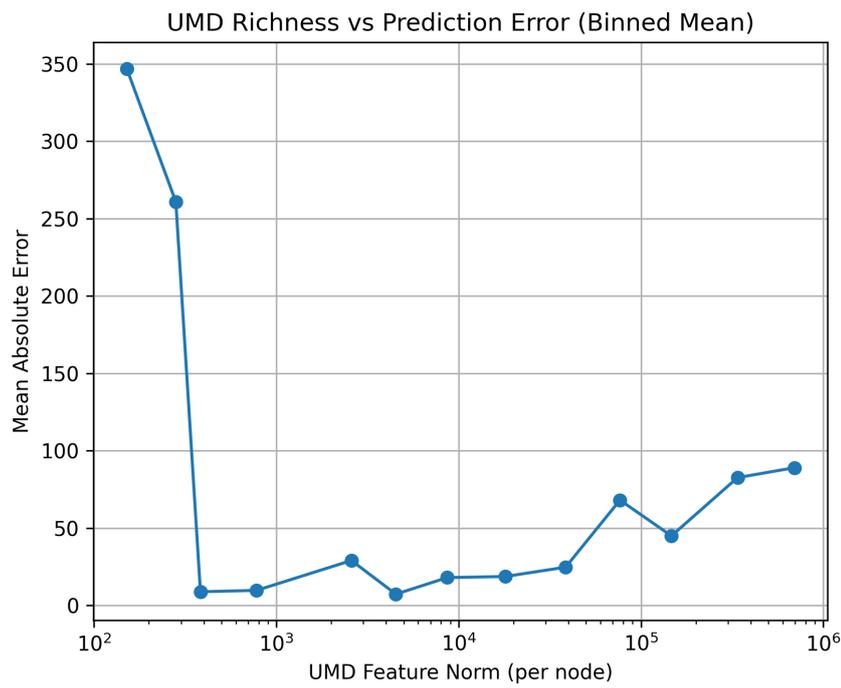


Figure 4.20.: UMD Feature Richness vs Prediction Error (ST-SimNet, Weekends Included)

## 4. Results

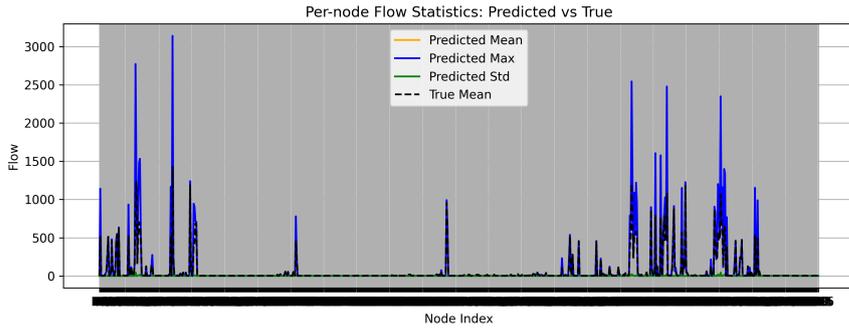


Figure 4.21.: Per-node Flow Statistics: Predicted vs True (ST-SimNet, Weekdays + Weekends)

In summary, mixing weekday and weekend data under a fixed input structure diminishes the model’s ability to generalise. A cleaner strategy would be to treat weekdays and weekends as distinct prediction domains, each with a tailored input length and possibly separate training sessions.

## 4.5. Application of ST-SimNet to a Larger Area

In this experiment, ST-SimNet was applied to a larger geographical area using the same UMD features as in the earlier scenario. The graph was regenerated based on the new nodes input with a corresponding adjacency matrix covering the center and peripheries (see Appendix B.1).

**Model Configuration** The model was trained with the following parameters: historical input window  $n_{\text{his}} = 24$ , prediction horizon  $n_{\text{pred}} = 6$ , time interval of 5 minutes, and two spatio-temporal convolutional blocks each with  $K_t = 3$  temporal filters. The spatial convolution employed Chebyshev graph convolution with a support size of  $K_s = 3$ , and random-walk normalised adjacency as the graph shift operator. Training utilised the AdamW optimiser with learning rate 0.002, weight decay 0.001, batch size 10, and dropout rate 0.1. A learning rate decay was applied every 10 epochs ( $\gamma = 0.6$ ), with early stopping patience of 10 epochs.

### 4.5.1. Training Performance

The learning rate schedule (Figure 4.22) shows a clear decay trend, facilitating stabilised training over the 100 epochs. Correspondingly, the training and validation loss curves (Figure 4.23) exhibit convergence with only minor fluctuations in the later stages. Both training and validation loss continued to decrease throughout the entire training. Notably, early stopping was not triggered, indicating that the model maintained generalisation throughout training.

#### 4.5. Application of ST-SimNet to a Larger Area

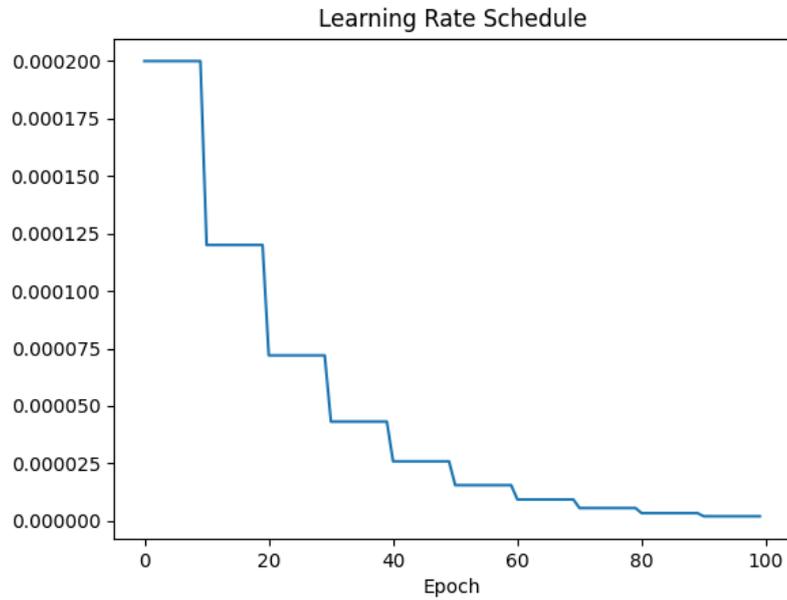


Figure 4.22.: Learning rate schedule during training for the full-area model.

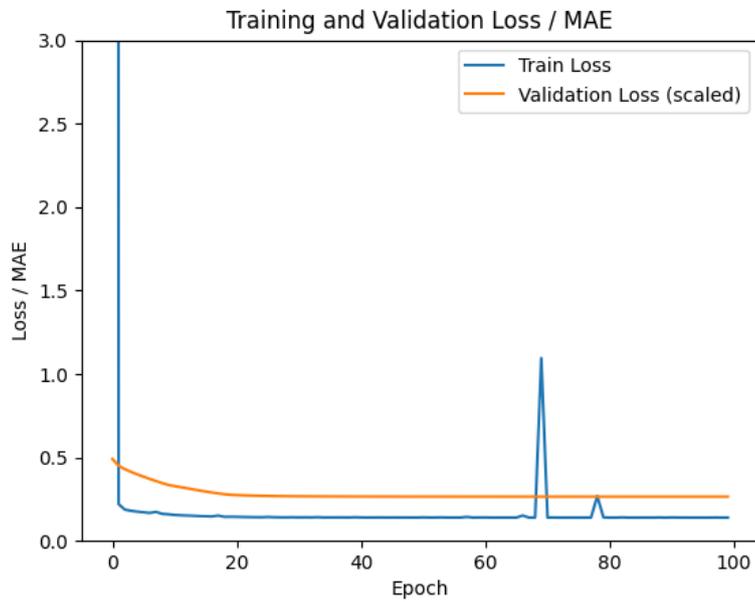


Figure 4.23.: Training and validation loss (MAE) over 100 epochs.

## 4. Results

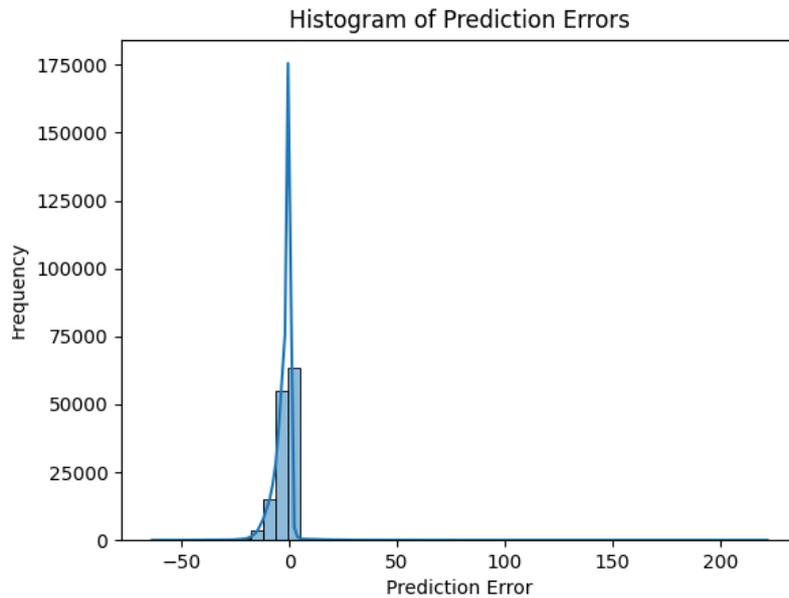


Figure 4.24.: Histogram of prediction errors. Most predictions are close to the ground truth.

### 4.5.2. Prediction Accuracy

The prediction histogram (Figure 4.24) shows a distribution tightly centred around zero, suggesting a low prediction bias. Most errors fall within a narrow range, with very few extreme deviations. The scatter plot of predicted vs. true values (Figure 4.25) illustrates strong alignment along the ideal diagonal, confirming high prediction fidelity.

### 4.5.3. Spatial Heterogeneity of Performance

Per-node statistics (Figure 4.27) reveal that the model accurately captures both peak and low-activity nodes, with predicted means, maxima, and standard deviations closely tracking the ground truth. A minority of nodes still display small variance under- or over-predictions, which may stem from feature sparsity, local anomalies in the freight signal or noise.

### 4.5.4. Feature Contribution Analysis

The contribution of urban morphological richness was further assessed by relating prediction errors to the norm of UMD feature vectors per node. As shown in Figure 4.26, nodes across all levels of morphological richness achieved remarkably low errors—typically below an MAE of 10. This marks a substantial improvement over earlier simulations, where nodes lacking UMD features often exceeded 800 in mean squared error. The observed stability confirms that even minimal morphological context helps anchor the predictions, and that [ST-SimNet](#) has generalised well to diverse spatial conditions. Elevated errors were observed

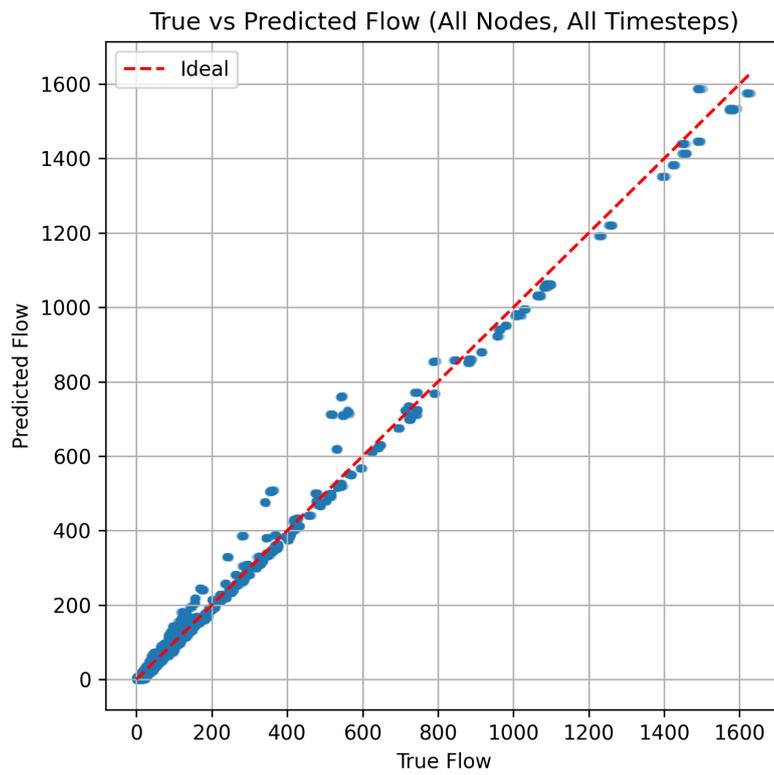


Figure 4.25.: Predicted vs. true flow values for all nodes and all time steps.

#### 4. Results

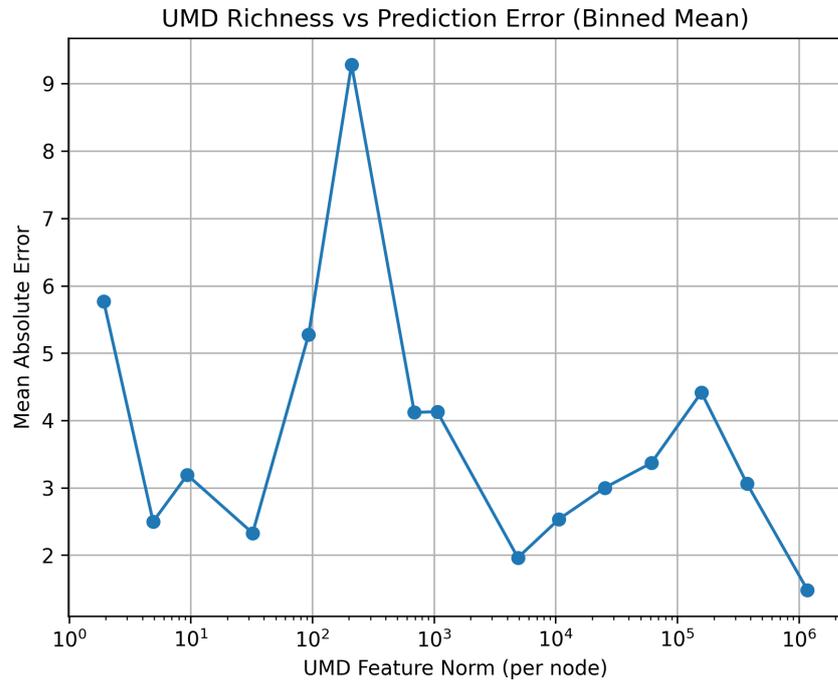


Figure 4.26.: Mean absolute error binned by UMD feature norm. Moderate feature richness leads to best performance.

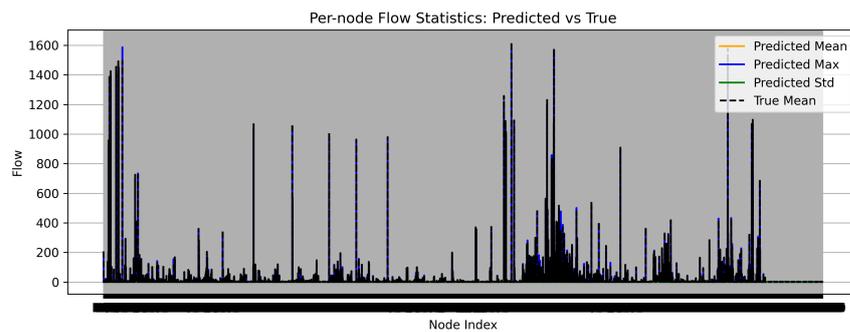


Figure 4.27.: Per-node prediction statistics. Predicted and true values exhibit strong correspondence.

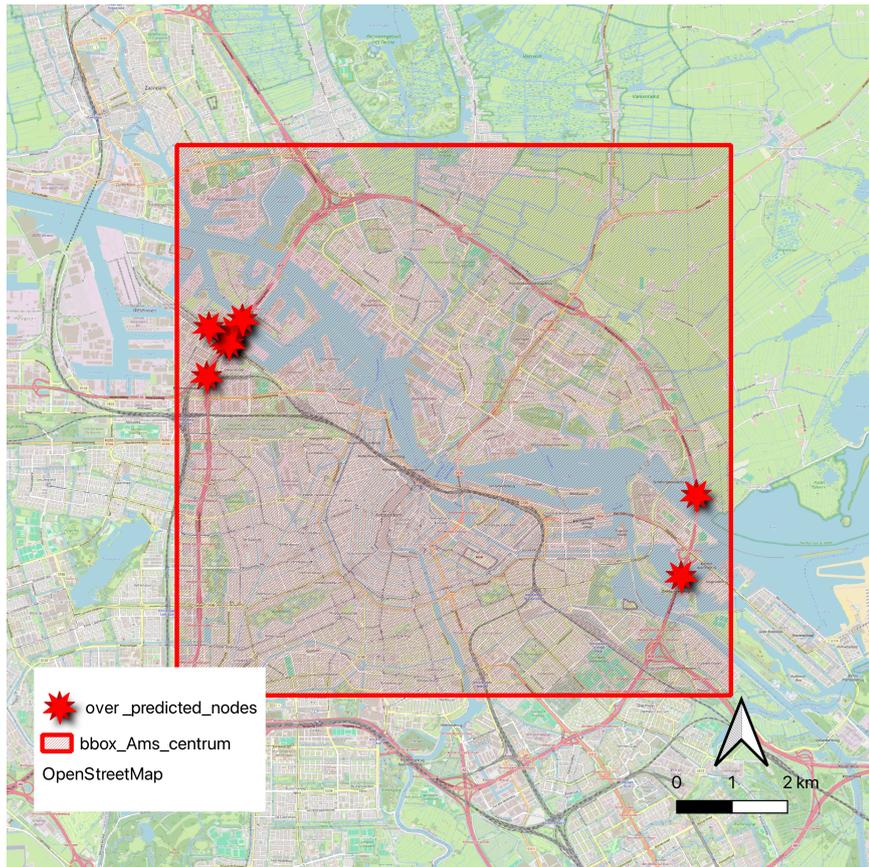


Figure 4.28.: Spatial distribution of over-predicted nodes, shown in red. Most are located along major highway infrastructure (A10)

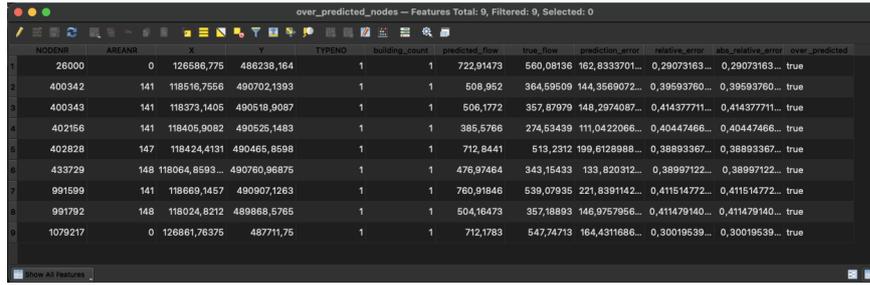
only for a small subset of isolated nodes, supporting the previous hypothesis that spatial remoteness and UMD sparsity lead to most prediction failures.

#### 4.5.5. Visual Inspection of Results in QGIS

To complement the quantitative model evaluation, a visual inspection was conducted in QGIS to spatially assess nodes with significant prediction errors. Figure 4.28 highlights the locations of nodes that were markedly over-predicted by the model. These nodes are predominantly situated along high-capacity highway segments, such as the A10 ring road surrounding Amsterdam. Despite their seemingly strategic location within the network, they lack sufficient urban morphology features - most of them inherit no building-based descriptors and have many missing PC6-level attributes (Figure 4.29), limiting the contextual information available to the model. The consistent over-prediction observed at these nodes, ranging from 129% to 142% of the true flow, can be primarily attributed to their peripheral location and minimal morphological anchoring.

Due to the absence of both building-level and postcode-level data, the over-predicted nodes

## 4. Results



NODEID	PC6NR	X	Y	TYPENO	building count	predicted_flow	true_flow	prediction_error	relative_error	abs_relative_error	over_predicted
26000	0	126586,775	486238,164	1	1	722,91473	560,08136	162,833701...	0,29073163...	0,29073163...	true
400342	141	118516,7556	490702,1393	1	1	508,952	364,59509	144,3569072...	0,39593760...	0,39593760...	true
400343	141	118373,1405	490518,9087	1	1	506,1772	357,87979	148,2974087...	0,414377711...	0,414377711...	true
402156	141	118405,9082	490525,1483	1	1	385,5766	274,53439	111,0422066...	0,40447466...	0,40447466...	true
402828	147	118424,4131	490465,8598	1	1	712,8441	513,2312	199,6128988...	0,38893367...	0,38893367...	true
433729	148	118064,8593...	490760,96875	1	1	476,97464	343,15433	133,820312...	0,38997122...	0,38997122...	true
991599	141	118669,1457	490907,1263	1	1	760,91846	539,07935	221,8391142...	0,411514772...	0,411514772...	true
991792	148	118024,6212	489868,5765	1	1	504,16473	357,18893	146,9757956...	0,411479140...	0,411479140...	true
1079217	0	126861,76375	487711,75	1	1	712,1783	547,74713	164,4311686...	0,30019539...	0,30019539...	true

Figure 4.29.: Summary of over-predicted nodes and corresponding features.

lacked any meaningful static input. All affected nodes had zero associated buildings, and most, if not all, of their PC6-level features were null. Consequently, the model was deprived of morphological context and relied exclusively on dynamic signals to generate predictions. While the dynamic component effectively captures temporal patterns, it does not provide sufficient spatial differentiation to account for local variations in freight activity, particularly in transitional zones such as highway ramps or peripheral ring roads. Furthermore, these nodes were spatially distant from others, limiting the ability of neighbourhood aggregation (via graph convolutions) to supplement missing context with information from better-informed neighbours.

### 4.5.6. Conclusion

The application of *ST-SimNet* to a significantly larger urban area demonstrates its scalability and robustness in generalising spatio-temporal freight flow predictions beyond the initial experimental scope. The model architecture featuring dual *ST-ConvBlock*, gated temporal convolutions, and a fusion mechanism for integrating static urban morphology descriptors exhibited stable convergence throughout training, as indicated by the smooth learning rate decay and alignment of training and validation loss curves (Figures 4.22, 4.23). Prediction accuracy remained high, with the majority of predicted values closely tracking the true flow values (Figure 4.25), and per-node flow statistics showed strong correspondence in mean and variance across the network (Figure 4.27).

Spatial inspection of the results uncovered key patterns in model behaviour. Nodes with richer morphological descriptors continued to benefit from increased contextual awareness, confirming earlier findings from the smaller-scale setting. Conversely, nodes located along peripheral motorways and industrial corridors with limited urban morphology features tended to be over-predicted (Figure 4.28). This behaviour is not indicative of model overfitting, but rather highlights the challenges of learning freight flow patterns in structurally homogeneous yet functionally complex regions. These areas lack sufficient feature richness to fully inform the model, which in turn defaults to stronger reliance on dynamic input signals.

Interestingly, results from the extended simulation reveal a mitigating effect of graph convolutional stacking. Nodes located in dense urban centres, such as Amsterdam’s inner districts, often benefit from neighbourhood aggregation, even when their own static features are sparse. Here, surrounding nodes rich in urban morphology descriptors contribute indirectly, enhancing the model’s understanding through multi-hop propagation. This effect is

#### *4.5. Application of ST-SimNet to a Larger Area*

particularly visible in the centre, where the interconnectedness of the road network and urban fabric allows nodes to inherit contextual awareness from their neighbours. Conversely, isolated motorway nodes on the periphery - lacking both direct and neighbouring static features, remain prone to overestimation. These findings emphasise that spatial embedding within the built environment, not just individual feature richness, plays a crucial role in freight flow prediction.



## 5. Conclusions and Future Work

### 5.1. Discussion

This thesis introduced ST-SimNet, a Spatio-Temporal Graph Neural Network architecture designed to integrate urban morphology with temporal freight data for short-term prediction of freight flows. The work sought to challenge conventional assumptions that freight forecasting must rely solely on dynamic data such as vehicle counts or precomputed flows. By encoding rich, static morphological descriptors derived from building characteristics and PC6-level demographic indicators into the GNN framework, ST-SimNet demonstrates the value of contextual urban features in guiding predictive freight analytics.

The experimental findings indicate several important insights. First, the inclusion of Urban Morphology Data significantly reduced prediction error, particularly in nodes that were structurally well-informed by surrounding morphology. Nodes with richer building-derived features, such as commercial zones or dense residential areas, exhibited improved forecast stability and alignment with ground-truth flows. These results support the hypothesis that the spatial context embedded in urban form plays a non-trivial role in shaping freight dynamics.

Second, the model showed strong scalability. When applied to a significantly larger area encompassing broader sections of Amsterdam, ST-SimNet maintained stable convergence and generalisation. Loss curves and prediction statistics affirmed the robustness of the architecture under more complex spatial variability. Notably, even in under-informed or peripheral nodes (e.g. those along highways), the model produced interpretable over-prediction patterns highlighting the need for additional context layers (e.g. road type, logistic hub proximity) rather than structural flaws in the model itself.

Third, the fusion strategy, governed by a global  $\alpha$  parameter, proved broadly effective in blending dynamic and static inputs. However, in spatially sparse regions - particularly highway, adjacent nodes with zero inherited building features and missing PC6 descriptors - the static component offers limited contextual value. In such cases, the model's reliance on static inputs may distort predictions. These findings do not undermine the utility of static fusion but rather highlight the potential for local adaptivity. Node-specific attention weighting, gating mechanisms, or residual modulation based on feature richness could help the model selectively downweight uninformative morphology signals, thereby improving robustness across heterogeneous urban contexts.

Interestingly, even in areas with sparse or missing UMD features, the spatial message passing inherent to GNNs appeared to partially compensate for static data sparsity. In dense urban regions such as central Amsterdam, neighbouring nodes often possessed rich morphological descriptors, allowing their information to propagate and inform nearby under-defined nodes through the stacked ST-Conv blocks. This emergent behaviour suggests that GNNs are not only sensitive to node-level input, but also capable of leveraging local urban structure to distribute contextual information spatially. Such interactions suggest that the

## 5. Conclusions and Future Work

chosen fusion design may be effective in certain urban contexts, but alternative integration strategies, such as injecting static features earlier in the architecture, remain to be explored.

The methodological contributions extend beyond implementation. This thesis illustrates that GNNs can be grounded in urban theory, capturing not only network connectivity but also spatial heterogeneity in freight demand. The results argue for a more integrated approach to digital twin development - one that bridges behavioural simulation and learning from spatial form. From a systems perspective, ST-SimNet represents a hybrid modelling approach that combines the rule-based precision of simulation models (e.g. MASS-GT) with the generalisation power of data-driven learning.

Lastly, the work underscores the interpretability potential of GNN-based systems. By correlating prediction accuracy with urban morphology richness and visually inspecting mis-predicted nodes, ST-SimNet enables both predictive analytics and spatial diagnosis. This duality makes the model not only a forecasting engine but also a diagnostic tool for understanding urban freight distribution logic.

### 5.2. Limitations

While the proposed framework demonstrated strong performance and flexibility, several limitations should be acknowledged to guide future improvements.

First, the aggregation of building-level features to nodes was based on simple spatial assignment, without enforcing a maximum distance threshold or applying distance-based weighting. As a result, buildings located far from a node may introduce noise into its static feature vector. Incorporating an inverse-distance weighting or setting a maximal aggregation radius could mitigate this issue and preserve local relevance.

Second, some nodes had no associated building-level features and relied solely on coarse PC6-level statistics. Rather than leaving these nodes sparsely populated, a more refined strategy could be employed: for example, interpolating missing static attributes from neighbouring nodes or constructing synthetic features based on nearby urban morphology patterns.

Third, the road network itself was treated as a homogeneous graph, without distinguishing between different road types (e.g. highways, arterial roads, local streets). Introducing categorical edge features or even continuous attributes (such as number of lanes, speed limits, or typical traffic volumes) could further enhance the model's understanding of traffic dynamics.

Moreover, the model does not currently differentiate between weekday and weekend traffic patterns during training. Introducing an additional "day type" indicator, potentially controlled by a hyperparameter, could allow ST-SimNet to adjust its temporal predictions according to the day's expected variability.

Another important limitation concerns the fusion of static and dynamic information. The current implementation employs a single global  $\alpha$  parameter to balance dynamic embeddings with static urban features uniformly across all nodes. While this approach performs robustly in most areas, it lacks flexibility in sparse or semantically misaligned regions. Nodes without inherited building features or with missing PC6 data cannot contribute meaningful static context, yet still receive a fixed static weighting. This mismatch

can lead to systematic overestimation. Introducing a locally adaptive mechanism, such as a node-specific  $\alpha_i$  or a learnable gating function, could dynamically adjust the contribution of static information based on local feature richness, improving model generalisability across heterogeneous urban forms.

Finally, missing or incomplete static data were handled using naive imputation strategies (e.g. filling NaNs with zeros). For some features, such as building year of construction, this can introduce unrealistic values. A more nuanced imputation approach could be explored, such as estimating missing erection years from energy labels, architectural characteristics, or nearest-neighbour interpolation based on similar buildings.

ST-SimNet effectively demonstrates the benefits of integrating static urban morphology with dynamic traffic data, addressing these limitations would further strengthen its accuracy, generalisation, and applicability across diverse urban contexts.

### 5.3. Conclusions

This thesis introduced *ST-SimNet*, a spatio-temporal GNN architecture tailored for urban freight flow prediction, and demonstrated its effectiveness in combining dynamic time-series data with static urban morphology descriptors. The model’s design—incorporating gated temporal convolutions, Chebyshev-based spatial GSOs, and a controlled fusion mechanism—proved capable of capturing complex, non-linear freight dynamics across both space and time in real urban settings.

Systematic experiments showed that integrating built environment features significantly enhances predictive performance, especially in spatially heterogeneous urban areas where pure time-series models struggle to generalise. Even minimal morphological context, such as association with a single building, was shown to meaningfully improve node-level predictions. At the same time, the architecture remained computationally efficient and robust, generalising well to larger city regions without modifications.

Beyond empirical results, this work fills a critical niche at the intersection of urban freight modelling, spatio-temporal deep learning, and urban morphology. It consolidates fragmented strands of the literature—ranging from grid-based forecasting, topological GNNs, and urban form analysis—into a unified framework. By embedding morphological context into predictive models, *ST-SimNet* advances the research frontier from abstract graph representations toward semantically enriched, context-aware forecasting tools.

Importantly, the approach aligns with the broader aims of digital twin initiatives by providing a scalable, data-driven engine for simulating and forecasting freight flows under realistic urban constraints. By surfacing both the opportunities and limitations of integrating static and dynamic data sources, this work lays the groundwork for future research and operational applications in urban logistics, planning, and policy.

### Answers to the Research Questions

**Main Research Question:**

## 5. Conclusions and Future Work

*To what extent can insights into urban morphology, modeled with Spatio-Temporal Graph Neural Networks, enhance the accuracy and adaptability of freight transportation predictions in the Netherlands?*

This research demonstrates that incorporating static UMDs based forecasting framework significantly enhances model performance in both accuracy and spatial generalisability. In spatially complex urban areas, such as Amsterdam, ST-SimNet consistently outperformed its dynamic-only counterparts, showing reduced MAE and more stable prediction curves. These improvements are especially pronounced at nodes with rich morphological context (e.g. commercial districts or mixed-use zones) where static features provide strong signals about freight generation and attraction.

The adaptability of ST-SimNet is evidenced by its ability to generalise across varying urban typologies. Despite the model being trained on flow data derived from a single day, it retained predictive consistency across multiple temporal configurations and spatial extents, including peripheries with lower data richness. Even in regions with sparse or incomplete UMD coverage, the model leveraged spatial message passing through the graph structure to interpolate context from well-instrumented neighbours. This emergent property underlines the benefit of embedding morphology into a graph-based architecture, where spatial dependencies can be propagated structurally rather than merely statistically.

Beyond accuracy, morphology-enhanced ST-GNNs also contribute to model interpretability. Visual inspection of residuals and correlation with feature richness revealed meaningful patterns: nodes with systematic over-predictions often coincided with poorly covered morphology data. These insights allow planners to not only trust the predictions but also diagnose areas where additional data or structural improvements are needed.

### **Architecture Suitability:**

*What are the key components and mechanisms of the ST-SimNet architecture required to capture both the spatial dependencies and temporal dynamics of urban freight flows?*

ST-SimNet is structured around two stacked ST-ConvBlocks, each of which sequentially applies a Gated Temporal Convolution, a GCN using a directional GSO, and a second temporal convolution with a GLU-based filter. This structure enables the model to learn both fine-grained short-term temporal patterns and spatial correlations informed by the urban road network. The use of GLUs allows for non-linear temporal filtering, dynamically modulating the flow of information across time steps to prioritise relevant input sequences such as freight peaks and off-peak lows.

A key innovation lies in the subsequent fusion block, where static UMDs are projected via an MLP into the same latent space as dynamic features and combined through a learnable fusion coefficient  $\alpha$ . This design allows the model to encode not only when and where freight is moving, but also why certain areas may act as attractors or generators of freight volume.

### **Graph Structure Design:**

*How should the graph representing the urban freight system be constructed using the available data (road network and spatial units), and what is the impact of different graph design choices on forecasting performance?*

The graph in ST-SimNet is constructed from the DT road network, with each node corresponding to a road junction and directed edges representing real-world traffic flows. This design preserves both directionality and topological fidelity, allowing freight movement dynamics

to be encoded through true infrastructural connectivity rather than artificial spatial grids. The adjacency matrix is normalised using a random-walk scheme, which retains flow asymmetry and supports realistic propagation of traffic signals through the network.

Empirical results show that this infrastructure-aligned graph significantly outperforms simplified proximity-based or uniform-grid approaches. Grid-based discretisation, while computationally convenient, often misrepresents local accessibility patterns and dilutes morphological context. In contrast, the road-network graph enables message passing to operate along actual freight pathways, improving both predictive sharpness and spatial consistency—particularly in areas where morphology or flow data are sparse.

Moreover, by leveraging a fine-grained node resolution ( $N = 10,691$ ) and directional edge encoding, the model is able to distinguish between inbound and outbound freight movements, a key requirement for realistically modelling urban logistics.

#### Feature Integration Strategies:

*What is the most effective way to integrate urban morphological features (e.g. building usage, density, and other CBS-derived statistics) into the ST-SimNet model, and how do different feature aggregation or encoding techniques influence the accuracy of freight flow forecasts?*

Urban Morphology Descriptors (UMDs) were aggregated at the node level using spatial joins between road junction buffers and PC6-level data. Features included both building-level attributes and population-level indicators. These were standardised and projected via a dedicated two-layer MLP, then combined with dynamic node embeddings using a convex fusion strategy governed by a learnable scalar  $\alpha \in [0, 1]$ .

This integration method proved effective in morphologically dense urban contexts, where even simple descriptors provided meaningful priors for freight generation and attraction. In sparse regions—especially near highways or mono-functional zones—the static features were often insufficient or uninformative, which in turn diluted model confidence and accuracy. Nonetheless, due to the graph’s spatial connectivity, contextual information could still propagate from adjacent nodes with richer morphological signatures.

## 5.4. Future Work

There are several promising directions for extending this work, both in terms of improving the model and broadening its applications:

- **Application-specific integration:** Beyond freight forecasting, this model architecture could be adapted to various domains where spatio-temporal dynamics are shaped by urban context. Examples include:
  - Energy demand prediction and grid optimisation, where energy demand is influenced by land use and building functions;
  - Military and emergency logistics, where robust predictions of movement in urban terrain are critical;
  - Delivery optimisation for e-commerce in areas with mixed urban typologies.

## 5. Conclusions and Future Work

- **Topographic augmentation:** A natural next step is to incorporate topography - elevation, slope, and accessibility, since freight patterns are often constrained by physical terrain. Integrating a digital elevation model (DEM) could help explain movement bottlenecks or preferred paths in hilly or coastal areas.
- **Improved fusion mechanisms:** While a global weighting parameter  $\alpha$  was used to balance static and dynamic features, this approach lacks flexibility for nodes with very sparse or very rich feature sets. Future work could explore attention-based fusion, node-specific gates, or temporal modulation layers that dynamically adjust how static context is used.
- **Generalisation and transfer learning:** Applying the model across multiple cities or temporal spans would help assess its generalisability. Domain adaptation techniques could enable pretraining on one region and fine-tuning on another, improving practical deployment in settings with limited data.
- **Probabilistic and multi-modal forecasting:** Finally, expanding the model to produce probabilistic predictions or jointly forecast related variables (e.g. speed, occupancy) would align it more closely with digital twin applications in logistics, mobility, and infrastructure planning.

# A. Reproducibility self-assessment

## A.1. Marks for each of the criteria

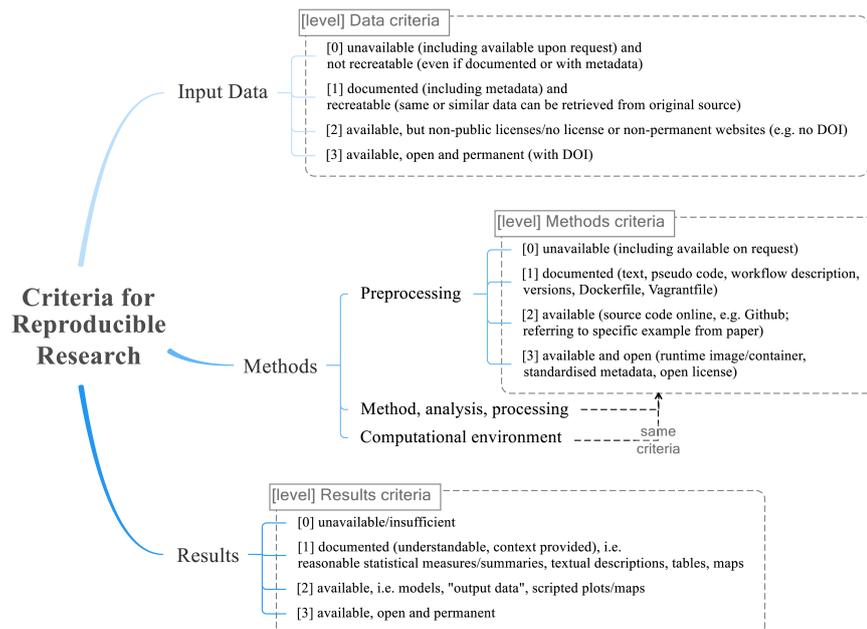


Figure A.1.: Reproducibility criteria to be assessed.

Grade/evaluate yourself for the 5 criteria (giving 0/1/2/3 for each):

1. input data: 0
2. preprocessing: 0
3. methods: 0
4. computational environment: 1
5. results: 1

## A.2. Personal Reflection

Working on this thesis has been one of the most enriching technical and conceptual experiences in my academic journey. What began as a modelling exercise quickly evolved into a

### *A. Reproducibility self-assessment*

deeper exploration of how cities operate, how data reflects (and sometimes distorts) reality, and how much contextual nuance is needed to make robust predictions. Integrating static urban morphology made me realise that even sophisticated models like STGCN are often blind to the very spatial heterogeneity that defines urban systems. Throughout this project, I also became more aware of the tension between engineering simplicity and real-world complexity. I was initially optimistic that simply adding static data would improve everything, but the experiments revealed that this process requires nuance. Some nodes benefited immensely, while others suffered due to noisy or sparse inputs. These moments, when the model over-predicted certain flows or failed to generalise were frustrating, but also where I learned the most.

Finally, I found joy in building something that didn't just "run," but produced insights. Visualising errors spatially, tweaking hyper-parameters, and interpreting model failures gave me confidence not only in my technical skills but also in my ability to think critically and iterate.

## B. Areas of Interest

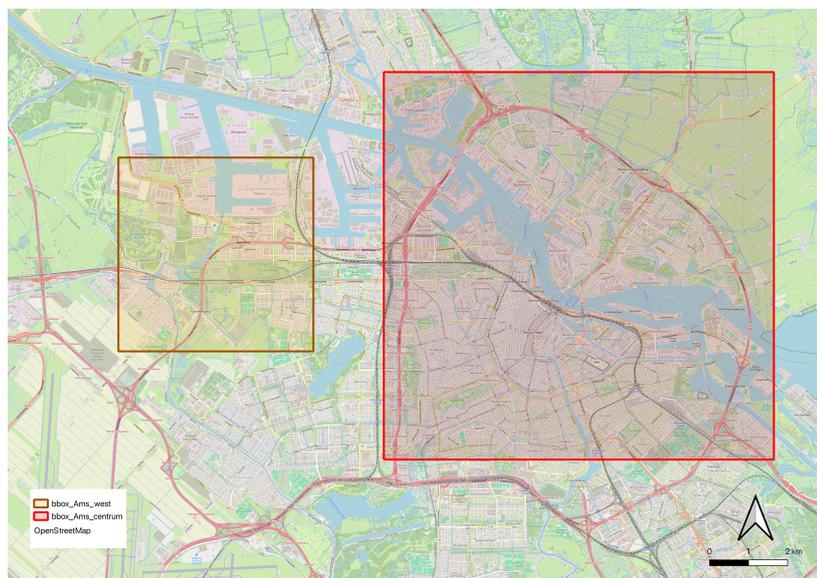


Figure B.1.: AOIs for trainings, Amsterdam, EPSG:28992. Amsterdam West (orange) with 817 nodes; Centre of Amsterdam (red) with 10691 nodes.



## C. MASS-GT parameters

### C.1. Vehicle Types

ID	Is Freight Type	Available in Parcel Module	Description
0	1	0	Truck (small)
1	1	0	Truck (medium)
2	1	0	Truck (large)
3	1	0	Truck+trailer (small)
4	1	0	Truck+trailer (large)
5	1	0	Tractor+trailer
6	1	0	Special vehicle
7	0	1	Van
8	0	1	LEVV
9	0	0	Moped

Table C.1.: Vehicle Type Classification

### C.2. NSTR Goods Classification

ID	Description
0	Agricultural products and live animals
1	Foodstuffs and animal fodder
2	Solid mineral fuels
3	Petroleum products
4	Ores and metal waste
5	Metal products
6	Crude and manufactured minerals, building materials
7	Fertilizers
8	Chemicals
9	Machinery, transport equipment, manufactured articles and miscellaneous articles
-1	Empty

Table C.2.: NSTR Goods Classification

### C.3. Logistic Segment Classification

ID	Description
0	Food (general cargo)
1	Miscellaneous (general cargo)
2	Temperature controlled
3	Facility logistics
4	Construction logistics
5	Waste
6	Parcel (consolidated flows)
7	Dangerous
8	Parcel (deliveries)

Table C.3.: Logistic Segment Classification

### C.4. Transport Flow Classification

ID	Is External	Description
1	0	Producer to Consumer
2	0	Producer to DC
3	0	DC to Consumer
4	0	Producer to TT
5	0	DC to DC
6	0	TT to Consumer
7	0	DC to TT
8	0	TT to DC
9	0	TT to TT
10	1	External Producer/Consumer to/from Producer/Consumer
11	1	External Producer/Consumer to/from DC
12	1	External Producer/Consumer to/from TT

Table C.4.: Transport Flow Classification

# Bibliography

- Al Sahili, Z. and Awad, M. (2023). Spatio-temporal graph neural networks: A survey. *arXiv preprint arXiv:2301.10569*.
- Aljohani, K. and Thompson, R. G. (2016). Impacts of logistics sprawl on the urban environment and logistics: Taxonomy and review of literature. *Journal of Transport Geography*, 57:255–263.
- Battaglia, P. W., Hamrick, J. B., Bapst, V., et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.
- Cruz-Daraviña, P. A. and Suescún, J. P. B. (2021). Freight operations in city centers: A land use conflict in urban planning. *Land Use Policy*, 108:105575–105575.
- Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. (2017). Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*.
- de Bok, M. and Tavasszy, L. A. (2018). An empirical agent-based simulation system for urban goods transport (mass-gt). *Procedia Computer Science*, 130:126–133.
- Gonzalez, J. and Smith, L. (2023). A framework for integrating freight transport, urban land planning, and economic geography. *Urban Science*, 8(2):30.
- Huang, R., Huang, C., Liu, Y., Dai, G., and Kong, W. (2020). Lsgcn: Long short-term traffic prediction with graph convolutional networks. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, pages 2355–2361. International Joint Conferences on Artificial Intelligence Organization.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 448–456. PMLR.
- Jain, A., Haghghat, E., Nabian, M., and Nelaturi, S. (2024). Using graph neural networks for additive manufacturing.
- Jepsen, T. S., Jensen, C. S., and Nielsen, T. D. (2019). Graph convolutional networks for road networks. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 460–463. ACM.
- Jiang, F., Ma, J., Webster, C. J., Chiaradia, A. J. F., Zhou, Y., Zhao, Z., and Zhang, X. (2023). Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *IEEE Transactions on Knowledge and Data Engineering*.
- Martin, A. and Reichmann, N. (2024). What is a knowledge graph?
- Merchant, A., Batzner, S., Schoenholz, S. S., Aykol, M., Cheon, G., and Cubuk, E. D. (2023). Scaling deep learning for materials discovery. *Nature*, 624:80–85.

## Bibliography

- Rahmani, S., Baghbani, A., Bouguila, N., and Patterson, Z. (2023). Graph neural networks for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 24(8):8846–8885.
- Spruijtenburg, D., Walraven, E., Sterkenburg, R., and van der Tuin, M. (2025). Parking in macroscopic transport models: Modelling parking capacities in traffic assignment. In McNally, C., Carroll, P., Martinez-Pastor, B., Ghosh, B., Efthymiou, M., and Valantasis-Kanellos, N., editors, *Transport Transitions: Advancing Sustainable and Inclusive Mobility*, Lecture Notes in Mobility, pages 105–119. Springer, Cham.
- TNO (2020). Decamod: Toolbox voor rekenen aan CO<sub>2</sub>-reductie in transport en logistiek. Technical Report TNO-2020-R11938, TNO.
- TNO (2023). Urban strategy: Digital twins for sustainable mobility and liveable cities. <https://www.tno.nl/en/newsroom/insights/2023/04/urban-strategy-digital-twins-mobility/>. Accessed: 2025-04-14.
- Wang, P., Luo, X., Tai, W., Zhang, K., Trajcevski, G., and Zhou, F. (2024). Urbanflow: A unified framework for urban flow prediction with graph neural networks. *ACM Transactions on Intelligent Systems and Technology*, 15(3):Article No.: 45, 1–25.
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24.
- Xiong, L., Yuan, X., Hu, Z., Huang, X., and Huang, P. (2024). Gated fusion adaptive graph neural network for urban road traffic flow prediction. *Neural Processing Letters*, 56(9).
- Xue, J., Jiang, N., Liang, S., et al. (2021). Quantifying spatial homogeneity of urban road networks via graph neural networks. *Nature Machine Intelligence*. Accepted for publication.
- Yu, B., Yin, H., and Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3634–3640. International Joint Conferences on Artificial Intelligence Organization.

## **Colophon**

This document was typeset using  $\text{\LaTeX}$ , using the KOMA-Script class `scrbook`. The main font is Palatino.



