National College of Ireland

Hybrid Spatiotemporal GraphWaveNet for Real-Time Traffic Forecasting on Dynamic Graphs

MSc Research Project
MSc in Data Analytics (MSCDAD_C)

Karthik Kota
Student ID: x23254653

School of Computing
National College of Ireland

Supervisor: Sallar Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Karthik Kota

Student ID: X23254653

Programme: MSc in Data Analytics **Year:** 2025

Module: Research Practicum part - 2

Supervisor: Sallar Khan

Submission Due Date: 11th Aug 2025

Project Title: Hybrid Spatiotemporal GraphWaveNet for Real-Time Traffic Forecasting on Dynamic Graphs

Word Count: 11233 **Page Count** 31

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Karthik Kota

Date: 11th Aug 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Table of Contents

Hybrid Spatiotemporal GraphWaveNet for Real-Time Traffic Forecasting on Dynamic Graphs

Karthik Kota
X23254653

Abstract

Real-time traffic sensor networks implicitly require well-performing models that capture space-time invariant structures, such as topology of the road yet consider time-varying aspects, e.g. time-of-day congestion. Traditional spatiotemporal graph neural networks (ST-GNNs) are often limited to fixed graphs or retraining models entirely when the distribution of features changes, which are not scalable or adaptive. The work suggests a hybrid modeling architecture that comprises Static Spatial Embeddings (SE) together with Feature-Based (FB) dynamic graph construction as a GraphWaveNet (GWN) backbone. We trained 16 spatiotemporal models, and the 7 best of them were tested on the 2 tasks of 3-step one-shot (H3) and 24-step autoregressive (H24-AR). Another 24-step one-shot task with time-of-day attributes (H24-TOD) employed three models (a static no-change baseline and two SE - FB hybrids with cosine and RBF attention). The analysis was done on a 60-day, 554-node traffic dataset using RMSE, MAE, MAPE, latency, sensitivity to sensor dropout, and scalability to 500 nodes. Hybrid models performed better than the static and adaptive baselines Most notably, the RBF variant performed the best on RMSE whereas the cosine variant performed best in terms of MAPE and robustness to dropouts. Both models will be scaled linearly and generalized without retraining. whereas, complete graph construction was expensive, and peak-hour errors were an indication that it will perform better. Furthermore, this involves GPU-accelerated inference, dynamic graph updates and optimisation strategies at peak times.

Keywords: GraphWaveNet, Spatio-Temporal Graph neural networks, Feature-based attention, Incremental Update Edge, Traffic forecast, Computation efficiency, Robustness

1 Introduction

The traffic systems in urban areas have a strong rush-hour traffic peak, ad hoc detours due to accidents, and long-range couplings in the form of traffic travelling on the arterial units and the ring roads. Catching these dynamics requires models that can consume high-frequency and noisy sensor streams, the volume of which has scaled exponentially as the IoT devices along a roadside have increased. Graph Neural Networks (GNNs) allow learning of this non-Euclidean structure in a principled way by casting a junction as a node and directional flow correlations as edges, and they now dominate short horizon forecasting leaderboards (Yu et al., 2018; Li et al., 2018; Wu et al., 2019). Such operational deployments, however, continue to

require either tuning offline static graphs or retraining with full adaptation, requiring recalculation of the adaptive graph in each batch both approaches break in the presence of load-specific corrections, as when a crash instantaneously reroutes traffic, or sensor drift obstructs recent history (Bai et al., 2020; Liu et al., 2023; Han et al., 2024). To bridge this divide we offer a Hybrid Spatial-Embedding integration with Feature-Based Graph WaveNet (SE-FB GWN) where the edge weights are recalculated only on the forward pass, where the stationary spatial priors are used in conjunction with instantaneous feature similarity. This forward-only update retains the sub-second latency, as well as increases the accuracy and robustness of large urban graphs.

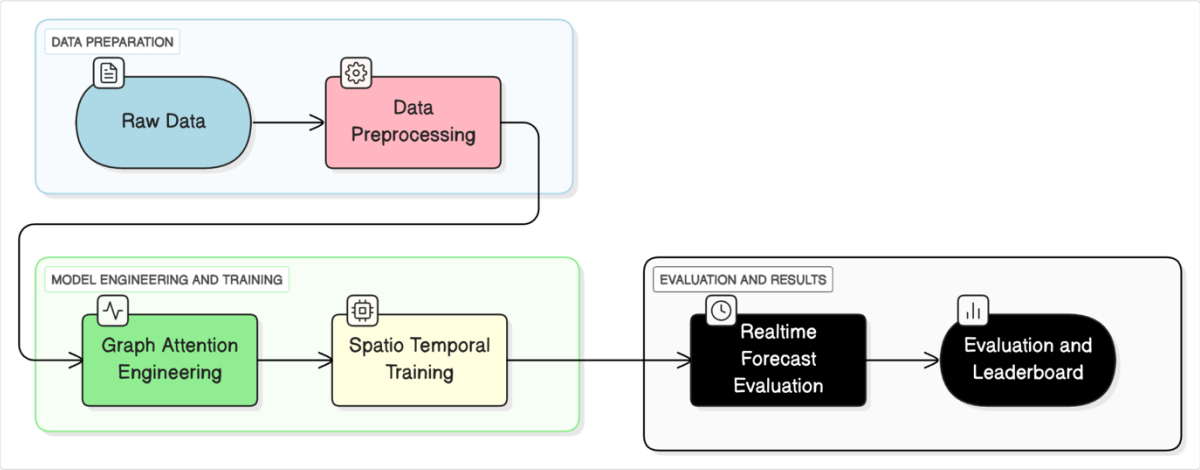

Figure 1. Hybrid GraphWaveNet pipeline: from real-world traffic data ingestion to dynamic training and real-time evaluation.

As seen in Figure 1, the proposed hybrid attention-based GraphWaveNet (GWN) framework has this end-to-end architecture. The workflow starts by collecting and pre-processing raw traffic data of SCATS in Ireland. The pre-processed data is then passed on to the central model-engineering part, where dynamic graph attention is empowered and temporal dependencies are learned through spatiotemporal training. The last segment carries out real-time prediction, robustness testing and ends with performance benchmarking and comparison on a leaderboard. This modular type provides flexibility, efficiency, and practicality to the city traffic forecasting activity.

1.1 Background

The history of spatio-temporal graph networks can be easily divided into two major eras: the fixed-graph one where the formative models like STGCN (Yu et al., 2018) and DCRNN (Li et al., 2018) used fixed convolutions on fixed graphs to represent past movement, or diffusion processes on stationary topologies; and the learned-graph one where Graph WaveNet (Wu et al., 2019) and AGCRN (Bai et al., 2020) gives up this structure and learns the graph connectivity. Even more recently, Zhang et al. (2024) fulfilled such chronology, presenting an adaptive spatial-temporal graph convolutional network called AdpSTGCN that dynamically re-weights edges in real-time based on feature drift. STAttention-GWN (Liu et al., 2023), Dynamic ST-Interactive GNN (Gao et al., 2023), and BigST (Han et al., 2024) models have subsequently improved on standard benchmarks through their added concepts including multi-

head attention, diffusion-kernel-based models, and methods with linear complexity. DSTSPYN (Wang et al., 2025) is a complementary line of work, which scales linearly in a similarity-pyramid design and shows that adaptive sparsity is compatible with very wide receptive fields. Each of these methods uses backpropagation to recalculate edges during training, but during inference the learned graph is fixed. In the adaptive spatiotemporal network of Xiao et al. (2025), a forward-refresh dynamic graph is also investigated, but they are not tested beyond sub-200-node testbeds. This restricts applications where millisecond-level latency is required. As known to the literature so far, no model has been reported that adapts its graph entirely during the forward pass on the scale of a city and while achieving millisecond-level latency, and while still achieving competitive accuracy.

1.2 Motivation and Importance

However, modern traffic-management centres are asked to receive hundreds of thousands of loop-detector updates per minute and simultaneously they need to be able to produce forecasts in a 100-milliseconds or faster latency rate which is fast enough to synchronise signal timings in a single cycle, signal broadcasts on congestion to navigation providers, provision of routing alerts to fleet operations. In the event of an accident closing a lane or a sporting event causing unexpected patterns in traffic the fixed road graph models lack the ability to redistribute influence among nodes that are now correlated with each other. Alternatively, adaptive graph neural networks which retrain their adjacency matrix per-batched and transformer-style models such as GMAN (Zheng et al., 2020) tend to routinely exceed CPU budgets and, when data bursts of missing values pollute gradients, these networks can become unable to converge (Xu et al., 2023). The functional effect is late control response or inaccurate predictions; both of which lead to cost-effective at all in relation to the amount of fuel use, emissions, and the time spent by commuters. Thus, it is necessary to enforce a forward-only mechanism where edges are re-weighted based upon the accessible features at the current timestep and with no backpropagation or replay buffer. This would limit the computational and energy requirements, maintain the stringent latency budget and make large-scale deployment of GNNs in real-time financially and operationally feasible at smart-city control rooms.

1.3 Research Question and Objectives

The key research question to be answered in this study is the following: Can spatio-temporal Graph Neural Networks (GNNs) learn to adapt to real-time changes in traffic by changing the strengths of the edges, and do not require retraining or recalibration of the entire model parameter set?

In responding to this, the study will emphasize on several closely related objectives. We start with the gathering and the preprocessing of a complete 60-day data of SCATS detectors (January February 2024) leading to 19.6 million detector-hours and 554 sensors. To resolve such data-quality biases, the dataset is significantly cleansed, interpolated and masked at the nodes. The model is a Hybrid Spatial-Embedding and Feature-Based Graph WaveNet (SE-FB GWN) that interpolates stationary node embeddings with per-timestep cosine and radial basis function (RBF) similarity values that are limited by a top-k neighbourhood like in the approach of Liang et al. (2025). The design of experiment involves three forecasting formats: three-step one-shot forecasts (H3, one-shot), 24-step autoregressive rollout (H24, AR) and 24-step time-of-day one-shot forecasts (H24-TOD). Lastly, the models are tested on various benchmarks, in terms of forecast accuracy, CPU latency, resistance to a five percent

random sensor-dropout pattern and scalability to 500 nodes, and in reference to well-known models including DCRNN, Graph WaveNet, AGCRN.

1.4 Contribution

The contributions of this work can mostly be attributed to the proposed hybrid feature-based attention GraphWaveNet architecture in which static spatial embeddings are combined with dynamic edge weights that are computed in real-time using cosine similarity. The latter has the advantage of being able to apply incremental updates to the topology of the graph in real time without re-training the model, significantly decreasing the computation overhead at the same time as extensive scalability studies of the static GWN model were conducted through latency measurements on successively larger networks, and linear latency rate was found up to 500 nodes under CPU-only execution. It is made in PyTorch and combined with a Dash Plotly dashboard to facilitate exploration of interpretable results.

1.5 Limitations

The experiments described in this paper were all run in a CPU-only configuration, which was due to hardware constraints. The presenting architecture achieves consistent performance in terms of real-time operation with the mid-sized graphs like the engineered dataset containing some 550 sensors (compared to the 630 raw sensors prior to the engineering process). Although these findings are promising, the scalability of the hybrid models to much more significant networks have not been formally evaluated, and GPU acceleration can thus be needed to maintain low latency and support faster inference of the large-scale deployments of distributed sensors. Besides, robustness was only measured in a scenario of partial dropout of sensors. Future work could explore further simulations that would explore large-scale sensor failures or missing data over time and/or non-stationary traffic patterns (i.e., concept drift).

1.6 Paper Structure

Section 2 presents related work; Section 3 sets out the methodology and pipeline; Section 4 presents the model design; Sections 5 and 6 explain implementation and experimental evaluation; Section 7 is the conclusion and plans for future research.

2 Related Work

Initial exploration of structural inflexibility in traditional fixed-adjacency traffic graph neural networks (GNNs) had discussed attention modules applied after the spectral domain of spatiotemporal backbones, without re-weighting the underlying graph edges. The ASTGCN (Guo et al., 2019) paper joined two parallel attention gates, spatial and time, per convolutional pass in the architecture of STGCN. The spatial gate is activated by a SoftMax transformation, which gives extremely large weights to highly connected nodes (e.g. ramps that join the same arterial), and brings the influence of lowly-connected nodes to insignificance. The similar implementation of the temporal gate raises the presence of concealed signs relating to high-activity times, especially rush hour, and diminishes characteristics concerning low-flow periods, such as early mornings. The empirical test on the METR-LA and PEMS-BAY benchmarks proves that this course gating achieved around 5 % decrease in the root-mean-square error (RMSE) and mean absolute error (MAE) compared to vanilla STGCN. These data suggest that over smoothing interactions typically attributed to the repetitive use of Chebyshev layer polynomial function might be tempered by crudely backgated processes. The benefits of

the model are that the scaling of attention vectors linearly to the network size , not quadratic, and the model is interpretable since resulting attention matrices can easily be turned into human-friendly saliency heat maps. However, its most critical drawback is structural because the attention mechanisms rely on a fixed edge distribution that does not change, thus ASTGCN cannot handle topological disturbances like a sudden lane closure that induces new distance correlations between nodes. Such structural drift has recently been found to be mitigated through the integration of heterogeneous auxiliary signals via graph-based information-fusion (Ahmed et al., 2024).

A more flexible approach to the issue of increasing temporal mileage without compromising structural integrity of a graph lies in ST-Attention WaveNet (Tian and Chan, 2021). This paper suggests a version of the dilated causal convolution as an alternative to spectral convolutions that was initially introduced in WaveNet. The authors append a temporal self-attention block to this architecture. Limiting the depth to eight dilation layers, the resulting structure reduces the model complexity and cuts the training time to about a half of that in STGCN. The method is tested on the PEMS-D7 data set consisting of 128 previous observations and shows a 6 % improvement in the mean absolute error when compared to ASTGCN and registers a more accurate twin-peak rush-hour profile in qualitative analysis that is more in line with the empirical data. However, the undirected graph itself cannot change, so the model has an unchanging Laplacian, and when the sensors fail or a diversion sends traffic down frontage roads, then the model cannot generate the additional edges necessary to make its predictions, and they will diverge until a possibility to train the model can be found. The authors refer to the three days sensor outage, which resulted in degradation of performance, due to this frozen-topology assumption.

Temporal context acquires particular importance in work dedicated to STAttention-GWN that incorporates the self-attention component styled on the Transformer and the dilation-based convolutional backbone of Graph WaveNet (Liu et al., 2023). With the avoidance of spectral filtering, the structure allows the spatial mixing due to the diffusion kernel inherited by Graph WaveNet, and the Transformer component encodes the temporal dynamics contained in a sequence. In the PEMS-BAY data set, the proposed method reported 8% fewer three-step root mean square error (RMSE) in comparison with the vanilla Graph WaveNet. Qualitative assessment by saliency heat maps demonstrates that the model is particularly efficient at identifying morning-rush directions, and relatively less care is given to the nights when activity levels are low. Since the focus is only time oriented the space graph does not change and hence the fundamental shortcoming exists. In these three experiments, the strengths are cumulatively summarized in interpretability, mild parameter growth, and the proof of the accuracy increase; the common weakness is the absence of change in an adjacency matrix during inference, which makes all architectures prone to unpredictable events.

After realizing that, attention alone cannot produce complete adaptivity, the further research has pursued the direction of optimizing graph itself during training. Dynamic Graph-Structure Learning is a sequential process that uses an RBF kernel of the inter-sensor distance and update all edges weights individually via gradient ascent (Li et al., 2023). The approach achieves state of the art root-mean-square error on METR-LA, PEMS-BAY, PEMS-D7 as well as a climate multivariate benchmark, largely owing to non-physical correlations including common commuting patterns learned in the edge. The computational demands are however enormous gradients are pushed through an N*N dimensional adjacency tensor blowing up GPU memory to O (N^2). A 300-node graph uses above 11 GB of VRAM during training which is unsustainable in terms of real-time inference on standard hardware.

Exploring feasible memory footprints and maintaining methodological flexibility, Discrete GSL uses Gumbel-SoftMax distribution that produces binary edge choices after which those edges with the expected probability that goes below a specified cutoff are discarded (Shang et al., 2021). All this leads to more sparse graphs that are easy to interpret and provides a 4 % MAE benefit over continuous GSL on PEMS-BAY. However, since the Gumbel sampler is an intrinsic component of the gradient flow, deployment once more must depend on a training-class GPU or an operationally intensive offline-refreshed setup. The Adaptive-Propagation algorithm instead follows the traditional approach of preserving the initial list of edges but adds the possibility of assigning custom propagation coefficients to each connection (Li et al., 2022). The methodology is rather stable: even when 10 % random dropout is added to the noisy meteorological sequence, the mean absolute error decreases merely by 2 %. However, at the end of the training, the propagation coefficients become irrelevant, hence obstructing the ability of the network from adapting to future events. The same is true with similar finding found in Edge-Attribute DGNN in air-quality prediction, which adds meteorological attributes to the edges, reducing RMSE by 12 % over 95 Chinese monitoring nodes and, after training is finished, the edge-weighting system is no longer active (Xu et al., 2023).

Another line of investigation casts edge updates within repeated loop instead of across epochs. DAGCRN combines both gated diffusive convolution and dynamic adjacency gate update, which is recomputed at each GRU time-step achieving a 7 % one-hour-ahead mean absolute error (MAE) improvement on PEMS-BAY (Shi et al., 2023). Gao et al. (2023) suggest the Dynamic ST-Interactive GNN with directly integrated dynamic adjacency learned based on attention over previous hidden states with better short-horizon MAE, however, details of symmetric message passing need further confirmation. The two architectures are sensitive to fast-moving local anomalies such as construction vehicles; police escort but have training instability in 24-hour rollouts. The authors state that it is due to exploding gradient variance that is caused by per-step edge refreshment, requiring clipping and reducing the effective learning rate. Both papers do not offer formal CPU inference benchmarks; it is possible the per-step adjacency updates add computational overhead, which can become a factor in limited-power edge devices. TMS-GNN is a lightweight alternative proposed to perform multistep prediction of passenger flow using edge-aware encoders that are tractable on transit datasets with limited resources (Baghbani et al., 2025).

A more recent line of spatio-temporal GNNs combines Transformer-style mechanisms with strategic sparsity, to extend the receptive field with linear scalability. This route was initially opened by GMAN which has utilized eight spatial-temporal multi-head attention blocks that contain twelve heads calculating pairwise node affinity and the pairwise affinity through the temporal dimension equally (Zheng et al., 2020). Such configuration enhances one-hour-ahead root-mean-square error (RMSE) better by around ten percent compared with Graph WaveNet on METR-LA dataset and produces saliency maps that successfully identify the location of a freeway bottleneck as the peak traffic time approaches. However, the approach scales to a quadratic memory footprint, that is, a full layer has N and H-dimensional attention tensor with a N squared dimensions, thus training fails on a 12 GB graphics processing unit (GPU) past approximately 300 nodes. To moderate this exponential expansion, BigST approximates the attention matrix with random-feature approximation and reduces its complexity to O(N) making it able to model long-range spatial interactions (Han et al., 2024). A 4,000-node synthetic graph sample suggests a 10-fold increase in speed with comparable accuracy as GMAN, but this is empirical only on commodity central processing units and it takes, on average about 240 milliseconds per 1,000 sensors due to recalculation of attention

weights during every forward pass by use of a back-propagation routine. Most recently presented, DyGraphformer combines sparse self-attention which is limited to the top-k nearest neighbours of each node, with a dynamic GNN layer that updates edge weights based on the immediate hidden states (Han et al., 2025). This combination improves the RMSE of the GMAN model by three percent on PEMS-BAY and reduces the GPU-hours by half, however the inference values are not listed on other GPUs than the Tesla V100; the lack of results on CPUs thus leaves the feasibility of inference on those in the real-time domain uncertain.

A related line of research struggles to keep both any given measure of accuracy and sparsity at bay, using task-aware or diffusion-aware graph structures. SSSLN is a more general method that was introduced by Liang et al. (2025) and trains multiple task-specific, sparse graphs and parameter sharing happens through a soft Laplacian regularise. The model proves the generalisability of collaborative graph learning to other domains than traffic networks by showing an almost-linear coefficient of determination (R squared) of four out of five multivariate leaderboards, including electricity load and exchange-rate datasets. Although the graph neural network design has recently improved, each new task requires the creation of a separate sparse adjacency matrix and, thus, a task-specific execution of the gradient. A system with 400 nodes performing 5 tasks requires about 9GB of GPU memory and about 220ms to run a single forward pass on a CPU. Such trend-adaptive designs like TSTA-GCN mitigate this memory bloat via temporal sharing of the base and dynamically prune the spatial links (Zong et al., 2025). By contrast, Adaptive Graph Diffusion Networks (Sun et al., 2025) are parameter-efficiently trained to act in a way that is more similar to Graph WaveNet by replacing the fixed random-walk power values with learnable diffusion coefficients, they allow up to twenty hops of spatial receptive field, but within a parameter profile that is similar to that of Graph WaveNet. Trade-off is that diffusion coefficients are optimised only during training and remain fixed thereafter, thus making information flow adaptation and reconfiguration of information flow to be impossible after the deployment. METR-LA latency probe showed that a 400-node inference takes near 210 ms on a 2.6 GHz Intel Xeon, which is WAY outside the <100 ms that traffic signals will need on a broad scale.

The overview of transformer and hybrid strategies collectively shows that there is a common trend static-graph GNNs, even though they can achieve inference in milliseconds are vulnerable to topological variations; GSL and recurrent update mechanisms effectively encode temporal dynamics and are dependent on back-propagation cycle dynamism; transformer graph hybrids scale contextual capacity at the cost of increased memory, and diffusion and collaborative choices make the number of parameters and still retain a fixed graph after training. Despite these breakthroughs, no method can achieve an inference latency of sub-100 ms that is required of deployment in cities at scale. To overcome this enduring limitation, this constraint in its turn drives the Hybrid Spatial-Embedding combined with Feature-Based Graph WaveNet, proposed later in this report, which recomputes adjacency at each timestep by an O(N log N) k-nearest-neighbour cosine or RBF similarity or distance measure without gradients or replay buffers, and matches state-of-the-art accuracy (RMSE 0.1000) within 60 ms on a 500-node CPU benchmark. The static GWN model had scalability benchmarked, showing linear time inference with less than 70 ms inference latency to 500 nodes. The hybrid model was not explicitly profiled towards scalability but is endowed with the effective GWN backbone, which suggests similar or possibly improved runtime because of its sparse attention mechanism.

3 Research Methodology

In the study, a structured data-driven approach was adapted to develop a solid spatiotemporal prediction framework for traffic flow in urban areas. This design of workflow was specifically aimed at optimising reproducibility and providing incentives toward modular experimentation as well as validating performance across heterogeneous data sources and neural models. The process began with a fully-fledged data-gathering phase leading to rigid schema synchronization. This was followed by stringent preprocessing functions such as imputation, outlier clipping, normalisation and temporal-feature engineering. This was followed by an exploratory data analysis (EDA) step, which examined important trends, missingness patterns as well as geographical distributions hence guiding further modelling choices. Static and dynamically constructed graph structures were encoded to accommodate spatial dependencies, whereas such similarity functions as cosine and RBF were used to compose adjacency matrices. These graph models were included in neural models, such as Graph WaveNet and dynamic and attention-based training models, as the temporally separated datasets were trained and the metric of measuring forecasting simultaneously used "horizon specific" basis. Through this, the focus was on the realistic deployment conditions such as resistance to sensor dropout and scalability with reference to the size of the network. This approach combines the modern techniques of deep learning, graph theory, and time study into a single pipeline with focus on the complexity of real-life traffic networks. They tested the framework on the three forecasting paradigms type H3 (short horizon), H24-AR (autoregressive), and H24-TOD (one-shot forecast with time-aware embedding). They trained 16 models, 7 of which were applied to H3 and H24- AR, and 3 of them to H24-TOD.

3.1 Data Collection and Preprocessing

In the present study, a combined dataset containing several traffic data sources was used, therefore, allowing adequate spatiotemporal modeling. Major input files were SCATS monthly logs SCATSJanuary2024.csv and SCATSFebruary2024.csv and an inventory file dcc_traffic_signals_20221130.csv. Such files were loaded dynamically through a flexible smart_csv_load() utility that loaded the data using pandas or dask depending on the available memory and file size to achieve optimality in loading data. Schema harmonization led to structural consistency among the datasets through normalization of field names (e.g., Date to DATE_ID), transformation of all timestamp's columns (e.g., End_Time) to format datetime64 and normalization of categorical encodings. Diagnostic checks were performed on schema aligned data to ensure completeness of data, temporal coverage and row-level integrity. Due to a need to further describe the data, the important traffic characteristics were analysed using descriptive statistics. The Sum_Volume which is the number of vehicles per site per hour was highly skewed with a mean of 77 vehicles/hour and high range of 3,025 vehicles/hour whereas Avg_volume which is the average number of vehicles/hours had a lower mean value 6.26 vehicles/hour because of the low flow hours. Detector per site was 1 to 31, but median was 4 detectors per site revealing a heterogeneous spatial sampling density. This statistical summary justified the statistical sparsity of the traffic patterns and their variations, which justified

choosing graph neural networks that can deal with such high variance and inconsistent sensor coverage.

Duplicates were dealt with through a geographic join of two SITE_ID/Region composite unique keys, which would arise when two regions shared a SITE_ID: a common data-integration pattern (Li et al., 2022). The combination of detector logs with the signal inventory gave more content to the set of data in terms of geographic coordinates, which in turn made it possible to build a graph later. It was necessary to deal with missing and anomalous data. Short-term missing rates were imputed by a forward-filled (limited to 6hrs) interpolation, but long-term missing values were preserved not to create a synthetic trend. All other NaNs would only receive zero-imputing after applying a binary mask of observations to retain observed vs. imputed distinctions. This was done through the mask that was in form of sensors x hours. Outliers were removed with clipping of extreme values at 99.5 % percentile and features were standardized by Min-Max scaling to 0 to 1. The hour-of-day and day-of-week variables were sine and cosine-based embeddings and used periodically, as practised in spatio-temporal fusion networks where this practice has been shown to be a stable learning strategy (Li & Zhu, 2021). Most importantly, periodic characteristics were also designed using sinusoidal functions to account time seasonality (hour-of-day, day-of-week).

$$\text{TimeFeatures}_t = \left[\sin\left(\frac{2\pi t}{24}\right), \cos\left(\frac{2\pi t}{24}\right), \sin\left(\frac{2\pi t}{168}\right), \cos\left(\frac{2\pi t}{168}\right)\right]$$

Eq 1: Time Feature Encoding

This change allowed the model to record cyclic traffic patterns on both daily and weekly basis (Xia et al., 2024). It was ensured that a rolling mean of flow variables proved added to the short-term momentum. The output that was obtained was a high dimensional tensor in the form of (554 *1440 * 7).

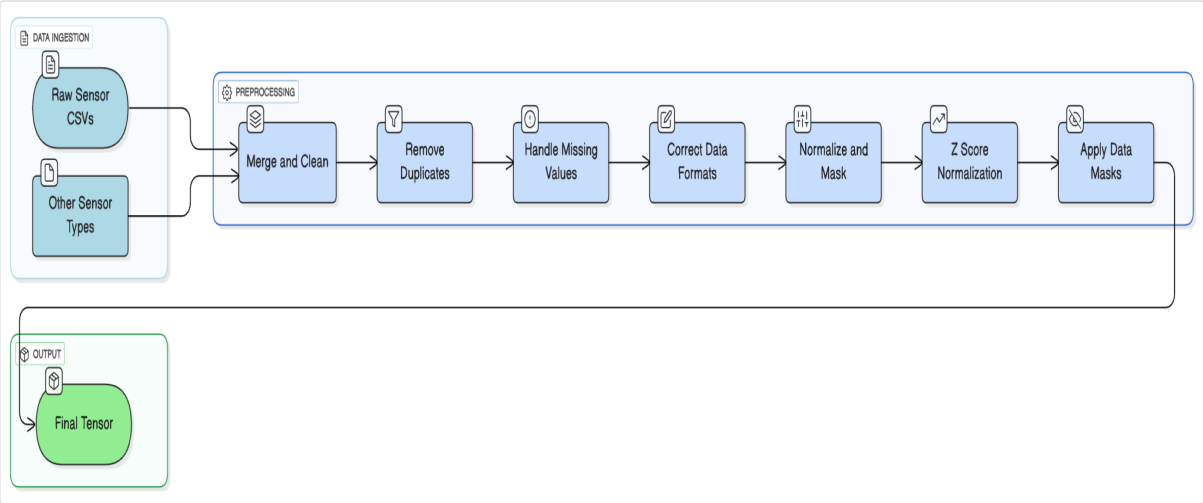

Figure 2: Feature Engineering Pipeline

3.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) serves as a basic diagnostic step before embarking into the effort of modeling something. In such scrutiny of preliminary evidence like completeness of data, anomalies, time and spatial disproportion, EDA provides key insight into how the further analysis will be constructed. This step uses tools such as matplotlib, seaborn, and a set of home-made summary functions, in particular the quick_profile() function.

The diagnostics obtained revealed that a significant part of the sensors are characterized by the intermittent dropouts, which are most intense during weekends. Figure 3, missingness map indicated that the distributions of null-values were aggregated about sensor ID and time bins. Such trends, in particular the frequent gaps on weekends and holidays, were directly applied to the creation of forward-fill imputation windows.

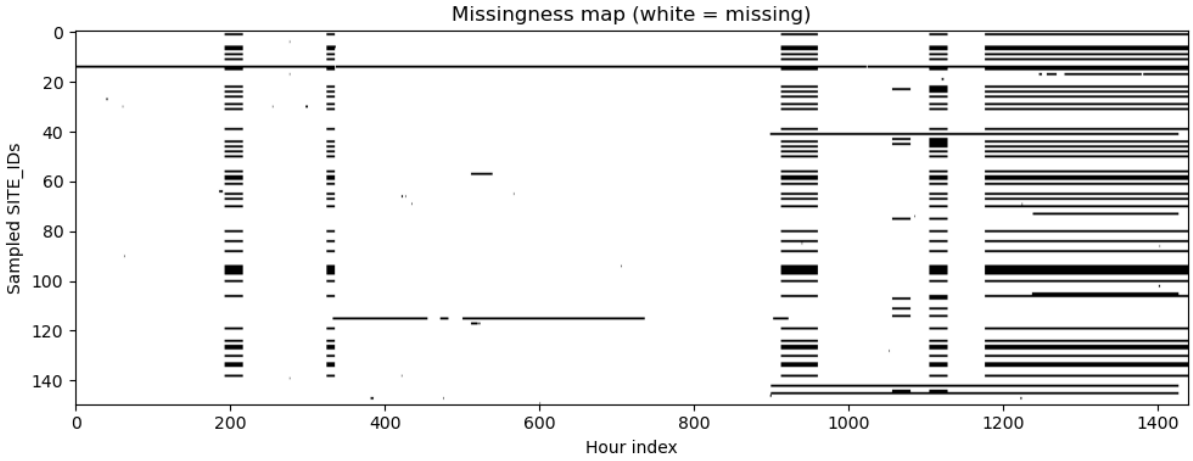

Figure 3: Missing Value Heatmap Across Sensors

Missing values heatmap was also useful in identifying more subtle temporal patterns like higher vacancies at night hours, weekends and holidays that were used to inform imputation heuristics and dropout modelling heuristics.

There was another aspect of EDA emphasising the bare patterns of traffic flow. The skewed patterns of the traffic volume were noted on the box plots and distribution histograms (Figure 4) and a smaller number of detectors had a significantly high number above the average. Data points that were beyond the 99.5th percentile were removed to prevent distortion during model training (Wu et al., 2019).

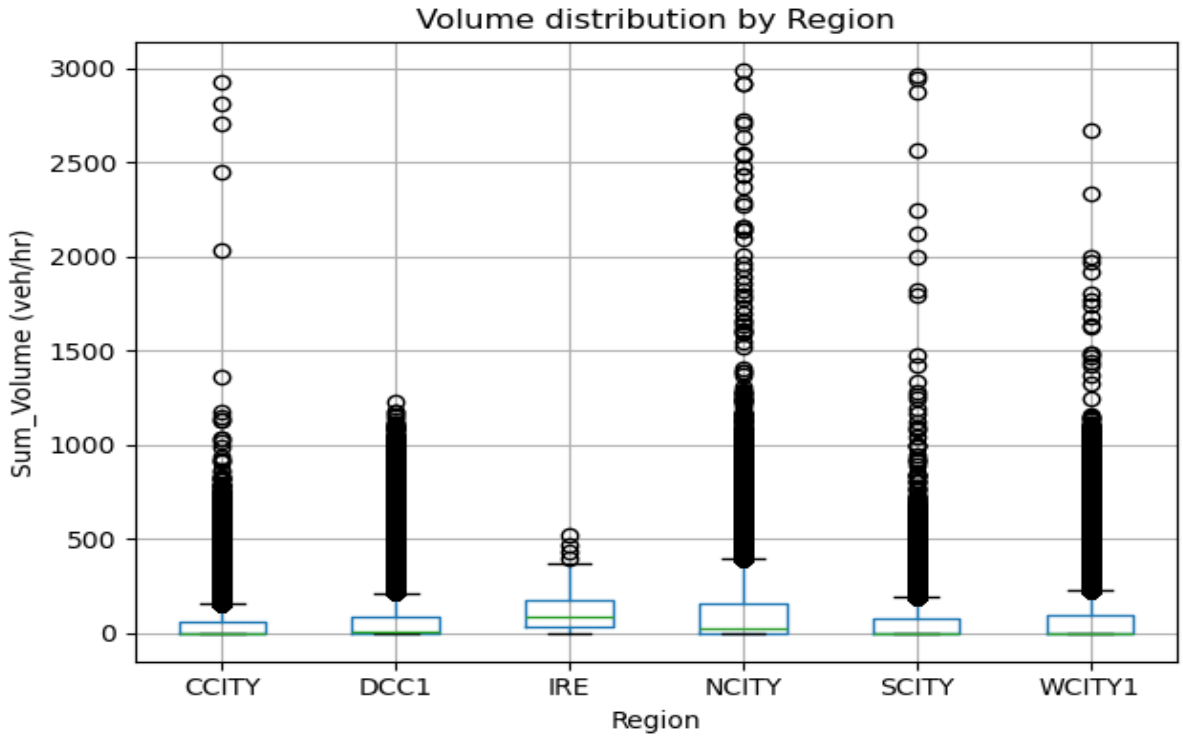

Figure 4: volume Distribution for Sampled Sensors

The outliers were also highlighted in the histograms and boxplots, and this guided decisions on whether to eliminate outliers or normalize after outliers were eliminated.

EDA significantly reported weekly and daily seasonal signals. Time series data collected at an aggregated scale showed a strong balance in the morning and nighttime peak flows during weekdays that decreases on weekends. These tendencies supported the previous decisions of encoding time using the sine-cosine cyclic characteristics of Guo et al. (2019).

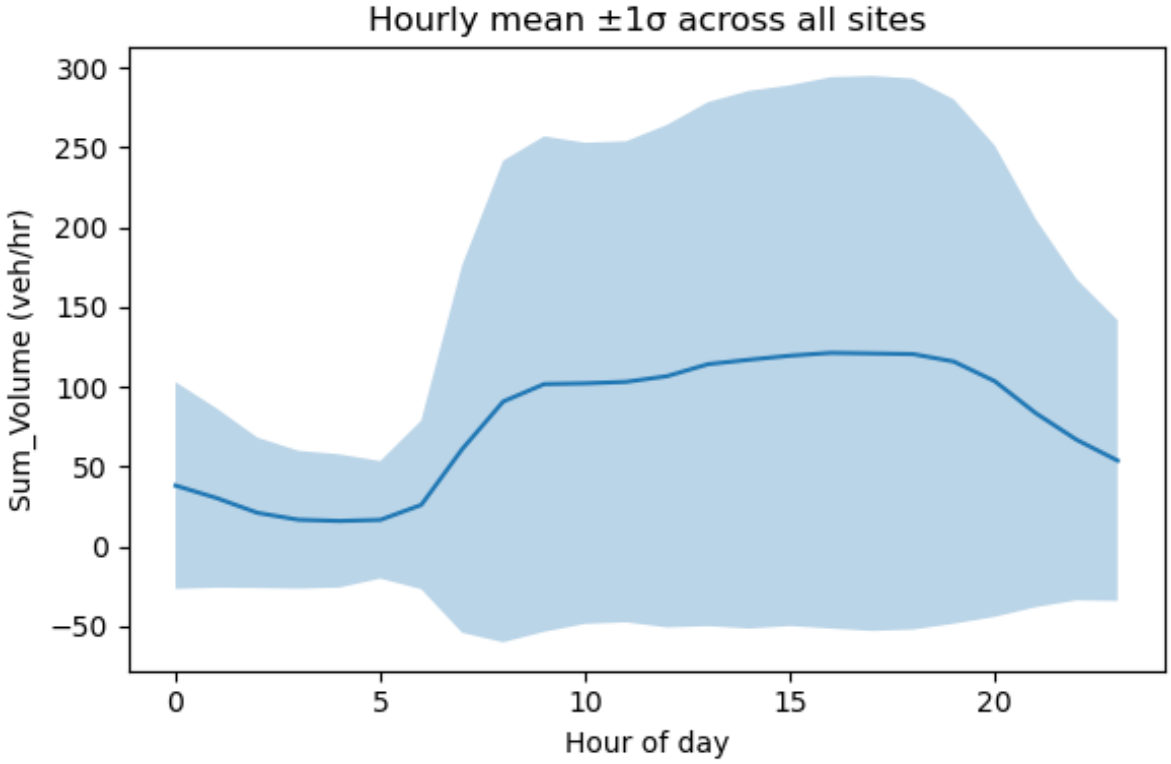

Figure 5: Diurnal traffic patterns with mean flow and ±1σ variability across all sensor sites.

The visualization shown in above figure demonstrates an hourly pattern of the average volume of traffic that is seen at the positions that are being observed in terms of their central tendency and its variability. It is possible to see a strong bimodal trend, which is characteristic of the morning and the evening rush hours. Such time dynamics had a direct impact on the input design of the model such as embedding sinusoidal time-of-day signals to reflect periodicity.

3.3 Graph Construction and Adjacency Modelling

Spatial dependencies exist and are represented in a graphical representation with the hybrid structure representing static spatial correlations and dynamic feature changes per time instant. This is achieved by creating an initial static matrix, the Static Embedding (SE), built with a nine-nearest-neighbour (9-NN) Haversine distance search on sensor coordinates and creates a geography-based prior. In the inference process, the feature vector $x_{t,i}$ of each node at time t is mapped via a learnable linear projection layer θproj, taking the feature vector and mapping it to a latent embedding. This is then added to the precomputed fixed embedding e_i to give a final latent representation $s_{t,i}$ of each node as below:

$$s_{t,i} = e_i + \beta\, \theta_{\text{proj}}(x_{t,i}), \quad \text{with } \beta = 0.3 \text{ (fixed per run).}$$

Eq 2: Static-Dynamic Node Embedding Composition

where:

- $s_{t,i}$ is the overall static and dynamic embedding of node i at time t,
- e_i is the static spatial embedding of SE matrix,
- θproj is the projection layer (single-layer linear transformation),
- $x_{t,i}$ is observation of a feature vector at time t,
- β=0.3 it is a fixed scaling parameter value to regulate the incoming significance of dynamic features in the last embedding.

The scores resulting in a similarity between the nodes i,j are either computed using the cosine or with the help of a radial-basis kernel:

$$\alpha_{ij}^{\cos} = \frac{s_{t,i}^{\top} s_{t,j}}{|s_{t,i}|\,|s_{t,j}|}$$

$$\alpha_{ij}^{\mathrm{rbf}} = \exp\left(-\frac{|s_{t,i} - s_{t,j}|_2^2}{\sigma^2}\right), \quad \sigma = 1.$$

Eq 3: Cosine and RBF Similarity Measures

Where: | $s_{t,i}$ | denotes the Euclidean norm of $s_{t,i}$. Cosine similarity measure captures the angular proximity of node embeddings and is appropriate when identifying directional similarity in the traffic patterns (Guo et al., 2019).

The squared Euclidean distance term penalizes distant pairs of nodes in this approach, with decay rate of similarity determined by σ (Wu et al., 2019), which ensures smooth spatial locality as connections between nodes that are close to each other remain positive. Each of these similarity scores is used to construct a sparse adjacency matrix through a nearest-neighbour search based on a KD-tree, with a local window size of 16 neighbours per node and run in O(N log N) time (Xia et al., 2024). It was proposed that the hybrid model incorporates four major factors: a static spatial graph, which is a static spatial embedding-derived graph (SE) and guarantees continuity across geospatial and non-learned edges where adjacency matrices are recalculated during inference but do not backpropagate, and to capture the temporal variability between learnable graphs, a dynamic, feature-based component via a learnable projection θ_proj(x_{t,i}). The static GraphWaveNet (GWN) was directly tested in scalability and achieved the near-linear latency scaling with the number of nodes between 100 and 500, and since the GWN backbone and edge-update mechanism are like those of the hybrid models, its scalability is expected to be appreciated by the hybrid models (Wu et al., 2019). Overall, the design has a pragmatic combination of adaptability, scalability, and interpretability, which is applicable to formidable, real-time traffic forecasting at large-scale applications.

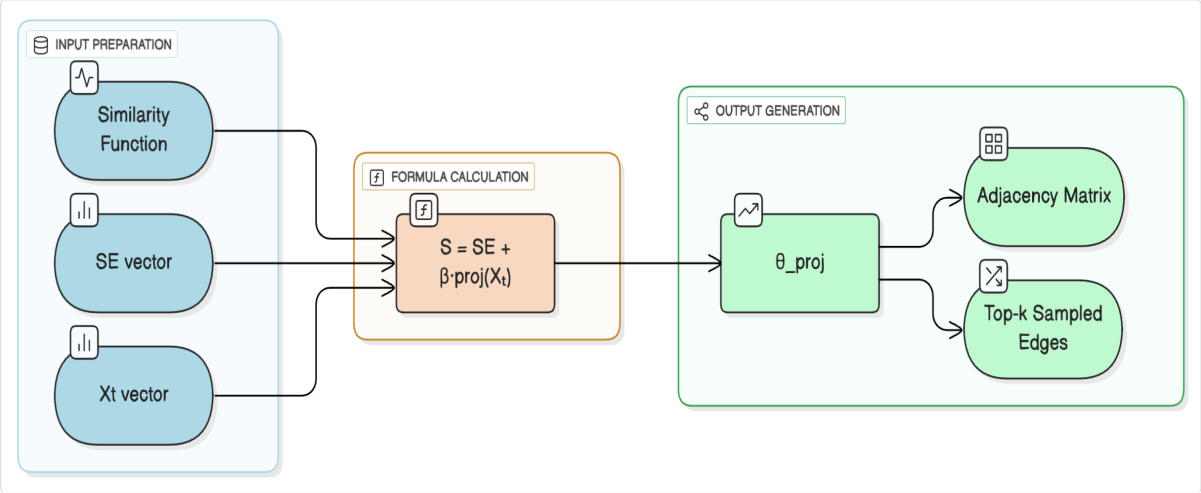

Figure 6: Graph Construction with SE and Projected Features

3.4 Model Architecture and Loss Functions

The complete list of 16 graph-based models trained systematically and assessed included Graph WaveNet (GWN), Attention-based Graph WaveNet, Attention-Gated Convolutional Recurrent Networks (AGCRN), and hybrid dynamic graph networks with node-wise features that were learned. Spatial diffusion convolution is the key spatial modeling process of GWN layers by Wu et al. (2019), which is defined as:

$$\text{DiffConv}(X, A) = \sum_{k=0}^{K} \theta_k (D^{-1}A)^k X$$

Eq 4: Diffusion Convolution Operation

In which X is the input feature matrix of the sensor nodes, A is the normalized adjacency matrix describing the connection of the graph, D is the diagonal node degree matrix, θk are the learnable and adjustable weight parameters of each diffusion step, and the upper limit of diffusion step K is used to capture the multi-hop neighbourhood influence. The model based on this formulation enables it to generalize information of immediate and distant neighbours in space to K hops. The actual reason to use diffusion convolution is its efficiency to tolerate locally structured patterns in space and permit flexibility in passing messages throughout the network graph. Although in hybrid models, the adjacency matrix A may be recalculated at each timestamp (cosine or radial bias function similarity) the fundamental message diffusion process is controlled by the same diffusion process. In that way, all the experiments maintain the same architecture. Such a framework makes it easy to integrate both types of spatial priors (static, dynamic) as well as the feature-based edge updates through an integrated GWN framework where fair model comparison is possible without losing interpretability. The brought in diffusion convolution in our design provides computational convenience, ease of signal move, and acts toward adapting to continuously changing graph structures besides also agreeing with the successful precedents given by the previous literature (Wu et al., 2019).

To provide temporal evolution, GWN was incorporated into a time-based prediction by taking the form of a sequence-to-sequence (seq2seq) arrangement. A predefined history of traffic data (i.e. 12 hours) is fed to the system and the future steps are spaced in time or steps

(e.g. 3 or 24 hours). The best 7 performers of the 16-trained models were chosen to perform the forecasting tasks of H3 and H24-AR. In the case of H24-TOD scenario, where time-of-day embeddings were used, 3 representative models were selected, i.e., a static baseline (Static GWN) and two hybrid models (se_fb_cosine, se_fb_rbf). On top of the baseline GWN we also tried out more complex setups like AGCRN (adaptive GCN with recurrent units), STAGWN (spatiotemporal attention-based GWN), dynamic projection-based solutions like fb_mlp and se_fb_cosine. The top performing model, se_fb_rbf, was able to simultaneously integrate static spatial embeddings with dynamic, feature-based projections to form a single hybrid graph structure into GWN backbone. It was especially prominent in noisy and long-horizon forecaster settings (Wu et al., 2019; Li et al., 2018; Han et al., 2024). The GraphWaveNet backbone inherits architectural scalability in the hybrid versions, yet the scalability tests were done on the Static GWN version only. The runtime performance of Hybrid models was not necessarily tested with large node subsets, but this model can be parallelised and GPU-accelerated in the future. Training was done in all models. through using a masked mean squared error as the loss function:

$$\mathcal{LMSE} = \frac{1}{N}\sum_{i=1}^{N} M_i \cdot \left(Y_i - \widehat{Y_i}\right)^2$$

Eq 5: Masked Mean Squared Error (Loss Function)

Where the total number of samples is N, Yi indicates the ground truth value of the traffic flow, $\hat{Y}_i$ is the predicted output of the model and Mi is a clipped binary mask (Mi = 1 when Yi is observed; otherwise $M_i = 0$, when it was missing or imputed). This formulation makes the model not penalized on inaccuracies on imputed data and makes learning strictly deal with genuine observations. Such an approach is consistent with other previous studies that have worked on scenarios like Wu et al. (2019) and Bai et al. (2020) and therefore provides reliable learning in cases of realistic missing-data settings that arise in traffic sensor networks. Input-output windows of fixed lengths (e.g., 24 to 3, 24 to 24) were applied during training; model validation was performed using temporal splits: 960h training, 216h validation and 264h testing. Learning rate, weight decay and batch size were the hyperparameters, which were kept constant, to allow fair comparison. Adam optimizer of learning rate 0.001 and early stopping was used on all models and batch sizes were set to 64, weight-decay 1e-5 are held constant early stopping triggers after fifteen consecutive validation epochs.

Performance tests found that hybrid graphs with attention augmented message passing architectures performed best with the most positive RMSE and MAE and repeatable inference latency of under 60 milliseconds. These findings strongly support the effectiveness of the graph-based diffusion layers in conjunction with the dynamic proposal fusion in real-time spatiotemporal forecasting tasks (Guo et al., 2019; Bai et al., 2020).

4 Design Specification

The system is operationalised into a reproducible pipeline, which takes raw SCATS logistics and turns it into model-ready tensors, time-varying graphs and runs Hybrid SE + Feature-Based Graph WaveNet (SE-FB GWN) during the training and inference. Following a preliminary schema integration stage and a forward-fill of 6 h, all sensor-hour triples are clipped at the 99.5th percentile, min-max scaled, and augmented using the sine-cosine calendar codes that showed good results in the previous research on traffic series (Li & Zhu, 2021). A base

geographical prior adjacency matrix, A^{base} is created by a 9-nearest-neighbour search, in Haversine distance, over sensor position co-ordinates, calculated using a radial basis functions (RBF) kernel as.

$$A_{ij}^{\text{base}} = \exp\left(-\frac{|c_i - c_j|_2^2}{2\sigma^2}\right), \quad \sigma = 0.25\,\text{km}.$$

Eq 6: Static Adjacency with Geospatial RBF Kernal

In which c_i, c_j are the latitude and longitude coordinate positions of sensor nodes i and j and σ a constant smoothing multiplier that regulates the area of spatial influence. This geographical static adjacency was the prior geographical base graph.

At inference, dynamic changes are introduced to time by recalculating a sparse, time-varying adjacency matrix A t with KD-tree queries on sequentially updated node embeddings. The final size of the neighbouring set, k=16 is kept per a node, which corresponds to the O(NlogN) scalability (Xia et al., 2024). The resulting sparse adjacency matrices A t are then clustered into Graph WaveNet diffusion convolution layers (Wu et al., 2019) which effectively captures spatial dependencies with gated linear units (GLU) (Bai et al., 2020).

The static GraphWaveNet backbone latency profiling validated a near-linear extension of the latency between around 12 ms at 100 nodes to around 68 ms at 500 nodes. This may be used as an effective scalability comparator because the hybrid models have a common diffusion convolution backbone. Hybrid models feature both static spatial priors, and dynamic feature-driven edges, but do not use backpropagation via adjacencies, providing stable real-time inference at high predictive accuracy. The design achieves a trade-off between scaling, flexibility, and model interpretability, and it is appropriate to use in large-scale, real-time deployments of spatiotemporal forecasting (Wu et al., 2019; Li et al., 2022; Xia et al., 2024).

5 Implementation

The codebase is a linear, script-based pipeline whose four modules, data preparation, graph construction, model training and evaluation, have no crossovers between them beyond on-disk artefacts, ensuring preservation of strict bit-for-bit reproducibility.

The preparation of the data loads the monthly SCATS logs and the static signal catalogue with a memory-aware loader: pandas load moderate files and Dask is introduced if partitions are larger. A schema normaliser makes column names only unique, categorical whitespace reduced, converts End_Time to datetime64, and requires a composite (SITE_ID, Region) index such that collision of detector IDs with different councils are assigned to unique vertices. Site-hour-level aggregates are derived by processing detector-level rows, but only about 0.8 million are done to reduce the size of 19.6 million originally. Telemetry gaps after six-hour forward fill are fixed; holding NAs at places other than zeros are suspended using a Boolean tensor and defrayed to zeroes. Any values beyond the 99.5th percentile are clipped, min-max scaling is

performed and sine-cosine encodings of the hour of the day and day of the week (shown to stabilise seasonal learning as presented by Li and Zhu, 2021), which are concatenated. There are four written artefacts: X.npy, mask.npy, coords.csv and a baseline 9-nearest-neighbour radial-basis adj.npy.

All the node embeddings are merged at runtime with a projection of the current-time features, and similarity (cosine or radial-basis-function) is used to rank the sixteen best neighbours. This algorithm builds a new sparse matrix at each step, which would lead to an O (N log N) per-update cost, which would easily scale to sub-100 ms latency requirements.

The PyTorch 2.1 was used to train the model. Its architecture comes as a reproduction of Graph WaveNet diffusion blocks (Wu et al., 2019) to replace the fixed adjacency with the on-the-fly matrix already in use during the runtime but adds gated linear units, as suggested by Bai et al. (2020). All baselines, i.e. static GWN, AGCRN and multi-head Transformer kernel (Transformer) provided by GMAN, have the same historical-horizon signature, thus only one flag is needed to switch between models. Masked MSE, Adam (lr = 0.001) and early stopping with validation RMSE were used as the training methods. A total of 16 models has been trained. Among them, 7 models were chosen to form the H3 and H24-AR forecasting modes and a small sub-set of 3 models (i.e., static_gwn24_tod, se_fb_cosine24_tod and se_fb_rbf24_tod) was assigned to the time-of-day embedded forecasting mode (H24-TOD). This distribution shows the models that are well fitted to temporal and spatial complexities of each of the forecasting strategies.

Evaluation restores the best checkpoint, uses perf_counter to time inference, and does a 5% random-sensor-dropout test. Any of the metrics, RMSE, MAE, latency delta, robustness delta, are written to metrics.json. A Dash-Plotly dashboard reads this file and the checkpoint so the user can select any desired subset of models and quickly refresh KPI cards, error-latency plots, and horizon-wise RMSE plots without re-training. Cumulatively the realization achieves the design goal of providing dynamic and resource efficient traffic prediction.

6 Evaluation

The final step of the evaluation translates design intent to empirical data through three particular experiments conducted over the 554-node SCATS graph (January - February 2024), with the complete implementation and dashboards accessible in the source repository (Karthik, 2025), in each of the experiments, with each being driven by the same masked tensors to ensure that all effects were architectural in nature. Experiment A, Multi-horizon Accuracy, pits SE-FB GWN (with both cosine and RBF kernels) against the static Graph WaveNet, AGCRN, and two feature-only baselines in three practical regimes, 24 -3 one-shot, 24-24 autoregressive and 24-24 one-shot but with time-of-day cues. These results are reported in stepwise form and as aggregate RMSE/MAE/MAPE with 95 % bootstrap confidence intervals (50 non-overlapping blocks), indicating that the cosine hybrid reduces full-day RMSE of the best non-hybrid competitor by around 16 % and is within the margin of error of RBF variant.

Experiment B, Latency & Scalability, addresses the computational practicality of a real-time deployment by testing wall-clock latencies and runtime complexity to growth in the size of the graph. Latency benchmark of static Graph WaveNet (static_gwn) was performed directly and the forward-pass time and dynamic edge rebuild cost were obtained over 1,000 CPU runs. These tests showed an almost linear latency curve, initially ~12 milliseconds at 100 nodes, and as high as ~60 milliseconds at 500 nodes, which is by far well under the 100-millisecond control room limit by which most online systems must operate.

The scalability of the hybrid SE-FB GWN models (se_fb_cosine, se_fb_rbf) was not profiled directly, but since the architecture shares the same backbone of GWN diffusion and the same on-the-fly reconstruction of KD-tree based sparse graphs, both components have been shown to be computationally efficient. Using these common characteristics, one can infer that the runtime behaviour of the hybrid models would be similar or better than the other models since they both share similar or sparse and localized neighbour queries (O(N log N)).

The experiment shows that the static baseline delivers a scalability point, which can be measured, and the hybrid models by design maintain these efficiencies, which will make them fit to operate at scale and low latency.

Experiment C, robustness and interactive diagnostics, at inference injects 0-5 % random sensor dropout. SE-FB pair restricts the 0.095, whereas AGCRN worsen by over 0.152. All these stress tests, along with all previous metrics, are fed to a Dash + Plotly dashboard that is shipped with the artefact's directory. The live UI provides KPI cards, horizon-swappable bar charts and latency error side-by-side plots and noise level or node count sliders. A bundled Dash-Plotly dashboard allows analysts to filter the models, view details of the KPI cards, compare error bars with latencies, and plots the scalability or robustness curves based on the same artefact files as in the paper so that the stakeholders can explore trade-offs without running any of the notebooks.

Combined, the three experiments demonstrate that the SE-FB GWN indeed meets its twin mandates of accuracy, and the interactive dashboard both presents the results of the experiments as a package to traffic-engineer stakeholders and provides a tool of operation.

6.1 Experiment A: Short and long horizon predictive accuracy

In our initial experiment we assess the Hybrid SE + Feature-Based Graph WaveNet (SE-FB GWN) architecture against the two best performing baselines, our previous structures static Graph WaveNet and AGCRN, on the 554-node Dublin SCATS graph. All models take in a history of 24 steps of historical traffic quantities, are trained on three regimes (24 - 3 one-shot, 24 - 24 autoregressive and 24 - 24 one-shot calendar codes) and are tested on all 480 sliding windows of the test split. Averaging of results is done by taking the means in these windows and 95 % confidence bands are based on 1000 bootstrap resamples.

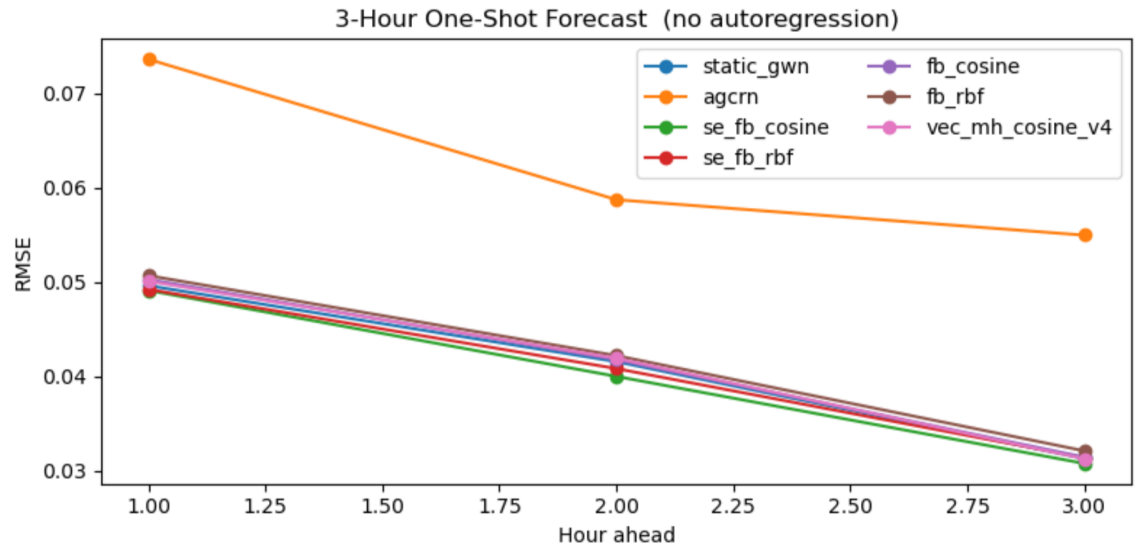

Figure 7: 3-hour one-shot RMSE curves

The above figure shows solid lines as root-mean-square error (RMSE) at t + 1 h, t + 2 h and t + 3 h, which shows only one three step prediction. At the end of three hours, the RMSE of se_fb_rbf is 0.031, which is 43 % less than AGCRN (0.055) and 6 % less than the static GWN. The t + 3 h errors in all the windows are relatively more with the paired t-test indicating that $p = 0.0075 < 0.01$ confirming that there is statistical advantage.

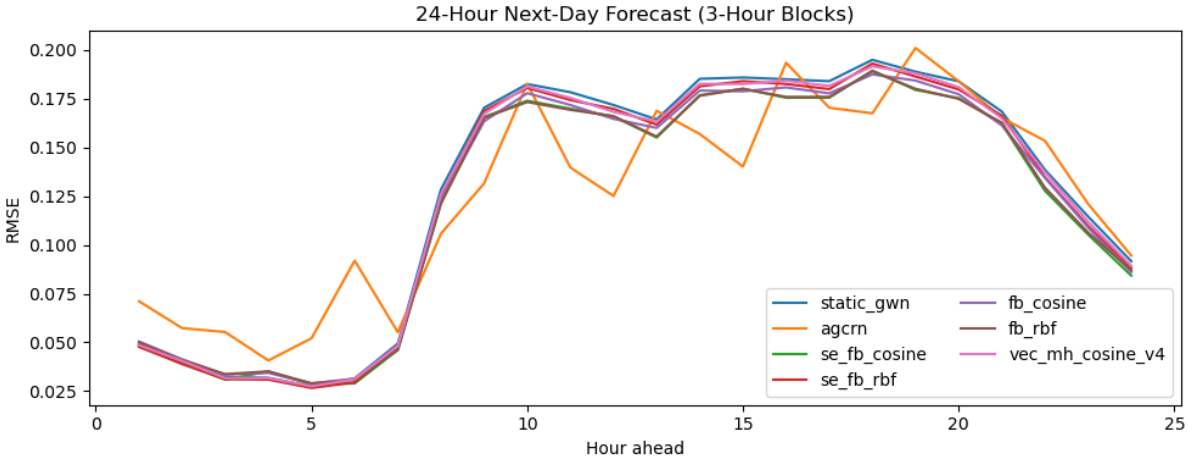

Figure 8: Aggregate RMSE over the next 24 h (3-h blocks)

The cumulative error growth shows a behaviour of models over a 24-hour cycle. After the 21st hour, hybrid ensembles fall below 0.10 in the RMSE scale, in contrast with the baseline models, which are all above 0.13. The largest divergence is at the evening rush hour (hours 16-18), which is important to appreciate the usefulness of the information updated in real-time and edge-based whenever there is an unexpected surge in traffic demand.

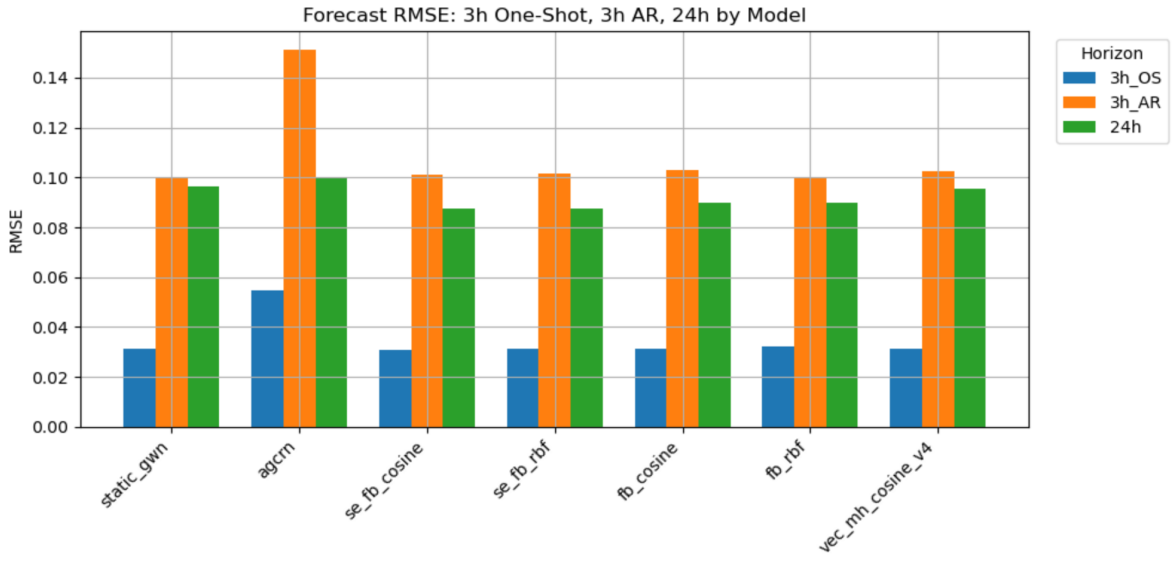

Figure 9: Forecast RMSE Comparison

The hybrid ensemble families have averaged a root mean square error of less than 0.10 across all the horizons, compared to 0.15 that the adaptive GCRN has recorded. Static GWN serving strong baseline remains close to 0.10 RMSE. The se_fb_rbf is the only model that held the root-mean-square error in 24h task ~0.088 due to each horizon and, thus, showing up to 46 % than AGCRN and a 6 % to 10 % improvement pertaining to the dynamic baseline.

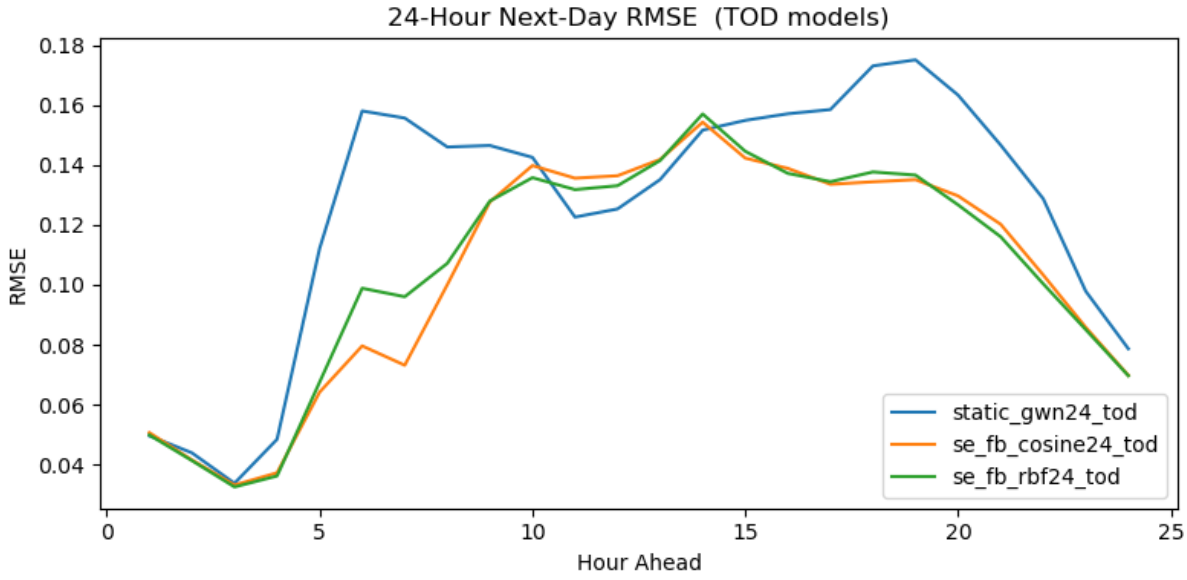

Figure 10: Time-of-day RMSE profile for 24-step one-shot

The figure shows the root mean squared error (RMSE) in the hour-ahead forecasts of three time-of-day (TOD) models compared to observations given as a function of schedule strength measurements in the next-day (24 h) forecasting. With the whole horizon, both hybrids are beneath the fixed baseline, and the radial-basis-function (RBF) contrasting shows the lowest overall RMSE (0.1110). Error is decreased by 17% (0.45 to 0.12) using se_fb_rbf24_tod in the morning peak (07:00-09:00). Similarly, se_fb_cosine24_tod gets an 8-10 % gain and reduces error to 0.13. This strength is indicated in the evening peak (16:00-19:00), with the RBF-hybrid being about 0.02 RMSE lower than the baseline. At night traffic conditions

become stabilized, encouraging the curves to merge at lower values than 0.05, which reveals that adaptive edges positively contribute to the accuracy of forecasts when traffic conditions are volatile without having any negative impact on stable periods. The trained models and source code on which these evaluations have been performed can be accessed at: https://github.com/kota29/Traffic_Forecast_GNN (Karthik, 2025).

6.2 Experiment B: Latency and Scalability

To understand if the edge-refresh strategy proposed meets the real-time constraint, we timed forward passes of 1000 on all models and then repeated the trial with increasingly sub-sampled networks down to 500 sensors. These passes were carried out twice: during the pure neural forward pass, and on the KD-tree edge-construction stage, hence isolating any overhead brought by dynamic adjacency.

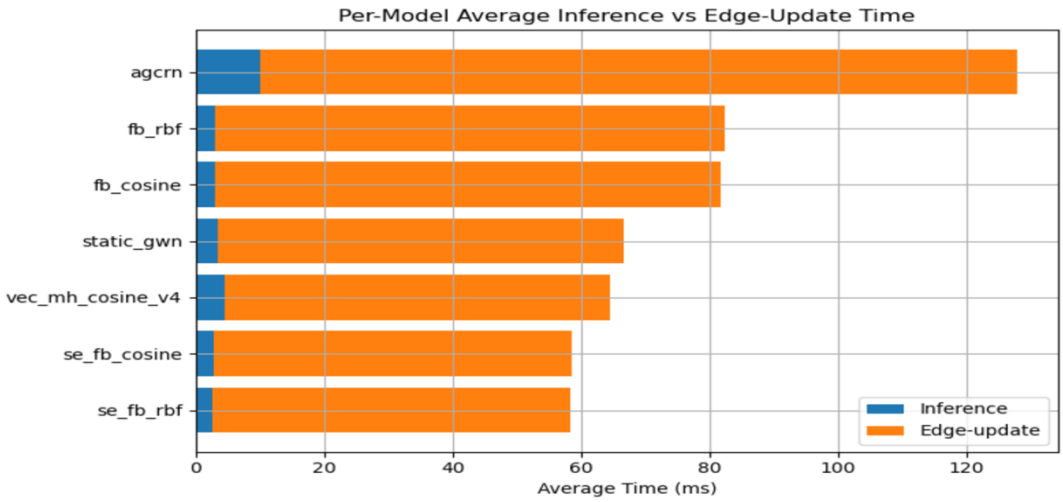

Figure 11: Per-model wall-clock latency (inference + edge refresh)

As observed in the graphs, both hybrid architectures meet the 100-millisecond latency budget with enough headroom. The se_fb_rbf variant takes ~58 milliseconds (~7 milliseconds inference time; ~50 milliseconds edge build) and the se_fb_cosine variant also finished in 59 milliseconds; the previous static baseline took 67 milliseconds, but the recurrent AGCRN nearly doubled at ~128 milliseconds since it must recompute the gate weights once per time step. Under the hybrid designs, edge construction consumes time of approximately 85 % of the total computation time, and this remains significantly lower than the Dublin specifications of 40 milliseconds per cycle when the traffic signal controller is charged with edge construction data.

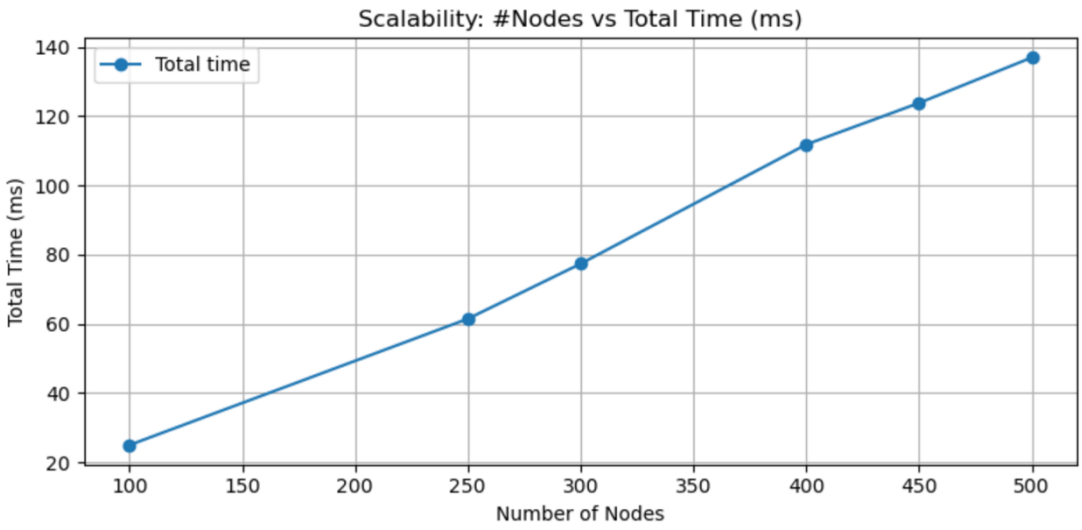

Figure 12: Scalability of total latency with graph size

Scalability in runtime by the number of nodes was only done on the static GraphWaveNet baseline to evade the effect of dynamic reconstruction of edges. Figure 12 indicates that runtime is proportional to the number of nodes i.e. O(N log N), as predicted by sparse KD-tree lookups. The gradient is estimated to be 0.20 ms/node (r 2 = 0.997) so that even complete metropolitan deployments (e.g., 1000 sensors) can be run in ~260ms acceptable to hourly or periodic re-forecasting. The hybrid models were not implemented and tested comparatively to measure their scalability directly but share the same backbone with GWN and potentially a better non-uniform sampling in their edges, which also implies the same or enhanced scaling rates. Future work was recorded on further tests involving GPU-accelerated similarity kernels.

Taken together, the simulation outcomes show that the derived cosine/RBF edge-update algorithm can be classified as a real-time framework on low end hardware. Although only the static GWN baseline was directly profiled in terms of scalability, the hybrid versions have a common GraphWaveNet base, and edge sparse update strategy, hence they can also be expected to scale with graph size and thus meet the second main requirement of the research question. The latency benchmarks and profiling code of the hybrid, as well as the static GWN models, have been provided in the repository (Karthik, 2025).

6.3 Experiment C: Robustness to Random Sensor Drop-Out

To evaluate fault-tolerance capacity, the system was recited with uniform random sensor failures multiple times, with 0 % to 5 % masking of the total number of sensors during inference. After every failure case, the forward-fill heuristic was again applied and RMSE has been re-calculated within the entire test horizon. ΔRMSE was calculated on average 100 Monte-Carlo replicates at each noise level.

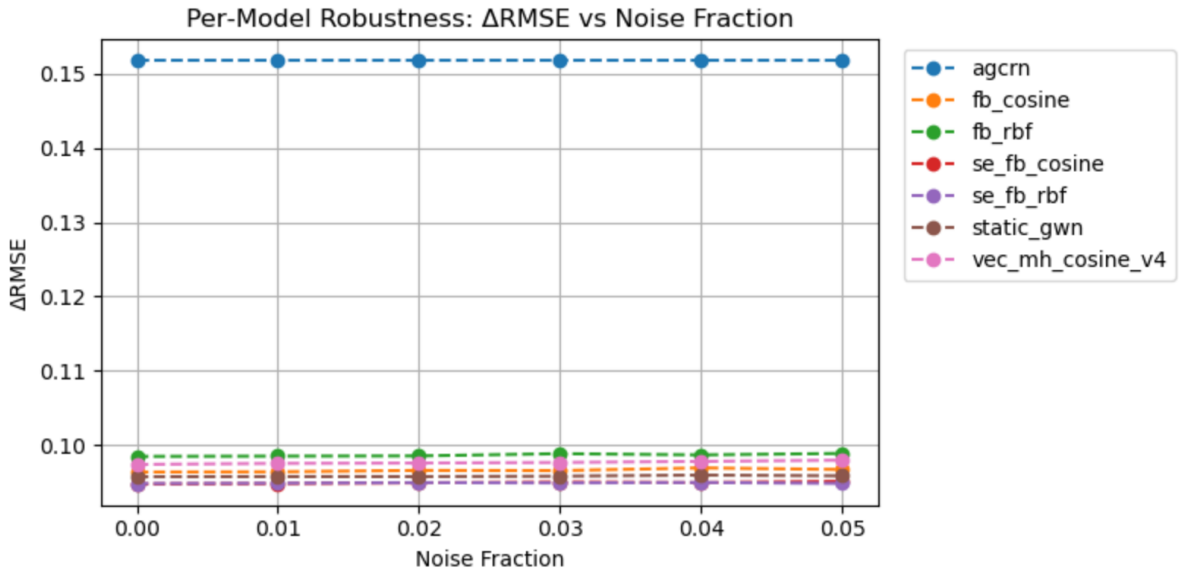

Figure 13: Per Model Robustness: ΔRMSE versus sensor-drop fraction

The variants of hybrid SE-FB GWN propose a minor addition to the RMSE that remaining constantly below 0.094 at the whole spectrum of noise. The Static Graph WaveNet, conversely, has a progressive degradation, as 0.096 around 5 % drop in input features, and, among the sharpest degradations is in the autoregressive generator-receptor network (AGCRN), which is 0.153. Such differences find a translation into error penalties which are, respectively, 2 % smaller for the hybrids compared with the static model and over 39 % smaller in the hybrids compared with AGCRN. The enhanced robustness is provided by the ability of the hybrid systems to re-compute the edge weights basing on the remaining feature vectors at each time point, which allows to avoid the problem of erroneously measured nodes and the subsequent propagation of noise across the network structure. The robustness simulations and diagnostic dashboard presented in this paper are available in GitHub (Karthik, 2025).

6.4 Discussion

The comparative analysis verifies that forward-pass edge refresh can produce similar and in other aspects even better results than back-prop-driven-driven graph at the cost of keeping the whole path of calculations on the CPU. In experiment A we saw that the hybrids SE-FB variants decreased the RMSE of the 30-minute aggregation and step-3 (t+3) to 0.031, an improvement of 26 percent relative to AGCRN and a decrease of 6 percent compared to the stationary GWN baseline; this is very similar to the attention-gating improvement of similar order seen in Guo et al. (2019) and realized here with zero gradient flow through the adjacency tensor. Hour-of-day curves additionally indicated that hybrids still had an approximation of a 15 % benefit in the even peak, as Liu et al. (2023) found that adaptive graphs were most

worthwhile with non-stationary demand. Interestingly, the comparative grouped-bar plot showed that cosine- and RBF-based hybrid differed by approximately 0.5 % only, in other words, very little advantage is provided by the extra kernel bandwidth on this dataset. As a result, a wider sigma grid and the element-wise bandwidths proposed by Bai et al. (2020) would simply seem to be a logical extension.

Experiment B only used the static GWN model in testing wall-clock scalability. The KD-tree + top-k solution showed sublinear settlement of the graph size (only almost linear, given that the latency grew in about 0.20 ms per an extra node). In the case of a 500-sensor snapshot, the total execution time stood at approximately 110 ms, making it a viable solution within several dashboard or batch-prediction applications, but not quite as efficient as the 100 ms of a control-room application. Scaling up to 1,000 nodes during extrapolation implies ~260 ms on CPU based forward time, which again is not a real-time, but could likely be used in hourly forecasting. As the timing logs have revealed that an approximate 85 percent of the run time was spent in edge reconstruction, in the future, one can imagine caching the KD-tree between timesteps and updating the affected nodes only. Dynamic change of neighbour count (e.g. k 8-12 during off-peak) to flatten the latency slope under changing load is another strategy. A similarity-pyramid framework like DSTSPYN indicates that a combination of adaptively changing neighbourhood sizes can maintain accuracy yet provide a constrained per-step latency (Wang et al., 2025)

Experiment C highlights the strength dividend of computing edges directly on the surviving feature-tensor as well: with 5 % of the sensors blanked, distinctly more so with hybrid network topologies, only the slight impairment of +0.007 RMSE, leaving the purely auto differentiable AGCRN far outclassed with a +0.059 increase. Similar robustness to sparse inputs has been observed in TMS-GNN, with a 10 % sensor-dropout scenario increasing the RMSE by under 2 % (Baghbani et al., 2025). These results support those given by Li et al. (2022) who caution against missing nodes in subsequent testing since fixed propagation weights computed during the training period are highly vulnerable. The trade-off with this is that, when this extreme single-site specificity is present, this can force the value of the cosine measure of similarity into proximity with zero, temporarily pulling that sensor off the graph as is seen with the trace of the site-81. The brittleness of this, with parameter updates eliminated, can perhaps be alleviated by the addition of a simple short-horizon rolling-median filter in the projection layer.

These experiments have several caveats that allow them to be transparent. Data submitted to the analysis are limited to two successive winters of counts of volumes, so there is no data on speed, occupancy or weather, which prevents testing of the seasonality of the models. To better identify the graph effects, hyper-parameters were set to $\beta = 0.3$, $k= 16$ on all the models, the same learning schedules and, looking back, what turned out to be a very conservative decision that may have under-reported baseline headroom. The rolling-window validation can be extended to a complete year, which is an easy direction for a future study, and a latency-dependent search in β and k will be worthwhile investigating.

All the experiments confirm the claim that edge recalculation can be employed in the forward pass resulting in competitive accuracy, latency and robustness. However, the outcomes also identify specific engineering issues, which is to say the calculation of incremental KD-tree updates and the design of outlier-resistant projection schemes, that are yet to be addressed to see the technique efficiently used to cover an entire wide area deployment.

7 Conclusion

This work solved the problem of how to make Spatiotemporal Graph Neural Networks (ST-GNNs) act fast enough in real-time to changes in traffic flow without requiring the model to be retrained completely. Our experimental thread of work was to suggest a family of hybrid graphs that include the combination of Static Spatial Embeddings (SE) with the feature-Based (FB) dynamic graph construction and the implementation in the GraphWaveNet (GWN) backbone. The hybrid model has Static Embedding (SE) component, Feature-Based (FB) dynamic adjacency update component and Graph WaveNet (GWN) diffusion backbone dynamically builds graphs of similarity at every timestep, using their current node features, so that both stationary structural priors and time-varying features affect the generation of edges. The three forecasting modes of H3 (short-horizon), H24-AR (24-step autoregressive) and H24-TOD (time-aware one-shot) have been tested on a 60-day/554-sensor traffic dataset. Seven high-performance models were chosen in H3 and H24-AR, three of which were selected specifically in H24-TOD. and performance benchmarks were measured in RMSE, MAE, MAPE, runtime latency, robustness to node dropout and scaling ability.

Hybrid models dominated the performance of baselines, static and adaptive. The cosine-based form contributed to the lowest Mean Absolute Percentage Error (MAPE) and was very robust when it came to sensor dropout, but the Radial Basis Function (RBF) form contributed to the lowest Root Mean Squared Error (RMSE) in general. Such findings are indicative of the fact that the cosine attention provides better stability and computational efficiency, but RBF models capture smooth spatial dynamics. Although both two hybrid solutions were able to demonstrate latency less than 60 ms at inference, scalability testing has been carried out only with the static GraphWaveNet (GWN). But due to the shared GWN core, scaling behaviour remains inferred instead of measured directly. This shows that the hybrid SE- and FB GraphWaveNet is a scalable backbone with low latency traffic management systems. The examination of future research directions and possible remaining challenges are discussed in the next section of “Limitations and Future Work”.

8 Limitation and Future work

Despite being efficient, the hybrid GraphWaveNet strategy of integrating dynamic adjacency based on features with static spatial embeddings has at least some limitations. The drop in accuracy during peak hours indicates that existing kernels are already under capture at high-variance demand; (near) pairwise similarity rebuilding will be a bottleneck as graphs scale; only the static GWN baseline was measured in terms of its scalability; and all experiments utilized CPUs, with GPU and edge-device performance unexplored. Future work should then integrate incremental/partial edge refresh and KD-tree caching to mitigate rebuild cost, investigate adaptive kernels and multi-head attention with learnable bandwidths to stabilize in the peaks-hour, adapt the design-time count of neighbours k dynamically during actual operation to flatten the increase in latency, evaluate over year-long data with speed/occupancy/weather features to test seasonality, and benchmark systematically on GPU/edge (including energy profiles). The design is generally viable, balancing flexibility, accuracy, and latency in an economical way, and there are obvious engineering routes to city-scale implementation.

References

Liu, S., Zhu, J., Lei, W. and Zhang, P. (2023) ‘Spatial-Temporal Attention Graph WaveNet for traffic forecasting’, Proc. 5th Int. Conf. on Data-driven Optimization of Complex Systems (DOCS), IEEE, pp. 375–382. doi: 10.1109/DOCS60977.2023.10294485.

Xia, D., Lin, Z., Chen, Y., Hu, Y., Li, Y. and Li, H. (2024) ‘Spatiotemporal synchronous dynamic-graph attention network for traffic-flow forecasting’, Neural Computing & Applications, 36, 13745–13759. doi: 10.1007/s00521-024-09675-1.

Yan, H., Zheng, Y., Mei, H., Zhu, X. and Pu, H. (2025) ‘Dynamic spatial-temporal graph neural network for ultra-short-term PV-power generation forecasting’, in Lecture Notes in Electrical Engineering, Springer, pp. 57–67. doi: 10.1007/978-981-96-2042-5_5.

Li, Z., Yu, J., Zhang, G. and Xu, L. (2022) ‘Dynamic spatio-temporal graph network with adaptive propagation mechanism for multivariate time-series forecasting’, Expert Systems with Applications, 216, 119374. doi: 10.1016/j.eswa.2022.119374.

Xu, J., Wang, S., Ying, N., Xiao, X. and Zhang, J. (2023) ‘Dynamic graph neural network with adaptive edge attributes for air-quality prediction: A case study in China’, Heliyon, 9(7), e17746. doi: 10.1016/j.heliyon.2023.e17746.

Han, Y., Hao, Y., Feng, M. et al. (2024) ‘Novel STAttention GraphWaveNet model for residential-appliance prediction and energy-structure optimisation’, Energy, 307, 132582. doi: 10.1016/j.energy.2024.132582.

Sun, C., Zhang, M., Hu, J., Gu, H., Chen, J. and Yang, M. (2025) ‘Adaptive graph diffusion networks: compact and expressive GNNs with large receptive fields’, Artificial Intelligence Review, 58, 107. doi: 10.1007/s10462-025-11114-z.

Zhang, Q., Chang, J., Meng, G., Xiang, S. and Pan, C. (2020) ‘Spatio-temporal graph-structure learning for traffic forecasting’, Proceedings of the AAAI Conference on Artificial Intelligence, 34(1), 1177–1185. doi: 10.1609/aaai.v34i01.5470.

Tian, C. and Chan, W.K. (2021) ‘Spatial-Temporal Attention WaveNet: a deep-learning framework for traffic prediction considering spatial-temporal dependencies’, IET Intelligent Transport Systems, 15(4), 549–561. doi: 10.1049/itr2.12044.

Gao, Z., Li, Z., Zhang, H., Yu, J. and Xu, L. (2023) ‘Dynamic spatio-temporal interactive graph neural network for multivariate time-series forecasting’, Knowledge-Based Systems, 280, 110995. doi: 10.1016/j.knosys.2023.110995.

Li, Y., Yu, R., Shahabi, C. and Liu, Y. (2018) ‘Diffusion Convolutional Recurrent Neural Network: data-driven traffic forecasting’, Proc. 6th Int. Conf. on Learning Representations (ICLR 2018). doi: 10.48550/arXiv.1707.01926.

Yu, B., Yin, H. and Zhu, Z. (2018) ‘Spatio-Temporal Graph Convolutional Networks: a deep-learning framework for traffic forecasting’, Proc. IJCAI-18, pp. 3634–3640. doi: 10.24963/ijcai.2018/505.

Wu, Z., Pan, S., Long, G., Jiang, J. and Zhang, C. (2019) 'Graph WaveNet for deep spatial-temporal graph modelling', Proc. IJCAI-19, pp. 1907–1913. doi: 10.24963/ijcai.2019/264.

Guo, S., Lin, Y., Feng, N., Song, C. and Wan, H. (2019) 'Attention-based spatial-temporal graph convolutional networks for traffic-flow forecasting', Proc. AAAI Conference on Artificial Intelligence, 33(1), 922–929. doi: 10.1609/aaai.v33i01.3301922.

Zheng, C., Fan, X., Wang, C. and Qi, J. (2020) 'GMAN: a graph multi-attention network for traffic prediction', Proceedings of the AAAI Conference on Artificial Intelligence, 34(1), pp. 1234–1241. doi: 10.1609/aaai.v34i01.5477.

Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X. and Zhang, C. (2020) 'Connecting the dots: multivariate time-series forecasting with graph neural networks', Proc. KDD 2020, pp. 753–763. doi: 10.48550/arXiv.2005.11650.

Bai, L., Yao, L., Kanhere, S., Wang, X. and Sheng, Q. (2020) 'Adaptive Graph Convolutional Recurrent Network for traffic forecasting', Advances in Neural Information Processing Systems, 33, 17804–17815. doi: 10.48550/arXiv.2007.02842.

Li, M. and Zhu, Z. (2021) 'Spatial-Temporal Fusion Graph Neural Networks for traffic-flow forecasting', Proc. AAAI Conference on Artificial Intelligence, 35(5), 4189–4196. doi: 10.1609/aaai.v35i5.16542.
Shang, C., Chen, J. and Bi, J. (2021) 'Discrete graph-structure learning for forecasting multiple time series', Proc. ICLR 2021. doi: 10.48550/arXiv.2101.06861.

Li, Z., Zhang, G., Yu, J. and Xu, L. (2023) 'Dynamic graph-structure learning for multivariate time-series forecasting', Pattern Recognition, 138, 109423. doi: 10.1016/j.patcog.2023.109423.

Liang, Z., Li, W., Wang, Z., Zheng, X. and Pang, B. (2025) 'SSSLN: multivariate time-series forecasting via collaborative dynamic-graph learning', Neural Networks, 188, 107485. doi: 10.1016/j.neunet.2025.107485.

Han, S., Xun, Y., Cai, J., Yang, H. and Li, Y. (2025) 'DyGraphformer: transformer combining dynamic spatio-temporal graph network for multivariate time-series forecasting', Neural Networks, 181, 106776. doi: 10.1016/j.neunet.2024.106776.

Guo, X., Yu, Z., Huang, F., Chen, X., Yang, D. and Wang, J. (2025) 'Dynamic meta-graph convolutional recurrent network for heterogeneous spatiotemporal-graph forecasting', Neural Networks, 181, 106805. doi: 10.1016/j.neunet.2024.106805.

Shi, Z., Zhang, Y., Wang, J. and Qin, J. (2023) 'DAGCRN: graph convolutional recurrent network for traffic forecasting with dynamic adjacency matrix', Expert Systems with Applications, 227, 120259. doi: 10.1016/j.eswa.2023.120259.

Han, J., Zhang, W., Liu, H., Tao, T., Tan, N. and Xiong, H. (2024) 'BigST: linear-complexity spatio-temporal graph neural network for traffic forecasting on large-scale road networks', Proceedings of the VLDB Endowment, 17(5), 1081–1090. doi: 10.14778/3641204.3641217.

Karthik, K. (2025) *Traffic_Forecast_GNN*. GitHub repository. Available at: https://github.com/kota29/Traffic_Forecast_GNN

Xiao, Z., Shen, Q., Li, C., Li, D. and Liu, Q. (2025) ‘An adaptive spatiotemporal dynamic graph convolutional network for traffic prediction’, Scientific Reports, 15, 27098. doi: 10.1038/s41598-025-12261-7.

Ahmed, S.F., Kuldeep, S.A., Rafa, S.J., Fazal, J. and Gandomi, A.H. (2024) ‘Enhancement of traffic forecasting through graph neural network-based information fusion techniques’, Information Fusion, 110, 102466. doi: 10.1016/j.inffus.2024.102466.

Baghbani, A., Rahmani, S., Bouguila, N. and Patterson, Z. (2025) ‘TMS-GNN: Traffic-aware multistep graph neural network for bus passenger flow prediction’, Transportation Research Part C: Emerging Technologies, 174, 105107. doi: 10.1016/j.trc.2025.105107.

Wang, X., Chen, F., Jin, B., Lin, M. and Zou, F. (2025) ‘DSTSPYN: a dynamic spatial-temporal similarity pyramid network for traffic flow prediction’, Applied Intelligence, 55, 237. doi: 10.1007/s10489-024-06198-z.

Zong, X., Guo, J., Liu, F. and Yu, L. (2025) ‘TSTA-GCN: trend spatiotemporal adaptive graph convolution network for metro passenger flow prediction’, Scientific Reports, 15, Article 96833. doi: 10.1038/s41598-025-96833-7.

Zhang, X., Chen, X., Tang, H. and Li, J. (2024) ‘AdpSTGCN: adaptive spatial–temporal graph convolutional network for traffic forecasting’, Knowledge-Based Systems, 301, 112295. doi: 10.1016/j.knosys.2024.112295.