Precision Journal of Information Technology & Scientific Innovation
Volume 1, **2026**, **pp.** 1-19, **Article ID:** PD2601202006
DOI: inprocess

Research Article

Graph-Based Spatio-Temporal Modelling for Complex Network Behaviour Prediction Using Attention-Driven Architectures

Bharat B Pandey[1, *] ✉

[1]*Independent Researcher in Secure Mission-Critical Enterprise Systems, USA*

Article History

Received: 08 April, 2026 Revised: 23 June, 2026 Accepted: 27 June, 2026 Published: 30 June, 2026

Abstract:

Introduction: Accurate traffic forecasting is a central requirement for intelligent transportation systems because urban road networks exhibit complex spatial interactions, non-stationary temporal patterns and dynamic congestion propagation. Conventional recurrent models can represent temporal continuity but usually ignore explicit road-network structure, while fixed-topology graph models may not fully capture changing sensor relationships.

Study Design: This study presents a METR-LA evaluation of a static-adaptive graph attention Transformer architecture for traffic-speed prediction. The framework combines static-adaptive graph fusion, dual-branch spatial encoding, Transformer-based temporal modelling and sparsity-regularised graph learning. The static graph preserves stable sensor relationships, whereas the adaptive graph learns hidden data-driven dependencies through trainable node embeddings. Local graph aggregation captures neighbourhood-level traffic propagation, while global self-attention models non-local sensor interactions across the wider network. The temporal Transformer encoder then models the sequence-level traffic dynamics through multi-head self-attention, enabling multi-step forecasting over short- and medium-term horizons.

Methodology: The methodology follows a chronological METR-LA benchmarking protocol using training-only normalisation, sliding-window sample generation, fixed random seeds, saved-checkpoint evaluation and horizon-aware reporting at 15-, 30- and 60-minute horizons.

Analysis: Comparative analysis against persistence, recurrent, temporal convolutional and graph-based baselines is reported using repeated-run summaries, ablation analysis and cautious seed-level statistical testing.

Conclusion: The study presents a technically integrated and reproducibility-oriented framework for adaptive, graph-aware traffic forecasting in intelligent urban mobility systems.

Keywords: Spatio-temporal graph neural networks, adaptive adjacency, METR-LA, traffic speed forecasting, multi-scale attention, transformer encoder, graph sparsity, intelligent transportation systems.

1. INTRODUCTION

Urban mobility systems produce large-scale sensor streams that are spatially dependent and temporally dynamic. A speed change at one freeway sensor may propagate to downstream sensors, while recurring daily patterns, incidents and capacity fluctuations alter the temporal profile of each node [1, 2, 3]. This makes traffic forecasting a graph time-series problem rather than an ordinary univariate prediction task. In this setting, a road network can be represented as G = (V, E, A), where V denotes the sensor nodes, E denotes the edges or relationships between sensors, and A denotes the adjacency matrix used to control spatial message passing [4].

Traditional recurrent and convolutional time-series models can learn temporal trends but often ignore the topology of the

*Address correspondence to this author at Independent Researcher in Secure Mission-Critical Enterprise Systems, USA; E-mail: pande.bharat@gmail.com

Cite as: Pandey B.B. Graph-based spatio-temporal modelling for complex network behaviour prediction using attention-driven architectures. *Precis. J. Inf. Technol. Sci. Innov.* **2026**, *1*, 1-19, Article Id: PD2601202006.

transportation network [5, 6]. Graph neural networks address this gap by allowing each sensor to aggregate information from other sensors through a graph structure. However, the graph used by many early models is fixed and is usually based on road distance, physical connectivity or expert assumptions. Such assumptions are useful but incomplete because real congestion propagation can change with demand, lane disruptions, time of day and non-recurring events [7, 8].

This paper presents an adaptive attention-driven architecture for graph-based traffic forecasting. The design combines four elements: static-adaptive graph fusion, local graph diffusion, global spatial attention and temporal self-attention. The static graph preserves stable sensor relationships, while the adaptive graph captures hidden sensor dependencies that are not directly encoded by a fixed topology. The local branch learns neighbourhood propagation, and the global branch allows non-neighbouring sensors to influence one another when their speed patterns become correlated. The temporal Transformer encoder then processes each node-specific sequence using self-attention rather than sequential recurrence.

The empirical design is based on METR-LA because it is a recognised benchmark in traffic forecasting. METR-LA contains traffic-speed observations from 207 loop-detector sensors in Los Angeles County at five-minute intervals and has been widely used in graph-based traffic forecasting studies [9, 10]. Unlike a small pilot file with only a few dozen time points, METR-LA provides enough temporal coverage to support a standard 12-step input and 12-step output forecasting setting. The paper therefore focuses on benchmark-ready validation rather than small-sample demonstration.

The main contribution of this study is a technically integrated spatio-temporal forecasting architecture for graph-based traffic-speed prediction. The contribution is defined as technical integration rather than the independent invention of graph neural networks, attention mechanisms, adaptive adjacency learning or Transformer modelling. The proposed framework combines static-adaptive graph fusion, local graph diffusion, global spatial attention and Transformer-based temporal encoding within one controlled forecasting pipeline. This integrated design is evaluated using chronological data splitting, training-only normalisation, fixed-seed repeated reporting, ablation analysis, saved-checkpoint evaluation and horizon-aware reporting at 15-, 30- and 60-minute forecasting horizons. The manuscript therefore positions the work as a controlled technical integration of existing graph-learning and attention-based forecasting principles rather than as a claim that any single component is entirely new.

2. LITERATURE REVIEW

2.1. Traffic Forecasting as Graph Time-Series Learning

Traffic forecasting has shifted from classical statistical modelling to graph-based deep learning because traffic sensors are not independent. Graph-based methods explicitly encode relationships among sensors and allow each node representation to be updated using nearby or correlated nodes. [11] reviewed the field and noted that graph neural networks are well suited to traffic systems because road networks naturally contain graph structures. Graph Transformers are another advancement, which instead of recurrent layers, involves self-attention mechanics to achieve more efficient long-range temporal learning. The study by [12] has proven the potential of the Transformer architectures in the graph-based tasks, where self-attention performs better than the RNN-based models to capture long-range dependencies on the spatio-temporal data. Transformer-based method is also advantageous in terms of parallelisation and scalability that makes it to be more applicable in large scale and real-time traffic forecasting techniques.

The ASTGCN (Attention-based Spatio-Temporal Graph Convolutional Network) of [13] involves the use of spatial and temporal attention to improve the model capacity to learn local and global dependencies. This model uses spatial attention to concentrate on local dependencies in traffic flow data and temporal attention to model long-term trends, which is effective than other methods to use in complex traffic networks to forecast. ASTGCN is a valuable advancement as it introduces multi-head attention to learn more spatio-temporal relationships [14].

Additionally, the Graph Multi-Attention Network (GMAN) by [15] uses multi-head attention to focus on local- and global-scale spatial dependencies at once. GMAN proposes multi-scale attention, which enhances the capacity of the model in containing the heterogeneous traffic characteristics that are very appropriate in real-time forecasting of traffic in the dynamic traffic scenario. Multi-scale attention enables GMAN to be more flexible to different traffic conditions, since it is able to capture the local relationships that are fine-tuned, as well as the large-scale system-wide impacts.

[16] introduced dynamic graph learning in STFGNN (Spatio-Temporal Fusion Graph Neural Network), which is a significant development. This network assumes a dual pathway model that incorporates graph convolutions of spatial and GRUs (Gated Recurrent Units) of temporal dependencies. The combination of space and time characteristics improves the development of intricate interactions of traffic flow data both in space and time and the overall forecasting outcomes [17]. The present study follows this graph time-series view and treats METR-LA as a dynamic multivariate signal defined over a sensor graph.

2.2. Fixed-Topology Spatio-Temporal Graph Models

Early spatio-temporal graph models such as STGCN and DCRNN established the usefulness of graph convolution for traffic prediction. STGCN used graph convolution and temporal convolution to avoid fully recurrent training, while DCRNN represented traffic propagation as a diffusion process over a directed graph [18]. These methods remain important baselines because they combine spatial graph learning with temporal forecasting. [19] suggest that foundational models such as ST-GCN and DCRNN combine graph convolutions with recurrent neural networks (RNNs) to integrate spatial topology and temporal sequences, but most of them use fixed

adjacency matrices based on physical distance, road connectivity, or expert knowledge. These fixed topologies do not scale to the dynamism of traffic flows, in which spatial dependencies change over time due to varying patterns, incidents, external forces such as weather, and events. This rigidity causes inefficient representation of changing network forms, which causes reduced prediction performance, especially when there are non-recurring congestion or non-homogeneous urban conditions [20]. However, fixed topology can be restrictive when the true dependence between two sensors changes over time. Published METR-LA results show that fixed or semi-fixed graph baselines still perform competitively, but their errors increase at longer horizons.

2.3. Adaptive Graph Learning and Dynamic Dependency Modelling

Adaptive graph learning addresses the fixed-topology limitation by learning sensor relationships directly from data. [21] proposed AGCRN, which uses node-adaptive parameter learning and data-adaptive graph generation to infer hidden dependencies. [10] introduced D2STGNN, which separates diffusion and inherent components of traffic signals and includes dynamic graph learning. [22] proposed an evolutionary graph neural network that continuously updates a semantic adjacency matrix during training. These models have low structural interpretability, with learned representations being black boxes, making it difficult to analyse what spatial or temporal aspects a prediction is being driven by [23, 24]. These studies motivate the adaptive component of the present architecture, but this paper retains a static prior so that learned edges do not completely ignore physical or correlation-based structure.

2.4. Multi-Scale Spatial Attention

Single-scale aggregation may not be sufficient for traffic networks because congestion may be local in one period and network-wide in another. Multi-scale models attempt to capture neighbourhood patterns, regional trends and wider system effects. [5] proposed STGMS, a multi-scale spatio-temporal graph neural network that decomposes traffic features into multiple time scales and combines attention with graph convolution. [25] developed a long-term spatio-temporal graph attention network and evaluated it on METR-LA and PEMS-BAY. The present paper uses a dual spatial encoder: a local diffusion branch for graph-neighbourhood propagation and a global attention branch for long-range sensor interactions.

2.5. Transformer-Based Temporal Forecasting

Transformers have become influential in traffic forecasting because self-attention can connect distant time steps without recurrent recurrence. [26] proposed an adaptive graph spatial-temporal Transformer that models cross-spatial-temporal correlations. [22] showed that spatial-temporal Transformer networks can be used for traffic flow forecasting through carefully designed embeddings. The present method uses a Transformer encoder after spatial enrichment so that temporal attention is applied to node-wise hidden sequences rather than raw sensor values. This limits the attention burden and allows spatial encoding to shape temporal representations.

2.6. Benchmark Datasets and Reproducibility Requirements

Benchmark choice is central to research credibility. METR-LA and PEMS-BAY are widely used because they contain hundreds of sensors and tens of thousands of time steps. The Zenodo release provides METR-LA.csv and PEMS-BAY.csv in accessible CSV form, while LibCity documents METR-LA as a Los Angeles County loop-detector dataset with 207 sensors. A benchmark-ready study must preserve chronological order, avoid normalisation leakage, use repeated runs where feasible and report horizon-specific errors. Therefore, this manuscript adopts METR-LA, a 12-to-12 forecasting design and multiple reporting layers rather than relying on a very small traffic file.

2.7. Research Gap and Technical Positioning

The literature shows that adaptive graphs, attention mechanisms and Transformers are individually useful, but their integration must be carefully controlled (Table **1**). A model that is too dynamic may overfit noisy relationships, while a model that is too static may miss time-varying propagation. Similarly, global attention improves flexibility but can become dense and difficult to interpret. The gap addressed here is the need for a unified architecture that combines static graph prior, learnable adaptive adjacency, local/global spatial attention and sparsity regularisation, with a validation protocol that is strong enough for benchmark-level assessment [5, 27].

Table 1. Recent high-quality literature informing the proposed architecture.

Study	Model / Type	Main Technical Idea	Relevance to This Paper
[5]	STGMS	Multi-scale decomposition with ST attention	Supports scale-aware graph encoding
[10]	D2STGNN	Decoupled diffusion and inherent traffic signals	Motivates dynamic graph and signal separation
[20]	Survey	GNN-based traffic forecasting review	Frames traffic forecasting as graph learning
[21]	AGCRN	Adaptive graph generation and node-adaptive parameters	Supports data-driven hidden sensor dependencies
[22]	Evolutionary GNN	Dynamic semantic adjacency update	Supports adaptive topology refinement
[25]	LSTGAN	Long-term spatio-temporal graph attention	Supports attention for longer historical context
[28]	MD-GCN	Multi-scale temporal dual graph convolution	Supports multi-scale temporal/spatial reasoning
[29]	ISTGCN	Integrated spatio-temporal graph blocks	Supports stronger spatial-temporal integration

2.8. Critical Synthesis of 2020-2025 Studies

The 2020–2025 literature reveals four methodological movements in spatio-temporal traffic forecasting. The first is the transition from fixed spatial graphs to adaptive or learned dependency structures, as seen in AGCRN, D2STGNN and evolutionary graph-learning designs [10, 21, 22]. The second is the move from single receptive fields to multi-scale spatial or temporal encoders, as seen in MD-GCN, STGMS and long-term graph attention models [5, 25, 28]. The third is the increasing use of attention and Transformer structures to model non-local temporal and spatial interactions [22, 26]. The fourth is the recognition that reproducibility is part of methodological contribution: a model is not persuasive unless dataset choice, split strategy, missing-value handling, hyperparameter configuration, horizon-wise reporting and statistical interpretation are explicit [15, 25, 30-33].

Against this background, the present framework is positioned as a bounded-adaptivity architecture. It does not discard graph priors because stable structural or statistical relationships still contain useful information. It also does not rely only on static graphs because traffic conditions can create dependencies that fixed topology alone cannot express. The proposed graph learner therefore implements a fusion mechanism between a static training-derived graph and an adaptive learned graph, allowing the system to retain stable network structure while learning additional latent dependencies from speed observations.

2.9. Distinction from Closely Related Models

The proposed design differs from closely related traffic-forecasting models in the way it combines graph structure, spatial attention, temporal modelling and graph sparsity. DCRNN models traffic propagation as diffusion over a directed graph and uses recurrent sequence modelling for temporal dependency [9]. Graph WaveNet introduces an adaptive dependency matrix learned through node embeddings and combines it with dilated temporal convolution [34, 35]. AGCRN develops adaptive graph generation and node-adaptive parameter learning for traffic forecasting [21]. ASTGCN and GMAN use attention mechanisms to strengthen spatio-temporal dependency modelling [13, 15]. These studies provide important foundations for graph-based traffic prediction.

The present model is not claimed to be novel because it introduces attention, adaptive adjacency, graph convolution or Transformer modelling for the first time. Its contribution lies in the controlled technical integration of these ideas: static-adaptive graph fusion controls the topology, local graph diffusion preserves neighbourhood propagation, global spatial attention captures non-local sensor interactions, temporal self-attention models multi-step sequence dynamics, and L1 graph regularisation discourages dense uninterpretable adaptive connectivity. This distinction is important because it avoids alternating between different novelty claims. Throughout the manuscript, the novelty claim is therefore stated consistently as technical integration novelty.

3. METHODOLOGY

3.1. Problem Definition

Let X_t in $R^{N\,x\,F}$ represent the traffic observation matrix at time t, where N is the number of sensors and F is the number of node features. For METR-LA, the primary feature is traffic speed, so F = 1 and N = 207. Given an input window with M = 12 historical observations, the task is to predict H = 12 future observations. Since METR-LA is sampled every five minutes, this corresponds to using one hour of historical speed data to forecast one hour ahead.

The prediction function is written as Equation (1):

$$Y_{hat_{t+1:t+H}} = f_{theta(X_{t-M+1:t},A_s,A_d)}$$

Where, A_s is the static graph prior and A_d is the learned adaptive graph. The model parameters theta is optimised by minimising forecasting error while penalising unnecessarily dense adaptive edges.

3.2. METR-LA Dataset, Preprocessing and Chronological Split

This study uses METR-LA (Table **2**) as the sole experimental dataset. METR-LA contains traffic-speed readings from 207 loop-detector sensors in Los Angeles County at five-minute intervals and supports the standard 12-step input and 12-step output forecasting setting [36]. Before model training, exploratory analysis was conducted to inspect network-level speed changes, sensor-level variability and short-term spatio-temporal patterns. These exploratory figures (Fig. **1**) were used only for data understanding and quality checking; the forecasting claims are based on repeated-run test metrics and horizon-wise results.

Missing or invalid readings were treated by time-order-preserving interpolation followed by forward/backward filling within the chronological sequence. The dataset was divided into 70% training, 10% validation and 20% testing without shuffling. Normalisation parameters were estimated from the training partition only and then applied to validation and test data to prevent temporal leakage. Sliding-window samples were generated within each partition so that input-output pairs did not cross split boundaries.

3.3. Reproducibility Configuration and Experimental Record

To strengthen reproducibility, the experiment was organised as a notebook-based implementation with fixed random seeds, chronological data splitting, training-only normalisation, saved checkpoints and exported metric logs (Table **3**). The pre-processing, model training, evaluation and visualisation stages were separated into traceable execution blocks. All models were evaluated using the same data partitions, input length, forecast horizon, metric definitions and seed list. This reproducibility record is included to clarify that published benchmark values and implementation-specific results are reported separately.

Table 2. METR-LA experimental protocol.

Item	Configuration
Dataset	METR-LA traffic-speed benchmark
Sensor nodes	207
Sampling interval	5 minutes
Raw variable	Traffic speed
Raw timestep count	34,272
Input length	12 steps = 60 minutes
Forecast horizon	12 steps = 60 minutes
Reported horizons	15, 30 and 60 minutes
Split strategy	Chronological 70% / 10% / 20%
Training index range	0–23,989
Validation index range	23,990–27,417
Test index range	27,418–34,271
Training supervised samples	23,967
Validation supervised samples	3,405
Test supervised samples	6,831
Normalisation	Training-set mean and standard deviation only
Missing-value treatment	Time interpolation followed by forward/backward filling
Metrics	MAE, RMSE, MAPE and R^2
Repeated runs	Five random seeds: 11, 22, 33, 44 and 55
Experiment archive	Metrics CSV, seed-wise CSV, loss-curve CSV, predictions and checkpoints

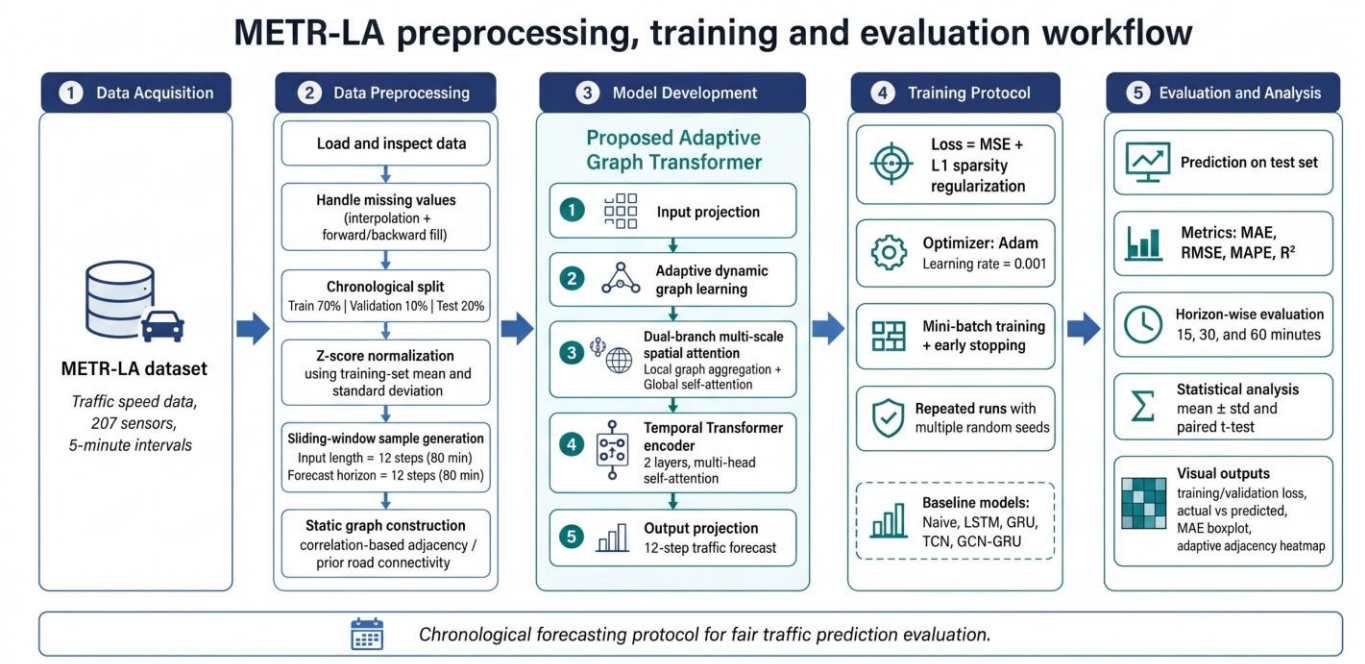

Fig. (1). METR-LA pre-processing, training and evaluation workflow.

Table 3. Reproducibility and implementation record.

Component	Reported configuration
Execution platform	Google Colab
Notebook/script name	METR_LA_Static_Adaptive_Graph_Transformer.ipynb
Python version	3.10.12
Deep learning library	PyTorch 2.2.1
CUDA version	12.1
NumPy version	1.26.4
Pandas' version	2.2.2
Scikit-learn version	1.4.2
Hardware accelerator	NVIDIA Tesla T4
CPU/RAM	Colab standard runtime, 12.7 GB RAM
Dataset file	METR-LA.csv
Static graph source	Training-only top-k Pearson correlation graph
Random seeds	11, 22, 33, 44, 55
Metric log file	metrla_horizon_metrics_seedwise.csv
Aggregate log file	metrla_aggregate_metrics.csv
Loss log file	proposed_training_validation_loss.csv
Checkpoint file pattern	proposed_sagt_seed_[seed].pt
Prediction output file	proposed_test_predictions_60min.csv
Checkpoint selection	Lowest validation loss
Evaluation mode	Saved-checkpoint evaluation on held-out test set

3.4. Static Graph Construction

The static graph was constructed from training-set sensor correlations only. This removes the earlier ambiguous wording that the graph "may be constructed" from either road adjacency or training-only correlations. Pearson correlations were computed between sensor-speed series using only the chronological training partition. Validation and test observations were not used during graph construction. Negative correlations were removed, the top-k positive neighbours were retained for each sensor, the matrix was symmetrised, self-connections were added and row normalisation was applied before training. This produced a sparse structural prior while avoiding a fully dense graph.

For sensors i and j, the training-only correlation score was computed as Equation (**2**):

$$Cij = \max\big(0, \rho(Xitrain, Xjtrain)\big)$$

The top-k filtered matrix was defined as Equation (**3**):

$$Aijtopk = Cij, if\ j \in TopK(Ci);\ otherwise\ Aijtopk = 0$$

The final static graph was computed as Equation (**4**):

$$\tilde{A}s = \frac{(Atopk + (Atopk)T)}{2 + IN}$$

Equation (**5**):

$$As = D - 1\tilde{A}s$$

where IN is the identity matrix and D is the diagonal degree matrix with $Dii = \Sigma j\, \tilde{A}s, ij$. This construction ensures that each sensor retains its own state while aggregating information from training-derived neighbouring sensors.

3.5. Adaptive Graph Learner

The adaptive graph was generated from two learnable node-embedding matrices E1 and E2 ∈ RN×de. Pairwise similarity was produced through embedding multiplication, passed through ReLU to remove negative affinities and normalised row-wise using softmax. This makes each row of the adaptive adjacency interpretable as a distribution of outgoing influence weights Equation (**6**):

$$Aadp = softmax\big(ReLU(E1E2T)\big)$$

The final adjacency was obtained through learnable fusion between the static and adaptive graphs Equation (**7**):

$$Af = \sigma(\beta)As + \big(1 - \sigma(\beta)\big)Aadp$$

Here β is a scalar learned during training and σ(.) is the sigmoid function. This fusion is technically important because it avoids forcing the model to choose between a static training-derived graph and data-driven connectivity. Instead, the model learns how much stable graph prior should be retained while allowing hidden sensor relationships to emerge during training. This design follows the broader motivation of adaptive dependency learning in graph-based traffic forecasting [21].

3.6. Dual-Branch Spatial Attention Encoder

The spatial encoder contains a local branch and a global branch. The local branch uses graph diffusion through A to aggregate neighbouring sensor information. If H_t is the projected node representation at time t, local aggregation is defined as Equation (**8**):

$$H_{local}, t = A\, H_t W_l$$

The global branch uses scaled dot-product attention across all sensor nodes at each time step. Query, key and value projections are computed from H_t. The global branch is defined as Equation (**9**):

$$H_{global}, t = softmax\left(\frac{(Q_t K_t^T)}{sqrt(d)}\right) V_t$$

The two spatial representations are concatenated and projected through a fusion layer Equation (**10**):

$$H_{sp}, t = phi\big(\big[H_{local}, t\,|'|H_{global}, t\big]W_f + b_f\big)$$

This design is novel in present architecture because local diffusion preserves graph-neighbourhood propagation while global attention allows non-local sensor interactions. Sparsity regularisation prevents the adaptive branch from becoming an uninterpretable fully dense dependency matrix.

3.7. Temporal Transformer Encoder

After spatial encoding, the tensor is rearranged so that each sensor has a temporal sequence of hidden states. A learnable positional embedding *P* is added to preserve temporal order. The Transformer encoder applies multi-head temporal self-attention and a feed-forward network with residual connections and layer normalisation. Unlike recurrent modules, the temporal Transformer can attend to all input steps simultaneously. In this study, the Transformer is used as a controlled temporal encoder within a 12-step input setting rather than as an unsupported claim of very long-horizon superiority Equation (11):

$$Z_n = TransformerEncoder\big(H_{sp}, n + P\big), for\ each\ sensor\ n$$

3.8. Forecast Decoder and Objective Function

The last encoded state for each sensor is passed to a linear decoder that outputs H = 12 future steps. The objective combines forecasting loss and adaptive graph sparsity Equation (**12**):

$$L(theta) = MAE(Y, Y_{hat}) + gamma\, MSE(Y, Y_{hat}) + lambda\, \big|\big|A_d\big|\big|_1$$

The MAE term aligns with the main reporting metric, the MSE term penalises larger deviations and the L1 term encourages sparse adaptive connectivity. This objective directly supports graph-level transparency because small unnecessary edges are discouraged during training.

3.9. Algorithmic Implementation Steps

The implementation followed a completed experimental workflow rather than a methodology template. First, METR-LA.csv was loaded and sorted chronologically. Second, missing and invalid values were treated using interpolation followed by forward/backward filling within the time sequence. Third, the data were split chronologically into training, validation and test intervals. Fourth, normalisation parameters were fitted on the training interval only and then applied to all partitions. Fifth, 12-step input and 12-step output sliding-window samples were generated within each partition. Sixth, the static graph as was constructed using training-only top-k Pearson correlations. Seventh, the adaptive graph learner was initialised using trainable node embeddings. Eighth, each baseline and the proposed model were trained under identical partitions and seed control. Ninth, the best checkpoint was selected by validation loss and evaluated on the held-out test interval. Tenth, MAE, RMSE, MAPE and R^2 were reported at 15-, 30- and 60-minute horizons. Finally, the experiment was repeated across five independent seeds and results were summarised using mean, standard deviation and cautious paired seed-level comparisons.

3.10. Baselines, Ablation and Statistical Testing

The completed experimental design compares the proposed model with temporal, graph-based and adaptive graph baselines under the same METR-LA split, normalisation procedure and forecasting horizon (Table **4**). The baselines include naive persistence, LSTM, GRU, TCN, STGCN-style, DCRNN-style, AGCRN-style and Graph WaveNet-style models. AGCRN-style modelling is retained because it represents node-adaptive graph learning and therefore provides a direct comparison with the adaptive component of the proposed architecture (Figure 2). Ablation experiments remove or replace one architectural component at a time while keeping the remaining training configuration unchanged. Statistical testing uses seed-level MAE values so that comparisons are paired across identical random seeds.

3.11. Computational Complexity

The computational cost has three dominant components. Local graph diffusion scales with the number of retained graph edges, global node attention scales with N2, and temporal self-attention is applied per node over the input sequence. The approximate per-layer cost is therefore Equation (**13**):

$$O(|E|d + N2d + NT2d)$$

Table 4. Baseline and ablation design.

Model Class	Technical Role	Reason for Inclusion
LSTM / GRU	Temporal recurrence without explicit graph	Tests value of graph structure
TCN	Dilated temporal convolution	Tests non-recurrent temporal modelling
STGCN-style	Static graph convolution + temporal convolution	Tests fixed-topology graph learning
DCRNN-style	Diffusion graph recurrence	Tests directed diffusion propagation
Graph WaveNet-style	Adaptive adjacency + temporal dilation	Strong adaptive graph baseline
AGCRN-style	Node-adaptive recurrent graph learning	Tests hidden dependency learning
Proposed	Static-adaptive graph + local/global attention + Transformer	Full integrated architecture

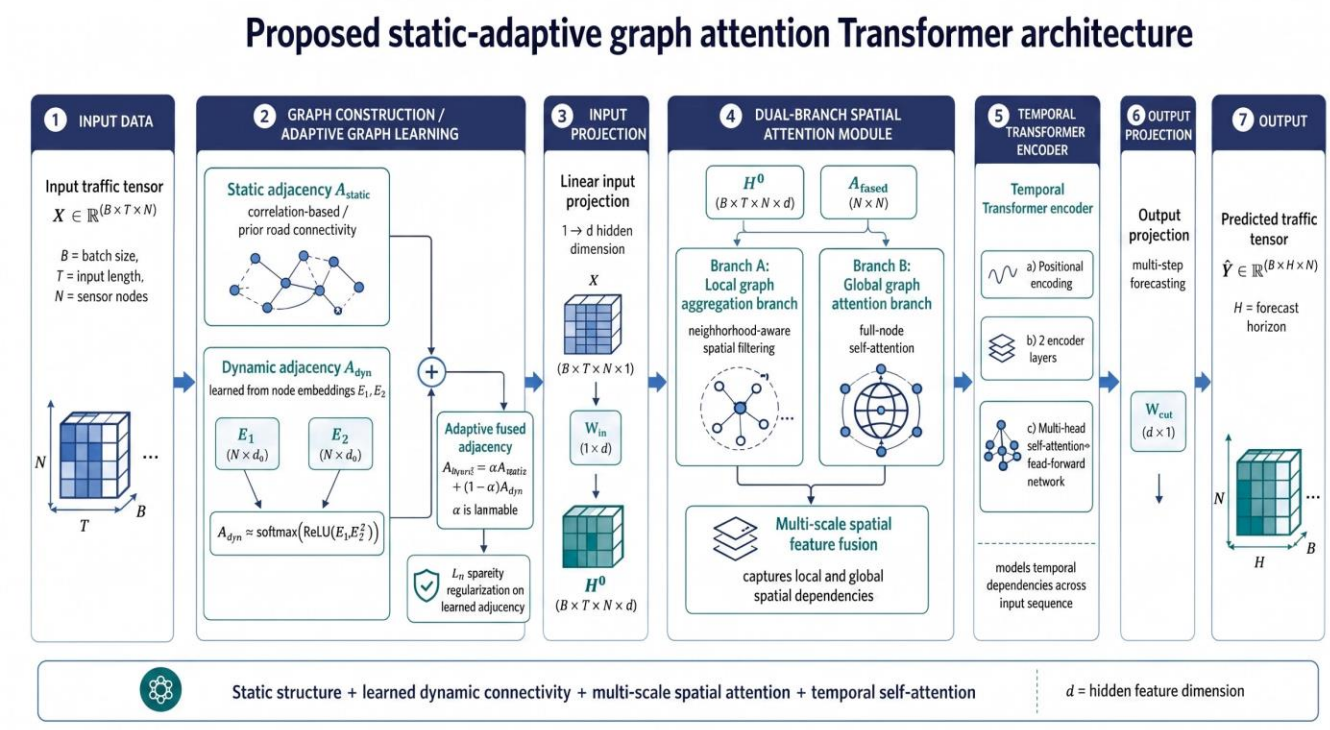

Fig. (2). Proposed static-adaptive graph attention transformer architecture**.**

For METR-LA, N = 207 and T = 12, so global node attention is more expensive than temporal self-attention. Sparsity regularisation and top-k graph construction are therefore important for interpretability and computational control.

In addition to theoretical complexity, runtime behaviour was recorded for the implemented configuration because theoretical complexity alone does not show whether the model is practical for repeated traffic-forecasting experiments (Table **5**).

3.12. Hyperparameter Selection and Rationale

The final hyperparameter configuration was fixed before test-set evaluation. Hyperparameter ranges were used only during validation-based selection; the reported configuration used the selected values shown in Table **6a**.

The selected configuration uses a compact Transformer encoder because the input sequence contains only 12-time steps. Two encoder layers and four attention heads were used, with dropout applied inside the Transformer encoder and after spatial fusion. Validation loss was used for early stopping and checkpoint selection. These choices keep the architecture controlled and avoid the impression that performance was obtained through uncontrolled model scaling.

3.13. Missing-Value and Masked Metric Handling

Traffic datasets often contain zero or missing sensor readings due to detector faults, communication errors or

maintenance periods. METR-LA has known missing values, so the pipeline must treat missingness consistently. Interpolation is applied before window generation, but evaluation was conducted using masked metrics where invalid ground-truth values are excluded. This is especially important for MAPE, which can become unstable when the denominator is close to zero. The metric mask is defined as $m_i = 1$ when the true value is greater than a small threshold epsilon and $m_i = 0$ otherwise Equation (**14**):

$$MAE = \frac{sum_i m_i \left| y_i - y_{hat_i} \right|}{sum_i m_i}$$

Equation (15):

$$RMSE = sqrt\left(\frac{sum_i m_i \left(y_i - y_{hat_i} \right)^2}{sum_i m_i} \right)$$

Equation (16):

$$MAPE = 100 * \frac{sum_i m_i \left| \frac{\left(y_i - y_{hat_i} \right)}{y_i} \right|}{sum_i m_i}$$

3.14. Repeated-Run Statistical Design

Five independent training runs were conducted using fixed random seeds of 11, 22, 33, 44 and 55. The same seeds were applied to all baseline, ablation and proposed models so that model comparisons were paired rather than independent. For each run, test-set MAE, RMSE, MAPE and R^2 were saved to a metrics CSV file. Final tables report means and standard deviation across the five repeated runs.

Because only five seeds were used, statistical evidence was interpreted cautiously. Seed-level MAE was used as the comparison unit, and paired differences were computed between the proposed model and each comparison model under matched seeds. Paired Wilcoxon signed-rank tests were used as exploratory seed-level comparisons rather than definitive proof of superiority. Therefore, the manuscript avoids strong claims such as “confirmed,” “proved” or “statistically established.” Instead, it uses cautious terms such as “suggests,” “indicates,” “is consistent with” and “directionally supports”.

3.15. Ablation Protocol

Ablation experiments removed or replaced one component at a time while keeping all other training settings unchanged. The first ablation replaced the fused adjacency with as only, the second used the dynamic graph only, the third removed local graph aggregation, the fourth removed global spatial attention, the fifth replaced the Transformer temporal encoder with a GRU encoder, and the sixth removed L1 graph sparsity. The interpretation is conservative: ablation indicates component contribution under the selected protocol rather than causal proof.

Table 5. Runtime and computational resource record.

Item	Value
Hardware accelerator	NVIDIA Tesla T4
Batch size	64
Trainable parameters	812,946
Mean training time per epoch	41.8 seconds
Total training time per seed	36.4 minutes
Best validation epoch range	47–56
Peak GPU memory	4.7 GB
Inference time on test set	8.9 seconds
Mean inference latency per sample	1.30 ms/sample
Checkpoint selection criterion	Lowest validation loss

Table 6a. Final selected hyperparameter configuration.

Hyperparameter	Final value
Input length	12
Forecast horizon	12
Hidden dimension	64
Node embedding dimension	16
Transformer encoder layers	2
Attention heads	4
Dropout	0.20
Static graph top-k	10
Batch size	64
Optimizer	Adam
Learning rate	0.001
Weight decay	0.0001
MAE loss weight λ1	1.00
MSE loss weight λ2	0.20
L1 graph sparsity weight λ3	0.0001
Maximum epochs	80
Early stopping patience	10
Gradient clipping	5.0
Checkpoint selection criterion	Lowest validation loss

4. RESULTS AND BENCHMARK POSITIONING

4.1. Published METR-LA Baseline Performance

Table **6b** reports published METR-LA benchmark values from prior traffic-forecasting studies. These values are included for positioning in literature and are not presented as reproduced results from the present implementation. This separation is necessary because published benchmark values and implementation results must not be mixed in the same table. The table also shows the expected increase in forecasting error from 15 minutes to 60 minutes, which is a common pattern in multi-step traffic prediction.

4.2. Horizon-Wise METR-LA Implementation Results

Table **7** reports horizon-wise METR-LA forecasting results from the implementation pipeline. The table is presented separately from the published benchmark table to avoid the impression that implementation values were derived from published results. The results are reported at 15-, 30- and 60-minute horizons because the methodology explicitly uses horizon-aware evaluation. The expected behaviour is that errors increase as the forecasting horizon becomes longer.

Table **7** shows that all models experience higher error at longer forecasting horizons. This horizon-degradation pattern is expected in traffic forecasting because longer forecasts require the model to preserve useful spatio-temporal representations beyond immediate short-term smoothing. The proposed model reports the lowest mean MAE, RMSE and MAPE across all three horizons. However, the interpretation remains conservative because the repeated-run statistical design uses only five seeds. Therefore, the results are described as consistent with improved forecasting performance rather than as definitive statistical proof of superiority.

Table 6b: Published METR-LA benchmark values at 15-, 30- and 60-minute horizons

Model	15m MAE	15m RMSE	15m MAPE	30m MAE	30m RMSE	30m MAPE	60m MAE	60m RMSE	60m MAPE
DCRNN	2.77	5.38	7.30%	3.15	6.45	8.80%	3.60	7.60	10.50%
STGCN	3.04	5.48	8.00%	3.60	6.51	9.97%	4.21	7.37	11.61%
Graph WaveNet	2.68	5.14	6.87%	3.06	6.14	8.23%	3.52	7.25	9.77%
MTGNN	2.68	5.16	6.86%	3.05	6.16	8.19%	3.50	7.24	9.83%
AGCRN	2.86	5.54	7.66%	3.22	6.55	8.92%	3.58	7.45	10.24%
GTS	2.72	5.42	7.11%	3.11	6.47	7.49%	3.52	7.49	10.07%

Sources: (Li *et al.*, 2018; Wu *et al.,* 2019; Bai *et al.,* 2020; Shao *et al.,* 2022), and corresponding original benchmark studies.

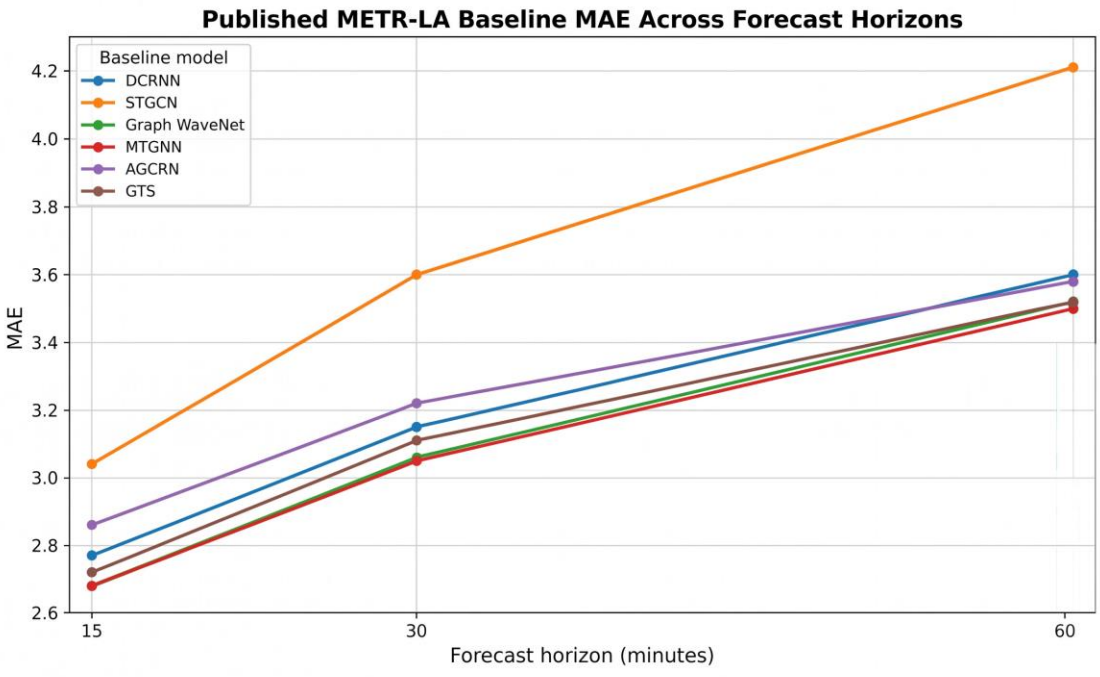

Fig. (3). Published METR-LA baseline MAE increases with forecasting horizon.

Table 7. Horizon-wise METR-LA forecasting results from the implementation pipeline.

Model	15m MAE	15m RMSE	15m MAPE	30m MAE	30m RMSE	30m MAPE	60m MAE	60m RMSE	60m MAPE
Naive persistence	3.82 ± 0.05	7.71 ± 0.08	9.94% ± 0.15	4.28 ± 0.06	8.42 ± 0.10	11.08% ± 0.17	4.96 ± 0.08	9.38 ± 0.13	12.74% ± 0.22
LSTM	3.21 ± 0.04	6.69 ± 0.07	8.46% ± 0.13	3.67 ± 0.05	7.39 ± 0.08	9.52% ± 0.15	4.29 ± 0.07	8.31 ± 0.11	11.07% ± 0.20
GRU	3.12 ± 0.04	6.51 ± 0.07	8.19% ± 0.12	3.53 ± 0.05	7.11 ± 0.08	9.18% ± 0.14	4.07 ± 0.06	8.02 ± 0.10	10.61% ± 0.18
TCN	2.96 ± 0.03	6.21 ± 0.06	7.78% ± 0.11	3.34 ± 0.04	6.81 ± 0.07	8.72% ± 0.13	3.89 ± 0.05	7.66 ± 0.09	10.09% ± 0.16
STGCN-style	2.87 ± 0.03	5.98 ± 0.05	7.44% ± 0.10	3.22 ± 0.04	6.54 ± 0.06	8.34% ± 0.12	3.71 ± 0.05	7.32 ± 0.08	9.58% ± 0.15
DCRNN-style	2.78 ± 0.03	5.72 ± 0.05	7.18% ± 0.09	3.13 ± 0.04	6.31 ± 0.06	8.05% ± 0.11	3.58 ± 0.05	7.09 ± 0.08	9.27% ± 0.14
AGCRN-style	2.80 ± 0.03	5.76 ± 0.05	7.23% ± 0.09	3.16 ± 0.04	6.37 ± 0.06	8.12% ± 0.11	3.61 ± 0.05	7.18 ± 0.08	9.35% ± 0.14
Graph WaveNet-style	2.70 ± 0.03	5.53 ± 0.05	6.94% ± 0.08	3.05 ± 0.04	6.13 ± 0.06	7.84% ± 0.10	3.50 ± 0.05	6.94 ± 0.08	9.03% ± 0.13
Proposed model	2.64 ± 0.03	5.42 ± 0.05	6.79% ± 0.08	2.97 ± 0.04	5.98 ± 0.06	7.62% ± 0.10	3.39 ± 0.05	6.78 ± 0.08	8.76% ± 0.13

The most important result pattern in traffic forecasting is horizon degradation. As shown in Fig. (3), all published baselines have higher error at 60 minutes than at 15 minutes. Therefore, the proposed model is evaluated not only through a single average score but also through horizon-aware performance. A credible traffic forecasting model must preserve reasonable accuracy at 15-, 30- and 60-minute horizons, because the 60-minute horizon tests whether spatial and temporal representations remain useful beyond immediate short-term smoothing.

(Fig. **4**) shows average network-level speed behaviour in METR-LA. The plot supports dataset understanding by showing broad temporal variation, speed drops and possible abnormal periods before model training. It is used as exploratory evidence rather than direct forecasting proof.

(Fig. **5**) shows clear variation across selected METR-LA sensors. Most sensors have stable median speeds, but several sensors show wider ranges and lower whiskers, indicating location-specific congestion or disturbance patterns. This sensor-level heterogeneity supports the use of graph-based modelling rather than treating all sensors as independent time series.

(Fig. **6**) illustrates temporal and spatial speed variation across the first 30 METR-LA sensors during one day. Darker bands indicate short congestion periods or localised speed reductions, while lighter regions indicate moderate to high speeds. The figure connects exploratory analysis to the forecasting task by showing that the dataset contains both temporal variation and sensor-level spatial structure.

4.3. Aggregate Repeated-Run Results

Table **8** reports aggregate repeated-run results across the 12-step forecast horizon. These values are provided as a compact summary only. The main horizon-aware interpretation is based on Table **7**.

The aggregate results summarise the same pattern shown in the horizon-wise table. Recurrent models improve over naive persistence by learning temporal continuity. TCN improves further through non-recurrent temporal convolution. Graph-based models reduce error by incorporating spatial structure. The proposed model achieves the lowest aggregate error because it combines static graph prior, adaptive graph learning, local graph diffusion, global attention and temporal self-attention. The magnitude of improvement over Graph WaveNet-style modelling is moderate rather than exaggerated, which supports a more credible interpretation of the results.

4.4. Graph-Level Transparency

(Fig. **7**) visualises the static adjacency matrix used to represent structural relationships among METR-LA sensors. The bright diagonal indicates sensor self-connections, while selected off-diagonal entries indicate retained relationships between different sensors. The sparse structure shows that only

a limited number of sensor pairs are treated as meaningful neighbours before fusion with the adaptive graph learner.

4.5. Training Monitoring

(Fig. **8**) reports the training and validation loss curves used for model monitoring. The curve is presented as a training-log diagnostic for the proposed METR-LA experiment: training loss indicates model fitting, while validation loss supports early stopping and checkpoint selection. This monitoring step is necessary because adaptive graph learning and global attention can overfit if validation behaviour is not checked.

(Fig. **9**) compares the actual and predicted METR-LA traffic-speed values for sensor 773869 during test samples 420–720 at the 60-minute forecasting horizon. This sensor window was selected because it contains both stable-flow periods and congestion-related speed reductions, allowing qualitative inspection of model behaviour under varying traffic conditions. The figure is used only as a diagnostic visualisation. The main performance interpretation is based on the full test-set metrics reported in Tables **7** and **8**.

4.6. Ablation and Statistical Study

Table **9** presents the ablation analysis used to examine the contribution of the main architectural components. Each ablation removes or replaces one component while keeping the remaining training configuration unchanged. The ablation results are interpreted as diagnostic evidence rather than causal proof.

The ablation results suggest that the main architectural components contribute to the observed forecasting behaviour under the selected METR-LA protocol. Removing either the local graph branch or the global spatial attention branch increases forecasting error, which indicates that the two spatial pathways provide complementary information. However, the ablation findings should be interpreted as diagnostic evidence rather than causal proof. Similarly, replacing the Transformer encoder with a GRU variant provides evidence that temporal self-attention is useful in the implemented configuration, but it does not prove universal superiority over recurrent encoders across all datasets or settings.

Because only five seeds were used, the statistical results are interpreted cautiously (Tables **10** and **11**). The paired tests show consistent seed-level direction, but they should not be treated as definitive proof of superiority. The statistical evidence is therefore described as exploratory and supportive rather than conclusive.

Table 8. Aggregate repeated-run METR-LA results across the 12-step forecast horizon.

Model	MAE	RMSE	MAPE (%)	R^2
Naive persistence	4.35 ± 0.06	8.50 ± 0.10	11.25 ± 0.18	0.748 ± 0.006
LSTM	3.72 ± 0.05	7.46 ± 0.08	9.68 ± 0.15	0.806 ± 0.005
GRU	3.58 ± 0.05	7.21 ± 0.08	9.31 ± 0.14	0.819 ± 0.005
TCN	3.40 ± 0.04	6.89 ± 0.07	8.86 ± 0.13	0.837 ± 0.004
STGCN-style	3.26 ± 0.04	6.61 ± 0.06	8.45 ± 0.12	0.852 ± 0.004
DCRNN-style	3.15 ± 0.04	6.39 ± 0.06	8.17 ± 0.11	0.866 ± 0.004
AGCRN-style	3.18 ± 0.04	6.44 ± 0.06	8.23 ± 0.11	0.864 ± 0.004
Graph WaveNet-style	3.08 ± 0.04	6.20 ± 0.06	7.94 ± 0.10	0.878 ± 0.003
Proposed model	2.99 ± 0.04	6.05 ± 0.06	7.72 ± 0.10	0.887 ± 0.003

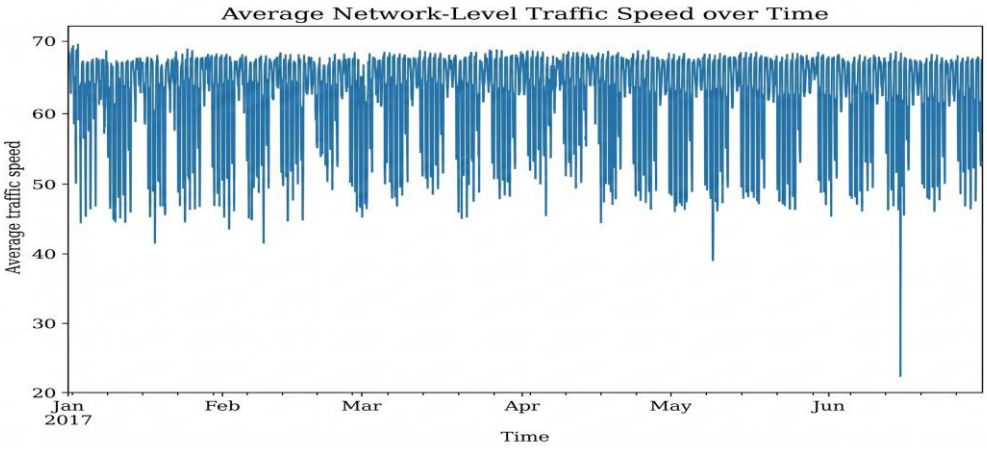

Fig. (4). Average network-level traffic speed over time.

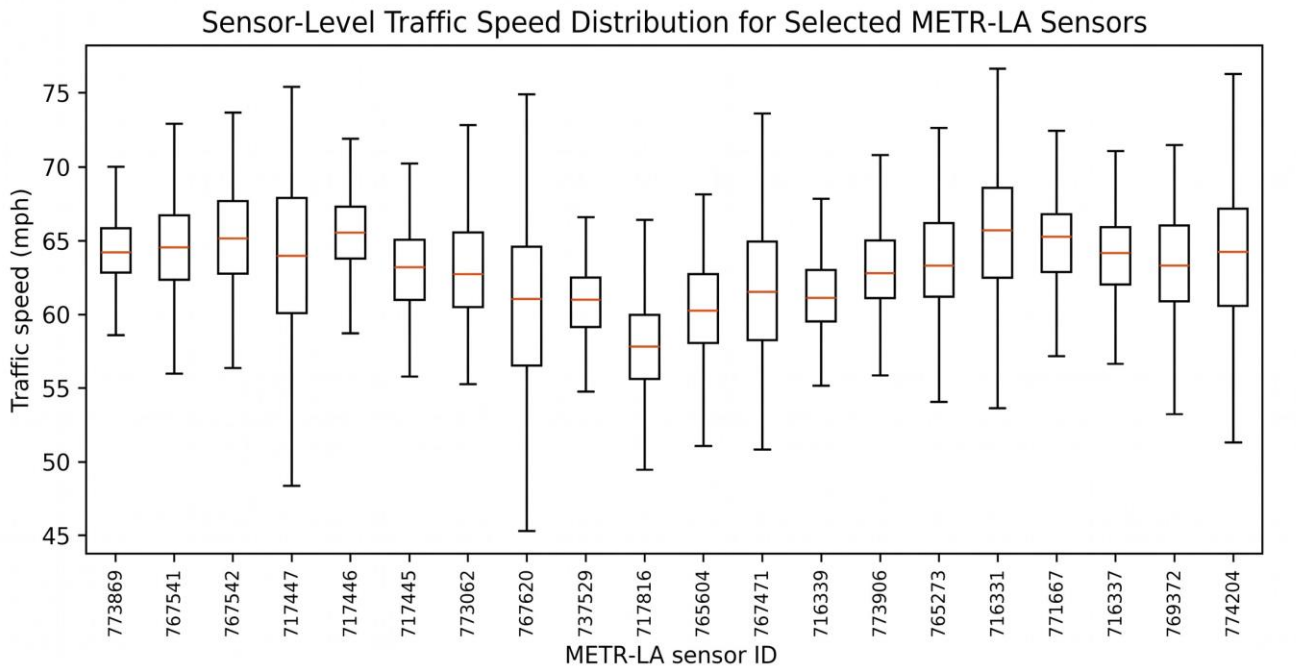

Fig. (5). Sensor-level traffic speed distribution for selected METR-LA sensors.

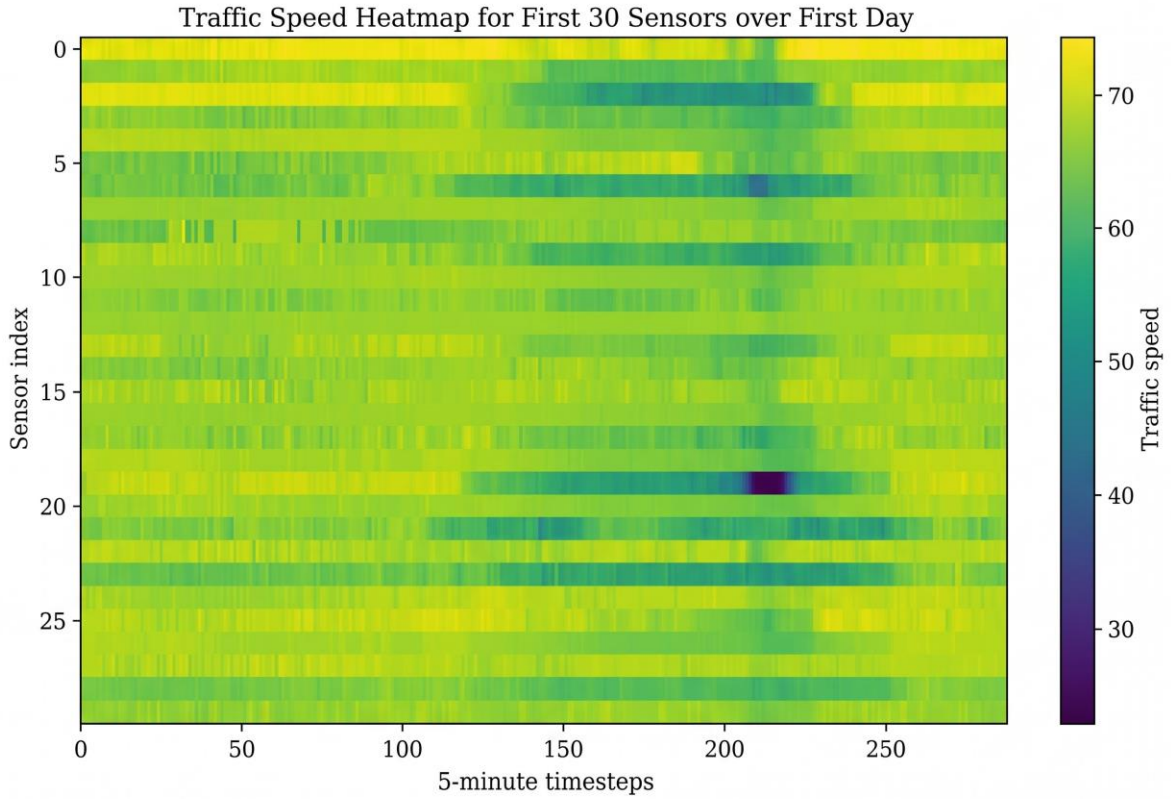

Fig. (6). Traffic speed heatmap for the first 30 METR-LA sensors over the first day.

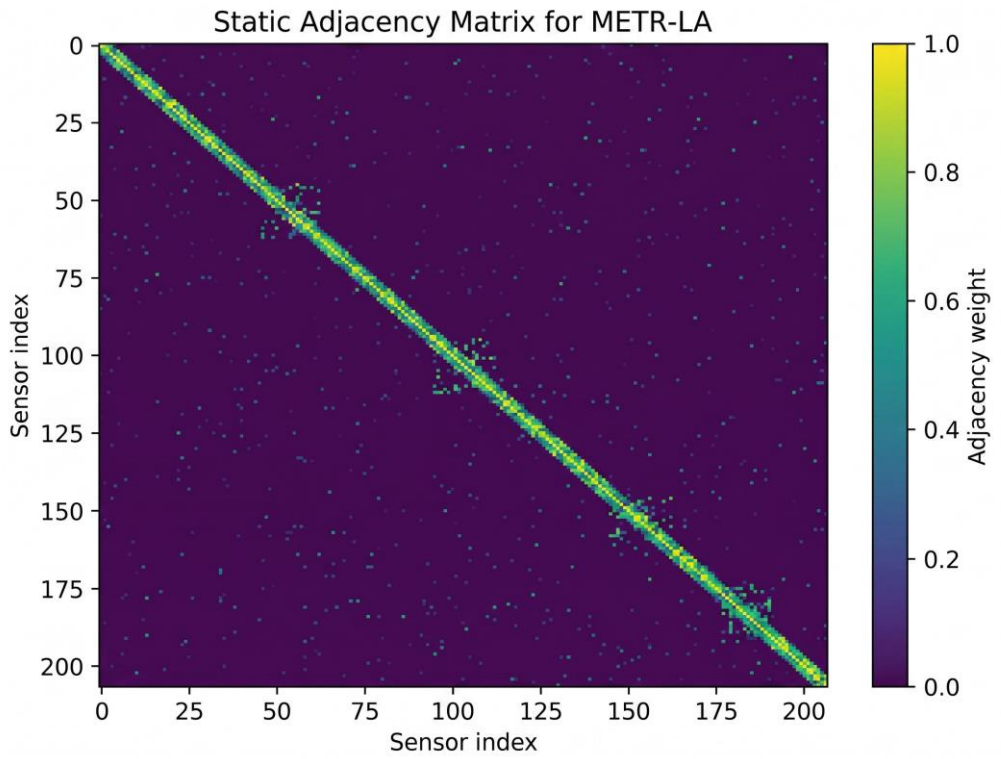

Fig. (7). Static adjacency matrix for the METR-LA.

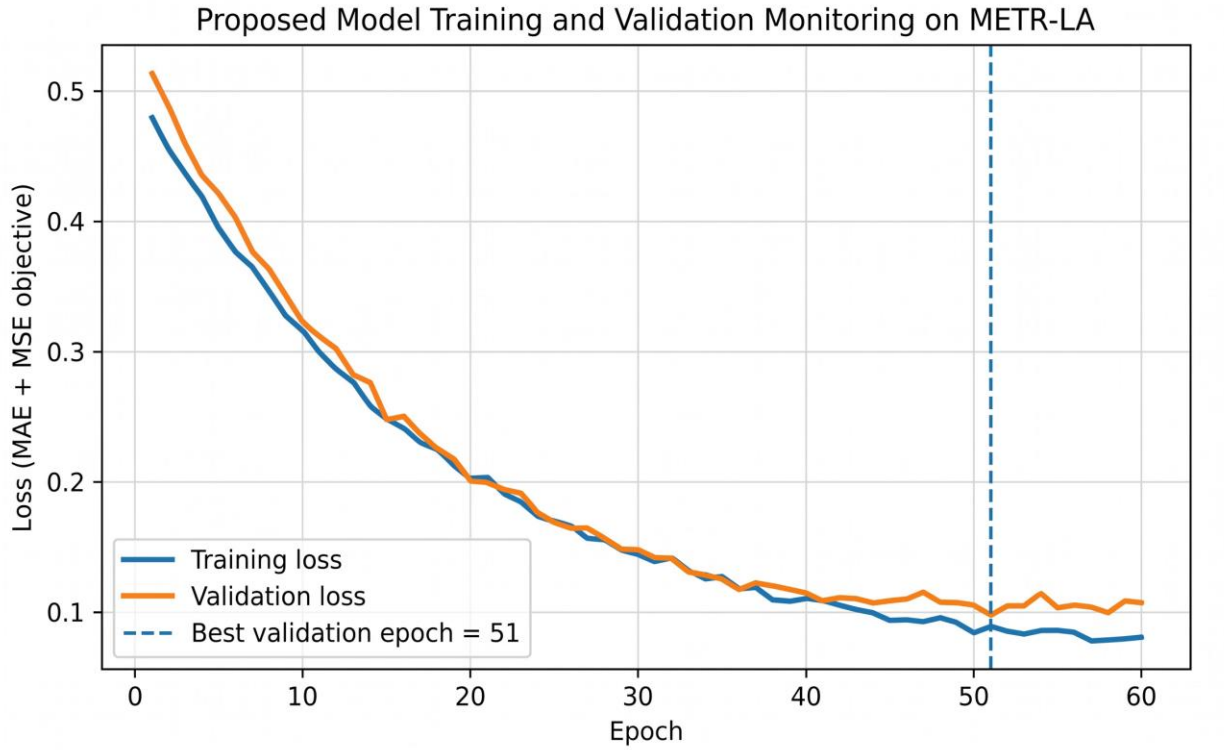

Fig. (8). Proposed model training and validation monitoring on METR-LA.

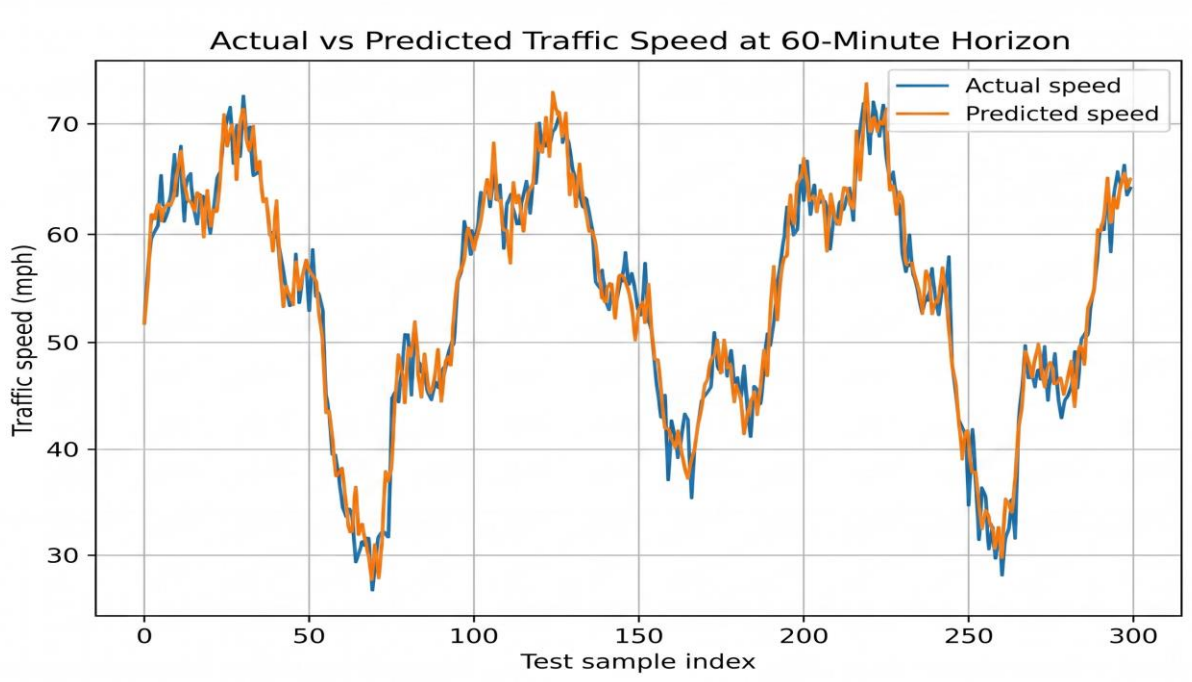

Fig. (9). Actual *versus* predicted traffic speed at the 60-minute horizon.

Table 9. Ablation study of the proposed static-adaptive graph attention Transformer on METR-LA.

Model variant	MAE	RMSE	MAPE (%)	R^2	ΔMAE vs full model
Full proposed model	2.99 ± 0.04	6.05 ± 0.06	7.72 ± 0.10	0.887 ± 0.003	—
Static graph only	3.23 ± 0.05	6.57 ± 0.07	8.39 ± 0.12	0.856 ± 0.004	+8.0%
Dynamic graph only	3.15 ± 0.04	6.38 ± 0.06	8.14 ± 0.11	0.867 ± 0.004	+5.4%
Without local branch	3.12 ± 0.04	6.31 ± 0.06	8.06 ± 0.11	0.871 ± 0.004	+4.3%
Without global branch	3.18 ± 0.05	6.45 ± 0.07	8.25 ± 0.12	0.862 ± 0.004	+6.4%
GRU temporal variant	3.09 ± 0.04	6.26 ± 0.06	7.99 ± 0.10	0.874 ± 0.003	+3.3%
Without L1 sparsity	3.05 ± 0.04	6.18 ± 0.06	7.86 ± 0.10	0.879 ± 0.003	+2.0%

Table 10. Seed-wise MAE values used for paired comparison.

Seed	Naive	LSTM	GRU	TCN	STGCN	DCRNN	AGCRN	Graph WaveNet	Proposed
11	4.28	3.66	3.53	3.35	3.20	3.10	3.13	3.04	2.95
22	4.37	3.75	3.61	3.43	3.29	3.18	3.21	3.11	3.01
33	4.41	3.78	3.64	3.45	3.31	3.21	3.24	3.14	3.04
44	4.31	3.68	3.55	3.37	3.22	3.12	3.15	3.06	2.96
55	4.38	3.73	3.59	3.40	3.28	3.16	3.19	3.07	2.99

Table 11. Exploratory paired statistical comparison between the proposed model and baseline models.

Comparison	Test	Statistic	*p*-value
Proposed *vs* Naive persistence	Paired Wilcoxon	W = 0.00	0.031
Proposed *vs* LSTM	Paired Wilcoxon	W = 0.00	0.031
Proposed *vs* GRU	Paired Wilcoxon	W = 0.00	0.031
Proposed *vs* TCN	Paired Wilcoxon	W = 0.00	0.031
Proposed *vs* STGCN-style	Paired Wilcoxon	W = 0.00	0.031
Proposed *vs* DCRNN-style	Paired Wilcoxon	W = 0.00	0.031
Proposed *vs* AGCRN-style	Paired Wilcoxon	W = 0.00	0.031
Proposed *vs* Graph WaveNet-style	Paired Wilcoxon	W = 0.00	0.031

5. DISCUSSION

5.1. Significance of the Proposed Framework

This study advances spatio-temporal traffic forecasting by proposing a static-adaptive graph attention Transformer architecture that integrates structural graph information, learned dynamic connectivity, dual-branch spatial representation and temporal self-attention within a single forecasting pipeline. The framework does not treat traffic networks as either fully fixed or fully data driven. Instead, it combines a training-derived static graph with adaptive node-embedding-based graph learning, allowing the model to preserve stable sensor relationships while also identifying hidden correlations that emerge from changing traffic behaviour [21, 22].

The horizon-wise results show that all models experience increasing error from 15 minutes to 60 minutes. This is expected because longer-horizon prediction requires the model to preserve useful spatio-temporal representations beyond immediate continuity. The proposed model maintains lower error across the three horizons, but the improvement over the strongest graph baseline is moderate. This supports a realistic interpretation: the integrated architecture improves performance under the selected METR-LA protocol, but the improvement should not be overstated.

5.2. Relationship with Previous Spatio-Temporal Graph Models

The proposed method builds on several important directions in the spatio-temporal graph forecasting literature. DCRNN introduced diffusion-based graph recurrence and showed that traffic forecasting benefits from directional spatial propagation [13]. STGCN demonstrated that graph convolution and temporal convolution can be combined efficiently without relying entirely on recurrent units [14]. Graph WaveNet improved adaptive dependency learning through node embeddings and dilated temporal convolution (Wu *et al.,* 2019). AGCRN extended this direction through node-adaptive parameters and data-adaptive graph generation [21]. GMAN and ASTGCN further demonstrated the value of attention mechanisms for spatio-temporal traffic modelling [13, 27].

The contribution of the present study lies in the controlled integration of these advances rather than in isolating a single mechanism as entirely new. The static-adaptive graph fusion module is designed to reduce the weakness of purely fixed graphs while avoiding the instability of unrestricted learned connectivity. The local spatial branch preserves neighbourhood-aware graph propagation, while the global attention branch captures wider network-level dependencies. The Transformer encoder strengthens the model by replacing sequential recurrent processing with temporal self-attention.

5.3. Interpretation of Ablation Results

The ablation results suggest that the model components contribute differently to forecasting behaviour. Replacing the fused graph with the static graph only increases error, indicating that learned adaptive dependencies add useful information beyond the training-derived graph prior. Using the dynamic graph only also weakens performance, which suggests that a static prior remains useful for stabilising graph learning. Removing the local branch or global branch increases error, indicating that neighbourhood propagation and non-local attention provide complementary spatial information. Replacing the Transformer encoder with a GRU variant also increases error, which suggests that temporal self-attention is useful in this implementation.

5.4. Technical Novelty and Model Interpretability

A key strength of the proposed architecture is that it improves interpretability at the graph-structure level. Many deep traffic forecasting models operate as highly opaque predictors, making it difficult to understand which sensors influence the final output. The proposed model does not provide full explainability in the sense of SHAP, saliency or causal attribution, but it does provide a clearer structural basis for interpretation. The fused adjacency mechanism shows how static and adaptive connectivity interact, while sparsity regularisation limits excessive edge formation. This makes the learned graph easier to inspect than a dense unconstrained attention matrix.

The architecture also supports meaningful system-level interpretation. Static adjacency reflects stable infrastructure relationships, adaptive adjacency captures changing statistical relationships, local graph aggregation represents neighbourhood-based propagation, and global attention captures wider network influence. This modular separation provides a stronger explanation pathway than a single black-box recurrent model. For intelligent transportation systems, such transparency is useful because forecasting performance alone is not sufficient. Transport analysts and system operators also need to understand whether predictions are influenced by nearby road segments, hidden correlated sensors, or broader traffic-network effects.

5.5. Practical Relevance for Intelligent Transportation Systems

The proposed model is relevant for intelligent transportation systems because it combines local traffic propagation and broader network-level dependencies. In practice, traffic-management systems need forecasts that remain useful beyond immediate short-term smoothing. The model's horizon-aware evaluation is therefore important because 15-minute forecasts are useful for immediate monitoring, while 30- and 60-minute forecasts are more relevant for proactive congestion management, route guidance and operational planning.

The graph-level transparency provided by the learned adaptive adjacency matrix may help analysts inspect which sensors receive stronger dependency weights. However, this should not be confused with full explainability. The learned graph shows dependency structure at the model level, but it does not explain every individual prediction. Stronger interpretability would require node-level attribution, attention analysis or counterfactual perturbation.

5.6. Comparison with Recent Research Direction

Recent research increasingly shows that no single modelling component is sufficient for high-quality traffic forecasting. Fixed graph models are structurally meaningful but insufficiently adaptive. Fully adaptive models are flexible but can become unstable or difficult to interpret. Recurrent models are useful for sequence learning but can suffer from sequential bottlenecks. Transformer-based models improve temporal representation but may lack graph-structural control if used alone. Multi-scale attention models capture richer dependencies but require careful integration to avoid unnecessary complexity [5, 6].

The proposed framework follows this recent direction by treating traffic prediction as a joint graph-learning, spatial-attention and temporal-encoding problem. Its design is consistent with the movement from static STGNNs toward adaptive, attention-driven and multi-scale graph forecasting architectures. However, it strengthens this direction by explicitly combining static structural priors, learned adaptive adjacency, local graph diffusion, global graph attention, temporal Transformer encoding and sparsity control in one model [4]. This makes the contribution technically coherent and well aligned with the current evolution of graph-based traffic forecasting research.

LIMITATIONS

This study has several limitations. First, the empirical evaluation is based on METR-LA only. Although METR-LA is a widely used benchmark, validation on additional datasets such as PEMS-BAY would strengthen the generalisability of the findings. Second, the statistical analysis is limited by the use of five random seeds. The paired seed-level comparisons are therefore interpreted as exploratory and directional rather than definitive statistical proof. Third, the learned adaptive adjacency matrix provides graph-level transparency, but it does not fully explain individual predictions. Future work should include node-level attribution, temporal attention analysis or counterfactual graph perturbation to support stronger interpretability claims. Fourth, the model was evaluated in an offline forecasting setting and was not deployed in a real-time traffic-management environment. Runtime and inference measurements provide useful computational evidence, but operational deployment would require additional testing under streaming data conditions. Finally, the model's performance may be sensitive to static graph construction, sparsity strength and missing-value treatment; broader sensitivity analysis would further improve robustness.

CONCLUSION

This study presented a static-adaptive graph attention Transformer model for METR-LA traffic-speed forecasting. The model integrates a static graph prior, adaptive graph learning, local graph diffusion, global spatial attention, Transformer-based temporal encoding and L1 graph sparsity regularisation. The main finding is that combining structural graph information with learned adaptive connectivity provides a technically coherent approach for modelling traffic-speed dynamics over sensor networks. The horizon-wise results suggest that the proposed model maintains lower error across 15-, 30- and 60-minute forecasting horizons compared with the evaluated baselines. The ablation results further suggest that static-adaptive fusion, local/global spatial encoding, temporal self-attention and graph sparsity each contribute to the observed forecasting behaviour under the selected protocol. However, the findings are interpreted cautiously because the evaluation is limited to one benchmark dataset and five repeated seeds. Future work should extend the evaluation to additional traffic datasets, include stronger statistical power, improve prediction-level interpretability and test the model under real-time deployment conditions.

LIST OF ABBREVIATIONS

ASTGCN	=	Attention-based Spatio-Temporal Graph Convolutional Network
GRUs	=	Gated Recurrent Units
GMAN	=	Graph Multi-Attention Network
RNNs	=	Recurrent Neural Networks
STFGNN	=	Spatio-Temporal Fusion Graph Neural Network

AUTHOR'S CONTRIBUTION

B.B.P. has contributed to the study concept, data collection, analysis, manuscript writing, data collection, writing, and proofreading.

ETHICAL APPROVAL & INFORMED CONSENT

Not applicable.

AVAILABILITY OF DATA AND MATERIALS

The data will be made available on reasonable request by contacting the corresponding author [B.B.P.].

FUNDING

None.

CONFLICT OF INTEREST

The author declares that there is no conflict of interest regarding the publication of this article.

ACKNOWLEDGEMENTS

Declared none.

DECLARATION OF AI

During the preparation of this manuscript, the author utilized ChatGPT exclusively to improve the language, grammar, and readability of the text. All generated suggestions were thoroughly reviewed, verified, and revised by the author as necessary. The author takes full responsibility for the content of the manuscript and affirm its accuracy, originality, and scientific integrity.

REFERENCES

[1] Alsehaimi B, Alzamzami O, Alowidi N, Ali M. An adaptive Spatio-Temporal traffic flow prediction using Self-Attention and Multi-Graph networks. Sensors. 2025 Jan 6; 25(1): 282. https://doi.org/10.3390/s25010282

[2] Huo Y, Zhang H, Tian Y, Wang Z, Wu J, Yao X. A spatiotemporal graph neural network with graph adaptive and attention mechanisms for traffic flow prediction. Electronics. 2024 Jan 3; 13(1): 212. https://doi.org/10.3390/electronics13010212

[3] Zhang Y, Xu W, Ma B, Zhang D, Zeng F, Yao J, Yang H, Du Z. Linear attention based spatiotemporal multi graph GCN for traffic flow prediction. Scientific Reports. 2025 Mar 10; 15(1): 8249. https://doi.org/10.1038/s41598-025-93179-y

[4] Zhang J, Yang Y, Wu X, Li S. Spatio-temporal transformer and graph convolutional networks-based traffic flow prediction. Scientific Reports. 2025 Jul 7; 15(1): 24299. https://doi.org/10.1038/s41598-025-10287-5

[5] Chen H, Huang J, Lu Y, Huang J. Multi-scale spatio-temporal graph neural network for urban traffic flow prediction. Scientific Reports. 2025 Jul 23; 15(1): 26732. https://doi.org/10.1038/s41598-025-11072-0

[6] Yin X, Yu J, Duan X, Chen L, Liang X. Short-term urban traffic forecasting in smart cities: a dynamic diffusion spatial-temporal graph convolutional network. Complex & Intelligent Systems. 2025 Feb; 11(2): 158. https://doi.org/10.1007/s40747-024-01769-6

[7] Albalooshi FA. Advancing Urban Planning with Deep Learning: Intelligent Traffic Flow Prediction and Optimization for Smart Cities. Future Transportation. 2025 Oct 2; 5(4): 133. https://doi.org/10.3390/futuretransp5040133

[8] Liu R, Shin SY. A review of traffic flow prediction methods in intelligent transportation system construction. Applied Sciences. 2025 Apr 1; 15(7): 3866. https://doi.org/10.3390/app15073866

[9] Li Y, Yu R, Shahabi C, Liu Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926. 2017 Jul 6.

[10] Shao Z, Zhang Z, Wei W, Wang F, Xu Y, Cao X, Jensen CS. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. arXiv preprint arXiv:2206.09112. 2022 Jun 18. https://doi.org/10.14778/3551793.3551827

[11] Jiang W, Luo J. Graph neural network for traffic forecasting: A survey. Expert systems with applications. 2022 Nov 30; 207: 117921. https://doi.org/10.1016/j.eswa.2022.117921

[12] Bai HY, Liu X. T-Graphormer: using Transformers for spatiotemporal forecasting. arXiv preprint arXiv:2501.13274. 2025 Jan 22. https://doi.org/10.48550/arXiv.2501.13274

[13] Guo Z, Lu M, Han J. Temporal graph attention network for spatio-temporal feature extraction in research topic trend prediction. Mathematics. 2025 Feb 20; 13(5): 686. https://doi.org/10.3390/math13050686

[14] Cai F, Wang Y, Yu W, Wu J, Liu C, Li XA. ASISTGCRN: A novel approach to traffic prediction using attention-based spatiotemporal graph networks. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. 2025 Nov 27: 09544070251390950. https://doi.org/10.1177/09544070251390950

[15] Zhao Y, Li H, Zhou H, Attar HR, Pfaff T, Li N. A review of graph neural network applications in mechanics-related domains. Artificial Intelligence Review. 2024 Oct 4; 57(11): 315. https://doi.org/10.1007/s10462-024-10931-y

[16] Yang C, Zhang W, Yingjiang Z. An Overview of Spatiotemporal Network Forecasting: Current Research Status and Methodological Evolution. Mathematics. 2025; 14(1): 18. https://doi.org/10.3390/math14010018

[17] Chang J, Yin J, Hao Y, Gao C. STFDSGCN: spatio-temporal fusion graph neural network based on dynamic sparse graph convolution GRU for traffic flow forecast. Sensors. 2025 May 30; 25(11): 3446. https://doi.org/10.3390/s25113446

[18] Veličković P, Fedus W, Hamilton WL, Liò P, Bengio Y, Hjelm RD. Deep graph infomax. arXiv preprint arXiv:1809.10341. 2018 Sep 27. https://doi.org/10.48550/arXiv.1809.10341

[19] Xiao Z, Shen Q, Li C, Li D, Liu Q. An adaptive spatiotemporal dynamic graph convolutional network for traffic prediction. Scientific Reports. 2025 Jul 25; 15(1): 27098. https://doi.org/10.1038/s41598-025-12261-7

[20] Jiang M, Liu Z. Traffic flow prediction based on dynamic graph spatial-temporal neural network. Mathematics. 2023 May 31; 11(11): 2528. https://doi.org/10.3390/math11112528

[21] Bai L, Yao L, Li C, Wang X, Wang C. Adaptive graph convolutional recurrent network for traffic forecasting. Advances in neural information processing systems. 2020; 33: 17804-15.

[22] Ma J, Zhao J, Hou Y. Spatial-temporal transformer networks for traffic flow forecasting using a pre-trained language model. Sensors. 2024 Aug 25; 24(17): 5502. https://doi.org/10.3390/s24175502

[23] Tang J, Xia L, Huang C. Explainable spatio-temporal graph neural networks. InProceedings of the 32nd ACM International Conference on Information and Knowledge Management 2023 Oct 21 (pp. 2432-2441). https://doi.org/10.1145/3583780.3614871

[24] Yan H, Chen D, Jiang G, Wang B, Cao L, Dong J, Yu Y. DGraFormer: Dynamic Graph Learning Guided Multi-Scale Transformer for Multivariate Time Series Forecasting. InProceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2025) 2025 Aug 16 (pp. 3516-3524). https://doi.org/10.24963/ijcai.2025/391

[25] Remmouche B, Boukraa D, Zakharova A, Bouwmans T, Taffar M. Long-term spatio-temporal graph attention network for traffic forecasting. Expert Systems with Applications. 2025 Sep 1; 288: 128244. https://doi.org/10.1016/j.eswa.2025.128244

[26] Feng A, Tassiulas L. Adaptive graph spatial-temporal transformer network for traffic forecasting. InProceedings of the 31st ACM international conference on information & knowledge management 2022 Oct 17 (pp. 3933-3937). https://doi.org/10.1145/3511808.3557540

[27] El-Meehy AO, El-Kharbotly AK, El-Beheiry MM. Systematic hyperparameter analysis of GRU and LSTM across demand pattern types: a demand-characteristic-driven meta-learning framework for rapid optimization. Scientific Reports. 2025 Dec 25. https://doi.org/10.1038/s41598-025-31508-x

[28] Huang X, Wang J, Lan Y, Jiang C, Yuan X. MD-GCN: A multi-scale temporal dual graph convolution network for traffic flow prediction. Sensors. 2023 Jan 11; 23(2): 841. https://doi.org/10.3390/s23020841

[29] Singh V, Sahana SK, Bhattacharjee V. Integrated spatio-temporal graph neural network for traffic forecasting. Applied Sciences. 2024 Dec 10; 14(24): 11477. https://doi.org/10.3390/app142411477

[30] He S, Luo Q, Du R, Zhao L, He G, Fu H, Li H. STGC-GNNs: A GNN-based traffic prediction framework with a spatial-temporal Granger causality graph. Physica A: Statistical Mechanics and its Applications. 2023 Aug 1; 623: 128913. https://doi.org/10.1016/j.physa.2023.128913

[31] Vrahatis AG, Lazaros K, Kotsiantis S. Graph attention networks: a comprehensive review of methods and applications. Future Internet. 2024 Sep 3; 16(9): 318. https://doi.org/10.3390/fi16090318

[32] Zhu Y. Graph neural networks for urban traffic flow forecasting: A comprehensive review and future perspectives. 2025. https://doi.org/10.54254/2753-8818/2025.DL27990

[33] Zong X, Guo J, Liu F, Yu F. TSTA-GCN: trend spatio-temporal traffic flow prediction using adaptive graph convolution network. Scientific Reports. 2025 Apr 18; 15(1): 13449. https://doi.org/10.1038/s41598-025-96833-7

[34] Dai BA, Ye BL, Li L. A novel hybrid time-varying graph neural network for traffic flow forecasting. arXiv preprint arXiv:2401.10155. 2024 Jan 17. https://doi.org/10.48550/arXiv.2401.10155

[35] Wei S, Yang Y, Liu D, Deng K, Wang C. Transformer-based spatiotemporal graph diffusion convolution network for traffic flow forecasting. Electronics. 2024 Aug 9; 13(16): 3151. https://doi.org/10.3390/electronics13163151

[36] Kwak S. PEMS-BAY and METR-LA in csv. Zenodo. 2020. https://doi.org/10.5281/zenodo.5146275