Latent-Topology Graph State-Space Model (LT-GSSM) for Robust Traffic Fore-Casting

Selma Kerdous
Department of Computer Science and Engineering, University of Quebec at Outaouais, Canada

***Abstract*—Accurate traffic forecasting remains challenging when sensor data are noisy, incomplete, or non-stationary. Recent advances in spatio-temporal learning have combined Graph Neural Networks (GNNs) with recurrent, convolutional, or attention mechanisms to capture spatio-temporal dependencies. However, most existing approaches remain largely deterministic and rely on fixed or pre-learned adjacency matrices, limiting their adaptability when network structures evolve or sensor reliability varies. Some methods further stack multiple adjacency matrices to represent complex spatial relations, yet still lack explicit mechanisms to model uncertainty, resulting in reduced robustness under degraded data conditions. This work introduces the Latent Topology Graph State-Space Model (LT-GSSM), a probabilistic framework designed to enhance robustness and adaptability in traffic forecasting. LT-GSSM represents the road network as a latent dynamic graph whose structure evolves over-time through dynamic adjacency learning based on past hidden states and observations, enabling the model to capture evolving spatial correlations such as congestion propagation. Temporal dependencies are modelled by a nonlinear state-space function implemented with a Temporal Convolutional Network (TCN), which captures long-range temporal patterns without recurrence. The probabilistic state-space formulation explicitly represents sensor noise and handles missing data through probabilistic estimation inspired by Kalman filtering. By jointly integrating dynamic graph learning, explicit noise modelling, and nonlinear temporal transitions, LT-GSSM achieves greater stability and resilience to data uncertainty. Experiments on SUMO simulations and real-world PeMS datasets show that LT-GSSM consistently outperforms static and adaptive-graph models, providing a strong foundation for robust spatio-temporal forecasting under uncertain conditions.**

Keywords*—*Traffic forecasting; graph neural networks; state-space models; latent topology; dynamic adjacency learning; spatio-temporal modeling; noise and missing data robustness; probabilistic modeling

I. Introduction

Despite major advances in artificial intelligence, spatio-temporal learning remains challenging due to complex spatial dependencies, dynamic temporal patterns, and inherent data uncertainty. These difficulties are particularly evident in traffic forecasting, where large-scale sensor data are often noisy, incomplete, and non-stationary [1].

Traditional prediction models perform well under ideal conditions but lack explicit mechanisms to handle noise, missing data, or uncertainty, leading to poor stability in real deployments. Moreover, traffic networks are non-Euclidean, governed by road connectivity rather than distance [2], and their correlations evolve dynamically with congestion, incidents, or weather [3,4].

Recent advances in spatio-temporal graph architectures—such as diffusion-based DCRNN [5], recurrent T-GCN [6], and attention-driven ASTGCN or ST-Transformer [7,8]—have improved prediction accuracy However, deterministic graph adaptation remains fundamentally limited under noisy or incomplete observations, as it cannot represent the structural uncertainty induced by sensor degradation or missing data.

To address these limitations, we propose the Latent Topology Graph State-Space Model (LT-GSSM), a probabilistic framework for robust and adaptive traffic forecasting. Unlike conventional adaptive GNNs (e.g., AGCRN, DGCRN), where adjacency updates depend on static node embeddings or deterministic functions, LT-GSSM learns a latent dynamic graph, where the adjacency is treated as a state-conditioned latent variable rather than a deterministic function of node embeddings. This enables joint modeling of temporal and structural uncertainty, resulting in a self-evolving spatio-temporal representation that better reflects real-world non-stationarity.

The adjacency matrix A_t evolves from past hidden states and observations, allowing the model to capture time-varying spatial correlations such as congestion propagation or structural shifts. Temporal dependencies are modeled by a nonlinear state-space transition using a Temporal Convolutional Network (TCN), which captures long-range patterns without recurrence. The probabilistic state-space formulation explicitly represents process and observation noise, enabling robust estimation under sensor degradation or missing data—an approach inspired by Kalman filtering.

The main contributions of this work are as follows:

- A unified probabilistic graph state-space formulation, in which both temporal dynamics and graph structure are modelled within a single latent state-space framework, rather than treating adaptive graphs and temporal uncertainty as separate components.
- State-conditioned latent topology modelling, where the adjacency matrix is inferred as a latent random variable conditioned on hidden states, instead of being deterministically parameterized from node embeddings as in existing adaptive GNNs.
- Explicit modeling of process and observation uncertainty within the state-space dynamics, enabling robust spatio-

temporal forecasting under sensor noise and missing data, without relying on deterministic graph updates.

The remainder of this paper is organized as follows: Section II reviews related work on spatio-temporal and probabilistic models; Section III describes the LT-GSSM methodology; Section IV presents experimental results and robustness analyses; Discussion is given in Section V and Section VI concludes with future research directions.

II. Related Works

Traffic forecasting is a complex spatio-temporal task shaped by irregular topologies, non-stationary dynamics, and noisy sensor conditions.

Classical time-series models such as ARIMA and VAR [7,8] capture temporal patterns but ignore spatial correlations among sensors. Early machine-learning methods [9] improved flexibility but struggled with high-dimensional dependencies across the road network.

Deep learning advanced the field through Recurrent Neural Networks (RNNs) such as LSTM and GRU [10,11], which model temporal sequences but process each sensor independently. Convolutional Neural Networks (CNNs) [12] extended modeling to spatial grids, yet their Euclidean structure limits generalization to real road networks [13].

To overcome these limitations, Graph Neural Networks (GNNs) [14] model traffic as a graph, where nodes represent sensors and edges capture road connectivity. Hybrid spatio-temporal architectures such as DCRNN, T-GCN, and attention-based models like ASTGCN [15] and ST-Transformer [16] jointly learn spatial and temporal dependencies. While accurate, these models assume fixed or pre-learned adjacency matrices, neglecting dynamic spatial relationships that vary with congestion, incidents, or weather.

Recent advances introduced adaptive and dynamic graph learning, including AGCRN [17], DGCRN [18], and RT-GCN [19], which update adjacency matrices using learned node embeddings or Gaussian-based convolutions. These designs improve flexibility but remain deterministic, failing to model uncertainty or stochastic graph evolution explicitly.

Parallel progress in probabilistic state-space models (SSMs) such as KalmanNet [20], Deep Kalman Filters [21], and Neural State-Space Models [22] has reintroduced uncertainty estimation and latent dynamics into deep learning. However, these frameworks are typically temporal only, lacking explicit graph reasoning. Attempts to combine GNNs with probabilistic methods, such as Probabilistic GNNs [23] and Graph Variational Filters [24], remain computationally costly and seldom applied to large-scale traffic networks.

In summary, prior models either:

- Learn dynamic graphs without probabilistic treatment of noise and uncertainty, or
- Apply probabilistic state-space reasoning without spatial modelling.

This gap motivates our Latent Topology Graph State-Space Model (LT-GSSM), a unified probabilistic framework that jointly models time-varying graph topology and temporal dynamics. By treating the adjacency A_t as a latent variable evolving from hidden states and observations, LT-GSSM captures both structural and stochastic uncertainty, achieving robust forecasting under noisy, incomplete, and dynamically changing conditions.

III. Methodology

A. Data Simulation and Preparation

A key element of our methodology is the use of both synthetic and real-world datasets to assess robustness. Real data often suffers from partial sensor coverage and measurement noise, whereas synthetic data generated with SUMO (Simulation of Urban Mobility) [24] allows full control over network coverage, traffic conditions, and injected noise. This controlled environment enables systematic evaluation of our GSSM under diverse and challenging scenarios.

1) Synthetic dataset (SUMO): We generated a two-month traffic dataset using SUMO for the Outaouais region near Ottawa (Fig. 1), based on real road networks extracted from OpenStreetMap. Traffic flows were dynamically managed through TraCI (Traffic Control Interface) [25] to emulate realistic conditions such as peak-hour congestion and weekend traffic reduction. Vehicle counts were aggregated into 5-minute intervals, following common benchmarking practices [26, 27].

The chosen duration aligns with established datasets such as METR-LA and PEMS-BAY [14, 15], which span 2–4 months. This ensures a balance between temporal coverage and computational efficiency, while remaining comparable to standard baselines. To assess robustness, we generated several perturbed versions of the SUMO dataset:

- Gaussian noise injected at 5–30 % levels to simulate sensor errors.
- Reduced sensor coverage through three controlled scenarios:
  - Sensors limited to main roads,
  - Random removal of 20 % of sensors, and
  - Random removal of 50 % of sensors.

These perturbations make it possible to study the model's resilience to measurement uncertainty and coverage sparsity, establishing SUMO as an effective robustness benchmark.

2) Validation of realism: To validate the realism of our synthetic dataset, we compared key traffic patterns with re-al-world datasets such as PEMS08 and UTD19 [28]. Despite differences in absolute magnitudes, our dataset reproduces essential traffic behaviors observed in real-world da-ta, including:

- morning and evening peak-hour congestion,
- sharp reductions in weekend traffic, and
- daily distribution patterns. These similarities, illustrated in Fig. 2 and Fig. 3, confirm that SUMO-based data capture fundamental traffic dynamics.

(a) Selected road network.

(b) Traffic flow visualization

Fig. 1. Road network and traffic flow visualization: (a) Selected road network, (b)Traffic flow visualization.

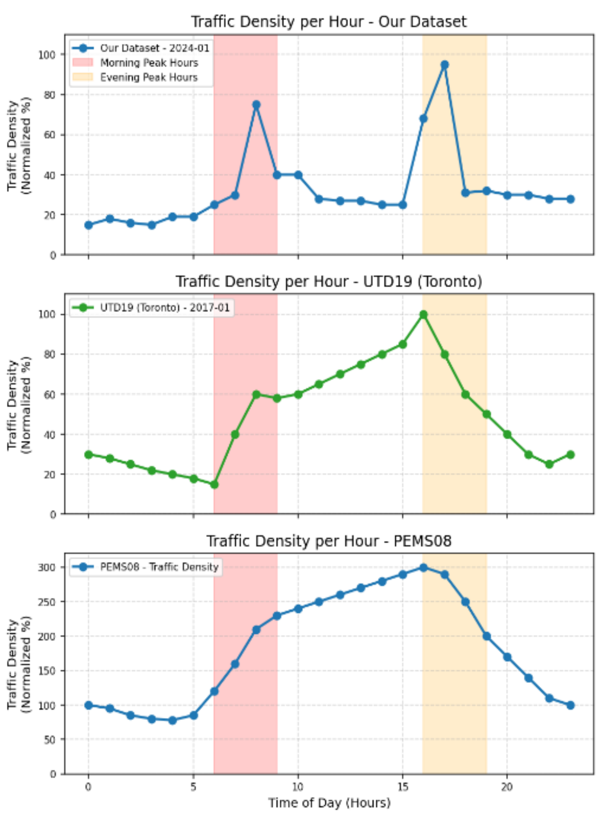


Fig. 2. Comparison of hourly traffic density trends: our dataset vs. Real-world datasets.

This SUMO-based dataset thus provides a reliable testbed for assessing the performance and robustness of GSSM under controlled scenarios of noise and sensor sparsity, complementing evaluations on real-world datasets.

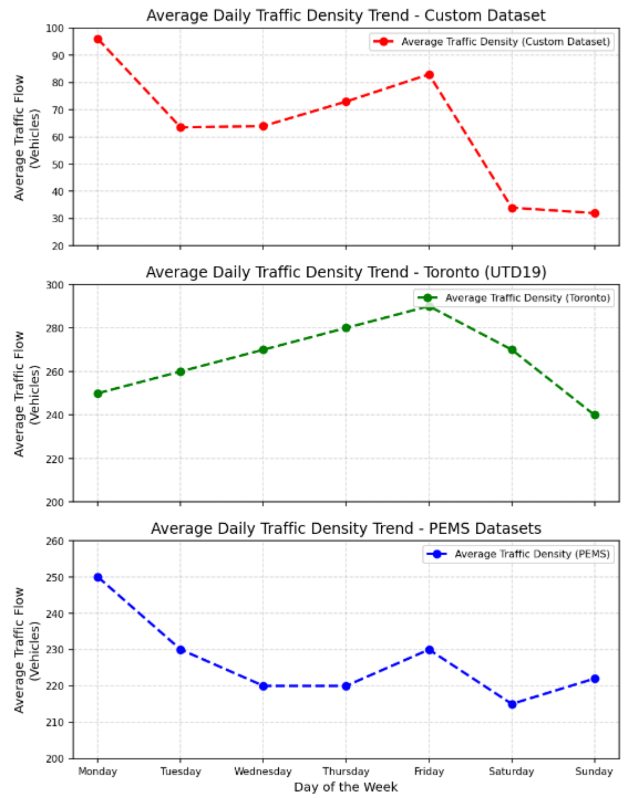


Fig. 3. Weekend traffic density comparison: our dataset vs. Real-world datasets.

B. Spatio-Temporal Data Modeling and Representation

To capture spatio-temporal dependencies in traffic data, we represent all datasets as three-dimensional tensors X ∈ ℝ(T×E×V), where T denotes the number of time steps (e.g., 5-minute intervals), E the number of road segments or sensors, and V the set of traffic variables such as vehicle density and average speed. This tensor format enables the joint modeling of temporal dynamics, spatial interactions, and multivariate dependencies, and is well-suited for graph-based neural architectures [29, 30].

Complementing this tensor structure, we construct an initial spatial adjacency matrix $A_0$to encode the physical connectivity between road segments. This matrix serves as a topological prior — it provides an initial structural representation of the network, which will later be refined dynamically during model training based on learned latent representations. Thus, while A_0defines the initial spatial configuration, the model does not assume a fixed topology, allowing subsequent updates to better capture evolving spatio-temporal correlations. The method used for constructing A_0differs depends on the dataset:

- SUMO (Synthetic Data): Since the simulated Outaouais network is relatively small and geographically localized, distances between connected roads are short and highly heterogeneous. To preserve strong locality while avoiding dense, fully connected graphs, we define adjacency entries inversely proportional to the physical distances between connected segments. This choice ensures that immediate neighbors exert stronger influence, reflecting realistic propagation of congestion within a compact road network. This ensures that spatial relationships reflect real-world topology rather than mere Euclidean proximity [31].

Our adjacency matrix has the following structure:

$$A_0 = \begin{array}{c|cccc} & \text{edge}_1 & \text{edge}_2 & \cdots & \text{edge}_N \\ \hline \text{edge}_1 & 0 & \frac{1}{d_{1,2}} & \cdots & \frac{1}{d_{1,N}} \\ \text{edge}_2 & \frac{1}{d_{2,1}} & 0 & \cdots & \frac{1}{d_{2,N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \text{edge}_N & \frac{1}{d_{N,1}} & \frac{1}{d_{N,2}} & \cdots & 0 \end{array}$$

- PEMS Datasets These datasets cover larger and more heterogeneous urban areas with complex traffic flows. To model smooth spatial dependencies while avoiding overly dense connections, we adopt a Gaussian kernel approach [14, 15], computing adjacency as:

$$A_{ij} = \exp(-\left(\frac{d_{ij}}{\sigma}\right)^2) \quad (1)$$

where, d_{ij} is the physical distance between sensors i and j, and σ is set to 1000 meters. This distinction reflects the different scales and characteristics of the networks: SUMO enables precise control and sparse adjacency, whereas PEMS requires soft, distance-based weighting to capture complex spatial patterns without excessive graph density. This adjacency matrix serves as an initial structural prior reflecting geographic connectivity. During LT-GSSM training, this matrix evolves into a latent, data-driven adjacency A_t that adapts to changing spatial correlations and temporal patterns in traffic dynamics.

C. Hybrid Graph–State–Space Model with Latent Topology (LT-GSSM)

The main innovation of this work is the integration of a latent, dynamically evolving graph topology within a state–space framework—a dimension rarely addressed in spatio-temporal modeling.

Unlike prior adaptive GNNs such as AGCRN [17] and DGCRN [18], which learn static or semi-static adjacency matrices, the proposed LT-GSSM continuously updates its graph structure during both training and inference.

The model begins with a static prior A_0, derived from spatial proximity (e.g., inverse distance or Gaussian similarity). As learning progresses, the adjacency matrix A_t is inferred from the previous hidden state h_{t-1}, coupling spatial evolution with temporal reasoning.

Unlike adaptive GNNs such as AGCRN or DGCRN, where adjacency updates are deterministic functions of node embeddings, LT-GSSM treats topology evolution as part of the latent state dynamics under uncertainty.

At each time step, LT-GSSM performs four operations:

- Latent topology inference — infers A_t from h_{t-1};
- Spatial feature extraction — applies GCN using A_t and input X_t;
- Temporal modeling — updates h_t via a Temporal Convolutional Network (TCN).
- Topology update — refines A_{t+1} using the new latent state.

This feedback loop makes the graph self-evolving, allowing spatial dependencies to adapt dynamically to non-stationary or uncertain conditions. By linking latent state evolution to graph adaptation, LT-GSSM (Fig. 4) jointly models spatial, temporal, and structural uncertainty, achieving a robust and adaptive representation for real-world spatio-temporal forecasting.

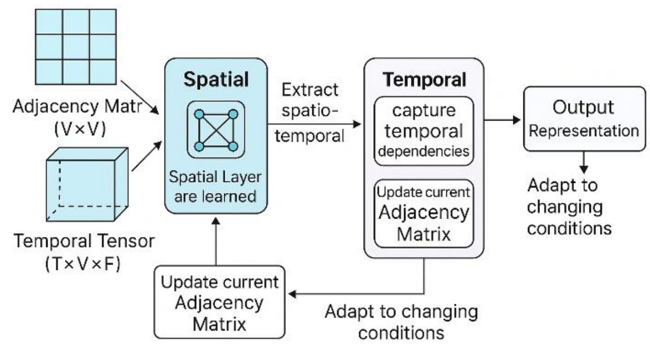


Fig. 4. Overview of LT-GSSM architecture.

1) Temporal layer — state-space model and uncertainty management: The LT-GSSM adopts a probabilistic state–space formulation in which latent states evolve under dynamic graph structures and stochastic perturbations. This allows the model to jointly capture temporal dependencies and uncertainty due to sensor noise, missing data, or non-stationary traffic.Two noise sources are modeled: transition uncertainty (latent fluctuations in dynamics) and observation uncertainty (measurement errors and delays).

In a classical linear SSM:

$$h_t = Ah_{t-1} + Bx_t + w_t, y_t = Hh_t + v_t \quad (2)$$

where, w_t and v_t denote Gaussian process and observation noise. To handle nonlinear and time-varying dynamics, the LT-GSSM replaces fixed matrices with a Temporal Convolutional Network (TCN):

$$h_t = \text{TCN}(Z_{1:t}) + w_t, w_t \sim \mathcal{N}(0, \sigma_t^2 I) \quad (3)$$

The TCN serves as the nonlinear transition function, capturing long-range temporal dependencies without recurrence while preserving probabilistic uncertainty through w_t.

2) Integration with dynamic graph topology: Temporal evolution is explicitly coupled with a time-varying adjacency.From the previous latent state, node embeddings are obtained via a learnable projection:

$$E_t = \phi_E(h_{t-1}), E_t \in \mathbb{R}^{V \times d} \quad (4)$$

Pairwise similarities define a latent affinity matrix:

$$S_t = \frac{E_t E_t^{\top}}{\sqrt{d}} \quad (5)$$

which is row-normalized to yield the dynamic adjacency:

$$A_t = \text{softmax}_{\text{ροω}}(S_t) \quad (6)$$

This attention-like normalization ensures each row forms a probability distribution over neighbors, making the graph context-dependent rather than fixed. The updated A_t guides the spatial layer, and the resulting latent state h_t generates A_{t+1}, creating a feedback loop where topology and dynamics co-evolve.

3) Graph Convolutional Network (GCN): At each step, the GCN extracts spatial dependencies from X_t using the dynamic A_t:

$$Z_t = \widetilde{D}_t^{-1/2}\tilde{A}_t\widetilde{D}_t^{-1/2}X_tW_{\text{GCN}} \quad (7)$$

where $\tilde{A}_t = A_t + I_V$ adds self-connections, $\widetilde{D}_t$ is the degree matrix, and W_{GCN} are learnable weights. A normalization layer stabilizes training, producing spatially filtered features Z_t that encode latent correlations among sensors.

4) Unified spatio-temporal formulation: The LT-GSSM integrates both modules within a unified probabilistic transition:

$$h_t = TCN(GCN(X_t, A_t)) + w_t, w_t \sim \mathcal{N}(0, \sigma_t^2 I) \quad (8)$$

and the observation equation:

$$\hat{y}_t = Hh_t + v_t, v_t \sim \mathcal{N}(0, \tau_t^2 I) \quad (10)$$

This formulation fuses dynamic spatial reasoning and nonlinear temporal modeling while explicitly quantifying uncertainty, yielding a robust framework for spatio-temporal forecasting under noisy and evolving conditions.

IV. Results

A. Dataset Summary

We evaluated the proposed LT-GSSM and all baseline models on both synthetic (SUMO) and real-world (PeMS) datasets, which differ in scale, variability, and network complexity.

The SUMO dataset provides a controlled environment to systematically assess robustness to noise and sensor sparsity, while PeMS datasets reflect real-world traffic dynamics across large urban regions. This dual setup ensures that robust trends observed in simulation can be validated under realistic, heterogeneous conditions.

All datasets span 2–4 months at 5-minute intervals, with traffic flow as the key variable. Table I summarizes the main characteristics of the synthetic (SUMO) and real-world (PeMS) datasets used in this study, including network size, temporal resolution, and observation period.

TABLE I. Summary of Datasets Used in This Study

Dataset	Nodes	Time Steps	Period
PeMSD03	358	26,208	09/2018–11/2018
PeMSD04	307	16,992	01/2018–02/2018
PeMSD07	883	28,224	05/2017–08/2017
PeMSD08	170	17,856	07/2016–08/2016
SUMO	127	16,992	Simulated (2 months)

B. Model Selection and Architectural Variants

We compared LT-GSSM against a broad spectrum of spatio-temporal architectures, ensuring fair and representative evaluation:

- Classical temporal baselines: ARIMA, VAR, and BiLSTM.
- Hybrid sequential models: LSTM–CNN, LSTM–GCN (T-GCN), and GRU-based variants.
- Convolutional models: 3D CNNs capturing local spatio-temporal dependencies.
- Attention and transformer architectures: ST-Transformer, attention-based GCNs.
- Diffusion and robustness models: DCRNN and RT-GCN, integrating diffusion or Gaussian convolutions for stability.
- Adaptive graph model: AGCRN as a dynamic-graph baseline using learnable node embeddings.
- Proposed model: LT-GSSM introduces a latent, probabilistic topology derived from hidden states, allowing state-driven graph evolution rather than deterministic embedding updates.

All baselines were re-implemented according to their original papers, with minimal adaptations for uniform preprocessing and input size. This guarantees that performance differences stem from model design, not implementation bias.

C. Experimental Framework and Hyperparameter Sensitivity

All models were trained under identical preprocessing, input windows (10 time steps), and forecast horizons (10 time steps, ≈50 minutes) to ensure fair comparison. Hyperparameters for each baseline—such as hidden dimensions, learning rates, and kernel sizes—were initialized following their original publications and fine-tuned within reported ranges. The LT-GSSM used a hidden size of 64–128, learning rate 0.0005–0.005, and learnable noise parameters to capture both process and observation uncertainty. This consistent setup guarantees that observed performance differences reflect architectural robustness rather than arbitrary tuning.

The following experimental settings and sensitivity analyses were considered to ensure fair comparison and to assess the robustness of LT-GSSM:

- We adopted a forecasting horizon of 10 time steps (≈ 50 minutes), a common setting in prior traffic studies [32 − 35].
- A hyperparameter sensitivity analysis was conducted exclusively on LT-GSSM, focusing on the hidden state dimension and the learnable noise parameter, as both directly affect model capacity and robustness.
- Hidden dimensions from 64 to 128 yielded the most stable results: smaller sizes caused underfitting, whereas larger ones amplified noise and led to overfitting. For the noise parameter, learnable variance consistently converged near $\sigma^2 \approx 0.2$, outperforming fixed settings.

- Low fixed values made the model too rigid, while high ones introduced instability. The adaptive formulation improved convergence and reduced RMSE by about 1–2 %, confirming that dynamic noise learning enhances robustness under noisy or incomplete data.

Fig. 5 summarizes these trends, showing steady performance for hidden sizes 64–128 and optimal stability when the learnable noise variance converges around 0.2.

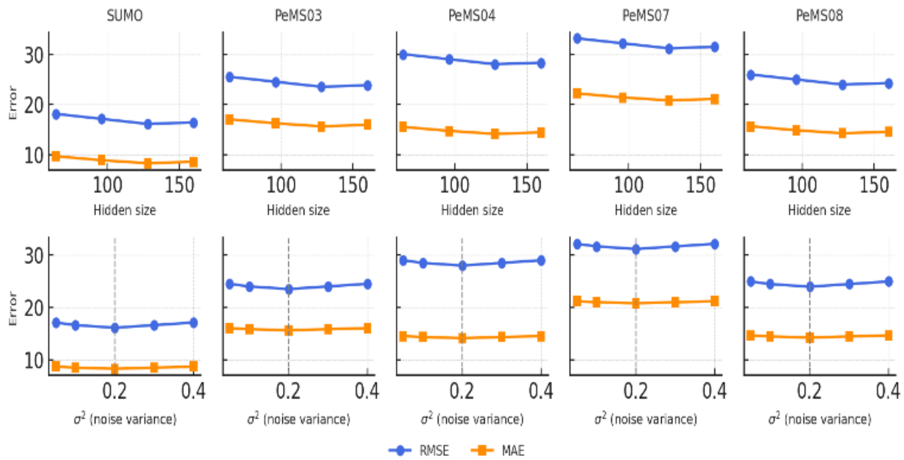


Fig. 5. Sensitivity of LT-GSSM to hidden state dimension (top) and noise variance σ^2 (bottom) across five datasets.

D. Baseline Performance Comparison

Table II reports the average results over seven runs with different random seeds. Variance remained below 2 %, confirming the stability and reproducibility of all experiments, and paired t-tests showed that LT-GSSM's improvements are statistically significant ($p < 0.05$).

Across both SUMO and PeMS datasets, LT-GSSM consistently achieves the lowest error among all models. Compared to classical temporal baselines (ARIMA, VAR), it reduces RMSE by 35–40 % (≈ 5.8 points on SUMO, ≈ 6.2 on PeMSD04). While recurrent models like BiLSTM (avg RMSE ≈ 22.1) perform better, they remain limited by the absence of spatial modeling. Hybrid spatio-temporal models (LSTM-GCN, GCN-GRU) further cut RMSE by ≈ 20 % on average, confirming the benefit of explicit spatial reasoning. 3D-CNN and LSTM-CNN improve temporal baselines but generalize poorly on irregular networks. Attention-based models show mixed behavior: ST-Transformer performs well on large datasets but underutilizes short input windows, while Attention-GCN offers only modest gains. Robust baselines like RT-GCN and AGCRN deliver strong accuracy under normal conditions but slightly trail LT-GSSM, especially on SUMO and PeMS08.

Overall, LT-GSSM's integration of a probabilistic state-space formulation with dynamic latent topology learning yields the best accuracy and stability across all settings, outperforming recent robust graph models while remaining computationally efficient. This establishes LT-GSSM as a strong benchmark for reliable spatio-temporal forecasting under noisy or incomplete data.

TABLE II. SUMMARY OF BASELINE MODELS AND EXPERIMENTAL

Model	SUMO RMSE	SUMO MAE	PeMS03 RMSE	PeMS03 MAE	PeMS04 RMSE	PeMS04 MAE	PeMS07 RMSE	PeMS07 MAE	PeMS08 RMSE	PeMS08 MAE
ARIMA	35.23	22.57	46.80	31.26	45.32	31.42	50.98	35.16	40.32	29.64
VAR	24.29	13.80	38.26	23.28	38.61	23.19	69.75	47.14	28.12	20.97
BiLSTM	20.29	11.76	31.09	19.13	36.87	22.72	43.12	25.90	31.39	22.12
GCN-GRU	16.67	8.93	26.52	15.67	28.48	15.91	33.15	22.24	24.73	14.15
LSTM-GCN	16.72	8.98	26.39	15.42	28.45	15.76	32.40	21.19	24.70	14.25
ST-Transformer	17. 98	9.31	24.16	14.11	30.74	16.32	35.23	23.40	24.95	14.02
Attention GCN	16.68	8.91	25.06	14.55	28.49	15.54	32.78	21.09	25.06	15.19
LSTM-CNN	17.73	9.83	29.15	17.38	28.72	15.5	36.93	22.80	27.76	15.9
3D-CNN	19. 56	10.1	30.33	17.89	30.94	17.32	38.64	24.18	28.34	17.63
STGCN	19. 99	9.26	28.17	19.95	30.23	16.45	38.12	24.45	26.91	15. 24
DCRNN	17.3	9.26	27.89	18.23	30.68	16.6	34.57	22.80	24.3	15.34
RT-GCN	16.90	8.90	26.80	16.25	28.55	15.60	31.40	21.40	24.80	14.40
AGCRN	16.73	9.11	26.82	16.45	29.72	16.93	31.28	22.51	24.76	15.52
LTGSSM	16.11	8.27	23.51	15.63	28.01	14.12	31.17	21.49	23.96	14.24

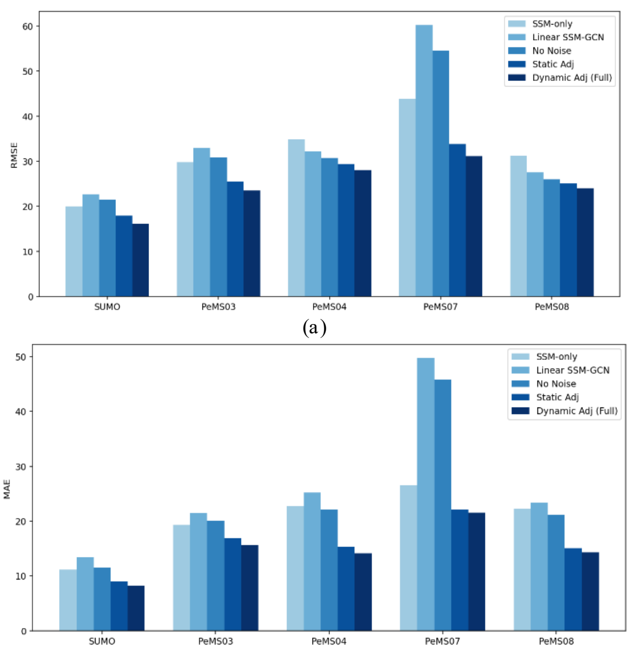
E. Ablation Study

To assess the contribution of each module within LT-GSSM, four ablation variants were evaluated: 1) a state-space only configuration (without GCN), 2) a classical SSM-GCN using linear temporal transitions instead of a TCN, 3) a noise-free variant without explicit process and observation noise, and 4) a comparison between static and dynamic adjacency learning. Fig. 6 summarizes the results across all datasets.

The SSM-only configuration already achieved competitive accuracy, outperforming purely temporal models such as LSTM and GRU on several datasets. This confirms that the recursive state-space formulation and process noise terms provide strong temporal modeling capacity, even without explicit spatial reasoning.

Replacing the nonlinear TCN with a linear update led to a consistent increase in error (≈ 20% higher RMSE on average), indicating that nonlinear temporal transitions are crucial for capturing complex latent traffic dynamics. Similarly, removing explicit noise modeling degraded robustness under noisy or incomplete conditions, with the full LT-GSSM achieving up to 25% lower RMSE compared to the deterministic version.

Finally, dynamic adjacency learning further improved performance relative to static topologies, reducing RMSE by approximately 10% across datasets. This confirms that allowing the graph structure to evolve with latent state dynamics better captures non-stationary spatial correlations such as congestion propagation or sensor drift.



(a)

(b)

Fig. 6. RMSE and MAE comparison of LT-GSSM ablation variants across SUMO and PeMS datasets.

Fig. 6 and Fig. 7 illustrate these trends: Fig. 6 shows RMSE and MAE across ablation variants, while Fig. 7 visualizes relative performance gains over the static baseline. Overall, each component—state-space recursion, nonlinear transition, explicit noise modeling, and dynamic topology—contributes significantly to the robustness and accuracy of LT-GSSM, with the complete model consistently achieving the best balance across datasets.

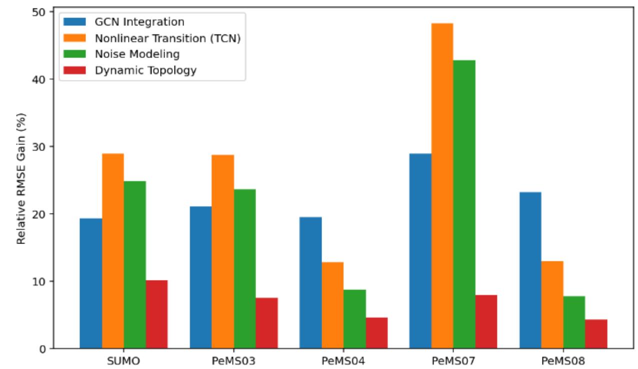


Fig. 7. Relative performance gain (%) of LT-GSSM components compared to the static baseline.

V. Discussion

A. Effect of Sensor Placement and Noise on Spatiotemporal Prediction

We used the custom SUMO network to evaluate how sensor placement and noise affect forecasting robustness. Two practical scenarios were simulated: 1) prioritizing sensors on major arterial roads, and 2) reducing overall sensor coverage to 50 % and 20 %.

Concentrating sensors along main corridors improved accuracy by capturing structured, correlated traffic flows, whereas reduced coverage increased errors across all models. Nevertheless, LT-GSSM remained the most stable, confirming that its dynamic adjacency learning effectively adapts spatial correlations even when observations are sparse.

By updating its graph topology from latent state representations, the model can infer missing spatial dependencies, rather than relying solely on static connectivity. Comparable resilience was observed in RT-GCN and AGCRN, yet LT-GSSM showed the smallest degradation under both sensor loss and noisy data, demonstrating the strength of its joint spatio-temporal uncertainty modeling.

Fig. 8 illustrates the impact of sensor coverage on prediction accuracy.

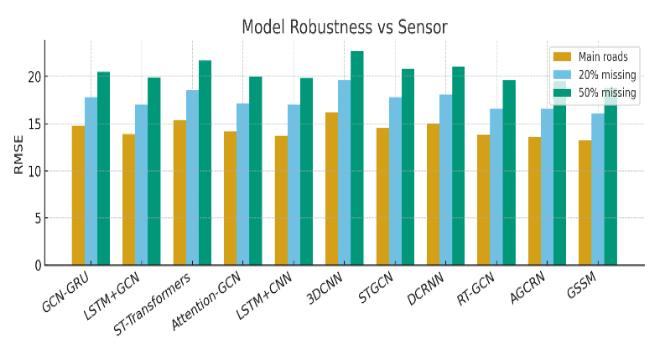


Fig. 8. Impact of sensor coverage on the performance of spatiotemporal prediction models.

B. Robustness Under Gaussian Noise

Beyond spatial sparsity, robustness was also tested under Gaussian sensor noise injected into traffic flow data at three levels (5 %, 15 %, 30 %). Noise was applied selectively to subsets of sensors, simulating real detector failures. As expected, higher noise levels increased errors for all models (Table III), but LT-GSSM exhibited only a ≈ 2–7 % RMSE increase at 30 % noise—significantly lower than competing architectures.

This resilience stems from its explicit uncertainty modeling: the transition function captures latent dynamic fluctuations, while the observation model accounts for measurement errors, allowing LT-GSSM to maintain stable performance even under severe data corruption.

TABLE III. IMPACT OF GAUSSIAN NOISE LEVELS ON MODEL PREDICTION ERRORS

Model	SUMO (5%)	15%	30%	PeMS08 (5%)	15%	30%
GCN-GRU	17.5	18.0	18.4	25.2	25.7	26.1
LSTM-GCN	17.1	17.8	18.2	25.1	25.6	26.0
ST-Transformer	17.6	18.3	18.8	25.5	26.1	26.6
Attention-GCN	17.2	17.8	18.3	25.5	25.9	26.4
LSTM-CNN	17.7	18.4	18.9	28.2	28.8	29.3
3D-CNN	18.1	18.7	19.2	28.9	29.4	29.9
STGCN	17.5	18.2	18.7	27.3	27.8	28.3
DCRNN	17.3	17.9	18.3	24.8	25.3	25.8
RT-GCN	16.3	17.1	17.9	25.3	25.9	26.5
AGCRN	17.4	17.6	18.2	25.0	25.4	25.9
LT-GSSM (ours)	16.2	16.8	17.4	24.2	24.6	25.1

C. Computational Complexity and Execution Time

LT-GSSM maintains complexity comparable to standard spatio-temporal GNNs. The TCN-based temporal layer replaces recurrent cells, reducing sequential dependencies and allowing efficient parallelization with per-step cost $O(H^2 \times L_t)$ (with $L_t \leq 3$). Dynamic adjacency learning adds a manageable $O(V^2)$term, negligible for medium-sized networks (hundreds of sensors). Overall, LT-GSSM offers a better trade-off between robustness and cost than diffusion- or attention-based models, which scale as $O(T^2 d)$.To validate this analysis, we measured execution times under identical hardware and software conditions (Intel i7-10700 CPU, 32 GB RAM, NVIDIA GTX 1660 GPU, PyTorch 2.0, CUDA 11.8).

Using a batch size of 32, input length T=12, and hidden size H=64, LT-GSSM consistently trained 10–20% faster per epoch than recurrent baselines such as LSTM-GCN, GRU-GCN, and DCRNN on the PeMS04 and PeMS08 datasets.

These gains arise mainly from the parallelizable temporal convolutions and the absence of recurrent backpropagation.

Compared to heavier architectures such as ST-Transformer, Attention-GCN, or 3D-CNN, LT-GSSM maintained a lightweight execution profile, ranking among the most efficient models overall.

Although inference latency was not extensively benchmarked, average forward-pass times remained below 100 ms. per prediction step, indicating that LT-GSSM can operate effectively in near real-time forecasting scenarios without sacrificing accuracy or stability. Table IV reports the average training time per epoch for LT-GSSM and competing spatio-temporal models under identical hardware and software settings.

TABLE IV. AVERAGE TRAINING TIME PER EPOCH ACROSS SPATIOTEMPORAL MODELS

Model	Avg. Time per Epoch (s)
ST-GCN (parallel)	235.2
ST-Transformer	168.7
Attention-GCN	144.5
3D CNN	129.1
DCRNN	117.3
RT-GCN	110.5
AGCRN	104.2
LTGSSM (ours)	99.8
LSTM-GCN	96.7
GRU-GCN	96.2
LSTM-CNN	70.3

VI. CONCLUSION

This paper presented the Latent Topology Graph State-Space Model (LT-GSSM), a probabilistic spatio-temporal framework that dynamically learns graph structures from latent representations rather than relying on fixed connectivity. By treating topology evolution as part of the latent state dynamics, LT-GSSM reframes graph adaptation itself as an uncertain latent process, enabling robust modeling of non-stationary spatial dependencies under noise and missing data.

The main scientific contribution of this work lies in explicitly modeling spatial structure uncertainty within a state-space formulation, rather than relying on deterministic or heuristically updated graph topologies. This perspective provides a principled explanation for robustness in spatio-temporal forecasting, complementing existing dynamic-graph architectures.

Experiments on SUMO and PeMS datasets confirm that this formulation yields stable and adaptive forecasting performance even with reduced sensor coverage or high noise levels, while maintaining a computational cost comparable to standard GCN-based hybrids (e.g., LSTM-GCN, AGCRN), making it suitable for practical deployment in intelligent transportation systems.

Despite these advantages, dynamic adjacency learning introduces a quadratic dependency in the number of nodes, which may require sparsification strategies for very large-scale networks. Robustness was evaluated under commonly adopted noise and missing-data settings, and further extensions are left for future work.

Beyond traffic forecasting, the proposed modeling paradigm is applicable to a broader class of spatio-temporal problems involving evolving relational structures, such as sensor networks, urban monitoring, and environmental dynamics. Future work will investigate scalability improvements, alternative uncertainty models, and extensions to other spatio-temporal domains.

ACKNOWLEDGMENT

The author would like to thank her doctoral supervisor for the opportunity to conduct this research within the PhD program.

REFERENCES

[1] L. Peng et al., “An Overview Based on the Overall Architecture of Traffic Forecasting,” Data Science and Engineering, vol. 9, no. 3, pp. 341–359, 2024.

[2] S. Wei et al., “Traffic Flow Prediction with Multi-Feature Spatio-Temporal Coupling Based on Peak Time Embedding,” The Journal of Supercomputing, vol. 80, no. 16, pp. 23442–23470, 2024.

[3] V. Singh, S. K. Sahana, and V. Bhattacharjee, “Integrated Spatio-Temporal Graph Neural Network for Traffic Forecasting,” Applied Sciences, vol. 14, 2024. DOI: 10.3390/app142411477.

[4] A. A. Karim and N. Nower, “Probabilistic Spatio-Temporal Graph Convolutional Network for Traffic Forecasting,” Applied Intelligence, vol. 54, no. 11, pp. 7070–7085, 2024.

[5] Y. Li et al., “Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting,” in Proc. Int. Conf. Learning Representations (ICLR), 2018.

[6] L. Zhao et al., “T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction,” IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 9, pp. 3848–3858, 2020.

[7] D. Li and J. Lasenby, “Spatiotemporal Attention-Based Graph Convolution Network for Segment-Level Traffic Prediction,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 8337–8345, 2022.

[8] L. Xiao and H. Chen, “Spatio-Temporal Transformer Graph Network for Traffic Flow Forecasting,” in Proc. 3rd Int. Joint Conf. Information and Communication Engineering (JCICE), 2024.

[9] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, “Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting,” in Advances in Neural Information Processing Systems (NeurIPS 2020), vol. 33, pp. 17804–17815, 2020. DOI: 10.48550/arXiv.2007.02842.

[10] Y. Kamarianakis and P. Prastacos, “Forecasting Traffic Flow Conditions in an Urban Network: Comparison of Multivariate and Univariate Approaches,” Transportation Research Record: Journal of the Transportation Research Board, vol. 1857, no. 1, pp. 74–84, 2003.

[11] Z. Lu et al., “Integrating Granger Causality and Vector Auto-Regression for Traffic Prediction of Large-Scale WLANs,” KSII Transactions on Internet and Information Systems (TIIS), vol. 10, no. 1, pp. 136–151, 2016.

[12] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] R. Fu, Z. Zhang, and L. Li, “Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction,” in Proc. 31st Youth Academic Annual Conf. Chinese Association of Automation (YAC), 2016.

[14] S. Natha et al., “Improving Traffic Surveillance: Deep Learning Approach for Road Anomaly Detection in Videos,” in Proc. IEEE 3rd Int. Conf. Computing and Machine Intelligence (ICMI), 2024.

[15] M. Lv et al., “Temporal Multi-Graph Convolutional Network for Traffic Flow Prediction,” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 6, pp. 3337–3348, 2021.

[16] Y. Liu et al., “RT-GCN: Gaussian-Based Spatiotemporal Graph Convolutional Network for Robust Traffic Prediction,” Information Fusion, vol. 102, p. 102078, 2024.

[17] G. Revach, D. Soudry, R. Giryes, and Y. C. Eldar, “KalmanNet: Neural Network-Aided Kalman Filtering for Partially Known Dynamics,” IEEE Transactions on Signal Processing, vol. 70, pp. 1532–1547, 2022. DOI: 10.1109/TSP.2022.3149823.

[18] R. G. Krishnan, U. Shalit, and D. Sontag, “Structured Inference Networks for Nonlinear State Space Models,” in Proc. AAAI Conf. Artificial Intelligence (AAAI 2017), pp. 2101–2109, 2017. arXiv:1609.09869.

[19] M. Karl, M. Soelch, J. Bayer, and P. van der Smagt, “Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data,” in Proc. Int. Conf. Learning Representations (ICLR 2017), 2017. arXiv:1605.06432.

[20] Z. Zhang, P. Cui, and W. Zhu, “Probabilistic Graph Neural Networks,” Communications of the ACM, vol. 63, no. 8, pp. 107–113, 2020. DOI: 10.1145/3363574.

[21] D. Valsesia, G. Fracastoro, and E. Magli, “Graph Neural Networks for Graph Signal Denoising,” IEEE Transactions on Signal Processing, vol. 67, no. 23, pp. 6176–6189, 2019. DOI: 10.1109/TSP.2019.2945544.

[22] P. A. Lopez et al., “Microscopic Traffic Simulation Using SUMO,” in Proc. 21st IEEE Int. Conf. Intelligent Transportation Systems (ITSC), Maui, HI, USA, 2018.

[23] A. Wegener et al., “TraCI: An Interface for Coupling Road Traffic and Network Simulators,” in Proc. 11th Communications and Networking Simulation Symposium (CNS’08), Ottawa, Canada, pp. 155–163, 2008.

[24] I. Juva et al., “Traffic Characterization for Traffic Engineering Purposes: Analysis of Funet Data,” in Proc. Next Generation Internet Networks (NGIN), 2005.

[25] G. Beliakov et al., “Measuring Traffic Congestion: An Approach Based on Learning Weighted Inequality, Spread and Aggregation Indices from Comparison Data,” Applied Soft Computing, vol. 67, pp. 910–919, 2018.

[26] S. Guo et al., “Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting,” in Proc. 33rd AAAI Conf. Artificial Intelligence (AAAI 2019), Honolulu, HI, USA, 2019.

[27] A. Loder et al., “Understanding Traffic Capacity of Urban Networks,” Scientific Reports, vol. 9, no. 1, p. 16283, 2019.

[28] Q. Wu et al., “Tensor-Based Recurrent Neural Network and Multi-Modal Prediction with Its Applications in Traffic Network Management,” IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 780–792, 2021.

[29] J. Xu et al., “Spatio-Temporal Multi-Task Learning via Tensor Decomposition,” IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 6, pp. 2764–2775, 2021.

[30] B. Yu, Y. Lee, and K. Sohn, “Forecasting Road Traffic Speeds by Considering Area-Wide Spatio-Temporal Dependencies Based on a Graph Convolutional Neural Network (GCN),” Transportation Research Part C: Emerging Technologies, vol. 114, pp. 189–204, 2020.

[31] A. Vaswani et al., “Attention Is All You Need,” in Advances in Neural Information Processing Systems (NeurIPS 2017), Curran Associates Inc., Long Beach, CA, USA, 2017.

[32] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” arXiv preprint arXiv:1609.02907, 2017.

[33] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, OTexts, 2018.

[34] C. Brooks, “The Impact of Seasonality on Forecasting,” International Journal of Forecasting, vol. 23, no. 1, pp. 210–226, 2007.

[35] G. Bontempi, C. Hodges, and N. D. Lawrence, “Forecasting with a Compounding Neural Network,” in Proc. Int. Conf. Machine Learning and Applications (ICMLA), 2013.