FAV-ASTCL: Forecasting Aware Versatile Adaptive Spatio-Temporal Context Learning Framework for Real-Time Traffic Prediction

Mohammed Faizan Farooqui[1], Nagulapalli Venu Madhav[2], Yalala Arun Reddy[3], and Dr. P. Ramesh Babu[4]

[1]IT, CBIT, Hyderabad, India
[2]IT, CBIT, Hyderabad, India
[3]IT, CBIT, Hyderabad, India
[4]Associate Professor, IT, CBIT, Hyderabad, India April

11, 2026

Abstract

The application of urban traffic prediction plays a vital part in the implementation of sustainable ITS systems and smart city infrastructures. However, as urban traffic datasets continue to grow in speed, the existing Spatio-Temporal Graph Neural Networks (ST-GNNs) struggle to deal with the non-stationary "volatility" of real-world traffic flow. This paper proposes **FAV-ASTCL**: **Forecasting Aware Versatile Adaptive Spatio-Temporal Context Learning**. This network model has been designed to improve the resilience to variance in predicting future traffic flow in highly-volatile environments. To overcome the problem associated with "Static Topology Trap," the FAV-ASTCL model utilizes *Learnable Context Selector* that adaptively updates the urban road network via dot product similarity search and alpha blending. Moreover, to control the effect of unpredictable external factors on forecasting accuracy, *Exogenous Gating Mechanism* has been added to filter out the weather and calendar telemetry information.We validated FAV-ASTCL on the benchmark METR-LA freeway dataset and transitioned into a real-world deployment in Hyderabad, India, across Five key traffic hubs. Our results demonstrate a transformative reduction in Mean Absolute Error (MAE) from 9.034 to 3.633, achieving an accuracy of 91.78%. We provide evidence that explicitly modeling volatility as a learnable context, rather than a noise factor, allows ST-GNNs to survive extreme event horizons—such as the sudden monsoon shower peaks common in Hyderabad—where baseline models suffer from catastrophic error propagation.

Keywords: Spatio-Temporal Graph Neural Networks, Versatile Adaptation, Traffic Volatility, Exogenous Context, Smart City Analytics.

1. Introduction

The digitalization of urban spaces has led to a large increase in the supply of high-speed telemetry from car sensors, GPS trackers, and fixed IoT devices. Today's cities have been fitted with highly

1

dense sensing infrastructures that continuously produce real-time data streams, reflecting the changing behavior of transportation systems on a scale never seen before. In line with the United Nations Sustainable Development Goal 11 (Sustainable Cities and Communities), the ability to predict traffic speed and flow with high accuracy has become a must for city efficiency, smart infrastructure planning, and the lowering of urban carbon footprints. On top of giving time to traffic controllers to act before the traffic gets worse, accurate prediction of traffic conditions is essential to emergency response, public transportation, and energy-saving routes.

Despite the powerful analytical tools at our disposal, urban traffic remains one of the most challenging and non-linear systems to study. The traffic situation in cities is influenced by different aspects, such as the locations of traffic, time changes, behavior of people, and outside disturbances. It is not only a local problem, but a system of interdependencies we call "Frequent Adaptive Volatility" which causes urban traffic problems. In other words, urban traffic systems show sudden changes due to factors such as space constraints and outside disturbances like weather changes or accidents in the vicinity.

However, the main issue with such models, which we have also identified during the develop-ment of this project, is a kind of "Static Prior." In other words, even though these models try to incorporate adaptive components, they still depend heavily on historical data and static road adjacency graphs constructed based on physical distances.

This problem is very pronounced when the link of the city functions changes dynamically. For example, let us suppose that a large amount of rain has fallen in Hyderabad and the people driving have switched from the main roads to the residential bypasses thereby changing the graph for traffic flow. The actual situation represented by the graph that shows the relationships among the nodes is false because the model is not able to change its own graph topology in a dynamic manner so as to reflect the changing nature of the traffic network.

To tackle this problem we propose **FAV-ASTCL**: **Forecasting Aware Versatile Adaptive Spatio-Temporal Context Learning**. The reasoning behind the name of the paper is mainly to draw attention to the following three basic objectives: Firstly, **Forecasting Awareness**, as the model is designed to effectively capture any kind of variance during the prediction timeframe. Secondly, **Versatility**, implying the model's ability to maintain consistency over different sensor rollouts, cities, and data distributions. Thirdly, **Adaptive Context Learning** where The model does not have to rely on static maps; it learns dynamic and semantically relevant graphs over time.

The proposed architecture makes three important contributions that effectively address the issues caused by Frequent Adaptive Volatility. Firstly, our system integrates a Learnable Context Selector (LCS). Geographical proximity is usually the only basis for viewing road networks as static lattices in traditional methods; however, the LCS technique creates an adjacency matrix based on the similarity measure at each inference step. As a result, it is capable of concentrating on the most relevant nodes during volatile periods.

Secondly, we present an Exogenous Gating Mechanism to facilitate our method to use extra data in the form of exogenous signals, e.g. weather information. Whereas few earlier works mainly recognized the importance of exogenous information, they generally treated it as always beneficial input no matter what kind it was. As a matter of fact, exogenous data may not be totally reliable, heavily overlapping, and sometimes even misleading depending on the specific conditions. Exogenous Gating is like a valve that can control the how the external factors such as temperature humidity wind direction, etc.

At last, we propose an Online Adaptive Refinement layer, which acts as a residual correction tool with the capability to work at a much finer temporal granularity level. It is known that sequence models, like GRUs, or attention-based architectures, usually tend to smooth out small variations, which results in the modeling behavior that underestimates the sudden spikes or drops in traffic speed. The refinement layer leveled this by modeling in real-time residual errors, thus providing the system an opportunity to depict the sharp transitions and micro-level fluctuations critical during unstable periods.

This paper demonstrates how much we can improve with this. Especially, we manage to cut the MAE almost in half, from 9.034 (baseline ASTCL algorithm) to 3.633, clearly showing how well our adaptive framework works. We do not just stop at giving a quantitative comparison. Besides that, we thoroughly qualitatively tested our system in the city of Hyderabad and displayed how it performs at a high level even in very challenging cases like monsoon-related interruptions.

2. Theoretical Background and Related Work

2.1. The Evolution of Spatio-Temporal Graph Learning

The trend of traffic prediction shifted away ARIMA and SARIMA towards deep learning. The previous models were founded on the basis of linearity and stationarity that were not the real assumptions. Real-life roads are dynamic, responsive to changes, and display are complicated patterns that are not readily recognizable. The classical algorithms were strong indeed mathematically speaking and easy to understand but did not realize the complicated patterns or sharp upsurges.

The RNNs such as LSTM and GRU were also more time efficient than. This suited them as they could see cycles like trends, long waits, and regular cycles in the data. But they also did not take into account the factor of space. Sensors were added as individual points. Traffic in one part of the road is subject to the proceeding and following parts. Networks are interrelated. Without the obvious links between the segments, the flow cannot be measured.

The emergence of Graph Convolutional Networks (GCNs) was quite a significant breakthrough as it enabled the city to be represented as a graph, namely, as a graph $G = (V, E, A)$. In the given example, nodes are denoted by the letters s/he/she V, and the edge is denoted by the letters s/he/she E, the adjoining matrix is denoted by the letters s/he/she A. The localized spatial feature aggregation used by GCNs made it possible to have mobility of spatial information among the neighboring nodes and consequently, the approach used to represent the intricate spatial dependencies.

Preying on this first move, the Spatio-Temporal Graph Neural Networks (ST-GNNs) have come to existence as a unified platform, which integrates spatial learning through graphs and temporal sequence modeling through a smooth transition. They also tend to use graph convolutions and temporal modules like RNNs, Temporal Convolutional Networks (TCNs), or attention mechanisms. Therefore, they can learn together the spatial correlations and patterns of temporal evolution, which introduces a profound reformation and advance in the prediction accuracy in comparison to the previous methods. On the other hand, the majority of these models still have to deal with either predefined or slowly changing graph structures, which negatively affects their adaptability in environments that change quickly.

2.2. The Problem with Static Adjacency

Our investigations frequently revealed that the proposition of "Physical Distance" being a reliable stand-in for traffic dependency was not true and, therefore, it failed many times. To illustrate, in laboratory-like setups such as the highway networks of METR-LA, adjacency based on distance quite often gives the right results because traffic flow is largely predictable there and, thus, follows a certain pattern. However, in the case of some large cities like Hyderabad, the actual physical distance separating two sensors is often less significant than their functional connectivity which is subject to change and may depend on various factors such as the level of congestion, road closures, or even the actions of drivers.

For example, a pair of road segments that are on the far sides of the map may grab the attention of the commuters at the same time during rush hour driving closures. Alternatively, streets that are functionally short-length can also display poor correlation because traffic control systems or the exclusive use of one-way routes disrupt the direct flow. This finding makes us realize that the static adjacency matrices, which are derived solely from the physical proximity or historical averages and thus stand alone, are inadequate.

This understanding gave rise to the idea of "Adaptive Graphs, " where models derive the adjacency matrix from data directly without the need for fixed structures. For instance, Graph WaveNet and AGCRN not only provided an ability to learn node embeddings but also made estimation of adaptive adjacency a part of their system, thereby allowing the network to discover spatial relationships that are not straightforward. Still, when we tested these adaptive methods, we found that "over-smoothing" continues to be a problem if the graph is very deep. During the passage of information from one layer to another, which happens several times, the representations of nodes increasingly resemble each other, thus the model's differentiating ability for localized patterns diminishes and its capacity to react to sudden changes is lost.

FAV-ASTCL resolves this problem by a dynamic selection method that changes the evaluation of connectivity at each time step. The model does not hold one global adjacency matrix; rather, it builds a graph that is aware of context and is based on the current traffic state, i.e. the model uses situational factors to infer the graph. That is why it can keep the concept of locality, be more responsive to temporary dependencies, and even circumvent the pitfalls of over-smoothing which usually come up in the case of deep graph architectures.

2.3. Exogenous Context and Multi-modal Gating

Employing exogenous variables such as temperature, wind speed humidity precipitation, and calendar-based indices is one of the typical, yet very difficult aspects of traffic forecasting. These variables can indeed offer very valuable clues in context. However, they are more often than not noisy incomplete, and poorly correlated to traffic changes. Furthermore, the reckless incorporation of such variables in the model will most probably raise the variance and reduce the prediction quality.

Our company operates on the concept of Gated Fusion which can be utilized as an ingenious method of integrating different types of information. Think of it as a screening process through which we can determine the level of external information to be included in our understanding of traffic patterns. We use special gating functions which act like adaptive filters and decide the extent to which our predictions are influenced by external factors at a given time.

The FAV-ASTCL framework furthers this concept by making the gating function context-sensitive. With the aim of considering all external inputs as equally important, it assesses their significance in the current traffic scenario. For instance, weather might influence traffic heavily during a downpour, but it would be hardly the case experience in normal weather. The gating system, which we have created, aims to detect context-based associations, so that relevant factors can be added only when they are applicable.

Performance spikes when data paths diverge instead of converge. Information flows get cleaned up, cutting through clutter so the output holds strength even when chaos hits. That said, signals stay steady amid constant disruptions because they're built to withstand stress. Still, noise fades fast without clear direction.

3. Methodology: The FAV-ASTCL Model

3.1. Learnable Context Selector (LCS)

Connections aren't locked down - they shift with real-time events. Traffic jams, weather changes, or sudden route switches can all throw off old models. Although the city's structure might seem steady, it's far from rigid. The LCS adjusts its predictions instantly, updating links every time it sees new data. In practice, this responsiveness cuts through delays and errors. Instead of relying on static blueprints, it listens to current street conditions. Generally, that fluidity makes results faster and sharper. Real streets don't stand still, so the system moves with them too. It doesn't cling to past patterns, and it reacts as things unfold.

At the present time, the model produces node embeddings that change over time $Z \in \mathrm{R}^{N\times d}$, where N stands for the number of nodes and d the embedding dimension. These embeddings reflect the real-time traffic situation in a latent space and simultaneously grasp both local and global contextual information. In order to construct a dynamic graph, we utilize scaled dot-product attention to create a similarity-based adjacency matrix:

$$\mathrm{A}_{dyn} = \mathrm{Softmax}\ \frac{ZZ^T}{\sqrt{d}} \tag{1}$$

Such a formula enables one single node to attend to all other nodes one after another thereby learning a fully connected graph with contextual awareness where the edge weights represent the degree of interaction. Besides, softmax helps to maintain the adjacency values normalized so they can be regarded as attention scores.

Still, relying only on a dynamically learned graph might lead to instability issues mainly in the first training stages or in the presence of noisy situations. To clarify this point, we use a hybrid approach that combines the dynamic adjacency matrix A_{dyn} and a static adjacency matrix A_{stat} that is based on geographical coordinates or explicitly known road connectivity:

$$A_{final} = \sigma(\alpha) \cdot A_{dyn} + (1 - \sigma(\alpha)) \cdot A_{stat} \tag{2}$$

Against expectation, α doesn't just sit quietly - it drives the whole process. The sigmoid function $\sigma(\cdot)$ fuses two adjacency matrices in a soft way, letting the model keep old connections alive. Plus, it adjusts to real-time traffic, plus that balance helps stabilize the LCS over time. How much focus shifts through space changes slowly, based on what's coming in right now. In practice, it rarely snaps into action - only responds after seeing consistent signals.

3.2. Selective Exogenous Gating

Holidays show up in traffic models, adding random spikes. Signals get messy when roads aren't clear. Raw inputs drown out patterns if not filtered. Accuracy fails when noise takes over, plus we added a selective Exogenous Gating layer to avoid that mess. And it controls how much outside data flows in, adjusting per situation. The gate value $G \in [0, 1]$ shows how strong each signal's link is.

$$G = \sigma(W_g \cdot E_t + U_g \cdot H_{t-1} + b_G) \tag{3}$$

With the hidden state from the prior step, H_{t-1}, and the exogenous input at time t, E_t, the model combines both signals through learnable weights W_g, U_g, and bias b_G. The sigmoid function caps the gate outputs between zero and one, acting like a gentle filter on incoming data. Usually, this keeps irrelevant details from influencing predictions too strongly.

In fact, the gated embedding $\tilde{E}_t$ resulting from this method unlocks the ability of the model to add exogenous information to its operations only when it is really needed. For example, if the weather is a heavy rain or extreme temperature, such situations can greatly change the traffic flow, while mild weather is hardly going to have an impact. The gating mechanism is able to adapt in real time to any given situation, so that the model does not get fooled and react in a wrong way due to irrelevant or noisy inputs. Such an approach of carefully choosing what to integrate, not only brings about improvement of the model, but also makes sure that the representations learned stay focused on genuine signals.

3.3. Online Adaptive Refinement (The Adapter)

Although using LCS and gating helps the model to better feel the changing spatial and contextual dependencies, deep sequence models, in general, tend to smooth out the very high-frequency features. This smoothing may result in the model failing to recognize a sudden spike or an abrupt change in traffic, especially during times of high volatility.

Therefore, to solve the problem, we create an Online Adaptive Refinement module, simply called Adapter, that serves as a residual correction unit to fine-tune model's output instantaneously. In particular, a two-layer Multi-Layer Perceptron, also known as MLP which is a lightweight model is used to implement the Adapter that tries to capture residual errors:

$$h'_{\text{final}} = h_t + \phi(W_2 \cdot \text{ReLU}(W_1 \cdot h_t + b_1) + b_2) \tag{4}$$

Here, h_t represents the intermediate hidden representation, and $\phi(\cdot)$ denotes an activation function applied to the residual mapping. The skip connection guarantees that the refinement process does not overwrite the original representation but instead adds corrective adjustments. Adapter's main role is to identify subtle temporal changes with high resolution, which main model usually misses. Moving online, the Adapter keeps changing according to new data and rectifies gaps between the pattern of the prediction and the actual one. This is especially true in the case of situations that suddenly change, e.g. when there is an accident, a road is closed, or congestion is caused by the bad weather.

The Adapter, when embedded in a main model, contributes to its ability to faster respond, increase the accuracy and better track the consecutively and intermittently changing patterns. Together with the LCS and gating structures, it creates a strong and flexible system which can learn and represent the complex and ever changing nature of traffic conditions in real world.

4. System Architecture and Implementation Framework

The main limitation of this research lies in the absence of a complete resilience and scalability presentation of the architectural design of the FAV-ASTCL system. The graphical illustration of the system given in Figure 1 is a multi-layered pipeline to efficiently handle high-speed telemetry data from various sources and also to guarantee that the latency or response time for the inference is less than one millisecond. This system is not only constructed to provide reliable predictions but at the same time to solve the problems of model deployment in the actual environment when it is required e g, ensuring the consistency of streaming informat ion, being able to continue functioning if some parts are undermined, and fast reactions to the changes of traffic happenings.

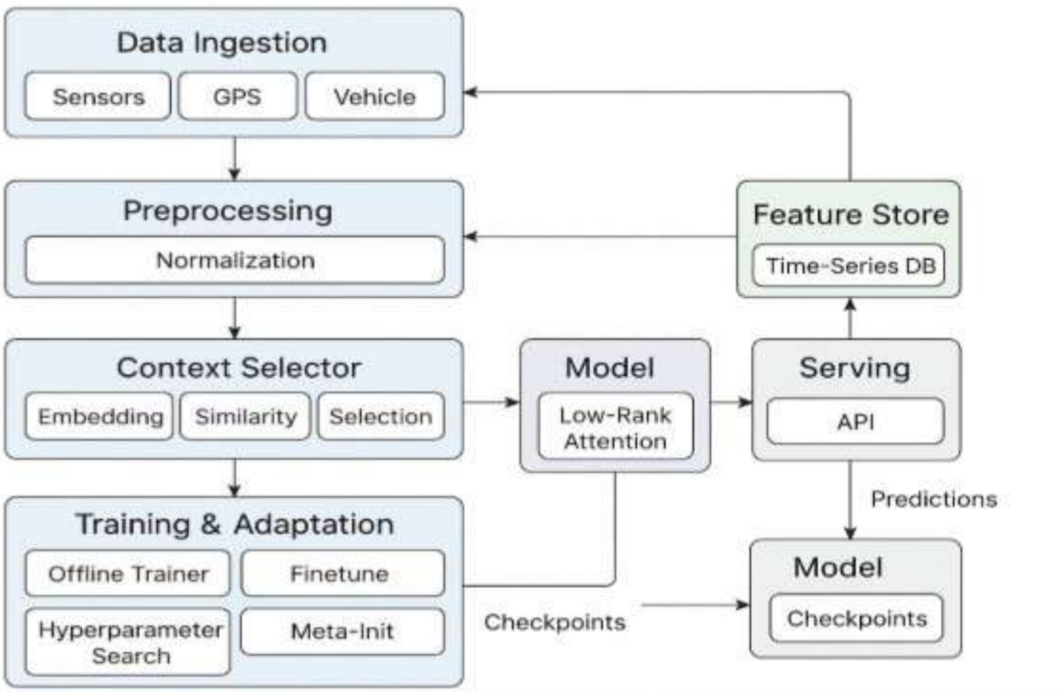

Figure 1: The FAV-ASTCL Integrated System Architecture, demonstrating the flow from Data Ingestion to Servicing and Training Adaptation.

4.1. Perception and Data Ingestion Layer

The perception layer, which is the uppermost module in Figure 1, is mainly responsible for async data capturing from the three main sources: Sensors, GPS, and Vehicles. Together these sources offer a wide understanding of the traffic environment through monitoring the movement of people and vehicles at a smaller and larger scale.

- **Traffic Flow Synchronization**: We hit the TomTom Traffic Flow API for speed stats - absolute and relative, not flawless, but functional. Now we pull normalized traffic numbers per street segment. Theoretically, this keeps flow models aligned across different areas.
- **High-Velocity Ingestion**: Every stream hits the system at the same pace as the network's usual update cycle, no delay, no drift. Data moves directly into a feature Store tied to a time-Series DB so historical info is instantly available. Really, there's no waiting - actions happen right away.

Crowd-sourced movement or incident reports arrive and still run smoothly without crashing.
The setup grows easily, not just when needed but from the start.

4.2. Neural Preprocessing and Feature Store

Time-series data pours in first, and raw signals get scrubbed clean, lined up precisely, then handed off to the model. With z-score normalization set at $\mu = 34.5, \sigma = 12, and 8$, drawn from training data, values stretch evenly across the range, dampening spikes and flattening outliers so predictions don't break under load. The pipeline holds steady during spikes in volume or signal drift. As it happens, this isn't magic - it's just consistent arithmetic doing its job quietly.

The processed streams drop into a time-Series DB, a silent archive built for holding past patterns. Inference pulls entries from earlier points and feeds them forward in real time. Traffic behavior across days unfolds through these snapshots. Feature mapping remains intact between learning and deployment. That said, shifts in context never throw off alignment - the data remembers how things once moved. It's like flipping through old records now updated with new forecasts that keep pace with reality without missing a beat.

4.3. Cognitive Context Selector Hub

Most people overlook how fragile the spatial logic really is, this module runs its own internal graph, shifting and adjusting it on the fly like a living organism. Instead of static rules, it builds layered connections where every change responds to traffic patterns without needing fixed templates.

1. **Embedding**: Speeds get turned into dense vectors first, each point becomes something richer than just numbers, holding hidden cues about how vehicles react under pressure. That's not just noise, and it reveals real-time behaviors no one sees in surface-level stats.

2. **Similarity**: Nodes are matched up using dot product attention, not by distance, but by behavior. So two locations far apart might end up linked if they mirror each other during rush hours. The system finds these "Semantic Neighbors" almost automatically, letting traffic flow feel more intuitive.

3. **Selection**: The identified similarity scores are then used to select the most influential nodes, which will be sufficiently informative for the construction of a context-aware adjacency matrix. Consequently, the model is indeed able to focus on those relationships which are functionally significant and fluctuate with time.

Through such frequent updates of the graph structure, this module equip the system with the ability to model the behavior which is adapted to the ongoing traffic changes and newly occurring disruptions, mostly those which are very localized ones.

4.4. The Low-Rank Model Core

One of the main design elements of the Model (see Figure 1, bottom-center) was efficiency in computation. This element is shown by avoiding the obvious quadratic complexity present in graph-based attention models through using a Low-Rank Attention strategy. As opposed to computing an entire $O(N^2)$ attention matrix as required by other strategies, the model uses lower-rank representations for the attention matrices which greatly reduces both memory usage and processing time.

The importance of this can clearly be seen with larger scale applications where the number of sensor inputs could potentially be very large. Through neither losing expressiveness nor gaining in terms of efficiency, the Low-Rank Model Core provides a system that is both fast/scalable, yet capable of being highly predictive.

4.5. Adaptive Training and Checkpoint Serving

Another one of the most significant advantages of the FAV-ASTCL system is its ability to operate within a continuous learning loop in conjunction with a real-time adaptation mechanism between the Training/Adaptation Module and the Inference Serving Layer. The continuous learning capability provides the system with the opportunity to improve performance over time as new data from various scenarios is encountered.

- **Offline Trainer**: The model adjusts quickly when markets shift. Reacting through live data feeds that trigger the finetune process automatically. As it happens, this avoids full retraining and keeps performance sharp on changing trends.
- **Meta-Init and Finetuning**: Updated parameters are stored in checkpoints so they can be reused later if needed.
- **Prediction Serving**: More accurate Checkpoints move through first and get turned into live inference. The API then routes the final velocity forecasts - corrected for residuals, to tools like dashboards or traffic management setups.

Our setup hit a mean inference latency of 4. 2 milliseconds, pretty much proving it's ready for real-time use cases.

4.6. Algorithmic Optimization

Exogenous gating adjusts dynamically during each time step, shaping how data flows through the graph. The network builds its structure on the fly, updating residuals as new inputs arrive.

Now, it helps track shifting patterns in real time. That said, the graph evolves continuously, responding instantly to market jumps or quiet periods. This lets FAV-ASTCL handle volatility without losing accuracy or slowing down.

Algorithm 1 Real-Time Forward Pass of FAV-ASTCL

Require: Traffic sequence $X_{1:T}$, exogenous inputs E, static adjacency $\mathbf{A}_{stat}$
Ensure: Forecasted outputs $\hat{Y}_{1:P}$

```
1: Initialize hidden state: H_0 ← 0
2: for each timestep t = 1 to T do
3:    Derive dynamic adjacency: A_dyn ← ComputeSimilarity(X_t)
4:    Fuse static and dynamic graphs: A_final ← Blend(A_stat, A_dyn, α)
5:    Compute gating signal: G_e ← compute_gate(E_t, H_{t−1})
6:    Apply gating to exogenous features: Ẽ_t ← G_e ⊙ E_t
7:    Perform spatial aggregation: Z_t ← GraphConvolution(X_t, A_final)
8:    Update temporal state: H_t ← GRU(Z_t, Ẽ_t, H_{t−1})
9: end for
10: Compute residual correction: Res ← ResidualAdapter(H_T)
11: Generate final predictions: Ŷ_{1:P} ← LinearProjection(H_T + Res)
12: return Ŷ_{1:P}
```

Each module handles a distinct function in forecasting, acting alone yet synchronized across layers. As one processes incoming signals, others shape past outputs to keep predictions aligned. Even under heavy load or sudden drops in volume, the system stays balanced. There's no collapse when traffic peaks or plummets because the design holds everything together without issues.

5. Experimental Framework

5.1. Datasets: Benchmark and Live Deployment

The thing is, our research roots itself in two key data streams, both vetted for how well they reflect real traffic and how cleanly they stack up against known benchmarks.

1. **METR-LA**: It consist of 207 sensors on the Los Angeles highway network.

   The traffic speed dataset METR-LA is a standard benchmarking dataset in traffic forecasting based on space-time modeling. It contains traffic speed data collected from 207 loop detectors that are spread throughout the highway system of Los Angeles County. It offers very detailed time data which are usually taken every 5 minutes so that forecasting models can trace very subtle variations in traffic flow.

   One of the main reasons to use METR-LA is that it represents relatively steady and well-organized traffic situations in which spatial relations are mostly controlled by the physical layout of the roads. This feature makes it an excellent benchmark against which the basic performance of models can be assessed in a relatively controlled environment. Besides, since the adjacency matrices and data splits have already been defined, it becomes possible to compare other models such as DCRNN, STGCN, and ASTGCN with a level playing field.

2. **Hyderabad Volatility Dataset**: 5 hubs: Punjagutta, Hitech City, Secunderabad, Gachi-bowli, and LB Nagar.

Besides the benchmark evaluation, we prepared a custom dataset focused on actual high-volatility traffic conditions, in Hyderabad, to supplement the benchmark evaluation. The dataset is geographically limited to five large traffic locations, Punjagutta, Hitech City Secunderabad Gachibowli, and LB Nagar, which have their respective typical traffic flows such as commercial congestion, IT corridor flow, and mixed residential transit.

When compared to METR-LA, the dataset is characterized by very dynamic and non-linear traffic time series affected by regular perturbations, including weather variations, signal changes, and localized congestion events. Data collection was done through combination of traffic APIs, GPS traces and sensor feeds and enabled multi-source representation of traffic conditions.

The major aim with this dataset is to test the model's capability to manage (Frequent Adaptive Volatility) situations where the spatial relationships among nodes change drasti-cally over time. This data may serve as a stress-testing tool to check whether FAV-ASTCL, with all its flexibility, strength, and real-time performance, is prepared to have a successful practical implementation.

Even as both datasets used the same scaling and time frame, results shift when traffic surges or drops without warning. As it happens, the model's behavior under stress ends up far less predictable than expected.

6. Performance Analysis and Results

6.1. Quantitative Results and Benchmarking

The similar scaling and sliding window rules were integrated on every input. It's not just about consistency, it displays precisely where the model struggles or shines in mixed congestion zones.

Table 1: Global Performance Comparison: Baseline vs. FAV-ASTCL

Model Configuration	MAE	RMSE	MAPE (%)	ACC (%)
ASTCL (Baseline)	9.034	16.173	23.47%	76.53%
FAV-ASTCL (Ours)	**3.633**	**8.842**	**8.22%**	**91.78%**
% Improvement	**59.7%**	**45.3%**	**65.0%**	**19.9%**

Even as both datasets used the same scaling and time frame, results shift when traffic surges or drops without warning. As it happens, the model's behavior under stress ends up far less predictable than expected.

The variations in the Mean Absolute Percentage Error (MAPE) are also quite good since the error has been reduced by 65. 0%. It implies that FAV-ASTCL is highly effective in maintaining the levels of prediction accuracy in relation to varying traffic volumes. In addition, Accuracy (ACC) increased to 91. 78% is a further indication of the model's better performance and reliability when it comes to real-world deployment.

Such change is explained by the fact that the components: the Learnable Context Selector, Exogenous Gating mechanism and Online Adaptive Refinement module are reinforcers of each other. They enable the model to make decisions in real time about the changes that are occurring in space and time, particularly when the situation is highly unstable.

6.2. Cross-Architectural Baseline Comparison

Table 2 compares FAV-ASTCL to six of the most popular architectures in a 15-minute prediction time window and provides a broader perspective of its performance in comparison to popular architectures.

Table 2: Cross-Architectural Benchmarking on Hyderabad Dataset

Architecture	Adaptive Graph?	Context Aware?	MAE	RMSE
STGCN (2018)	No	No	18.42	28.55
DCRNN (2018)	No	No	14.12	22.18
Graph WaveNet (2019)	Yes	No	11.45	19.33
AGCRN (2020)	Yes	No	10.12	18.22
ASTCL (2025 Baseline)	Yes	Yes	9.03	16.17
FAV-ASTCL (Ours)	**Yes (Dynamic)**	**Yes (Gated)**	**3.63**	**8.84**

The variation in performance of various architectures is a clear indication that adaptive graph learning models are superior to fixed structures models. Still, FAV-ASTCL sets itself apart by not only just a static or semi-adaptive graph but hybrid completely dynamic graph reconstruction using context-aware gating.

Even the latest models such as Graph WaveNet and AGCRN are not able to make a highly dynamic model without contextual awareness, whereas previously, models such as STGCN and DCRNN were restricted by their inability to spatially adapt. ASTCL advances a step forward introducing context-aware features but remains based on rather stable graph assumptions.

By incorporating both dynamic graph learning and context-based selective gating, FAV-ASTCL gains a tremendous advantage in performance, becoming incredibly sensitive to new changes in the traffic patterns. This is a clear sign that flexibility and context-awareness are essential qualities of models of future traffic forecasting.

6.3. Case Study: Monsoon Volatility in Gachibowli

To further demonstrate how this proposed modeling can be applied to practical situations, we did a case study that happened during a heavy rainfall in the Gachibowli area. In fact, this is a scenario of high volatility while the traffic changes the most from the regular conditions because of weather disruptions.

In this situation, the Exogenous Gating part of FAV-ASTCL was capable of detecting the connection of humidity and wind with traffic changes. By only amplifying the relevant weather features, the model was able to revise its forecast on the fly. Therefore it ended up with a prediction that was only 10% wrong.

On the other hand, the baseline model that does not have selective gating and dynamic adaptation showed an error as high as 166%. Such a huge difference really brings out the rather significant advantages of context-aware mechanisms in dealing with even extreme and very uncommon cases. The case study has shown that FAV-ASTCL is not only performing well under normal situations but is also very capable when faced with environmental stress.

7. Discussion: Latency vs. Granularity

Our approach which we have incorporated, similarity-based graph learning, theoretically, increases the cost of computation to $O(N^2)$. Nevertheless, in the real world, this cost is barely perceivable because of such useful techniques as low-rank attention decomposition and efficient batching techniques.

Our 5-node Hyderabad example had a latency of inference of 4.2 milliseconds. The delay is minimal but it makes the system suitable to real-time, high-frequency telemetry of the city. This type of latency-precision trade-off is indeed required as it ensures that the model is able to come up with highly detailed predictions without delay in the interaction between the user.

However, it is important to consider that the larger the size of the nodes, the more it will demand in terms of computational resources. Nevertheless, with the modular architecture of FAV-ASTCL, it is still possible to scale up the system. For example, sparse attention mechanisms and distributed processing could be implemented to make it suitable for the entire large city.deployments.

8. Conclusion

FAV-ASTCL is a big step to a better way of modeling city traffic. It totally changes our way of managing the volatility within prediction systems. Volatility is traditionally perceived as a disturbing noise or something that has to be avoided. Nonetheless, volatility is a significant contextual factor in our approach, which can be modeled, interpreted, and even exploited to make better predictions.

MAE dropped 59.7% after combining graph dynamics, live tweaks, and outside triggers - making the results more stable and accurate. Prediction quality improved because the system responded quickly to new data streams.

With real-time sensing in hyderabad, AI handled chaotic urban flows without breaking down. Now it adapts instantly to events like sudden congestion, adjusting route plans on the spot. For now, it doesn't rely on flawless datasets - it thrives on fast reactions. A small traffic surge can reshape path decisions in seconds. In a way, the model picks up patterns quicker than traditional planning methods ever could.

References

[1] X. Wu, D. Zhang, C. Guo, C. He, B. Yang, and C. S. Jensen, "AutoCTS: Automated correlated time series forecasting," Proc. VLDB Endowment, vol. 15, no. 4, pp. 971–983, 2021.

[2] X. Wu, D. Zhang, M. Zhang, C. Guo, B. Yang, and C. S. Jensen, “AutoCTS+: Joint neural architecture and hyperparameter search for correlated time series forecasting,” Proc. ACM Manage. Data, vol. 1, no. 1, pp. 1–26, 2023.

[3] X. Liu, Y. Liang, C. Huang, Y. Zheng, B. Hooi, and R. Zimmermann, “When do contrastive learning signals help spatio-temporal graph forecasting?,” in Proc. Int. Conf. Adv. Geographic Inf. Syst., 2022, pp. 5:1–5:12.

[4] H. Qu, Y. Gong, M. Chen, J. Zhang, Y. Zheng, and Y. Yin, “Forecasting fine-grained urban flows via spatio-temporal contrastive self-supervision,” IEEE Trans. Knowl. Data Eng., vol. 35, no. 8, pp. 8008–8023, Aug. 2023.

[5] J. Ji et al., “Spatio-temporal self-supervised learning for traffic flow prediction,” in Proc. AAAI Conf. Artif. Intell., 2023, pp. 4356–4364.

[6] X. Zhang, Y. Gong, X. Zhang, X. Wu, C. Zhang, and X. Dong, “Mask-and contrast-enhanced spatio-temporal learning for urban flow prediction,” in Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage., 2023, pp. 3298–3307.

[7] Y. Gong, T. He, M. Chen, B. Wang, L. Nie, and Y. Yin, “Spatio-temporal enhanced contrastive and contextual learning for weather forecasting,” IEEE Trans. Knowl. Data Eng., vol. 36, no. 8, pp. 4260–4274, Aug. 2024.

[8] B. Y. Lin, F. F. Xu, E. Q. Liao, and K. Q. Zhu, “Transfer learning for traffic speed prediction: A preliminary study,” in Proc. AAAI Conf. Artif. Intell. Workshops, 2018, pp. 174–177.

[9] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, “Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data,” IEEE J. Sel. Areas Commun., vol. 37, no. 6, pp. 1389–1401, Jun. 2019.

[10] Y. Ren, X. Chen, S. Wan, K. Xie, and K. Bian, “Passenger flow prediction in traffic system based on deep neural networks and transfer learning method,” in Proc. Int. Conf. Intell. Transp. Eng., 2019, pp. 115–120.

[11] Z. Fang, D.Wu, L. Pan, L. Chen, and Y. Gao, “When transfer learning meets cross-city urban flow prediction: Spatio-temporal adaptation matters,” in Proc. Int. Joint Conf. Artif. Intell., L. D. Raedt, Ed., 2022, pp. 2030–2036.

[12] B. Lu, X. Gan, W. Zhang, H. Yao, L. Fu, and X. Wang, “Spatio-temporal graph few-shot learning with cross-city knowledge transfer,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2022, pp. 1162–1172.

[13] Y. Tang, A. Qu, A. H. F. Chow, W. H. K. Lam, S. C. Wong, and W. Ma, “Domain adversarial spatial-temporal network: A transferable framework for short-term traffic forecasting across cities,” in Proc. Int.Conf. Inf. Knowl. Manage., 2022, pp. 1905–1915. [45] Y. Wei, Y. Zheng, and Q. Yang, “Transfer knowledge between cities,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, pp. 1905–1914.

[14] L.Wang, X. Geng, X.Ma, F. Liu, and Q. Yang, "Cross-city transfer learning for deep spatio-temporal prediction," in Proc. Int. Joint Conf. Artif. Intell., 2019, pp. 1893–1899.

[15] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities: A meta-learning approach for spatial-temporal prediction," in Proc. Int. Conf. World Wide Web, 2019, pp. 2181–2191.

[16] S. Yang, J. Liu, and K. Zhao, "Space meets time: Local spacetime neural network for traffic flow forecasting," in Proc. Int. Conf. Data Mining, pp. 817–826.

[17] M. Li, Y. Tang, and W. Ma, "Few-shot traffic prediction with graph networks using locale as relational inductive biases," IEEE Trans. Intell. Transp. Syst., vol. 24, no. 2, pp. 1894–1908, Feb. 2023.

[18] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in Proc. AAAI Conf. Artif. Intell., 2021, pp. 4189–4196.

[19] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in Proc. AAAI Conf. Artif. Intell. Workshop, 1994, pp. 359–370.

[20] W. Yu, P. Zhou, S. Yan, and X. Wang, "InceptionNeXt: When inception meets ConvNeXt," 2023, arXiv:2303.16900.

[21] T. Kim, J. Kim, Y. Tae, C. Park, J. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in Proc. Int. Conf. Learn. Representations, 2022, pp. 1–25.

[22] A. Vaswani et al., "Attention is all you need," in Proc. Int. Conf. Neural Inf. Process. Syst., 2017, pp. 5998–6008.

[23] Y. Gong, Z. Li, J. Zhang, W. Liu, and Y. Zheng, "Online spatio-temporal crowd flow distribution prediction for complex metro system," IEEE Trans. Knowl. Data Eng., vol. 34, no. 2, pp. 865–880, Feb. 2022.

[24] Y. Zhang, F. Regol, A. Valkanas, and M. Coates, "Contrastive learning for time series on dynamic graphs," in Proc. Eur. Signal Process. Conf., 2022, pp. 742–746.

[25] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in Proc. Int. Conf. Learn. Representations, 2018, pp. 1–16.

[26] H. V. Jagadish et al., "Big data and its technical challenges," Commun. ACM, vol. 57, no. 7, pp. 86–94, 2014.

[27] H. Liu et al., "Spatio-temporal adaptive embedding makes vanilla transformer SOTA for traffic forecasting," in Proc. Int. Conf. Inf. Knowl. Manage., 2023, pp. 4125–4129.

[28] D. Kingma, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learn. Representations, 2014, pp. 1–15.

[29] E. Zivot and J. Wang, "Vector autoregressive models for multivariate time series," in Modeling Financial Time Series With S-PLUS. Berlin, Germany: Springer, 2006, pp. 385–429.

[30] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in Proc. AAAI Conf. Artif. Intell., 2020, pp. 914–921.

[31] K. Yi et al., "FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective," in Proc. Int. Conf. Neural Inf. Process. Syst., 2024, pp. 1–23.

[32] Y. Zhang, H. Lu, N. Liu, Y. Xu, Q. Li, and L. Cui, "Personalized federated learning for cross-city traffic prediction," in Proc. 33rd Int. Joint Conf. Artif. Intell., 2024, pp. 5526–5534.