# SemCast: Bridging Semantic Reasoning and Probabilistic Forecasting for Traffic Intelligence

Haoyang Yan[1], Yuquan Xu[1], Jing Bian[1], Yi Li[3], Ziyuan Pu[4,5], and Xiaolei Ma*[1,2]

[1]School of Transportation Science and Engineering, Beihang University, Beijing, China.

[2]Key Laboratory of Intelligent Transportation Technology and System of the Ministry of Education, Beihang University, Beijing 102206, China

[3]AutoNavi Software Company, Ltd, Beijing 100020, China

[4]School of Transportation, Southeast University, Nanjing, 211189, China

[5]Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Southeast University, Nanjing, 211189, China

*Address correspondence to: xiaolei@buaa.edu.cn

**Abstract**

Accurate traffic state prediction is fundamental to Intelligent Transportation Systems (ITS). Although deep learning models, particularly Graph Neural Networks (GNNs) and Transformers, have achieved state-of-the-art performance in routine forecasting, they often exhibit limited robustness under anomalous conditions such as accidents, extreme weather, or public events, primarily due to their dependence on historical patterns. Moreover, the inherent opacity of these models hinders interpretability, thereby constraining their practical utility. To overcome these limitations, we present SemCast, a hybrid framework that integrates probabilistic time-series forecasting with the semantic reasoning capabilities of Large Language Models (LLMs). The proposed methodology consists of three stages: (1) A foundation model employing discrete tokenization and sequence-to-sequence learning produces a probabilistic distribution of future trajectories; (2) A cross-modal reasoning module, powered by an LLM, processes heterogeneous textual data to perform contextual adjustments on candidate trajectories; (3) An operational interface generates interpretable diagnostics and actionable control strategies. Experiments conducted on large-scale real-world datasets from Beijing show that SemCast reduces prediction error by up to 15% during non-recurring congestion events relative to baseline methods, while demonstrating zero-shot generalization to previously unseen event types.

# 1 Introduction

Accurate traffic prediction is fundamental to the efficacy of Intelligent Transportation Systems (ITS), enabling critical functions such as dynamic route guidance, adaptive traffic signal control, and proactive incident management to mitigate congestion, reduce emissions, and enhance urban mobility resilience [1]. Congestion alone costs economies billions annually and degrades quality of life in urban areas [2]. Reliable traffic forecasts are therefore crucial for improving transportation efficiency and sustainability. Recent advances in deep learning, particularly graph neural networks (GNNs) for modeling spatial dependencies across road networks [3] and sophisticated sequence models (e.g., temporal convolution networks, attention mechanisms) for capturing temporal dynamics [4, 5], have significantly improved short-term forecasting accuracy under typical, recurring traffic conditions [6]. These methods learn patterns from extensive historical datasets, establishing a foundation for next-generation ITS applications in predictable scenarios.

Despite these advances, state-of-the-art traffic forecasting methods exhibit several limitations that constrain their practical utility, especially under non-routine conditions [7–9]. First, predictive performance often degrades sharply during anomalous events such as road accidents, unexpected closures, severe weather, or large-scale public gatherings [10, 11]. Models trained primarily on routine historical patterns often exhibit poor generalization when faced with data distributions shifted by such irregular occurrences [12, 13]. This limitation compromises reliability precisely when accurate prediction is most critical for effective incident response. Second, the fixed input encoding mechanisms of many deep learning models limit their ability to incorporate diverse, unstructured, or dynamically updated information, such as road work schedules, which often provide crucial context for anticipating traffic impacts. Integrating textual incident reports, event schedules, social media alerts, or other unforeseen disruptions typically demands complex feature engineering or extensive model retraining, hindering adaptation to novel event types [14]. Third, and perhaps most crucially for translation into practice, the standard output of these models, typically a high-dimensional matrix or tensor representing predicted speeds or flows, lacks direct interpretability. It fails to convey the underlying reasons for the predicted state or provide actionable guidance for traffic operators and decision-makers [15]. Consequently, even statistically accurate forecasts may not readily translate into effective, timely, and context-aware traffic management interventions, limiting the practical impact of these advanced techniques.

Large language models (LLMs) have emerged as powerful tools demonstrating remarkable capabilities in natural language understanding, contextual reasoning, and generalization across diverse tasks [16]. Their ability to process unstructured text, synthesize information from multiple sources, and generate explanatory text offers potential solutions to the challenges of context integration and interpretability in ITS [7, 17, 18]. However, applying LLMs directly to numerical time-series forecasting presents inherent difficulties. Architectures optimized for sequential token generation often struggle with the precise numerical regression required for traffic state prediction and may be inefficient in capturing the complex spatio-temporal statistical dependencies inherent in traffic flow [14, 19]. Moreover, training or fine-tuning large LLMs for specialized forecasting tasks demands considerable computational resources and large-scale, domain-specific datasets, which can be impractical for widespread deployment in operational ITS settings where data characteristics vary across locations and time [20].

To address these challenges, we introduce SemCast, a hybrid forecasting framework that integrates probabilistic time-series modeling with the contextual reasoning capabilities of LLMs. SemCast employs a multi-stage approach that leverages the complementary strengths of each component. First, a dedicated spatio-temporal foundation model, pre-trained on extensive historical traffic data, generates multiple candidate traffic state trajectories along with associated uncertainty estimates, ensuring statistical rigor and capturing baseline traffic dynamics. Second, an LLM, conditioned on flexible natural language prompts, processes these candidate trajectories. The prompts can incorporate both structured data (e.g., quantitative weather forecasts, road closure notices) and unstructured descriptions (e.g., event updates or incident reports). The LLM evaluates the candidate trajectories within this broader context, reasoning about likely impacts to select or adjust towards the most plausible outcome given real-time information. Importantly, the LLM also generates explanatory text and actionable recommendations tailored for traffic management personnel, drawing on insights from historical operational data. This architecture avoids burdening the LLM with direct numerical prediction, instead utilizing its strengths in semantic comprehension, causal inference, and context-aware reasoning.

Through comprehensive experiments encompassing both routine traffic patterns and diverse simulated and real-world anomalous scenarios (including construction, accidents, and public events), we demonstrate that SemCast significantly outperforms conventional deep learning baseline models during irregular events, reducing prediction errors by up to 15% under certain conditions, while matching state-of-the-art accuracy under normal conditions. Case studies further illustrate the framework's ability to generalize zero-shot to unseen event types described only via text prompts and to deliver context-aware, actionable insights (e.g., suggesting specific signal timing adjustments, disseminating targeted traveler advisories, or recommending dynamic routing strategies). By integrating the statistical power of probabilistic forecasting with the semantic understanding and reasoning capabilities of language-based AI, SemCast offers a new paradigm for traffic prediction that is accurate, adaptive, explainable, and aligned with the practical needs of transportation practitioners for effective real-world ITS deployment.

# 2 Literature Review

## 2.1 Deep Learning for Traffic Forecasting

Traffic forecasting has evolved from traditional statistical and regression-based approaches to deep learning models capable of capturing complex temporal dynamics. Early studies primarily relied on Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) architectures, to model sequential dependencies in traffic speed and flow data [21, 22]. While effective in temporal modeling, these approaches are limited in their ability to explicitly represent the non-Euclidean spatial topology of road networks.

To address this limitation, Spatio-Temporal Graph Neural Networks (STGNNs) have been widely adopted. Representative models such as DCRNN [23] and Graph WaveNet [24] combine graph convolutions with temporal sequence modeling to capture spatial correlations and temporal dynamics simultaneously. Subsequent works have extended these frameworks by relaxing the assumption of static spatial dependencies. For instance, Traffic Transformer introduces global–local decoders to hierarchically aggregate spatial features [25], while PDFormer incorporates propagation delay-aware attention to explicitly model temporal lags in traffic interactions [26].

Despite their expressive power, the necessity of complex graph convolutions has recently been questioned. Empirical studies indicate that simplified spatial modeling strategies can achieve comparable performance with substantially reduced computational overhead. SimST demonstrates that lightweight spatial aggregation approximates the effectiveness of GCN-based methods [27]. Similarly, MLP-based architectures such as ST-MLP [28] and STID [29] show that concise spatio-temporal identity mappings can outperform complex GNNs by reducing overfitting. Furthermore, classical machine learning approaches remain highly viable for specific estimation tasks; for instance, Yuan et al. demonstrated that a Random Forest model, when enriched with sparse GPS, weather, and signal control data, effectively estimates bicycle delays at intersections without the need for deep architectures [30].

In parallel, Transformer-based models have gained attention for their ability to capture long-range temporal dependencies. PatchTST segments time series into patches to preserve local temporal semantics [31], whereas iTransformer inverts the attention mechanism to better model multivariate correlations [32]. To further account for uncertainty and stochasticity in traffic systems, probabilistic generative approaches such as SpecSTG [33] and Diffusion-TS [15] have been proposed, enabling uncertainty-aware forecasting and data imputation.

## 2.2 Foundation Models for Time Series

Inspired by the "pre-train and fine-tune" paradigm in Natural Language Processing, recent research has shifted toward the development of foundation models for time series analysis [34, 35]. These models aim to learn universal temporal representations that generalize across datasets and tasks, enabling zero-shot or few-shot inference.

Chronos adapts T5-style architectures by discretizing continuous values into token sequences, achieving strong zero-shot performance across diverse domains [36]. Lag-Llama adopts a probabilistic modeling framework to capture scaling behaviors over large-scale time-series corpora [37]. To model multi-periodicity, TimesNet reformulates one-dimensional time series into two-dimensional representations, facilitating variation modeling via convolutional kernels [38].

Recent efforts further emphasize unified modeling across heterogeneous tasks. UniTS proposes a prompt-based backbone capable of jointly addressing forecasting, classification, and imputation [39], while Moirai-MoE introduces a Mixture-of-Experts architecture to handle diverse temporal resolutions without manual frequency alignment [40]. Another research direction explores the reuse of frozen pre-trained language or vision models. One Fits All demonstrates that large language models can be adapted to time series with minimal parameter updates [41]. Nevertheless, most existing foundation models operate exclusively on numerical signals, limiting their ability to incorporate unstructured contextual information, such as event descriptions, that is often critical for interpreting anomalies in intelligent transportation systems [42].

### 2.3 Large Language Models in Transportation

The application of Large Language Models (LLMs) in transportation research introduces a paradigm shift from purely numerical modeling toward semantic reasoning, multi-modal integration, and agent-based decision making [43, 44]. Comprehensive surveys categorize these emerging roles into information processors, knowledge encoders, and decision facilitators, highlighting the potential of LLMs to bridge the gap between heterogeneous data sources and actionable traffic insights [43]. However, a central challenge lies in aligning continuous time series data with the discrete token-based representations of LLMs. Time-LLM and LLM4TS address this issue through reprogramming and fine-tuning strategies that encode numerical sequences as language tokens [20, 45].

In urban computing scenarios, UrbanGPT integrates spatio-temporal dependency encoders with instruction tuning to improve generalization under data scarcity [46], while ST-LLM reformulates spatio-temporal observations as token sequences to capture global network dependencies [47]. Beyond forecasting, LLMs have been explored for explanation, simulation, and decision support. TF-LLM and ChatTraffic generate natural language interpretations of traffic conditions and congestion causes, enhancing model interpretability [48, 49]. In the domain of safety, CrashSage leverages LLMs to transform tabular crash data into textual narratives, enabling interpretable severity inference and uncovering complex risk factor interactions [50]. Furthermore, CityGPT extends these capabilities by constructing a city-scale world model with LLM-agents [18], while Liu et al. propose a framework where LLM-agents act as behaviorally rich proxies for human travelers to enhance microsimulation and travel demand modeling [51].

More advanced frameworks adopt “LLM-in-the-loop” architectures. TimeCAP and TimeXL employ multi-agent systems in which LLMs generate contextual summaries or reasoning paths that guide downstream numerical predictors [52, 53]. Additionally, external information sources such as social events and news reports have been incorporated via generative agents to stabilize forecasting under non-stationary conditions [54]. Nevertheless, a fundamental challenge remains in effectively reconciling the numerical accuracy of specialized time-series models with the high-level semantic reasoning capabilities of LLMs, particularly for real-time anomaly detection and intervention.

## 3 Preliminary

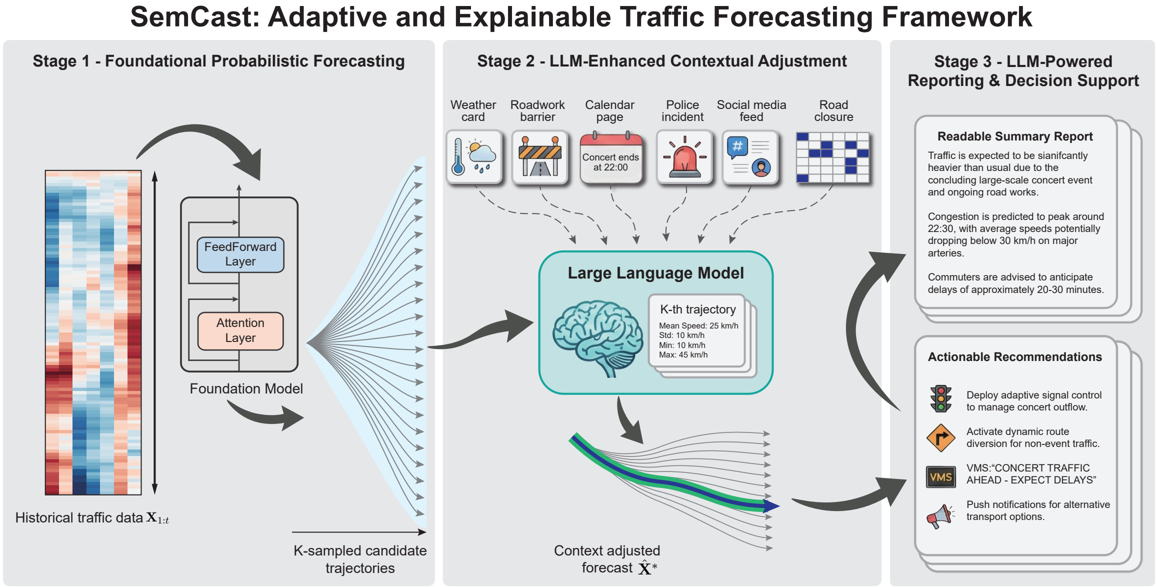
Traffic prediction is typically framed as a short-term time-series forecasting task, where future values $\mathbf{X}_{T+1:T+n}$ are predicted based on historical observations $\mathbf{X}_{1:T}$. This paper tackles a multi-modal version of this problem, recognizing that real-world traffic dynamics are influenced not only by past traffic states but also by a plethora of contextual factors often conveyed through textual or structured non-time-series data. We work with input instances $(\mathbf{X}_{1:T}, \mathbf{s})$, consisting of historical time series data $\mathbf{X}_{1:T} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$, where each $\mathbf{x}_t \in \mathbb{R}^N$ captures $D$ features of traffic states (e.g., speed, flow, occupancy) for $N$ spatial locations (e.g., road segments, sensors) over $T$ historical time steps, and auxiliary contextual information $\mathbf{s}$. This contextual information $\mathbf{s}$ can be diverse, including structured data (e.g., weather parameters, event schedules, road work logs) and unstructured natural language text (e.g., incident reports, social media alerts, news feeds) that potentially influences the time series and provides valuable context for improving forecast accuracy, especially during non-routine conditions. Our objective is to develop a model $\mathcal{F}$ that maps these multi-modal inputs to accurate and reliable predictions of future traffic states, potentially including uncertainty quantification. This is formalized as:

$$\mathbf{X}_{T+1:T+n} = \{\mathbf{x}_{T+1}, \mathbf{x}_{T+2}, \ldots, \mathbf{x}_{T+n}\} = \mathcal{F}(\mathbf{X}_{1:T}, \mathbf{s}), \tag{1}$$

where $\mathbf{X}_{T+1:T+n}$ is the predicted sequence of $n$ future state vectors or distributions. The ultimate goal is to identify an optimal model $\mathcal{F}$ that delivers accurate and reliable predictions while also being explainable and effectively leveraging the contextual information from $\mathbf{s}$ to adapt to both routine and non-routine conditions.

# 4 Methodology

## 4.1 Overall Framework


SemCast: Adaptive and Explainable Traffic Forecasting Framework
Stage 1 - Foundational Probabilistic Forecasting
FeedForward Layer
Attention Layer
Foundation Model
Historical traffic data $\mathbf{X}_{1:t}$
K-sampled candidate trajectories
Stage 2 - LLM-Enhanced Contextual Adjustment
Weather card
Roadwork barrier
Calendar page
Concert ends at 22:00
Police incident
Social media feed
Road closure
Large Language Model
K-th trajectory
Mean Speed: 25 km/h
Std: 10 km/h
Min: 10 km/h
Max: 45 km/h
Context adjusted forecast $\hat{\mathbf{X}}^*$
Stage 3 - LLM-Powered Reporting & Decision Support
Readable Summary Report
Actionable Recommendations
VMS

Traffic is expected to be sianifcantly heavier than usual due to the concluding large-scale concert event and ongoing road works.

Congestion is predicted to peak around 22:30, with average speeds potentially dropping below 30 km/h on major arteries.

Commuters are advised to anticipate delays of approximately 20-30 minutes.

Deploy adaptive signal control to manage concert outflow.

Activate dynamic route diversion for non-event traffic.

VMS:"CONCERT TRAFFIC AHEAD - EXPECT DELAYS"

Push notifications for alternative transport options.

Figure 1: Overall architecture of the SemCast framework. (A) Stage 1: A pre-trained spatio-temporal foundation model processes historical traffic data $\mathbf{X}_{1:T}$ to generate multiple candidate future trajectories $\{\hat{\mathbf{X}}^{(k)}\}$ representing a baseline probabilistic forecast. (B) Stage 2: Real-time contextual information $\mathbf{s}$, including structured and unstructured event data, is processed by an LLM. The LLM reasons about the event's impact and evaluates the candidate trajectories, selecting or adjusting to the most plausible event-conditioned forecast $\hat{\mathbf{X}}^*$. (C) Stage 3: The adjusted forecast, along with historical dispatch patterns, feeds into the LLM to generate human-readable summary reports and actionable traffic management recommendations.

The SemCast framework, illustrated in Fig.1, comprises three core stages that integrate advanced time-series modeling with large language models: (1) Foundational Probabilistic Forecasting, (2) LLM-Enhanced Contextual Adjustment, and (3) LLM-Powered Reporting and Decision Support.

- **Stage 1: Foundational Probabilistic Forecasting (Fig.1 A)**: This stage employs a robust forecasting model trained on extensive historical traffic data to capture complex dependencies and provide probabilistic outputs. The model architecture is pre-trained on curated data subsets, including time-series from high-volume road segments or grid cells representing typical urban traffic dynamics, as well as traffic data aggregated around key venues (e.g., stadiums, transport hubs, event centers) known to exhibit non-standard patterns. Inspired by architectures such as Chronos [36], which adapt language transformer-based models for time-series, our foundation model processes historical input to generate multiple trajectory samples $\{\hat{\mathbf{X}}^{(k)}_{T+1:T+n}\}_{k=1}^{K}$. These samples collectively approximate the predictive distribution $P(\mathbf{X}_{T+1:T+n}|\mathbf{X}_{1:T})$, providing a baseline probabilistic forecast with inherent uncertainty quantification, which is essential for representing the range of possibilities under routine conditions.

- **Stage 2: LLM-Enhanced Contextual Adjustment (Fig.1 B)**: This stage refines the baseline forecast by incorporating real-time contextual information $\mathbf{s}$, addressing the limitations of models that rely solely on historical patterns. The contextual information $\mathbf{s}$, which may include structured data and unstructured text, is processed by an LLM. The LLM follows a reasoning process: it first summarizes the event information, then assesses the likely causal impact on traffic flow (considering location, severity, and duration), and finally evaluates the quantitative effect. Based on this reasoning, the LLM selects the most plausible trajectory $\hat{\mathbf{X}}^{*}_{T+1:T+n}$ or generates an adjusted trajectory that better reflects the anticipated impact of the event. This step utilizes the LLM's capacity to interpret and reason about novel or complex situations described in natural language, thereby modulating the initial probabilistic forecast based on real-time context.

- **Stage 3: LLM-Powered Reporting and Decision Support (Fig.1 C)**: The final stage translates the adjusted forecast $\hat{\mathbf{X}}^{*}_{T+1:T+n}$ into practical outputs for end-users. The LLM receives the context-adjusted forecast along with relevant historical traffic guidance data, which enables it to incorporate implicit operational preferences and common responses from past similar situations. Based on the adjusted forecast, contextual information, and learned operational patterns, the LLM generates: (i) a concise, human-readable summary report describing anticipated traffic conditions, highlighting potential issues (e.g., specific bottlenecks, expected delay increases), and explaining the reasoning based on contextual factors; and (ii) actionable recommendations for traffic management (e.g., adjusting signal timing plans, disseminating advisories regarding lane closures, or preparing diversion routes). This stage bridges the gap between numerical prediction and practical operational utility, providing interpretable insights and decision support.

## 4.2 Stage 1: Foundational Probabilistic Forecasting

In this stage, we reformulate the time-series forecasting problem as a specialized language modeling task. By mapping continuous traffic dynamics into discrete semantic tokens, we leverage the robust reasoning capabilities of the Transformer architecture to capture complex temporal dependencies and model the aleatoric uncertainty inherent in stochastic traffic flows.

### 4.2.1 Sequence Serialization and Tokenization Strategy

Unlike traditional point-wise forecasting models, we adopt a patch-based tokenization strategy inspired by the Chronos paradigm [36]. This approach reduces the sequence length complexity from $O(T^2)$ to $O((T/S)^2)$ while preserving local temporal semantics.

**Temporal Patching** Given a univariate time series $\mathbf{x}^i = \{x_1^i, \dots, x_T^i\} \in \mathbb{R}^T$ for a spatial node $i$, we first decompose the sequence into a series of overlapping patches. Let $P$ denote the patch length and $S$ the stride. The sequence is unfolded into a matrix of patches $\mathcal{P}^i = \{\mathbf{p}_1^i, \dots, \mathbf{p}_N^i\}$, where the $j$-th patch $\mathbf{p}_j^i \in \mathbb{R}^P$ is defined as:

$$\mathbf{p}_j^i = [x^i_{(j-1)S+1}, \dots, x^i_{(j-1)S+P}] \tag{2}$$

where $N = \lfloor (T-P)/S \rfloor + 1$ is the number of tokens.

**Local Scaling for Stationarity** Traffic data exhibits significant non-stationarity (e.g., varying peak hours across days). To ensure the distribution of values within each patch falls into a learnable range for the quantizer, we apply instance-level mean scaling. For each patch $\mathbf{p}_j^i$, we compute a local scale factor $s_j = \frac{1}{P}\sum_{k=1}^{P} |\mathbf{p}_{j,k}^i| + \epsilon$. The scaled patch is obtained via:

$$\tilde{\mathbf{p}}_j^i = \frac{\mathbf{p}_j^i}{s_j} \tag{3}$$

This normalization allows the model to learn scale-invariant temporal patterns, generalizing across different traffic volume levels.

**Quantization and Vocabulary Mapping** To interface with the categorical nature of language models, we map the continuous scaled domain $\mathbb{R}$ to a discrete codebook $\mathcal{C} = \{c_1, \dots, c_V\}$ of size $V$. We employ a quantization function $Q : \mathbb{R} \to \{1, \dots, V\}$ using uniform binning within a fixed range $[\min, \max]$. The token ID $z_{j,k}$ for the $k$-th element of the $j$-th patch is derived as:

$$z_{j,k} = Q(\tilde{p}_{j,k}^i) = \text{clip}\left(\left\lfloor \frac{\tilde{p}_{j,k}^i - \min}{\max - \min} \times (V-1) \right\rfloor, 0, V-1\right) \tag{4}$$

Consequently, the continuous time series $\mathbf{x}^i$ is transformed into a sequence of discrete token IDs $\mathbf{Z}^i = \{z_{1,1}, \dots, z_{N,P}\}$, which serves as the "sentence" input to the Transformer.

### 4.2.2 Encoder-Decoder Architecture

We employ a modified T5 (Text-to-Text Transfer Transformer) backbone [55] to process the tokenized traffic sequences. The architecture consists of an encoder that maps the historical token sequence to a latent representation, and a decoder that autoregressively generates future tokens.

**Self-Attention with Relative Position Bias** The core mechanism is the Multi-Head Self-Attention (MHSA). Unlike standard Transformers that use absolute sinusoidal positional encodings, T5 utilizes relative positional embeddings, which is crucial for time series as it naturally models the "time lag" distance between patches. The attention score between query $q$ and key $k$ is computed as:

$$\mathcal{A}_{q,k} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_{model}}} + \mathbf{B}_{q,k}\right)\mathbf{V} \tag{5}$$

where $\mathbf{B}_{q,k}$ is a learnable scalar bias added to the attention logit, representing the relative temporal distance between position $q$ and $k$.

**Objective Function** The model is trained to minimize the standard Cross-Entropy loss over the vocabulary $V$. Given the historical context $\mathbf{Z}_{<t}$, the model predicts the probability distribution of the next token $z_t$:

$$\mathcal{L}_{\text{CE}} = -\sum_{t=1}^{L} \log P_\theta(z_t|\mathbf{Z}_{<t}) \tag{6}$$

We incorporate a multi-task learning objective by combining the primary forecasting task with a random masking reconstruction task (BERT-style) to enhance the model's robustness against missing data and noise.

### 4.2.3 Probabilistic Trajectory Generation

To capture the aleatoric uncertainty inherent in future traffic states, we move beyond point estimation to probabilistic trajectory generation.

**Nucleus Sampling (Top-$p$)** During inference, instead of greedy decoding (selecting the token with max probability), we approximate the posterior distribution $P(\mathbf{X}_{\text{future}}|\mathbf{X}_{\text{history}})$ by sampling $K$ independent hypotheses (trajectories). We employ Nucleus Sampling, which truncates the tail of the distribution, considering only the smallest set of top tokens whose cumulative probability exceeds a threshold $p$:

$$\mathcal{V}^{(p)} = \{z \in \mathcal{C} \mid \sum_{z' \in \mathcal{V}^{(p)}} P_\theta(z'|z_{<t}) \geq p\} \tag{7}$$

At each step $t$, the next token $z_t^{(k)}$ for the $k$-th hypothesis is sampled from the re-normalized distribution over $\mathcal{V}^{(p)}$. The stochasticity of the generated trajectories originates exclusively from this sampling procedure: each of the $K$ decoding passes draws tokens independently from the truncated categorical distribution using distinct random seeds, and the autoregressive nature of generation propagates this randomness across all subsequent time steps. Consequently, even with a fixed input, different sampling runs produce diverse trajectory hypotheses that collectively approximate the model's learned predictive distribution.

**De-tokenization and Aggregation** The generated token sequences are mapped back to the continuous domain via inverse quantization and inverse scaling using the stored local scale factors $s_j$. Since patches overlap, the value at a specific time step $t$ is reconstructed by averaging the predictions from all patches covering that step, ensuring smoothness at patch boundaries:

$$\hat{x}_t^{(k)} = \frac{1}{|\Omega_t|} \sum_{j \in \Omega_t} s_j \cdot Q^{-1}(z_{j,\mathrm{idx}(t)}^{(k)}) \tag{8}$$

where $\Omega_t$ is the set of patches containing time step $t$. This process yields a set of candidate trajectories $\mathcal{H} = \{\hat{\mathbf{X}}^{(1)}, \ldots, \hat{\mathbf{X}}^{(K)}\}$ representing plausible future scenarios.

## 4.3 Stage 2: Logic-Enhanced Contextual Trajectory Selection

While Stage 1 provides a robust probabilistic baseline based on historical patterns, it lacks the semantic understanding to account for varying external disruptions (e.g., accidents, extreme weather). Stage 2 bridges this gap by employing a Large Language Model (LLM) as a logic-driven reasoner to evaluate and select the most plausible trajectory $\hat{\mathbf{X}}^*$ from the candidate set $\mathcal{H}$ based on real-time context $\mathbf{s}$.

### 4.3.1 Semantic Serialization of Probabilistic Forecasts

Since LLMs operate in the semantic space rather than the numerical tensor space, we must project the candidate set $\mathcal{H} = \{\hat{\mathbf{X}}^{(1)}, \ldots, \hat{\mathbf{X}}^{(K)}\}$ into a textual representation compatible with the LLM's context window. We define a serialization function $\psi : \mathbb{R}^n \to \mathcal{S}$ that aggregates high-dimensional trajectory data into descriptive statistics. For each candidate $k$, we compute the statistical summary vector $\mathbf{v}^{(k)}$ over key corridors, including the minimum, median, and maximum speeds across the prediction horizon. This is formatted into a structured prompt segment:

$$\mathcal{T}_{traj}^{(k)} = \text{“Candidate } k : \text{Trend } \in [\min(\mathbf{v}^{(k)}), \max(\mathbf{v}^{(k)})], \text{Dynamics: Description}(\nabla \mathbf{v}^{(k)})\text{”} \tag{9}$$

where Description$(\cdot)$ maps numerical gradients to linguistic descriptors. Specifically, we define a set of threshold-based mapping rules for the average speed gradient $\bar{g}^{(k)} = \frac{1}{n-1} \sum_t (v_{t+1}^{(k)} - v_t^{(k)})$ computed across the prediction horizon: $\bar{g} > 5$ km/h per step → “rapidly increasing”; $2 < \bar{g} \leq 5$ → “gradually increasing”; $-2 \leq \bar{g} \leq 2$ → “stable”; $-5 \leq \bar{g} < -2$ → “gradually decreasing”; $\bar{g} < -5$ → “rapidly decreasing.” For example, a candidate with speed range $[25, 55]$ km/h and $\bar{g} = -6.2$ would be serialized as: “Candidate 3: Trend ∈ [25, 55] km/h, Dynamics: rapidly decreasing.” This deterministic mapping ensures reproducibility and allows the LLM to comprehend the traffic dynamics implied by each hypothesis without processing raw tensors.

### 4.3.2 Chain-of-Thought Reasoning and Selection

We leverage the reasoning capabilities of the Qwen2.5-14B-Instruct model [56] to model the causal relationship between the context $\mathbf{s}$ and traffic states. Critically, the LLM in this stage is restricted to a *selection* operation over the pre-generated candidate set $\mathcal{H}$; it does not generate new numerical values or modify existing trajectories. This design ensures that all forecast outputs remain within the realistic traffic distributions learned by the foundation model, while the LLM contributes its semantic reasoning strength. The inference process is structured as a conditional probability maximization problem via Chain-of-Thought (CoT) prompting.

Let $\mathcal{P}_{sys}$ be the system instruction and $\mathcal{T}_{ctx}$ be the textualized real-time context (e.g., incident logs, weather reports). The LLM generates a reasoning path $\mathcal{R}$ followed by a selection index $k^*$:

$$(\mathcal{R}, k^*) \sim P_{LLM}(\cdot \mid \mathcal{P}_{sys}, \mathcal{T}_{ctx}, \{\mathcal{T}_{traj}^{(k)}\}_{k=1}^{K}) \tag{10}$$

The reasoning path $\mathcal{R}$ explicitly decomposes the task into three logical steps:

1. **Event Impact Analysis:** The LLM parses $\mathcal{T}_{ctx}$ to extract event attributes (severity, location) and infers the spatiotemporal scope of the impact (e.g., "Lane closure on Highway A will cause upstream congestion propagation").

2. **Hypothesis Verification:** The model compares the inferred impact against the statistical properties of each candidate $\mathcal{T}_{traj}^{(k)}$. For instance, if the event implies a significant speed drop, candidates showing "stable high speed" are rejected.

3. **Optimal Selection:** The index $k^*$ corresponding to the candidate that maximizes semantic alignment with the reasoning $\mathcal{R}$ is selected.

The final output is the specific trajectory $\hat{\mathbf{X}}^* = \hat{\mathbf{X}}^{(k^*)}$, which represents the event-conditioned forecast. We emphasize that Equation (10) defines a pure selection mechanism: $k^* \in \{1, \ldots, K\}$ is an index into the candidate set, and the output $\hat{\mathbf{X}}^*$ is identical to one of the pre-generated trajectories without any numerical modification.

## 4.4 Stage 3: LLM-Powered Reporting and Decision Support

The objective of Stage 3 is to transform the selected traffic forecast $\hat{\mathbf{X}}^*$ into structured, actionable outputs by incorporating historical traffic management experience. Rather than relying solely on generative inference, this stage conditions the generation process on retrieved historical cases, thereby grounding the outputs in previously observed traffic contexts and response patterns.

### 4.4.1 Historical Case Retrieval

We construct a historical case repository $\mathcal{K} = \{(c_m, a_m)\}_{m=1}^{M}$, where each entry consists of a past traffic context $c_m$ (e.g., incident type and observed traffic state) and the corresponding management action $a_m$ (e.g., signal control strategies or information dissemination measures). This repository serves as a source of empirical reference for the current scenario.

Given the current system state $\mathbf{s}$ and the selected forecast summary $\hat{\mathbf{X}}^*$, we form a joint query representation $q$ by concatenation. A dense retrieval model is then employed to identify historical cases that are semantically similar to the current situation. Specifically, cosine similarity is computed between the query embedding $\phi(q)$ and the stored context embeddings $\phi(c_m)$:

$$\mathcal{S}_{rel} = \{(c_m, a_m) \mid \cos(\phi(q), \phi(c_m)) > \tau\}, \tag{11}$$

where $\phi(\cdot)$ denotes a pre-trained sentence encoder and $\tau$ is a similarity threshold. The top-$N$ retrieved cases constitute the reference set $\mathcal{S}_{rel}$, which provides contextual constraints for the subsequent generation stage. It is important to note that the retrieved cases serve exclusively as *in-context references* appended to the LLM prompt; they do not impose hard constraints or force the model to replicate historical actions. The LLM synthesizes information from the retrieved cases alongside the current forecast and context, enabling it to draw upon proven management patterns while adapting recommendations to the specific characteristics of the present scenario. This retrieval-augmented generation (RAG) approach also reduces the risk of hallucination by anchoring the LLM's outputs in documented operational precedents.

### 4.4.2 Structured Output Generation

The final output is produced by a conditional generator $G$, which integrates the traffic forecast, the current contextual description, and the retrieved historical cases:

$$\mathbf{Y} = G(\hat{\mathbf{X}}^*, \mathcal{T}_{ctx}, \mathcal{S}_{rel}). \tag{12}$$

The generated output $\mathbf{Y}$ is organized into two complementary components:

- *Situation Summary* ($\mathbf{Y}_{rep}$): a concise description of the anticipated traffic evolution derived from $\hat{\mathbf{X}}^*$, highlighting key temporal and spatial characteristics relevant to traffic management.
- *Action Suggestions* ($\mathbf{Y}_{rec}$): a set of recommended response measures informed by the retrieved cases in $\mathcal{S}_{rel}$ and adapted to the current forecasted conditions.

By conditioning the generation process on historically similar traffic scenarios, this framework encourages consistency with prior management patterns while allowing flexibility to accommodate the specific characteristics of the current forecast.

## 5 Experiments & Results

To rigorously evaluate the efficacy of the SemCast framework, we present a comprehensive experimental analysis. We first detail the experimental setup—including datasets, implementation details, baselines, and metrics—in Section 5.1. Subsequently, we report the empirical results under routine traffic conditions (Section 5.2), during anomalous events (Section 5.3), and analyze the system's explainability (Section 5.4) and component contributions (Section 5.5).

## 5.1 Materials and Methods

### 5.1.1 Datasets and Preprocessing

This study utilizes multiple large-scale, real-world traffic and contextual datasets collected by Amap across China, with a primary focus on Beijing for foundational model training and broader regions for event context and specific analyses. All datasets cover the period from September 2023 to May 2024, unless otherwise indicated.

**Spatio-Temporal Traffic Data:** The core dataset for training the probabilistic foundation model consists of high-resolution traffic state information for the urban core of Beijing. Raw traffic data were aggregated onto a regular grid with a spatial resolution of $500m \times 500m$, resulting in $N = 5,797$ distinct spatial grid cells covering the main urban road network. For each cell, key traffic state variables, including traffic volume (vehicles per interval) and average speed (km/h), were computed and aggregated into 5-minute intervals. This dataset contains approximately 1.3 billion data points, providing a comprehensive representation of urban traffic dynamics that is much larger than many commonly used open-source benchmarks [57].

**Venue-Centric Traffic Data:** To capture traffic patterns influenced by large-scale public events, supplementary datasets focused on major venues were compiled.

- *Aggregated Venue Flow:* Derived from location-based service data, aggregated traffic volume information was obtained for the precise geographical boundaries and surrounding buffer areas of over 300 major venues (sports stadiums, concert halls, major tourist attractions, transport hubs) across multiple cities in China. This dataset facilitates modeling of event-specific demand surges and dispersion patterns.

- *Fine-Grained Venue Grid Data:* For a subset of the venues above, traffic volume was aggregated onto a finer $100m \times 100m$ grid covering each venue. Due to the high granularity leading to sparsity, data were primarily extracted from the top-20 grid cells exhibiting the highest average historical traffic volume within each venue's defined area, focusing analysis on the most relevant micro-locations.

**Contextual Event Data:** Real-time and historical event information, essential for the LLM reasoning stage (Stage 2) and evaluation during anomalies, was primarily obtained from the Amap open platform APIs and associated historical logs for the relevant periods and geographical areas. This included:

- *Structured Construction Data:* A curated dataset encompassing over 10,000 construction events on highways and major arterials. Each entry typically includes precise location information (coordinates or road segment identifiers), scheduled start and end times, number and type of lanes affected (e.g., closure, partial blockage), and nature of the work.

- *Unstructured Anomaly Reports:* Traffic incident information disseminated via Amap, originating from user reports or official traffic authority alerts. These reports typically contain a natural language description of the incident (e.g., "Accident involving two cars on Ring Road eastbound near Exit 5, blocking right lane"), an approximate location, and a timestamp. This text serves as direct input for the LLM.

Table S1 and Table S2 in the Supplementary Materials provide representative examples of these two data types. In summary, the structured construction dataset contains 10,247 events with an average duration of 6.3 hours, spanning 892 distinct road segments across Beijing. Event types include full closure (18%), partial closure (52%), and shoulder work (30%). The unstructured anomaly report dataset comprises 8,631 incident records, covering accidents (41%), vehicle breakdowns (23%), road hazards (19%), and other incidents (17%), with an average report length of 45 characters.

**Preprocessing:** Standard preprocessing steps were applied to the traffic datasets before model training and evaluation. Missing values in the time-series data (less than 5% of points) were imputed using linear interpolation, while others were excluded. Traffic state features (speed, volume) were normalized using Z-score normalization based on the mean and standard deviation calculated from the training portion of the primary Beijing dataset. For GNN-based baselines, the spatial graph adjacency matrix was constructed based on a road network distance threshold, with edge weights typically defined by inverse distance. Unstructured text data from anomaly reports and event information were cleaned to remove irrelevant artifacts before being incorporated into the LLM prompts.

### 5.1.2 Baseline Implementations

The chosen baselines represent a broad spectrum of modern time series forecasting methodologies, ensuring a robust and comprehensive comparison against different architectures:

- DLinear[13]: Represents simple yet surprisingly effective linear models, serving as a strong benchmark against more complex architectures by decomposing the time series and applying separate linear layers. It challenges the necessity of intricate designs for certain forecasting tasks.

- FiLM [58]: Represents linear models enhanced with frequency analysis, designed to improve forecasting by better capturing periodicity through specific decomposition techniques applied in the frequency domain.

- Informer [59]: A prominent Transformer-based model optimized for long sequence time-series forecasting (LSTF) efficiency through a ProbSparse self-attention mechanism and distilling operation, representing efficient Transformer variants.

- PatchTST [31]: Represents channel-independent Transformer approaches utilizing patching, where input time series are divided into subseries-level patches that are fed as tokens to the Transformer, capturing local semantic information.

- Chronos [36]: Represents recent large pre-trained foundation models for time series, leveraging language model architectures scaled to time series data for zero-shot or few-shot forecasting, showcasing the potential of large-scale pre-training.

- iTransformer [32]: An innovative Transformer architecture that inverts the standard process by applying attention to embedded variates across the entire time series length, designed to better capture multivariate correlations.

For implementation, we utilized established framework TSLib [38, 60] or the official public code repositories associated with each baseline model. To guarantee a fair and direct comparison, all baseline models were trained using the identical historical dataset that was employed for training the SemCast foundation model. The sole exception was Chronos, for which we leveraged the publicly available pre-trained weights, applying it directly as a zero-shot forecaster without fine-tuning on our specific dataset. We note that this introduces a fundamental difference in experimental conditions: SemCast and the other baselines benefit from domain-specific pre-training on the Beijing traffic data, whereas Chronos relies entirely on its general-purpose pre-trained representations. The observed performance gap between Chronos and domain-trained models should therefore be interpreted as reflecting the challenge of zero-shot transfer to a specialized traffic forecasting domain, rather than a deficiency of the Chronos architecture itself. We include Chronos to illustrate the current limitations of general foundation models when applied to domain-specific traffic prediction without adaptation.

### 5.1.3 Evaluation Details

We evaluated the model's performance using standard time-series forecasting metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Weighted Absolute Percentage Error (WAPE). MAE measures the average absolute difference between predictions and actual values. MSE computes the average of the squared errors, which emphasizes larger deviations more heavily. RMSE is the square root of MSE, bringing the error back to the original scale of the data. MAPE and WAPE assess the relative size of errors compared to actual values, providing scale-independent evaluation.

It is worth noting that we preferred WAPE over MAPE for traffic volume forecasting tasks, where the data often contains zero or near-zero values. In such scenarios, MAPE can become undefined or unstable due to division by zero or extremely small actual values, leading to misleading evaluations. Additionally, in cases with large variance in traffic volumes, MAPE tends to overemphasize errors in low-volume periods while underrepresenting high-volume ones. In contrast, WAPE normalizes total absolute error by the sum of actual values across all time steps and locations, offering a more stable and representative metric under these conditions.

The specific metrics are defined as (13), (14), (15), (16), and (17):

$$\text{MAE} = \frac{1}{nN} \sum_{t=T+1}^{T+n} \sum_{i=1}^{N} |x_{t,i} - \hat{x}_{t,i}| \tag{13}$$

$$\text{MSE} = \frac{1}{nN} \sum_{t=T+1}^{T+n} \sum_{i=1}^{N} (x_{t,i} - \hat{x}_{t,i})^2 \tag{14}$$

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{nN} \sum_{t=T+1}^{T+n} \sum_{i=1}^{N} (x_{t,i} - \hat{x}_{t,i})^2} \tag{15}$$

$$\mathrm{MAPE} = \frac{1}{nN} \sum_{t=T+1}^{T+n} \sum_{i=1}^{N} \left| \frac{x_{t,i} - \hat{x}_{t,i}}{x_{t,i}} \right| \tag{16}$$

$$\mathrm{WAPE} = \frac{\sum_{t=T+1}^{T+n} \sum_{i=1}^{N} |x_{t,i} - \hat{x}_{t,i}|}{\sum_{t=T+1}^{T+n} \sum_{i=1}^{N} |x_{t,i}|} \tag{17}$$

**Anomaly Identification**: For evaluating performance during anomalies, event periods were identified using timestamps from the Amap/Gaode construction and incident logs. Anomalous periods were defined as 1 hour before to 1 hour after the logged event time for relevant locations. For public events, the anomalous period covered 2 hours before the event start to 2 hours after the event end. Zero-shot evaluation used events from categories completely held out during any training/fine-tuning.

**Data Splits**: Data was split chronologically for each city/dataset. Typically, the first 70% was used for pre-training the foundation model, the next 10% for validation, and the final 20% for testing.

**Foundation Model Training Details**: The SemCast foundation model was trained jointly on the Beijing spatio-temporal grid data and the venue-centric traffic data described above, without incorporating external datasets. The model was trained for 50 epochs using the AdamW optimizer with an initial learning rate of $5 \times 10^{-4}$, a cosine annealing schedule with warm-up over the first 5 epochs, and a batch size of 256. The vocabulary size was set to $V = 4096$ bins, the patch length to $P = 12$ (corresponding to 1 hour), and the stride to $S = 6$ (30 minutes). The number of candidate trajectories was set to $K = 20$ with a nucleus sampling threshold of $p = 0.9$. Training was conducted on 4× NVIDIA A100 (80GB) GPUs over approximately 36 hours. For the LLM components in Stages 2 and 3, we employed the Qwen2.5-14B-Instruct model deployed via vLLM for efficient batch inference, without any fine-tuning.

## 5.2 Baseline Performance under Routine Conditions

To establish the foundational capability of our framework, we evaluated the performance of the pre-trained spatio-temporal model (Stage 1 output) under routine traffic conditions, comparing it against established baselines. The evaluation used Beijing datasets (Jan-May 2024), excluding major anomalies.

Our results, summarized in Table 1, demonstrate that the SemCast foundational model substantially outperforms all evaluated deep learning baselines under these normal conditions. Across all prediction horizons (15, 30, and 60 minutes) and all metrics (MAE, MSE, WAPE), our model consistently achieved the lowest error rates. Notably, while Informer emerged as the most competitive baseline, our model still surpassed its performance considerably. For instance, the average MAE for our model (0.153) was markedly lower than Informer's (0.166) and substantially better than PatchTST (0.259) or FiLM (0.527).

Furthermore, our model displayed remarkable stability across prediction horizons, maintaining consistently low error values even for 60-minute forecasts. This contrasts with baselines like DLinear and PatchTST, which exhibited pronounced accuracy degradation as the horizon increased.

This confirms that the foundation model provides a robust starting point for subsequent contextual adjustment.

Table 1: Comparison of forecasting metrics (MAE, MSE and WAPE) under routine traffic conditions for different prediction horizons (15, 30, 60 min, and average). Models compared include our pre-trained spatio-temporal model and selected deep learning baselines (e.g., DLinear, FiLM, Informer, PatchTST, Chronos, iTransformer) on the Beijing dataset. Best results for each metric and horizon are highlighted.

| | **15 min** | | | **30 min** | | | **60 min** | | | **Average** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **MAE** | **MSE** | **WAPE** | **MAE** | **MSE** | **WAPE** | **MAE** | **MSE** | **WAPE** | **MAE** | **MSE** | **WAPE** |
| **DLinear** | 0.270 | 0.139 | 0.384 | 0.374 | 0.257 | 0.532 | 0.542 | 0.508 | 0.769 | 0.383 | 0.285 | 0.545 |
| **FiLM** | 0.423 | 0.301 | 0.602 | 0.534 | 0.491 | 0.759 | 0.669 | 0.772 | 0.949 | 0.527 | 0.496 | 0.748 |
| **Informer** | 0.161 | 0.060 | 0.229 | 0.167 | 0.066 | 0.237 | 0.176 | 0.074 | 0.249 | 0.166 | 0.066 | 0.236 |
| **PatchTST** | 0.201 | 0.090 | 0.286 | 0.251 | 0.147 | 0.356 | 0.347 | 0.285 | 0.491 | 0.259 | 0.164 | 0.368 |
| **chronos-b** | 0.423 | 0.427 | 0.561 | 0.342 | 0.323 | 0.495 | 0.291 | 0.325 | 0.425 | 0.424 | 0.489 | 0.602 |
| **chronos-m** | 0.433 | 0.426 | 0.574 | 0.351 | 0.328 | 0.508 | 0.301 | 0.343 | 0.440 | 0.431 | 0.495 | 0.613 |
| **chronos-s** | 0.439 | 0.443 | 0.582 | 0.349 | 0.325 | 0.505 | 0.311 | 0.374 | 0.454 | 0.436 | 0.504 | 0.620 |
| **iTransformer** | 0.194 | 0.088 | 0.275 | 0.226 | 0.130 | 0.321 | 0.293 | 0.234 | 0.416 | 0.233 | 0.143 | 0.331 |
| **SemCast** | **0.153** | **0.055** | **0.217** | **0.151** | **0.054** | **0.215** | **0.158** | **0.058** | **0.225** | **0.153** | **0.055** | **0.217** |

## 5.3 Enhanced Prediction Accuracy across Diverse Anomalous Events

We evaluate the effectiveness of incorporating textual context into traffic forecasting under four representative anomalous scenarios, covering both routine disruptions and semantically ambiguous events. The proposed SemCast model is compared with a foundation-model baseline that relies solely on historical traffic observations. Figure 2 shows that, when exposed to anomalous inputs, the baseline frequently produces predictions with large uncertainty, whereas SemCast yields more concentrated and accurate forecasts.

Figure 2 presents illustrative examples across different event types. In common disruption scenarios such as highway construction and traffic incidents (Fig. 2(a) and Fig. 2(b)), SemCast accurately captures the expected reductions in traffic volume or speed and closely follows the ground-truth trajectories. In contrast, the baseline model exhibits substantial predictive uncertainty and struggles to estimate the severity of the impact.

More pronounced differences are observed in semantically nuanced scenarios. In the parking control case (Fig. 2(c)), the baseline extrapolates historical stability and fails to anticipate a complete stop, resulting in a wide predictive interval. By incorporating event descriptions related to destination adjustment, SemCast correctly forecasts a sharp speed reduction approaching zero. Similarly, in the accident with rerouting scenario (Fig. 2(d)), the baseline prediction reflects a strong prior association between accidents and congestion, leading to lower-speed estimates. SemCast, however, accounts for the rerouting information and predicts sustained high-speed flow, consistent with the observed outcome.

Overall, these results indicate that integrating event-level textual context enables the model to disambiguate anomalous conditions and to revise statistically dominant patterns when additional semantic information is available.

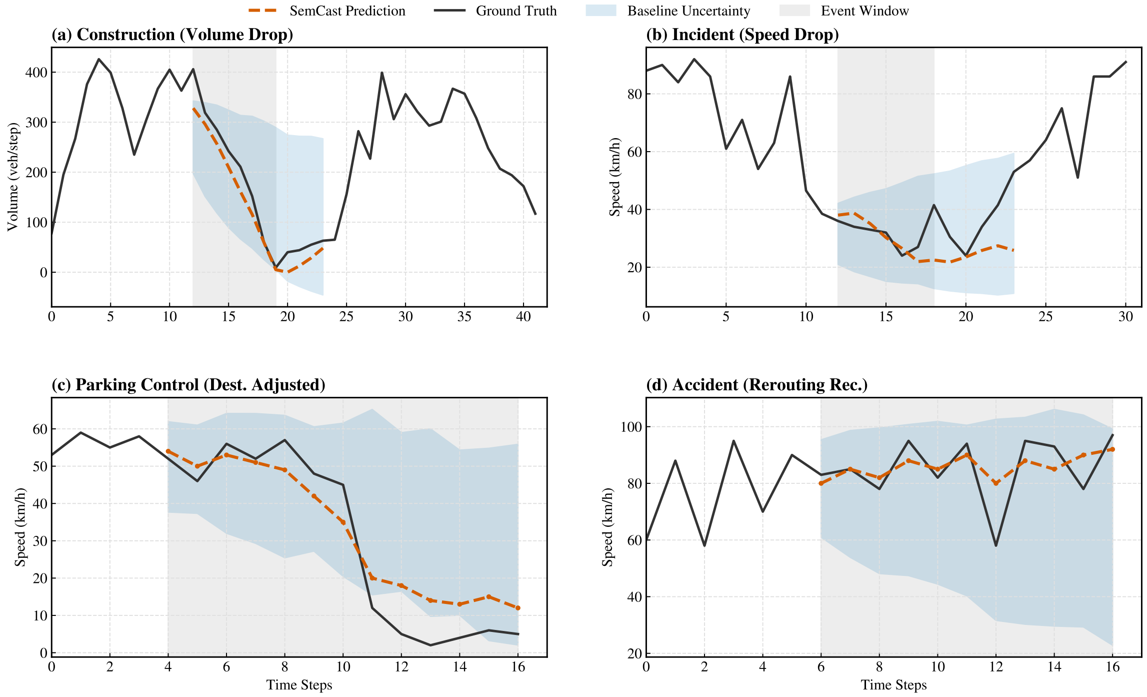


Figure 2: Time-series predictions under different anomalous traffic scenarios. Results are shown for (a) highway construction, (b) traffic incident, (c) parking control, and (d) accident with rerouting. The blue shaded region denotes the predictive uncertainty of the baseline model without textual input. Predictions from SemCast are shown as an orange dashed line, while the ground truth is shown as a black solid line.

## 5.4 Explainable Reporting and Actionable Recommendations

Stage 3 of the framework utilizes the LLM to translate quantitative forecasts into human-readable reports. Figure 3 showcases this capability:

- **Routine Scenario:** The LLM provides concise confirmation ("Expect typical heavy congestion") and prompts standard checks ("Ensure ramp metering active"), building trust.
- **Construction Event:** The output is analytical, synthesizing the speed drop with the "single lane closure" context. Recommendations are proactive, suggesting "deploying VMS advisory" and "adjusting signal timings" on alternate routes, demonstrating reasoning about network-wide ripple effects.
- **Public Event:** The report highlights specific egress congestion timing. Recommendations like "Monitor parking lot clearance rates" are highly tailored to the event type.

These outputs demonstrate SemCast's capacity to bridge the gap between prediction and action, transforming numerical data into operationally relevant intelligence. While comprehensive automated evaluation of LLM-generated text remains an open challenge, we assessed report quality through two complementary approaches. First, a panel of three domain experts (traffic engineers

Table 2: Forecasting accuracy improvements during anomalous events. Comparison of prediction accuracy (MAE and RMSE in km/h, MAPE in %) for SemCast (Context-Adjusted) against baseline models (Zero-shot and Few-shot) during periods affected by scheduled construction and unplanned incidents. Data evaluated on the anomaly datasets across different prediction horizons. SemCast consistently outperforms baselines, demonstrating the effectiveness of LLM-based contextual adjustment for handling diverse disruptions compared to unadjusted forecasts or standard fine-tuning.

| | **15 min** | | | **30 min** | | | **60 min** | | | **Average** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **MAE** | **RMSE** | **MAPE** | **MAE** | **RMSE** | **MAPE** | **MAE** | **RMSE** | **MAPE** | **MAE** | **RMSE** | **MAPE** |
| **Few-shot** | 5.11 | 8.25 | 18.11% | 5.63 | 9.27 | 21.18% | 6.36 | 10.57 | 23.56% | 5.59 | 9.25 | 20.59% |
| **Zero-shot** | 5.19 | 8.33 | 18.71% | 5.69 | 9.33 | 21.69% | 6.47 | 10.72 | 25.90% | 5.67 | 9.34 | 21.38% |
| **SemCast** | **4.86** | **8.03** | **16.51%** | **5.31** | **8.98** | **18.45%** | **5.94** | **10.19** | **20.58%** | **5.27** | **8.97** | **18.18%** |

with over 5 years of operational experience) independently evaluated 50 randomly sampled reports on four dimensions: factual consistency with the forecast data, actionability of recommendations, reasoning coherence, and language clarity, each on a 5-point Likert scale. The average scores were 4.3, 4.1, 4.2, and 4.5, respectively, indicating that the generated reports were considered reliable and practically useful. Second, we observed that the retrieval-augmented generation approach in Stage 3 substantially reduced instances of generic or hallucinated recommendations compared to a vanilla LLM baseline without case retrieval (reduction from 23% to 6% of outputs containing non-actionable suggestions). We acknowledge that more rigorous, large-scale evaluation protocols—including user studies in operational settings—remain an important direction for future work.

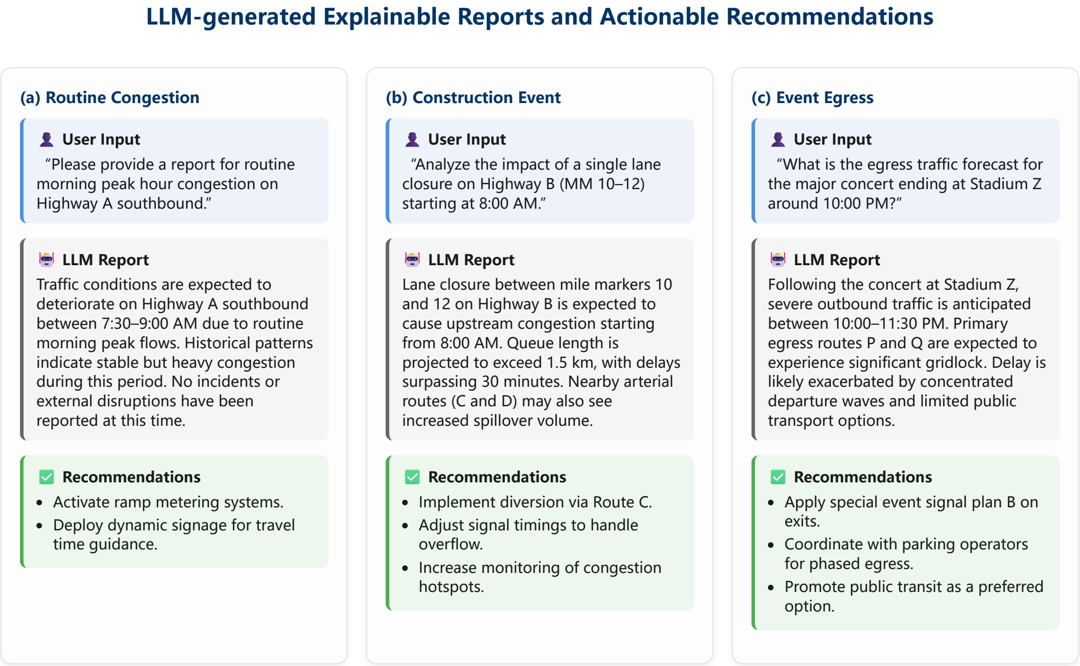
## 5.5 Ablation Studies

We performed ablation studies on the anomaly dataset to validate component contributions:

1. **Necessity of LLM Adjustment:** Removing Stage 2 and relying solely on the foundation model resulted in noticeable accuracy degradation during anomalies (Figure 4a), confirming the essential role of contextual reasoning.

2. **Probabilistic vs. Deterministic Input:** Providing the LLM with multiple probabilistic candidate trajectories (Stage 1 output) yielded better performance than providing a single deterministic mean forecast (Figure 4b). This suggests the probabilistic distribution offers a richer search space for the LLM to select the most contextually plausible outcome.

3. **Context Quality:** While performance dropped when input context was intentionally degraded (vague/incorrect), SemCast still outperformed context-unaware baselines, indicating a degree of robustness (Figure 4c). Specifically, the *vague* context condition was generated by stripping detailed attributes from event descriptions, retaining only the event category (e.g., replacing “Two-vehicle collision on Ring Road 3 eastbound near Shuangjing Exit, blocking the rightmost lane” with “Traffic incident on Ring Road 3”). The *incorrect* context condition was constructed by substituting contradictory event information (e.g., providing a “road clear, normal flow” description during an actual accident event). These degradation strategies test the model’s sensitivity to context specificity and its resilience against misinformation, respectively.



LLM-generated Explainable Reports and Actionable Recommendations

(a) Routine Congestion

User Input

"Please provide a report for routine morning peak hour congestion on Highway A southbound."

LLM Report

Traffic conditions are expected to deteriorate on Highway A southbound between 7:30–9:00 AM due to routine morning peak flows. Historical patterns indicate stable but heavy congestion during this period. No incidents or external disruptions have been reported at this time.

☑ Recommendations

- Activate ramp metering systems.
- Deploy dynamic signage for travel time guidance.

(b) Construction Event

User Input

"Analyze the impact of a single lane closure on Highway B (MM 10–12) starting at 8:00 AM."

LLM Report

Lane closure between mile markers 10 and 12 on Highway B is expected to cause upstream congestion starting from 8:00 AM. Queue length is projected to exceed 1.5 km, with delays surpassing 30 minutes. Nearby arterial routes (C and D) may also see increased spillover volume.

☑ Recommendations

- Implement diversion via Route C.
- Adjust signal timings to handle overflow.
- Increase monitoring of congestion hotspots.

(c) Event Egress

User Input

"What is the egress traffic forecast for the major concert ending at Stadium Z around 10:00 PM?"

LLM Report

Following the concert at Stadium Z, severe outbound traffic is anticipated between 10:00–11:30 PM. Primary egress routes P and Q are expected to experience significant gridlock. Delay is likely exacerbated by concentrated departure waves and limited public transport options.

☑ Recommendations

- Apply special event signal plan B on exits.
- Coordinate with parking operators for phased egress.
- Promote public transit as a preferred option.

Figure 3: **LLM-generated explainable reports and actionable recommendations.** Examples regarding (a) Routine Congestion, (b) Construction Event, and (c) Public Event Egress, showing the translation from context-aware prediction to operational intelligence.

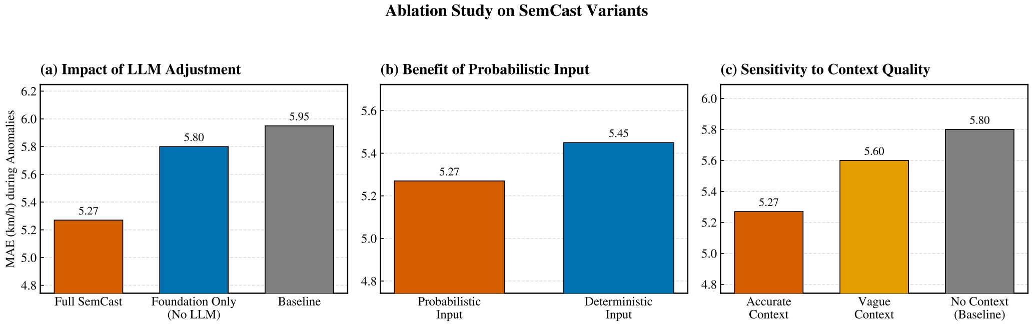


Figure 4: **Ablation study results.** (a) Impact of removing LLM adjustment. (b) Benefit of probabilistic candidates vs. deterministic input. (c) Sensitivity to context quality.

# 6 Discussion

In this study, we present SemCast, a hybrid framework that integrates a pre-trained spatio-temporal foundation model for probabilistic forecasting with the contextual reasoning capabilities of Large Language Models to address key limitations in current traffic prediction systems. Our results indicate that SemCast achieves forecasting accuracy comparable to state-of-the-art deep learning methods under routine traffic conditions. More importantly, it significantly outperforms these baselines during anomalous events, such as construction, incidents, and public gatherings, by effectively incorporating real-time structured and unstructured contextual information through LLM processing. We observed substantial error reductions (up to 15%) during such events, underscoring the framework's enhanced adaptability and resilience. Additionally, case studies demonstrate SemCast's promising zero-shot generalization capability, enabling it to interpret and respond to novel event types described solely through natural language prompts. Finally, we show that the framework can generate interpretable, human-readable summary reports and actionable traffic management recommendations, bridging the critical gap between prediction and operational decision-making.

The effectiveness of SemCast derives from its hybrid design, which leverages the complementary strengths of deep learning for pattern recognition in high-dimensional spatio-temporal data and LLMs for flexible context understanding, reasoning, and natural language generation. The foundation model provides a statistically robust baseline forecast that captures complex recurring dynamics and uncertainty. The LLM, rather than being tasked with direct numerical prediction, focuses on its core strengths: interpreting diverse inputs (text, structured data), inferring causal impacts of events, evaluating scenarios based on context, and communicating findings effectively. This division of labor enables SemCast to overcome the limitations of purely data-driven models during anomalies and the challenges of purely LLM-based approaches in precise numerical forecasting. The integration of historical dispatch patterns further enhances the practical relevance of the generated recommendations. This synergistic approach represents a significant step toward ITS that are not only predictive but also adaptive, interpretable, and operationally relevant.

The distinctive feature of SemCast is its integration of two powerful AI paradigms: deep learning for robust forecasting and LLMs for contextual reasoning. While deep learning models have advanced accuracy on benchmark datasets, they often lack robustness to out-of-distribution events and fail to provide actionable insights. Previous attempts to incorporate auxiliary data have frequently relied on rigid input structures and struggled with unstructured text. LLM applications in transportation have primarily focused on tasks such as route planning, dialogue systems, or summarizing traffic reports, but their direct application to end-to-end numerical forecasting remains challenging. SemCast integrates these two paradigms in a manner that mitigates their respective weaknesses while harnessing their combined potential for context-aware, interpretable, and actionable traffic prediction.

The performance of SemCast during anomalies is inherently dependent on the availability and quality of real-time contextual information ($\mathbf{s}$), where we use $\mathbf{s}$ to denote the raw contextual data (e.g., event logs, weather records) and $\mathcal{T}_{ctx}$ to refer to its processed textual representation used in the LLM prompts. Inaccurate or delayed event reports may limit the effectiveness of the LLM adjustment stage. Although we demonstrated some robustness, further research is needed to address noisy or conflicting contextual inputs.

A key assumption of the framework is that the candidate trajectory set $\mathcal{H}$ generated by Stage 1 covers the event-induced traffic dynamics sufficiently for the LLM to select an appropriate forecast. This assumption is supported by two design choices: (1) nucleus sampling with $K = 20$ trajectories and $p = 0.9$ generates diverse hypotheses spanning a broad range of the learned predictive distribution; and (2) the foundation model is pre-trained on venue-centric data that includes non-standard traffic patterns (demand surges, rapid dispersal), thereby expanding the model's capacity to generate trajectories reflecting event-like dynamics even without explicit event conditioning. Our experimental results in Figure 2 confirm that appropriate candidates are available across diverse anomaly types. Nevertheless, we acknowledge that for extreme or entirely novel event types whose traffic signatures differ fundamentally from any training data, the candidate set may fail to include sufficiently representative trajectories. Addressing this limitation—for instance, through guided sampling conditioned on event features or candidate augmentation strategies—is an important direction for future work.

Regarding computational feasibility, we provide a breakdown of the end-to-end inference latency measured on a single NVIDIA A100 GPU. Stage 1 (foundation model inference for $K = 20$ trajectories) requires approximately 0.8 seconds per prediction window. Stage 2 (LLM-based trajectory selection via Qwen2.5-14B-Instruct, deployed with vLLM) requires approximately 2.1 seconds per query, including prompt construction and decoding. Stage 3 (report generation) requires approximately 3.5 seconds. The total pipeline latency of approximately 6.4 seconds is well within the 5-minute forecasting interval used in our system, confirming real-time feasibility. We note that further latency reductions are achievable through model quantization (e.g., INT4/INT8), smaller LLMs (e.g., 7B parameter variants), or parallel processing of Stages 2 and 3 across multiple sensors.

Potential issues related to LLM biases or hallucinations, while mitigated by grounding the LLM's task in evaluating pre-generated trajectories, require ongoing attention and potentially safety layers in operational deployment. Moreover, effective prompt engineering is crucial for eliciting the desired reasoning and output from the LLM, which may necessitate domain expertise. Scalability to extremely large, city-wide networks with tens of thousands of sensors also needs further investigation; however, since Stage 1 operates independently per sensor and the LLM stages process serialized summaries rather than raw tensors, the framework is amenable to parallelization across spatial locations.

Future research will focus on integrating SemCast with real-time traffic control systems (e.g., adaptive signal control, variable speed limits) to enable fully automated, context-aware traffic management. We also aim to incorporate a broader range of contextual data sources, such as real-time social media feeds, advanced weather nowcasting, or connected vehicle data, to further enhance situational awareness. Developing more sophisticated methods for the LLM to not only select but also actively modify forecast trajectories based on context could yield additional accuracy gains. Finally, conducting user studies with traffic operators to evaluate the usability and effectiveness of the generated reports and recommendations in real-world control room settings is essential for practical validation and refinement.

## Acknowledgments

This work was supported by Beijing Natural Science Foundation (No. JQ24051).

## 7 Data Availability

The datasets central to this study, including the large-scale traffic network data and the anomalous event logs, were provided by Amap. Due to the proprietary nature of this information, which encompasses commercial sensitivities and privacy considerations, these datasets are not publicly available. We acknowledge the importance of reproducibility and regret that these necessary restrictions prevent the public dissemination of these materials. Enquiries regarding the methodology or potential collaborations may be directed to the corresponding author.

## 8 Supplementary Materials

- **Text S1 and Table S1.** Detailed description and illustrative examples of the Structured Construction Data used in this study.
- **Text S2 and Table S2.** Detailed description and examples of the Unstructured Anomaly Reports.

## References

1. Wu K, Ding J, Lin J, et al. Big-data empowered traffic signal control could reduce urban carbon emission. Nature Communications 2025;16:2013.
2. Avila AM and Mezić I. Data-driven analysis and forecasting of highway traffic dynamics. Nature Communications 2020;11:2090.
3. Shao Z, Zhang Z, Wang F, and Xu Y. Pre-training Enhanced Spatial-temporal Graph Neural Network for Multivariate Time Series Forecasting. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 2022:1567–77. DOI: 10.1145/3534678.3539396. arXiv: 2206.09113.
4. Li Z, Cai R, Fu TZJ, Hao Z, and Zhang K. Transferable Time-Series Forecasting Under Causal Conditional Shift. IEEE Transactions on Pattern Analysis and Machine Intelligence 2024;46:1932–49.
5. Runge J et al. Inferring causation from time series in Earth system sciences. Nature Communications 2019;10:2553.
6. Yuan Y, Ding J, Feng J, Jin D, and Li Y. UniST: A Prompt-Empowered Universal Model for Urban Spatio-Temporal Prediction. 2024. DOI: 10.1145/3637528.3671662. arXiv: 2402.11838.

7. Guo X, Zhang Q, Jiang J, Peng M, Zhu M, and Yang HF. Towards explainable traffic flow prediction with large language models. Communications in Transportation Research 2024;4:100150.
8. Williams AR et al. Context is Key: A Benchmark for Forecasting with Essential Textual Information. 2024. DOI: 10.48550/arXiv.2410.18959. arXiv: 2410.18959.
9. Liu H et al. Time-MMD: A New Multi-Domain Multimodal Dataset for Time Series Analysis. 2024. DOI: 10.48550/arXiv.2406.08627. arXiv: 2406.08627.
10. Li Z et al. Language in the Flow of Time: Time-Series-Paired Texts Weaved into a Unified Temporal Narrative. 2025. DOI: 10.48550/arXiv.2502.08942. arXiv: 2502.08942.
11. Xue H and Salim FD. PromptCast: A New Prompt-based Learning Paradigm for Time Series Forecasting. 2023. DOI: 10.48550/arXiv.2210.08964. arXiv: 2210.08964.
12. Arango SP et al. ChronosX: Adapting Pretrained Time Series Models with Exogenous Variables. 2025. DOI: 10.48550/arXiv.2503.12107. arXiv: 2503.12107.
13. Zeng A, Chen M, Zhang L, and Xu Q. Are Transformers Effective for Time Series Forecasting? 2022. arXiv: 2205.13504.
14. Tan M, Merrill MA, Gupta V, Althoff T, and Hartvigsen T. Are Language Models Actually Useful for Time Series Forecasting? 2024. DOI: 10.48550/arXiv.2406.16964. arXiv: 2406.16964.
15. Yuan X and Qiao Y. Diffusion-TS: Interpretable Diffusion for General Time Series Generation. In: *The Twelfth International Conference on Learning Representations*. 2023.
16. Su J et al. Large Language Models for Forecasting and Anomaly Detection: A Systematic Literature Review. 2024. DOI: 10.48550/arXiv.2402.10350. arXiv: 2402.10350.
17. Wang X, Fang M, Zeng Z, and Cheng T. Where Would I Go Next? Large Language Models as Human Mobility Predictors. 2023. DOI: 10.48550/arXiv.2308.15197. arXiv: 2308.15197.
18. Feng J, Du Y, Liu T, Guo S, Lin Y, and Li Y. CityGPT: Empowering Urban Spatial Cognition of Large Language Models. 2024. DOI: 10.48550/arXiv.2406.13948. arXiv: 2406.13948.
19. Liu P et al. Taming Pre-trained LLMs for Generalised Time Series Forecasting via Cross-modal Knowledge Distillation. 2024. DOI: 10.48550/arXiv.2403.07300. arXiv: 2403.07300.
20. Jin M et al. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. 2023. DOI: 10.48550/arXiv.2310.01728. arXiv: 2310.01728.
21. Ma X, Tao Z, Wang Y, Yu H, and Wang Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies 2015;54:187–97.
22. Ma X, Zhong H, Li Y, Ma J, Cui Z, and Wang Y. Forecasting Transportation Network Speed Using Deep Capsule Networks With Nested LSTM Models. IEEE Transactions on Intelligent Transportation Systems 2021;22:4813–24.
23. Li Y, Yu R, Shahabi C, and Liu Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 2017.

24. Wu Z, Pan S, Long G, Jiang J, and Zhang C. Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121 2019.

25. Yan H, Ma X, and Pu Z. Learning Dynamic and Hierarchical Traffic Spatiotemporal Features With Transformer. IEEE Transactions on Intelligent Transportation Systems 2022;23:22386–99.

26. Jiang J, Han C, Zhao WX, and Wang J. PDFormer: Propagation Delay-Aware Dynamic Long-Range Transformer for Traffic Flow Prediction. 2023. DOI: 10.48550/arXiv.2301.07945. arXiv: 2301.07945.

27. Liu X, Liang Y, Huang C, et al. Do We Really Need Graph Neural Networks for Traffic Forecasting? 2023. DOI: 10.48550/arXiv.2301.12603. arXiv: 2301.12603.

28. Wang Z, Nie Y, Sun P, Nguyen NH, Mulvey J, and Poor HV. ST-MLP: A Cascaded Spatio-Temporal Linear Framework with Channel-Independence Strategy for Traffic Forecasting. 2023. DOI: 10.48550/arXiv.2308.07496. arXiv: 2308.07496.

29. Shao Z, Zhang Z, Wang F, Wei W, and Xu Y. Spatial-Temporal Identity: A Simple yet Effective Baseline for Multivariate Time Series Forecasting. 2022. DOI: 10.48550/arXiv.2208.05233. arXiv: 2208.05233.

30. Yuan Y, Wang K, Duives D, Daamen W, and Hoogendoorn SP. Machine learning-based bicycle delay estimation at signalized intersections using sparse GPS data and traffic control signals: A Dutch case study using random forest algorithm. Artificial Intelligence for Transportation 2025;3-4:100037.

31. Nie Y, Nguyen NH, Sinthong P, and Kalagnanam J. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. 2023. arXiv: 2211.14730.

32. Liu Y, Hu T, Zhang H, et al. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. 2023. DOI: 10.48550/arXiv.2310.06625. arXiv: 2310.06625.

33. Lin L, Shi D, Han A, and Gao J. SpecSTG: A Fast Spectral Diffusion Framework for Probabilistic Spatio-Temporal Traffic Forecasting. 2024. DOI: 10.48550/arXiv.2401.08119. arXiv: 2401.08119.

34. Liang Y et al. Foundation Models for Time Series Analysis: A Tutorial and Survey. 2024. DOI: 10.48550/arXiv.2403.14735. arXiv: 2403.14735.

35. Jin M et al. Large Models for Time Series and Spatio-Temporal Data: A Survey and Outlook. 2023. DOI: 10.48550/arXiv.2310.10196. arXiv: 2310.10196.

36. Ansari AF et al. Chronos: Learning the Language of Time Series. 2024. arXiv: 2403.07815.

37. Rasul K et al. Lag-Llama: Towards Foundation Models for Time Series Forecasting. 2023. DOI: 10.48550/arXiv.2310.08278. arXiv: 2310.08278.

38. Wu H, Hu T, Liu Y, Zhou H, Wang J, and Long M. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In: *International Conference on Learning Representations*. 2023.

39. Gao S, Koker T, Queen O, Hartvigsen T, Tsiligkaridis T, and Zitnik M. UniTS: Building a Unified Time Series Model. 2024. arXiv: 2403.00131.

40. Liu X et al. Moirai-MoE: Empowering Time Series Foundation Models with Sparse Mixture of Experts. 2024. DOI: 10.48550/arXiv.2410.10469. arXiv: 2410.10469.

41. Zhou T, Niu P, Wang X, Sun L, and Jin R. One Fits All: Power General Time Series Analysis by Pretrained LM. 2023. arXiv: 2302.11939.

42. Jin M et al. Time Series Forecasting with LLMs: Understanding and Enhancing Model Capabilities. 2024. arXiv: 2402.10835v2. URL: https://arxiv.org/abs/2402.10835v2.

43. Nie T, Sun J, and Ma W. Exploring the roles of large language models in reshaping transportation systems: A survey, framework, and roadmap. Artificial Intelligence for Transportation 2025;1:100003.

44. Karim MM, Shi Y, Zhang S, Wang B, Nasri M, and Wang Y. Large language models and their applications in roadway safety and mobility enhancement: A comprehensive review. Artificial Intelligence for Transportation 2025;1:100004.

45. Chang C, Peng WC, and Chen TF. LLM4TS: Two-Stage Fine-Tuning for Time-Series Forecasting with Pre-Trained LLMs. 2023. DOI: 10.48550/arXiv.2308.08469. arXiv: 2308.08469.

46. Li Z et al. UrbanGPT: Spatio-Temporal Large Language Models. 2024. DOI: 10.48550/arXiv.2403.00813. arXiv: 2403.00813.

47. Liu C, Yang S, Xu Q, et al. Spatial-Temporal Large Language Model for Traffic Prediction. 2024. DOI: 10.48550/arXiv.2401.10134. arXiv: 2401.10134.

48. Guo X, Zhang Q, Peng M, Zhu M, Hao, and Yang. Explainable Traffic Flow Prediction with Large Language Models. 2024. arXiv: 2404.02937.

49. Zhang C, Zhang Y, Shao Q, et al. ChatTraffic: Text-to-Traffic Generation via Diffusion Model. 2024. DOI: 10.48550/arXiv.2311.16203. arXiv: 2311.16203.

50. Zhen H and Yang JJ. CrashSage: A large language model-centered framework for contextual and interpretable traffic crash analysis. Artificial Intelligence for Transportation 2025;3-4:100030.

51. Liu T, Yang J, and Yin Y. Toward LLM-agent-based modeling of transportation systems: A conceptual framework. Artificial Intelligence for Transportation 2025;1:100001.

52. Lee G, Yu W, Shin K, Cheng W, and Chen H. TimeCAP: Learning to Contextualize, Augment, and Predict Time Series Events with Large Language Model Agents. 2025. DOI: 10.48550/arXiv.2502.11418. arXiv: 2502.11418.

53. Jiang Y et al. Explainable Multi-modal Time Series Prediction with LLM-in-the-Loop. 2025. DOI: 10.48550/arXiv.2503.01013. arXiv: 2503.01013.

54. Wang X, Feng M, Qiu J, Gu J, and Zhao J. From News to Forecast: Integrating Event Analysis in LLM-Based Time Series Forecasting with Reflection. 2024. DOI: 10.48550/arXiv.2409.17515. arXiv: 2409.17515.

55. Raffel C et al. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research 2020;21:1–67.

56. Bai J et al. Qwen Technical Report. 2023. arXiv: 2309.16609.

57. Liu X et al. LargeST: A Benchmark Dataset for Large-Scale Traffic Forecasting. 2023. arXiv: 2306.08259.

58. Zhou T, Ma Z, Wen Q, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. In: *Advances in neural information processing systems (NeurIPS)*. Vol. 35. 2022:12677–90.

59. Zhou H, Zhang S, Peng J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 12. 2021:11106–15.

60. Wang Y, Wu H, Dong J, Liu Y, Long M, and Wang J. Deep Time Series Models: A Comprehensive Survey and Benchmark. arXiv preprint arXiv:2407.13278 2024.