Embracing Large Language Models in Traffic Flow Forecasting

Yusheng Zhao[♡], **Xiao Luo**[♠†], **Haomin Wen**[♢], **Zhiping Xiao**[♣†], **Wei Ju**[♡], **Ming Zhang**[♡†]
[♡] State Key Laboratory for Multimedia Information Processing,
School of Computer Science, PKU-Anker LLM Lab, Peking University
[♠] Department of Computer Science, University of California, Los Angeles
[♢] Carnegie Mellon University
[♣] Paul G. Allen School of Computer Science and Engineering, University of Washington
yusheng.zhao@stu.pku.edu.cn, xiaoluo@cs.ucla.edu,
wenhaomin.whm@gmail.com, patxiao@uw.edu, {juwei,mzhang_cs}@pku.edu.cn

Abstract

Traffic flow forecasting aims to predict future traffic flows based on historical traffic conditions and the road network. It is an important problem in intelligent transportation systems, with a plethora of methods being proposed. Existing efforts mainly focus on capturing and utilizing spatio-temporal dependencies to predict future traffic flows. Though promising, they fall short in adapting to test-time environmental changes in traffic conditions. To tackle this challenge, we propose to introduce large language models (LLMs) to help traffic flow forecasting and design a novel method named Large Language Model Enhanced Traffic Flow Predictor (LEAF). LEAF adopts two branches, capturing different spatio-temporal relations using graph and hypergraph structures, respectively. The two branches are first pre-trained individually, and during test time, they yield different predictions. Based on these predictions, a large language model is used to select the most likely result. Then, a ranking loss is applied as the learning objective to enhance the prediction ability of the two branches. Extensive experiments on several datasets demonstrate the effectiveness of LEAF. Our code is available at https://github.com/YushengZhao/LEAF.

1 Introduction

Traffic flow forecasting is an integral part of intelligent transportation systems (Dimitrakopoulos and Demestichas, 2010; Zhang et al., 2011) and smart cities (Shahid et al., 2021; Dai et al., 2022). The target of traffic flow forecasting is to predict future traffic flows using historical data and spatial information (*i.e.* the road network), which has a wide range of applications including traffic signal control (Jiang et al., 2021), route planning (Liebig et al., 2017), and congestion management (Fouladgar et al., 2017).

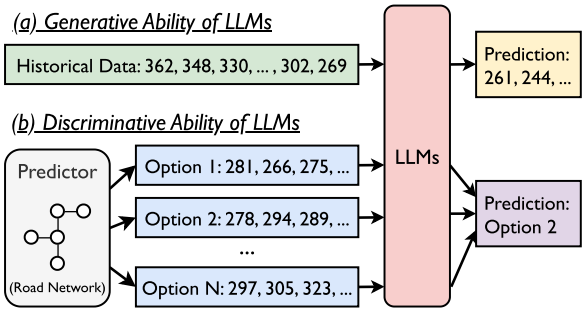

Figure 1: To use LLMs for traffic flow forecasting, a naive solution is to utilize their *generative* ability (a), which is hard to incorporate spatial relations. By comparison, LEAF utilizes the discriminative ability of LLMs (b), making it easier for LLMs to predict.

Due to its value in real-world applications, great efforts have been made to resolve the problem of traffic flow forecasting (Smith and Demetsky, 1997; Sun et al., 2006; Guo et al., 2019; Li and Zhu, 2021). Early works mainly model the traffic systems using physical rules or shallow models (Ghosh et al., 2009; Tchrakian et al., 2011; Hong et al., 2011; Li et al., 2012). With the advent of deep learning, the main-stream of traffic flow forecasting methods utilizes graph neural networks (Kipf and Welling, 2016; Hamilton et al., 2017; Xu et al., 2018; Veličković et al., 2018), recurrent neural networks (Hochreiter, 1997; Chung et al., 2014), and transformers (Vaswani, 2017; Jiang et al., 2023) to capture the rich spatio-temporal relations (Yu et al., 2018; Li et al., 2018; Guo et al., 2019; Li and Zhu, 2021; Zhang et al., 2021; Chen et al., 2022a; Liu et al., 2023; Ma et al., 2024).

Despite their success, existing traffic flow forecasting methods have two limitations, which hinder their applications in the real world.

Firstly, existing methods are generally unable to adapt to environmental changes of traffic conditions during test time. Most existing methods make the assumption that test data follow the same distri-

† Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2025, pages 8108–8123
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

bution as the training data, which may fail to hold in the real world (Lu et al., 2022; Zhao et al., 2025a,c), especially for time series data (Kim et al., 2021; Fan et al., 2023; Chen et al., 2024c; Zhang et al., 2024). Traffic conditions change over time due to a variety of factors like special events, changes in weather, or the shift of eras. Traditional models (Song et al., 2020; Li and Zhu, 2021), including fusion models (Kashinath et al., 2021) have poor generalization ability, suffering from performance degradation under distribution shifts. In contrast, LLMs have **strong generalization ability** to adapt well under such distribution shifts (Chang et al., 2024; Minaee et al., 2024; Beniwal et al., 2024). Moreover, LLMs have **strong reasoning ability**, which enables them to infer the current environment from the data to benefit forecasting. Due to the **high cost of collecting large-scale data**, the zero-shot generalization/reasoning ability is especially useful. To utilize LLMs, a naive solution is to use the *generative* ability of LLMs to make direct predictions (Li et al., 2024c; Ren et al., 2024; Liang et al., 2024), as shown in Figure 1 (upper part). However, this is too challenging for language models, as accurate forecasting relies on both historical data and complex spatio-temporal relations.

Secondly, existing methods are weak in capturing the rich structure of spatio-temporal relations in traffic data. The traffic network is complicated and the temporal dimension adds another layer of complexity. Prior works focus on capturing the complex spatio-temporal relations, using graph structures (Song et al., 2020; Zheng et al., 2023a) or hypergraph structures (Wang et al., 2022; Wang and Zhu, 2022; Zhao et al., 2023). Graphs capture pair-wise relations, while hypergraphs model non-pair-wise relations. Adopting only one of them is not enough, as the spatio-temporal relations in traffic data are rich by nature. For example, traffic congestion at one vertex affects adjacent vertices (pair-wise relations), whereas road closures affect a large set of vertices (non-pair-wise relations). Modeling the rich structure of spatio-temporal relations is challenging in predicting future traffic flows.

To that end, this paper proposes a novel method termed Large Language Model Enhanced Traffic Flow Predictor (LEAF) for adaptive and structure-perspective traffic flow forecasting. The core idea of LEAF is to utilize the discriminative ability of LLMs to enhance the task of traffic flow forecasting using a predictor and a selector, where the predictor generates predictions and the selector chooses the most likely result. To enhance adaptability, we build an LLM-based selector that selects from a range of possible future traffic flows using the discriminative ability of an LLM, as shown in Figure 1 (lower part). The selection results are used to guide the predictor with a ranking loss (Weinberger and Saul, 2009; Sohn, 2016), which only requires that the positive candidate (the ones chosen by the LLM) is better than the negative candidates (the ones not chosen). The LLM is good at understanding the changing traffic conditions and is open to further information provided by humans, making it an adaptable predictor. To better capture the rich structures of spatio-temporal relations, we build a dual-branch predictor composed of a graph branch which captures pair-wise relations of spatio-temporal traffic data, and a hypergraph branch, which captures non-pair-wise relations. During test time, the dual-branch predictor generates different forecasting results, and subsequently, a set of transformations is applied to obtain a wealth of choices of future traffic flows.

Our contribution is summarized as follows:

- We propose an LLM-enhanced traffic flow forecasting framework that introduces LLMs in test time to enhance the adaptability under distribution shift of traffic flow forecasting models.
- We propose a dual-branch predictor that captures both pair-wise and non-pair-wise relations of spatio-temporal traffic data, and an LLM-based selector that chooses from possible prediction results generated by the predictor. The selection results further guide the adaptation of the predictor with a ranking loss.
- Extensive experiments on several datasets verify the effectiveness of the proposed method.

2 Related Works

2.1 Traffic Flow Forecasting

Traffic flow forecasting is a topic that has been studied for several decades (Smith and Demetsky, 1997; Sun et al., 2006; Yang et al., 2016; Song et al., 2020; Li et al., 2024c,a). Early efforts mainly focus on traditional models (Smith and Demetsky, 1997; Asadi et al., 2012). With the success of deep learning, deep neural networks have become mainstream in this field. One line of research adopts recurrent neural networks (RNNs) (Hochreiter, 1997) and graph neural networks (GNNs) (Kipf and Welling, 2016), where the GNNs and RNNs capture the spa-

tial and temporal relations, respectively (Li et al., 2018; Wang et al., 2020; Chen et al., 2022b; Li et al., 2023; Weng et al., 2023).

To jointly model spatial and temporal relations, another line of research utilizes GNNs in both dimensions (Song et al., 2020; Li and Zhu, 2021; Lan et al., 2022; Chen et al., 2024a). As simple graphs only capture pair-wise relations, some works take a step further, introducing hypergraph neural networks (HGNNs) to capture non-pair-wise spatio-temporal relations (Luo et al., 2022; Wang and Zhu, 2022; Sun et al., 2022; Zhao et al., 2023; Wang et al., 2024). In this work, we take advantage of the benefits from both sides, designing a dual-branch architecture, which better captures the rich structures of spatio-temporal relations.

2.2 Large Language Models

In recent years, large language models have drawn increased attention, both within the community of natural language processing (Chen et al., 2024b; Yin et al., 2024; Luo et al., 2025) and beyond, including healthcare (Liévin et al., 2024; Van Veen et al., 2024; Labrak et al., 2024), education (Milano et al., 2023), legal technology (Lai et al., 2024), economics (Li et al., 2024b), recommendation (Chen et al., 2024b; Bao et al., 2024), code understanding (Zhao et al., 2025b), and transportation (Liu et al., 2024a; Guo et al., 2024b; Ren et al., 2024).

Among these applications, traffic flow forecasting is an important one, as it serves as the foundation of many downstream tasks in the field of intelligent transportation systems (Boukerche et al., 2020). The success of LLMs in language has inspired a plethora of works in this task. Some works adopt the architecture of LLMs for building transformer-based traffic flow predictors (Cai et al., 2020; Xu et al., 2020; Chen et al., 2022a; Liu et al., 2023; Jiang et al., 2023; Zou et al., 2024). However, they typically require a large amount of data for training.

Another line of research attempts to equip LLMs with the ability to predict future traffic flows based on the history and specific situations (Zheng et al., 2023b; Guo et al., 2024a; Li et al., 2024c; Yuan et al., 2024; Han et al., 2024). Although LLMs have shown promising results in understanding time series (Yu et al., 2023; Jin et al., 2023; Gruver et al., 2024; Koval et al., 2024; Gruver et al., 2024; Ansari et al., 2024; Liu et al., 2024b; Woo et al., 2024) or temporal events (Xia et al., 2024; Hu et al., 2024), the traffic data involves complex spatio-temporal relations challenging LLMs' *generative* ability. In this work, we show that one can instead utilize their *discriminative* ability to enhance existing traffic flow forecasting models.

3 Methodology

Problem Setup. We follow the standard setup for traffic flow forecasting (Song et al., 2020), where there is a road network denoted as $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \boldsymbol{A} \rangle$. $\mathcal{V}$ denotes the set of N vertices (*i.e.* the sensors in the city), $\mathcal{E}$ denotes the set of edges, and $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ represents the adjacency matrix. In this road network, historical traffic flows can be represented as $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_T) \in \mathbb{R}^{T \times N \times F}$, in which T is the length of historical observation and F is the dimension of input features. The goal is to predict the future of traffic flows with the length of T', denoted as $\boldsymbol{X}' = (\boldsymbol{X}'_1, \boldsymbol{X}'_2, \cdots, \boldsymbol{X}'_{T'}) \in \mathbb{R}^{T' \times N \times F}$.

Framework Overview. To solve the problem with the help of LLMs, we propose a novel framework termed Large Language Model Enhanced Traffic Flow Predictor (LEAF), whose overview is illustrated in Figure 2. Specifically, to achieve LLM-enhanced prediction (Section 3.3), we design a dual-branch traffic flow predictor (Section 3.1) and an LLM-based selector (Section 3.2). The predictor consists of a graph neural network branch capturing pair-wise spatio-temporal relations, and a hypergraph neural network branch capturing non-pair-wise relations. During training, the predictor is first pretrained using the training data. During test time, we apply transformations to the forecasting results of the predictor to obtain a variety of choices, among which the selector chooses the best one with a frozen LLM. The selection results are then used to fine-tune the dual-branch predictor.

3.1 Dual-branch Traffic Flow Predictor

Previous works on traffic flow forecasting adopt *the graph perspective* (Song et al., 2020; Zheng et al., 2023a) or *the hypergraph perspective* (Wang et al., 2022; Zhao et al., 2023). The graph perspective propagates messages between pairs of nodes, which makes them adept in modeling pair-wise spatio-temporal relations (*e.g.* an accident affects adjacent locations). On the other hand, the hypergraph perspective propagates messages among groups of nodes, making them proficient in modeling non-pair-wise spatio-temporal relations (*e.g.* people move from the residential area to the busi-

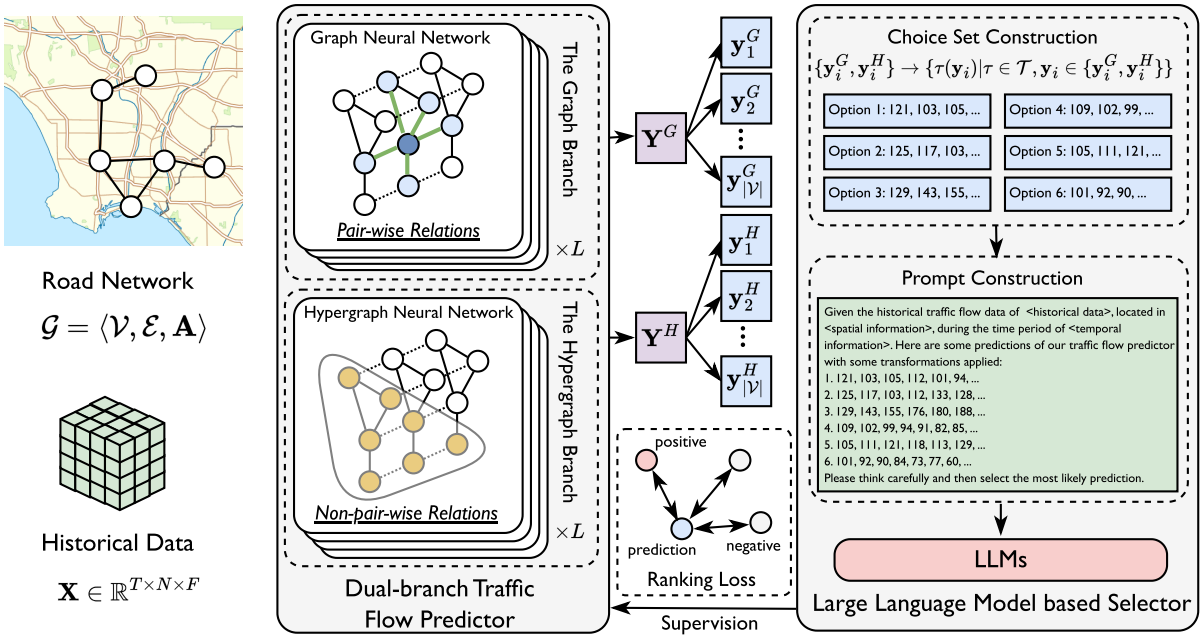

Figure 2: The framework of the proposed LEAF, consisting of a dual-branch predictor and an LLM-based selector. The predictor generates forecasts of traffic flows with the graph and hypergraph branches. The selector constructs a choice set and selects the best option using an LLM. The selection results are used to supervise the predictor.

ness area). For LEAF, we aim to take the benefits from both sides by constructing a dual-branch predictor and letting the LLM select.

Spatio-temporal Graph Construction. To utilize graph neural networks and hypergraph neural networks, we first construct a spatio-temporal graph corresponding to the input tensor $\boldsymbol{X} \in \mathbb{R}^{T \times N \times F}$. Particularly, the length of historical data T yields a set of TN spatio-temporal nodes, denoted as:

$$\mathcal{V}^{ST} = \left\{ v_i^t \mid i = 1, \ldots, N, t = 1, \ldots, T \right\}, \quad (1)$$

and we add temporal edges in addition to spatial edges $\mathcal{E}$ to obtain the edge set $\mathcal{E}^{ST}$:

$$\mathcal{E}^{ST} = \big\{ \langle v_i^{t_1}, v_j^{t_2} \rangle \mid (|t_1 - t_2| = 1 \wedge i = j) \vee (t_1 = t_2 \wedge \langle i, j \rangle \in \mathcal{E}) \big\}. \quad (2)$$

The spatio-temporal graph can then be represented as $\mathcal{G}^{ST} = \langle \mathcal{V}^{ST}, \mathcal{E}^{ST}, \boldsymbol{A}^{ST} \rangle$, where $\boldsymbol{A}^{ST} \in \mathbb{R}^{TN \times TN}$. The spatio-temporal features $\boldsymbol{X}^{(0)} \in \mathbb{R}^{TN \times d}$ is derived from $\boldsymbol{X} \in \mathbb{R}^{T \times N \times F}$ with a linear mapping and a reshape operation, where d is the dimension of the hidden space.

The Graph Branch. Based on the constructed spatio-temporal graph, we first adopt a graph neural network to model pair-wise spatio-temporal relations. Concretely, given the spatio-temporal feature inputs $\boldsymbol{X}^{(0)} \in \mathbb{R}^{TN \times d}$, we adopt convolution layers to process the features, which is

$$\boldsymbol{X}^{(l)} = \sigma\left(\widehat{\boldsymbol{A}^{ST}} \boldsymbol{X}^{(l-1)} \boldsymbol{W}_G^{(l)} \right), \quad (3)$$

where $\sigma(\cdot)$ is the activation layer and $\boldsymbol{W}_G^{(l)} \in \mathbb{R}^{d \times d}$ is the weight matrix. $\widehat{\boldsymbol{A}^{ST}} = \boldsymbol{D}^{-1/2} \boldsymbol{A}^{ST} \boldsymbol{D}^{-1/2}$ denotes the normalized version of the adjacency matrix, where $\boldsymbol{D}$ is the degree matrix (Kipf and Welling, 2016; Song et al., 2020). By adopting graph convolutions in Eq. 3, the information from one spatio-temporal vertex can be propagated to its neighbors as defined in $\mathcal{E}^{ST}$ in Eq. 2, and thus this branch models pair-wise spatio-temporal relations.

The Hypergraph Branch. Although the graph branch is adept at capturing pair-wise relations, the complex traffic patterns contain non-pair-wise relations. For example, in the morning rush hours, people move from the residential area (which is a set of vertices in a hyperedge) to the business area (which is another hyperedge). The vertices in one hyperedge share common patterns and the hyperedges affect each other. To model non-pair-wise relations, hypergraphs are adopted. For a hypergraph, its incidence matrix $\boldsymbol{I}_H \in \mathbb{R}^{NT \times m}$ describes the assignment of NT vertices to m hyperedges. As the incidence matrix is not given as input, we resort to a learnable one with low-rank decomposition (Zhao et al., 2023):

$$\boldsymbol{I}_H = \mathrm{softmax}(\boldsymbol{X}_H^{(l-1)} \boldsymbol{W}_H), \quad (4)$$

where $\boldsymbol{X}_H^{(l-1)} \in \mathbb{R}^{NT \times d}$ is the hidden features, and $\boldsymbol{W}_H \in \mathbb{R}^{d \times m}$ is the weight matrix. $\mathrm{softmax}(\cdot)$ is

System Prompt:
You are a helpful assistant.
User Prompt:
Given historical data for traffic flow over 12 time steps at <spatial information of the vertex>, the recorded traffic flows are <the historical data>, from <history start time> to <history end time>. The data points are recorded at 5-minute intervals. We aim to predict the traffic flow in the next 12 time steps from <future start time> to <future end time>. <special events>. We use two branches: the graph branch (description: ...) and the hypergraph branch (description: ...) to predict the future traffic flow and perform some transformations to the predictions. Here are some choices for you. Please consider historical values, models' predictions, and the time of the data (e.g. rush hours).
1. <choice content>, generated by <branch name>, with transformation <transformation name>.
2. ...
...
Please first think carefully and then select the most likely one:

Figure 3: An illustration of the prompt template.

applied for normalization. Subsequently, the output features can be computed as:

$$\boldsymbol{X}_H^{(l)} = \boldsymbol{I}_H \left(\boldsymbol{I}_H^\top \boldsymbol{X}_H^{(l-1)} + \sigma \left(\boldsymbol{W}_E \boldsymbol{I}_H^\top \boldsymbol{X}_H^{(l-1)} \right) \right), \tag{5}$$

where $\boldsymbol{W}_E \in \mathbb{R}^{m \times m}$ models the interactions of the hyperedges. In this way, the hypergraph branch considers both (a) the interactions within a set of vertices (within a hyperedge) through the first term of Eq. 5, and (b) the interactions among groups of vertices (among the hyperedges) through the second term of Eq. 5.

3.2 LLM-Based Selector

In Section 3.1, we obtain different prediction results from different branches, denoted as $\boldsymbol{Y}^G$ and $\boldsymbol{Y}^H$. Most previous works (Li et al., 2024c; Ren et al., 2024; Liang et al., 2024) directly generate these predictions, which is challenging since the complex spatio-temporal relations are hard to express in texts. By comparison, the LLM-based selector aims to choose the best prediction, using the internal knowledge of a frozen LLM and the constructed prompts. As the traffic networks are usually large, we break the result $\boldsymbol{Y} \in \{\boldsymbol{Y}^G, \boldsymbol{Y}^H\} \subset \mathbb{R}^{T' \times N}$ for individual vertices, *i.e.* $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N]$, where $\boldsymbol{y}_i \in \mathbb{R}^{T'}$.

Choice Set Construction. In practice, we want to give the LLM-based selector more choices so that it has the potential to make better predictions. Therefore, we introduce several transformations: smoothing, upward trend, downward trend, overestimating, and underestimating. The set of transformations is denoted as $\mathcal{T}$. For vertex $i \in \mathcal{V}$, the choice set is determined as follows:

$$\mathcal{C}_i = \{\tau(\boldsymbol{y}_i) | \tau \in \mathcal{T}, \boldsymbol{y}_i \in \{\boldsymbol{y}_i^G, \boldsymbol{y}_i^H\}\} \cup \{\boldsymbol{y}_i^G, \boldsymbol{y}_i^H\}. \tag{6}$$

By adopting transformations, the choice set is expanded and the selector has the potential to deal with more complex situations. For instance, if it believes that both of the branches underestimate the traffic flow on a Monday morning, the selector can choose an option with an upward trend.

Prompt Construction. When constructing the prompt, we consider the following aspects. (1) General information about the data, including the meaning of the numbers, the way traffic data are selected, etc. (2) The spatial information of the vertex, including the sensor ID, and the geometric location information. (3) The temporal information, including the time historical data are collected, the time we want to forecast, and whether there are special events. (4) The historical data. (5) The task instructions. (6) The choice set constructed in Eq. 6, including the names of specific branches and the description of augmentations. An illustration of this is shown in Figure 3.

3.3 LLM-Enhanced Prediction

The LEAF framework consists of a predictor and a selector. The predictor (the graph branch and the hypergraph branch) is pre-trained on the training set. During test time, the predictor first predicts, and the selector then selects. The selection results are used to supervise the predictor, and thus the two modules benefit from each other, achieving LLM-enhanced prediction. Concretely, given the input data $\boldsymbol{X} \in \mathbb{R}^{T \times N \times F}$, the predictor generates two forecasts, *i.e.* $\boldsymbol{Y}^G$ and $\boldsymbol{Y}^H$. Subsequently, the selector constructs choice sets and uses the LLM to find the best option for each individual vertex. The selection results are denoted as $\hat{\boldsymbol{y}}_i, i \in \mathcal{V}$, which are then used to supervise the predictor. Conceivably, $\hat{\boldsymbol{y}}_i$ may not be the optimal choice in the choice set, and therefore, directly using MAE or MSE losses may lead to noises in the supervision signals. By comparison, the ranking loss (Weinberger and Saul, 2009; Sohn, 2016) only requires that the positive candidate (the ones chosen by the LLM) is better than the negative candidates (the ones not chosen), described as follows:

$$\mathcal{L}^G = [\Delta(\boldsymbol{y}_i^G, \hat{\boldsymbol{y}}_i) - \inf_{\boldsymbol{y}_i' \in \mathcal{C}_i \setminus \{\hat{\boldsymbol{y}}_i\}} \Delta(\boldsymbol{y}_i^G, \boldsymbol{y}_i') + \epsilon]_+, \tag{7}$$

Algorithm 1 The Algorithm of LEAF

Requires: The road network $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \boldsymbol{A} \rangle$, historical data $\boldsymbol{X}$, the number of iterations K, the graph branch $\mathcal{B}^G$, and the hypergraph branch $\mathcal{B}^H$.
Ensures: The forecasting result $\hat{\boldsymbol{Y}}$.

```
Construct spatio-temporal graph G^ST.
for j = 1, 2, ..., K do
    Compute the prediction of the graph branch B^G as Y^G.
    Compute the prediction of the hypergraph branch B^H as Y^H.
    Construct the choice set using Eq. 6.
    Use LLM to select the best option for each vertex as ŷ_i where i ∈ V.
    Use ŷ_i to supervise the predictor (B^G and B^H) with Eq. 7 and Eq. 8.
end for
Stack ŷ_i, i ∈ V to obtain Ŷ
```

where $[\cdot]_+$ is the hinge function, $\Delta(\cdot, \cdot)$ is a distance measure, and ϵ is the margin. Similarly, we can define the loss function $\mathcal{L}^H$ using $\boldsymbol{y}_i^H$. The final objective is written as:

$$\mathcal{L} = \mathcal{L}^G + \mathcal{L}^H. \tag{8}$$

In Eq. 7 and Eq. 8, we encourage the forecasts of the predictor (*i.e.* $\boldsymbol{y}_i^G$ and $\boldsymbol{y}_i^H$) to be closer to the selected prediction (*i.e.* $\hat{\boldsymbol{y}}_i$) than the closest one in suboptimal predictions (*i.e.* $\mathcal{C}_i \setminus \{\hat{\boldsymbol{y}}_i\}$). Since the ground truth may not be covered by the choice set, this objective is better than directly minimizing the distance between the predictions and the selected forecast, as it allows the model to learn from a better choice compared to suboptimal choices.

By supervising the predictor, the two modules (*i.e.* the predictor and the selector) benefit from each other. A better predictor yields better choices, which benefits the selector; better selection results provide better supervision signals for the predictor. Through the iteration of prediction and selection, we can achieve LLM-enhanced prediction. The LEAF framework during inference is summarized in Algorithm 1.

4 Experiments

4.1 Experimental Setup

Datasets. We adopt three widely used datasets in traffic flow forecasting, including PEMS03, PEMS04, and PEMS08. The datasets are publicly available and collected by California Transportation Agencies (CalTrans) Performance Measurement Systems (PEMS) [1]. These records come from sensors on the roads of various places in California, and they are counted every five minutes. Additional details about the datasets can be found in Appendix A.

Evaluation Metrics. We follow standard setting (Li et al., 2018) that use one-hour historical data (*i.e.* $T = 12$ timesteps) to forecast one-hour future (*i.e.* $T' = 12$ timesteps). To signify the test-time distribution shift, we adopt small training sets of 10% of data and another 10% for validation. We choose a subset of non-overlapping slices in the test set. We adopt three standard metrics for evaluation, i.e., Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) (Song et al., 2020). Formal definitions about these metrics are presented in Appendix B.

Baseline Methods. We compare LEAF with a variety of baselines, including DCRNN (Li et al., 2018), ASTGCN (Guo et al., 2019), STSGCN (Song et al., 2020), HGCN (Wang et al., 2022), DyHSL (Zhao et al., 2023), STAEformer (Liu et al., 2023), COOL (Ju et al., 2024), and LLM-MPE (Liang et al., 2024). Among these methods, DCRNN is a combination of GNN and RNN. ASTGCN, STSGCN, and COOL are based on GNNs, while HGCN and DyHSL are based on hypergraph neural networks. STAEformer uses the transformer architecture, while LLM-MPE uses LLMs to understand and predict traffic data with textual guidance. Additional information about the baseline methods can be found in Appendix C.

Implementation Details. For the dual-branch predictor, the number of layers L for both branches is set to 7. We use linear mapping as the input embedding, and the hidden dimension d is set to 64, which is also shared across all baselines. We also use a two-layer MLP to map the last hidden embedding to the output. In choice set construction, smoothing is implemented with an average filter, upward/downward trend increases/decreases the traffic flow by 1% to 12% (12 timesteps, linearly increasing), overestimate/underestimate increases/decreases the traffic flow by 5% (for all 12 timesteps). As for the loss function in Eq. 7, we adopt Huber distance for $\Delta(\cdot, \cdot)$ and the margin ϵ is set to 0. When training with this loss function,

[1] https://pems.dot.ca.gov/

Model	PEMS03			PEMS04			PEMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
DCRNN (Li et al., 2018)	29.99	39.52	21.33	34.36	46.19	24.73	31.41	43.91	15.44
ASTGCN (Guo et al., 2019)	28.4	41.94	15.78	33.09	46.08	18.19	29.20	41.16	12.76
STSGNN (Song et al., 2020)	28.21	43.43	15.49	33.43	45.69	18.89	29.58	41.95	12.90
HGCN (Wang et al., 2022)	28.43	43.92	15.39	35.77	49.92	20.11	28.83	39.65	12.63
DyHSL (Zhao et al., 2023)	27.10	41.59	14.31	33.36	46.96	19.64	27.34	39.05	11.56
STAEformer (Liu et al., 2023)	27.87	37.31	16.49	33.77	45.50	18.36	27.43	38.16	11.36
COOL (Ju et al., 2024)	27.51	41.11	14.73	34.68	47.22	19.72	27.22	38.47	11.72
LLM-MPE (Liang et al., 2024)	33.82	47.06	20.40	35.63	51.41	18.19	26.42	40.02	10.61
LEAF	**25.46**	**35.17**	**14.22**	**31.49**	**44.45**	**17.53**	**24.68**	**36.07**	**10.56**

Table 1: Forecasting errors on PEMS03, PEMS04, and PEMS08 datasets.

we update the parameters for M iterations, where M is set to 5. For the prediction-selection loop, we set K to 2. For the LLM, we use LLaMA 3 70B (AI@Meta, 2024) and vLLM (Kwon et al., 2023) as the inference software. Additional details about the experiments are presented in Appendix D.

4.2 Main Results

The performance of LEAF in comparison with prior methods is shown in Table 1. According to the metrics in the table, we have several observations:

Firstly, the proposed LEAF demonstrates consistent improvement in all three datasets, showing the effectiveness of the proposed framework that adopts a dual-branch traffic flow predictor and an LLM-based selector. Smaller models often fail to provide satisfactory predictions under test-time distribution changes, as they fall short in reason and generalization, leaving room for improvement.

Secondly, the methods that utilize the generative ability of LLMs (*i.e.* LLM-MPE) do not perform well on all datasets. As we can see on the PEMS03 and PEMS04 datasets, their predictions are generally worse or similar compared to simple methods using graph neural networks (*e.g.* STSGCN, COOL). Since LLMs are not adept at capturing complex spatio-temporal relations, it is reasonable that we see such results on datasets with larger networks like PEMS03.

Thirdly, the proposed LEAF generally performs better than graph-based methods (*e.g.* ASTGCN, STSGCN) and hypergraph-based methods (*e.g.* HGCN, DyHSL), which shows that our method has the potential to take advantage of both complex spatio-temporal relations captured by the predictor and the knowledge of large language models, achieving LLM enhanced traffic flow forecasting.

Experiments	MAE	RMSE	MAPE
E1: Graph branch	29.12	41.36	13.54
E2: Hypergraph branch	27.94	39.11	11.82
E3: *w/o* hypergraph branch	26.29	38.18	12.83
E4: *w/o* graph branch	25.80	37.23	11.00
E5: *w/o* transformation	25.47	36.47	11.01
E6: *w/o* ranking loss	25.41	37.00	11.34
LEAF	**24.68**	**36.07**	**10.56**

Table 2: Ablation study on the PEMS08 dataset.

4.3 Ablation Study

We also perform ablation studies on the PEMS08 dataset, and the results are shown in Table 2. Specifically, we perform the following experiments. **E1** measures the performance of the graph branch only without the LLM-based selector. **E2** measures the performance of the hypergraph branch without the selector. The vanilla version (E1 and E2) of both branches performs much worse than LEAF. **E3** uses the graph branch in conjunction with the selector, which leads to performance degradation compared to LEAF. This suggests that non-pair-wise relations are important in traffic flow forecasting. **E4** only uses the hypergraph branch together with the selector. Similarly, it performs worse than LEAF, which shows that pair-wise and non-pair-wise relations are both important. Moreover, E3 and E4 perform better than E1 and E2, which shows the effect of the LLM-based selector. **E5** removes the transformations, *i.e.* $\mathcal{T} = \emptyset$. This leads to slightly worse performance, showing the effectiveness of providing more choices. **E6** removes the ranking loss, which means that the predictor is not further trained with the results from the selector (reducing to $K = 1$). This experiment demonstrates the effectiveness of supervising the predictor with a

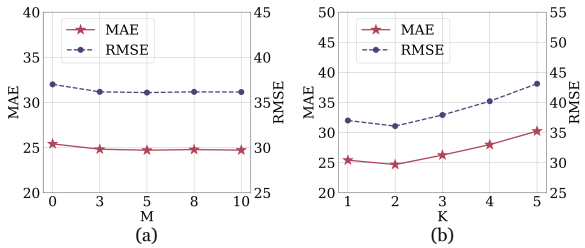

Figure 4: The forecasting errors under different hyper-parameters, *i.e.* Ms (left) and Ks (right).

ranking loss using the selection results.

4.4 Hyper-parameter Analysis

We perform experiments with respect to two hyper-parameters: **(a)** M, *i.e.* the number of iterations when training with the ranking loss in Eq. 7 and Eq. 8, and **(b)** the number of K, *i.e.* the number of prediction-selection iterations in Algorithm 1. The results on the PEMS08 dataset are shown in Figure 4. As can be seen from the figure, when M increases, both MAE and RMSE decrease, and then plateau after around 5. This shows that the predictor converges, and therefore, we set M to 5. For another hyper-parameter K, the optimal performance is achieved when K is set to 2, and with more iterations, the error increases. One reason is that we use the same prompt across iterations without contexts of previous selections to save computation, so the same factor may be considered multiple times, leading to decreased accuracy. In practice, $K = 2$ is good enough without too much computation for the LLM-based selector.

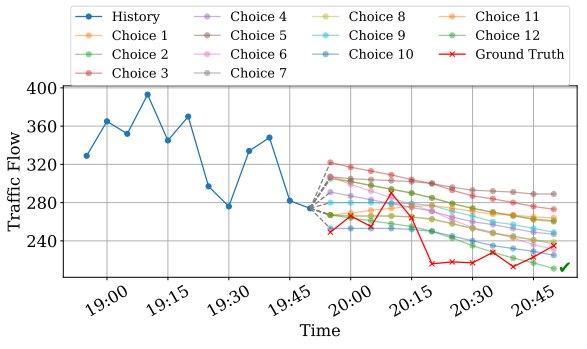

Figure 5: Visualization of historical data, choices, and the prediction in different time periods. The selected choice is marked with a green tick.

4.5 Visualization

Visualization of the Selection. We first provide visualizations of the historical traffic flow data, the choices in $\mathcal{C}_i$, and the option selected by the LLM-based selector (marked with a green tick). The results on the PEMS03 dataset are shown in Figure 5. Moreover, we also provide the analysis of the LLM-based selector, which is shown in Figure 6. According to the results, we can see that the traffic flow is generally going downward, since it is the end of the rush hour. In Figure 5, we can see that the LLM-based selector chooses the option with the lowest traffic flows. From its analysis in Figure 6, we can see that it understands that the time period to forecast is around the end of the rush hours, which is the reason why it selects the lowest option. This suggests that the LLM-based selector is able to understand changing traffic conditions. Another visualization example can be found in Appendix E.

To determine the most likely traffic flow for the next 12 time steps, we need to consider the historical data, the time of day, and the models' predictions.
First, let's analyze the historical data:
* The average traffic flow is 330.4, but there's a noticeable decrease in traffic flow from 18:55 to 19:50, with a minimum value of 274.
* The time period is during the afternoon rush hour, and the traffic flow is decreasing, indicating that the rush hour might be ending.
Considering the models' predictions and augmentations:
* The graph branch predictions are generally higher than the hypergraph branch predictions.
* The augmentations (smoothing, 5% higher/lower, upward/downward trend) provide different variations of the predictions.
Given the time of day and the historical data, we can make an educated guess:
* Since the rush hour is likely ending, the traffic flow might continue to decrease.
* The graph branch predictions seem too high, considering the historical data's decreasing trend.
* The hypergraph branch predictions with a downward trend (option 12) seem more plausible, as they align with the expected decrease in traffic flow during the end of the rush hour.
Therefore, the most likely traffic flow for the next 12 time steps is: Option 12: [267, 264, 261, 258, 255, 250, 243, 235, 228, 222, 217, 211] (average: 242.6), Hypergraph branch, Augmentation: prediction with downward trend

Figure 6: The LLM's analysis when selecting data.

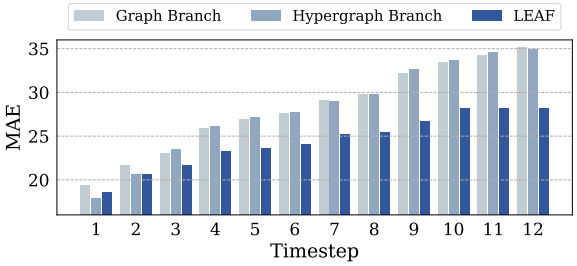

Figure 7: The Mean Absolute Error (MAE) under different timesteps. LEAF better reduces long-term errors.

Visualization of Errors in Different Timesteps. We then provide visualization of the mean absolute error (MAE) under different forecasting timesteps

Prompt:
Given historical data for traffic flow over 12 time steps at sensor <sensor id> of District <district number>, California (<spatial information of the vertex>), the recorded traffic flows are <historical data> (average:<average of historical data>), from <history start time> to <history end time>, with data points recorded at 5-minute intervals. We aim to predict the traffic flow in the next 12 time steps from <future start time> to <future end time>. We use two branches: the graph branch (which captures pair-wise relations), the hypergraph branch (which captures non-pair-wise relations) to predict the future traffic flow and perform some augmentations (transformations) to the predictions. Note that smoothing does not reduce MAE error. Here are some choices for you. Please consider: (1) historical values, (2) models' predictions, and (3) the time of the data (most importantly, whether they are in beginning / end of the morning / afternoon rush hours. Note that different nodes have different rush hours, if you have seen significant changes in history, you can identify the beginning / end of the rush hour).
Please first think carefully and then select the most likely one:
1. <choice data> (average: <average value>), the graph branch, Augmentation: none
2. <choice data> (average: <average value>), the graph branch, Augmentation: smoothing the prediction of the model
3. <choice data> (average: <average value>), the graph branch, Augmentation: 5% higher than the prediction
4. <choice data> (average: <average value>), the graph branch, Augmentation: 5% lower than the prediction
5. <choice data> (average: <average value>), the graph branch, Augmentation: prediction with upward trend
6. <choice data> (average: <average value>), the graph branch, Augmentation: prediction with downward trend
7. <choice data> (average: <average value>), the hypergraph branch, Augmentation: none
8. <choice data> (average: <average value>), the hypergraph branch, Augmentation: smoothing the prediction of the model
9. <choice data> (average: <average value>), the hypergraph branch, Augmentation: 5% higher than the prediction
10. <choice data> (average: <average value>), the hypergraph branch, Augmentation: 5% lower than the prediction
11. <choice data> (average: <average value>), the hypergraph branch, Augmentation: prediction with upward trend
12. <choice data> (average: <average value>), the hypergraph branch, Augmentation: prediction with downward trend

Figure 8: The details about the prompt.

in Figure 7. The experiments are performed on the PEMS03 dataset, where we compare the MAE values of our framework to its two branches (*i.e.* the graph branch and the hypergraph branch). The results show that LEAF reduces forecasting errors generally. Although the errors are similar in the first few timesteps, LEAF quickly diverges from the two branches, resulting in significantly lower errors in the long term. This suggests that with the help of the discriminative ability of the LLM-based selector, our method can select predictions that are more accurate in the long run.

Visualization of the Prompt. We provide an example of the prompt in Figure 8. The prompt contains general information about the task and the traffic data, spatio-temporal information, the historical data, the task instructions, and the choice set constructed in Eq. 6. Additionally, we also describe the transformations applied to the prediction of the dual-branch predictor. These sources of information will be helpful for the LLM to decide the most appropriate choice.

5 Conclusion

In this paper, we propose a novel framework named Large Language Model Enhanced Traffic Flow Predictor (LEAF), consisting of a dual-branch traffic flow predictor and an LLM-based selector. The predictor adopts two branches: the graph branch and the hypergraph branch, capturing the pair-wise and non-pair-wise relations, respectively. The selector uses the discriminative ability of the LLM to choose the best forecast of the predictor. The selection results are then used to supervise the predictor. We perform extensive experiments to demonstrate the effectiveness of LEAF.

Limitations

One limitation of this work is that we only focus on the traffic flow forecasting domain, due to the scope of this paper and the limited computational resources. However, this framework can be extended to more generalized spatio-temporal forecasting problems. Besides, we have not evaluated our methods on other traffic flow forecasting datasets due to limited resources.

Another limitation is that the LLM is not fine-tuned during the process. The LLM selector in the LEAF framework can be further optimized throughout the process. Since the traffic datasets are relatively small, a potential solution is to use parameter-efficient fine-tuning strategies, including LoRA, adapter layers, or prefix-tuning to enhance the performance of the LLM selector. More specifically, given the final prediction results from the iterative prediction-selection cycles, the LLM is then fine-tuned with the answer (in the choices) nearest to the final prediction. This will potentially enhance the ability of the LLM selector.

Acknowledgment

This paper is partially supported by grants from the National Key Research and Development Program of China with Grant No. 2023YFC3341203 and the National Natural Science Foundation of China (NSFC Grant Number 62276002). The authors are grateful to the anonymous reviewers for critically reading this article and for giving important suggestions to improve this article.

References

AI@Meta. 2024. Llama 3 model card.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

Shahrokh Asadi, Akbar Tavakoli, and Seyed Reza Hejazi. 2012. A new hybrid for improvement of autoregressive integrated moving average models applying particle swarm optimization. *Expert Systems with Applications*, 39(5):5332–5337.

Keqin Bao, Jizhi Zhang, Xinyu Lin, Yang Zhang, Wenjie Wang, and Fuli Feng. 2024. Large language models for recommendation: Past, present, and future. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2993–2996.

Himanshu Beniwal, Dishant Patel, D Kowsik, Hritik Ladia, Ankit Yadav, and Mayank Singh. 2024. Remember this event that year? assessing temporal information and understanding in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16239–16348.

Azzedine Boukerche, Yanjie Tao, and Peng Sun. 2020. Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems. *Computer networks*, 182:107484.

Ling Cai, Krzysztof Janowicz, Gengchen Mai, Bo Yan, and Rui Zhu. 2020. Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS*, 24(3):736–755.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Changlu Chen, Yanbin Liu, Ling Chen, and Chengqi Zhang. 2022a. Bidirectional spatial-temporal adaptive transformer for urban traffic flow forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, 34(10):6913–6925.

Jian Chen, Li Zheng, Yuzhu Hu, Wei Wang, Hongxing Zhang, and Xiping Hu. 2024a. Traffic flow matrix-based graph neural network with attention mechanism for traffic flow prediction. *Information Fusion*, 104:102146.

Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. 2024b. When large language models meet personalization: Perspectives of challenges and opportunities. *World Wide Web*, 27(4):42.

Ling Chen, Wei Shao, Mingqi Lv, Weiqi Chen, Youdong Zhang, and Chenghu Yang. 2022b. Aargnn: An attentive attributed recurrent graph neural network for traffic flow prediction considering multiple dynamic factors. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):17201–17211.

Mouxiang Chen, Lefei Shen, Han Fu, Zhuo Li, Jianling Sun, and Chenghao Liu. 2024c. Calibration of time-series forecasting: Detecting and adapting context-driven distribution shift. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 341–352.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Fei Dai, Penggui Huang, Qi Mo, Xiaolong Xu, Muhammad Bilal, and Houbing Song. 2022. St-innet: Deep spatio-temporal inception networks for traffic flow prediction in smart cities. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):19782–19794.

George Dimitrakopoulos and Panagiotis Demestichas. 2010. Intelligent transportation systems. *IEEE Vehicular Technology Magazine*, 5(1):77–84.

Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. 2023. Dishts: a general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 7522–7529.

Mohammadhani Fouladgar, Mostafa Parchami, Ramez Elmasri, and Amir Ghaderi. 2017. Scalable deep traffic flow neural networks for urban traffic congestion prediction. In *2017 international joint conference on neural networks (IJCNN)*, pages 2251–2258. IEEE.

Bidisha Ghosh, Biswajit Basu, and Margaret O'Mahony. 2009. Multivariate short-term traffic flow forecasting using time-series analysis. *IEEE transactions on intelligent transportation systems*, 10(2):246–254.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2024. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 922–929.

Xusen Guo, Qiming Zhang, Junyue Jiang, Mingxing Peng, Meixin Zhu, and Hao Frank Yang. 2024a. Towards explainable traffic flow prediction with large language models. *Communications in Transportation Research*, 4:100150.

Xusen Guo, Qiming Zhang, Mingxing Peng, Meixin Zhua, et al. 2024b. Explainable traffic flow prediction with large language models. *arXiv preprint arXiv:2404.02937*.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Xiao Han, Zhenduo Zhang, Yiling Wu, Xinfeng Zhang, and Zhe Wu. 2024. Event traffic forecasting with sparse multimodal data. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8855–8864.

S Hochreiter. 1997. Long short-term memory. *Neural Computation MIT-Press*.

Wei-Chiang Hong, Yucheng Dong, Feifeng Zheng, and Shih Yung Wei. 2011. Hybrid evolutionary algorithms in a svr traffic flow forecasting model. *Applied Mathematics and Computation*, 217(15):6733–6747.

Qisheng Hu, Geonsik Moon, and Hwee Tou Ng. 2024. From moments to milestones: Incremental timeline summarization leveraging large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7232–7246.

Chun-Yao Jiang, Xiao-Min Hu, and Wei-Neng Chen. 2021. An urban traffic signal control system based on traffic flow prediction. In *2021 13th international conference on advanced computational intelligence (ICACI)*, pages 259–265. IEEE.

Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. 2023. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 4365–4373.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.

Wei Ju, Yusheng Zhao, Yifang Qin, Siyu Yi, Jingyang Yuan, Zhiping Xiao, Xiao Luo, Xiting Yan, and Ming Zhang. 2024. Cool: a conjoint perspective on spatio-temporal graph neural network for traffic forecasting. *Information Fusion*, 107:102341.

Shafiza Ariffin Kashinath, Salama A Mostafa, Aida Mustapha, Hairulnizam Mahdin, David Lim, Moamin A Mahmoud, Mazin Abed Mohammed, Bander Ali Saleh Al-Rimy, Mohd Farhan Md Fudzee, and Tan Jhon Yang. 2021. Review of data fusion methods for real-time and multi-sensor traffic flow analysis. *IEEE Access*, 9:51258–51276.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. 2021. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Ross Koval, Nicholas Andrews, and Xifeng Yan. 2024. Financial forecasting from textual and tabular time series. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8289–8300.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. *arXiv preprint arXiv:2402.10373*.

Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and S Yu Philip. 2024. Large language models in law: A survey. *AI Open*.

Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. 2022. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International conference on machine learning*, pages 11906–11917. PMLR.

Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. 2023. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data*, 17(1):1–21.

Hourun Li, Yusheng Zhao, Zhengyang Mao, Yifang Qin, Zhiping Xiao, Jiaqi Feng, Yiyang Gu, Wei Ju, Xiao Luo, and Ming Zhang. 2024a. Graph neural networks in intelligent transportation systems: Advances, applications and trends. *arXiv preprint arXiv:2401.0713*.

Mengzhang Li and Zhanxing Zhu. 2021. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4189–4196.

Nian Li, Chen Gao, Mingyu Li, Yong Li, and Qingmin Liao. 2024b. Econagent: large language model-empowered agents for simulating macroeconomic activities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15523–15536.

Shuangshuang Li, Zhen Shen, and Gang Xiong. 2012. A k-nearest neighbor locally weighted regression method for short-term traffic flow forecasting. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1596–1601. IEEE.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*.

Zhonghang Li, Lianghao Xia, Jiabin Tang, Yong Xu, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. 2024c. Urbangpt: Spatio-temporal large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5351–5362.

Yuebing Liang, Yichao Liu, Xiaohan Wang, and Zhan Zhao. 2024. Exploring large language models for human mobility prediction under public events. *Computers, Environment and Urban Systems*, 112:102153.

Thomas Liebig, Nico Piatkowski, Christian Bockermann, and Katharina Morik. 2017. Dynamic route planning with real-time traffic predictions. *Information Systems*, 64:258–265.

Valentin Liévin, Christoffer Egeberg Hother, Andreas Geert Motzfeldt, and Ole Winther. 2024. Can large language models reason about medical questions? *Patterns*, 5(3).

Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao. 2024a. Spatial-temporal large language model for traffic prediction. *arXiv preprint arXiv:2401.10134*.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. 2023. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 4125–4129.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2024b. Timer: Generative pre-trained transformers are large time series models. *arXiv preprint arXiv:2402.02368*.

Wang Lu, Jindong Wang, Xinwei Sun, Yiqiang Chen, and Xing Xie. 2022. Out-of-distribution representation learning for time series classification. *arXiv preprint arXiv:2209.07027*.

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, et al. 2025. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*.

Xiaoyi Luo, Jiaheng Peng, and Jun Liang. 2022. Directed hypergraph attention network for traffic forecasting. *IET Intelligent Transport Systems*, 16(1):85–98.

Ying Ma, Haijie Lou, Ming Yan, Fanghui Sun, and Guoqi Li. 2024. Spatio-temporal fusion graph convolutional network for traffic flow forecasting. *Information Fusion*, 104:102196.

Silvia Milano, Joshua A McGrane, and Sabina Leonelli. 2023. Large language models challenge the future of higher education. *Nature Machine Intelligence*, 5(4):333–334.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Yilong Ren, Yue Chen, Shuai Liu, Boyue Wang, Haiyang Yu, and Zhiyong Cui. 2024. Tpllm: A traffic prediction framework based on pretrained large language models. *arXiv preprint arXiv:2403.02221*.

Nimra Shahid, Munam Ali Shah, Abid Khan, Carsten Maple, and Gwanggil Jeon. 2021. Towards greener smart cities and road traffic forecasting using air pollution data. *Sustainable Cities and Society*, 72:103062.

Brian L Smith and Michael J Demetsky. 1997. Traffic flow forecasting: comparison of modeling approaches. *Journal of transportation engineering*, 123(4):261–266.

Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 914–921.

Shiliang Sun, Changshui Zhang, and Guoqiang Yu. 2006. A bayesian network approach to traffic flow forecasting. *IEEE Transactions on intelligent transportation systems*, 7(1):124–132.

Yanfeng Sun, Xiangheng Jiang, Yongli Hu, Fuqing Duan, Kan Guo, Boyue Wang, Junbin Gao, and Baocai Yin. 2022. Dual dynamic spatial-temporal graph convolution network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):23680–23693.

Tigran T Tchrakian, Biswajit Basu, and Margaret O'Mahony. 2011. Real-time traffic flow forecasting using spectral analysis. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):519–526.

Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, Eduardo Pontes Reis, Anna Seehofnerová, et al. 2024. Adapted large language models can outperform medical experts in clinical text summarization. *Nature medicine*, 30(4):1134–1142.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Jingcheng Wang, Yong Zhang, Yongli Hu, and Baocai Yin. 2024. Large-scale traffic prediction with hierarchical hypergraph message passing networks. *IEEE Transactions on Computational Social Systems*.

Jingcheng Wang, Yong Zhang, Lixun Wang, Yongli Hu, Xinglin Piao, and Baocai Yin. 2022. Multitask hypergraph convolutional networks: A heterogeneous traffic prediction framework. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):18557–18567.

Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. 2020. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020*, pages 1082–1092.

Yi Wang and Di Zhu. 2022. Shgcn: a hypergraph-based deep learning model for spatiotemporal traffic flow prediction. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, pages 30–39.

Kilian Q Weinberger and Lawrence K Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2).

Wenchao Weng, Jin Fan, Huifeng Wu, Yujie Hu, Hao Tian, Fu Zhu, and Jia Wu. 2023. A decomposition dynamic graph convolutional recurrent network for traffic forecasting. *Pattern Recognition*, 142:109670.

G Woo, C Liu, A Kumar, C Xiong, S Savarese, and D Sahoo. 2024. Unified training of universal time series forecasting transformers. arxiv 2024. *arXiv preprint arXiv:2402.02592*.

Yuwei Xia, Ding Wang, Qiang Liu, Liang Wang, Shu Wu, and Xiao-Yu Zhang. 2024. Chain-of-history reasoning for temporal knowledge graph forecasting. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 16144–16159.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? In *International Conference on Learning Representations*.

Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. 2020. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*.

Hao-Fan Yang, Tharam S Dillon, and Yi-Ping Phoebe Chen. 2016. Optimized structure of the traffic flow forecasting model with a deep learning approach. *IEEE transactions on neural networks and learning systems*, 28(10):2371–2381.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review*, page nwae403.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3634–3640.

Xinli Yu, Zheng Chen, and Yanbin Lu. 2023. Harnessing llms for temporal data-a study on explainable financial time series forecasting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 739–753.

Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. 2024. Unist: a prompt-empowered universal model for urban spatio-temporal prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4095–4106.

Junping Zhang, Fei-Yue Wang, Kunfeng Wang, Wei-Hua Lin, Xin Xu, and Cheng Chen. 2011. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1624–1639.

Xiyue Zhang, Chao Huang, Yong Xu, Lianghao Xia, Peng Dai, Liefeng Bo, Junbo Zhang, and Yu Zheng. 2021. Traffic flow forecasting with spatial-temporal graph diffusion network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 15008–15015.

Zhiqiang Zhang, Dandan Zhang, and Yun Wang. 2024. Fast long sequence time-series forecasting for edge service running state based on data drift and non-stationarity. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6181–6194.

Yusheng Zhao, Junyu Luo, Xiao Luo, Jinsheng Huang, Jingyang Yuan, Zhiping Xiao, and Ming Zhang. 2025a. Attention bootstrapping for multi-modal test-time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22849–22857.

Yusheng Zhao, Xiao Luo, Wei Ju, Chong Chen, Xian-Sheng Hua, and Ming Zhang. 2023. Dynamic hypergraph structure learning for traffic flow forecasting. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2303–2316. IEEE.

Yusheng Zhao, Xiao Luo, Weizhi Zhang, Wei Ju, Zhiping Xiao, Philip S. Yu, and Ming Zhang. 2025b. Marco: Meta-reflection with cross-referencing for code reasoning. *arXiv preprint arXiv:2505.17481*.

Yusheng Zhao, Changhu Wang, Xiao Luo, Junyu Luo, Wei Ju, Zhiping Xiao, and Ming Zhang. 2025c. Traci:

A data-centric approach for multi-domain generalization on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13401–13409.

Chuanpan Zheng, Xiaoliang Fan, Shirui Pan, Haibing Jin, Zhaopeng Peng, Zonghan Wu, Cheng Wang, and S Yu Philip. 2023a. Spatio-temporal joint graph convolutional networks for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(1):372–385.

Ou Zheng, Mohamed Abdel-Aty, Dongdong Wang, Zijin Wang, and Shengxuan Ding. 2023b. Chatgpt is on the horizon: could a large language model be suitable for intelligent traffic safety research and applications? *arXiv preprint arXiv:2303.05382*.

Dongcheng Zou, Senzhang Wang, Xuefeng Li, Hao Peng, Yuandong Wang, Chunyang Liu, Kehua Sheng, and Bo Zhang. 2024. Multispans: A multi-range spatial-temporal transformer network for traffic forecast via structural entropy optimization. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 1032–1041.

A More Details about the Datasets

In the experiments, three widely used datasets in traffic flow forecasting are adopted, including PEMS03, PEMS04, and PEMS08. They are described as follows:

- **PEMS03**: This dataset consists of traffic flow data collected from California's PeMS (Performance Measurement System) network. It includes data from 358 sensors, recorded every 5 minutes. The data includes various attributes such as traffic speeds, flow rates, and occupancy rates, which are essential for traffic flow forecasting.
- **PEMS04**: The PEMS04 dataset is also collected from California's freeway system, but from a different region compared to PEMS03. It contains traffic flow and speed data from 307 sensors, collected every 5 minutes over several months.
- **PEMS08**: PEMS08 is another dataset derived from California's freeway system, covering a different geographic area than both PEMS03 and PEMS04. It includes data from 170 sensors, with recordings of traffic speeds and volumes over time.

B More Details about the Metrics

In this paper, we use three metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). They are defined as follows:

$$\text{MAE} = \frac{1}{TNF}\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{f=1}^{F}\left|\hat{\boldsymbol{Y}}_{tnf} - \boldsymbol{Y}_{tnf}\right|,$$

$$\text{RMSE} = \sqrt{\frac{1}{TNF}\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{f=1}^{F}\left(\hat{\boldsymbol{Y}}_{tnf} - \boldsymbol{Y}_{tnf}\right)^2},$$

$$\text{MAPE} = \frac{1}{TNF}\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{f=1}^{F}\left|\frac{\hat{\boldsymbol{Y}}_{tnf} - \boldsymbol{Y}_{tnf}}{\boldsymbol{Y}_{tnf}}\right| \times 100\%,$$

where $\hat{\boldsymbol{Y}} \in \mathbb{R}^{T\times N\times F}$ is the forecasting result, and $\boldsymbol{Y}$ is the ground truth data.

C More Details about the Baseline Methods

In the experiments, we use the following baselines, including DCRNN (Li et al., 2018), ASTGCN (Guo et al., 2019), STSGCN (Song et al., 2020), HGCN (Wang et al., 2022), DyHSL (Zhao et al., 2023), STAEformer (Liu et al., 2023), COOL (Ju et al., 2024), and LLM-MPE (Liang et al., 2024). We provide more details about them in the following:

- **DCRNN (Diffusion Convolutional Recurrent Neural Network)** (Li et al., 2018): DCRNN combines recurrent neural networks with graph neural networks to model spatial dependencies in traffic flow data. It captures both temporal and spatial correlations by incorporating graph-based diffusion processes to model the movement of traffic across different locations.
- **ASTGCN (Attention-based Spatial-Temporal Graph Convolutional Network)** (Guo et al., 2019): ASTGCN integrates graph convolutional networks with attention mechanisms to model both spatial and temporal dependencies in traffic data.
- **STSGCN (Spatial-Temporal Synchronous Graph Convolutional Network)** (Song et al., 2020): STSGCN is a graph-based model that jointly captures spatial and temporal correlations in traffic data.
- **HGCN (Hypergraph Convolutional Neural Network)** (Wang et al., 2022): HGCN is designed to model complex relationships between traffic sensors using hypergraphs. It leverages non-pair-wise dependencies that are difficult to capture with traditional graph-based models for traffic flow forecasting.
- **DyHSL (Dynamic Hypergraph Structure Learning)** (Zhao et al., 2023): DyHSL focuses on learning dynamic hypergraph structures to learn non-pair-wise relations in hypergraphs. Moreover, it utilizes interactive graph convolutions to capture higher-order relations.
- **STAEformer (Spatio-Temporal Adaptive Embedding transformer)** (Liu et al., 2023): STAEformer is a transformer-based model that uses spatio-temporal adaptive embeddings together with spatial and temporal attention mechanism to capture long-range dependencies.

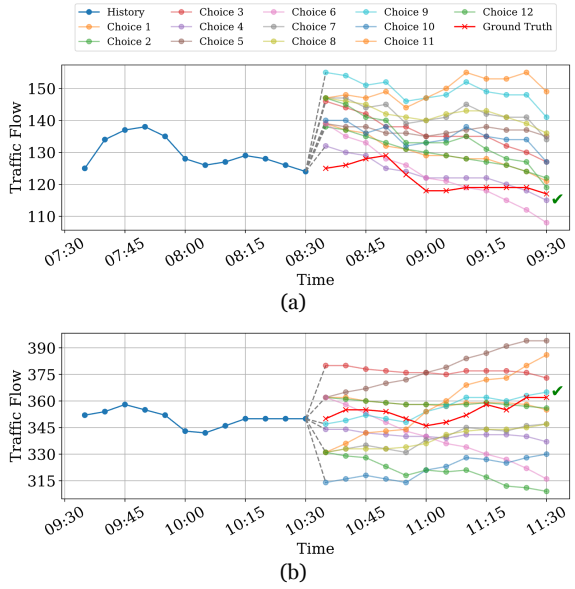

Figure 9: Additional visualizations of historical data, choices, and the prediction of the LLM-based selector.

- **COOL (Conjoint Spatio-Temporal graph neural network)** (Ju et al., 2024): COOL models traffic graphs from prior and posterior perspectives to capture high-order spatio-temporal relationships from a conjoint perspective.

- **LLM-MPE (human mobility prediction under public events based on LLMs)** (Liang et al., 2024): LLM-MPE is a recent approach that uses LLMs in conjunction with unstructured event descriptions to predict future traffic flows.

D Details about the Experiments

With regard to the time of the framework, for smaller datasets like PEMS08, the test time of a single slice is within 3 minutes on two A100 GPUs with LLaMA 70B, 4-bit quantization. For larger datasets, the framework can infer within 5 minutes for a single time slice, making it possible to predict traffic flows in real-time (since the data are collected every 5 minutes and the time for dual-branch predictor is negligible). In the experiments, we only perform a single run to save computational resources.

In the experiments, we use LLaMA 3 (AI@Meta, 2024) and vLLM (Kwon et al., 2023). Both of them can be used for research purposes. The data are mostly numerical and we manually checked that they do not contain personally identifying info or offensive content.

E More visualization of selection

We provide additional visualization of the selection results from the LLM-based selector, which is shown in Figure 9. Specifically, sub-figure (a) displays a road possibly in the suburb or near the residential area, since it exhibits lower traffic flows and a small morning peak near 8 a.m. During the morning rush hour, people move from their homes to the working places, and therefore after a small peak, the traffic flow generally decreases, which aligns with the selection result and the ground truth data. For sub-figure (b), it possibly describes a bustling place in the city center, as the traffic flow is fairly high. The traffic flow is relatively stable after the morning rush hour, and after 11 a.m., some people may go out to have lunch, leading to a slight increase of traffic flow. This aligns with the selected option as well as the ground truth data.