Multi-Graph Convolutional Recurrent Network for Fine-Grained Lane-Level Traffic Flow Imputation

Jingci Ming[1,2], Le Zhang[2*], Wei Fan[3], Weijia Zhang[4], Yu Mei[5], Weicen Ling[5], Hui Xiong[4,6*]
[1]Rutgers University, [2]Baidu Research, Baidu Inc,
[3]University of Central Florida, [4]The Hong Kong University of Science and Technology (Guangzhou),
[5]Department of Intelligent Transportation System, Baidu Inc
[6]Guangzhou HKUST Fok Ying Tung Research Institute
jingci.ming@rutgers.edu, {zhangle0202, vegazhang3, weicen.ling}@gmail.com,
weifan@knights.ucf.edu, whqyqy@hotmail.com, xionghui@ust.hk

***Abstract*—Traffic flow imputation provides a more-complete view of traffic flows, and thus is a fundamental function in building Intelligent Transportation Systems. The performance of traffic flow imputation has a big impact on a wide range of downstream applications, such as traffic forecasting and control. Therefore, in this paper, we propose a Multi-grAph Convolutional Recurrent netwOrk (MACRO) framework for supporting fine-grained lane-level traffic flow imputation, which can help to reconstruct more complete traffic flows at the lane level. Specifically, we first design a spatial dependency module to model the diversified spatial correlations within traffic flows, where multi-relation graphs are first constructed to consider correlations from various perspective, then a multi-graph convolution neural network is proposed to capture the integrated spatial dependencies of traffic flows and adequately propagate the observed traffic values to mitigate data sparsity problem from spatial domain. Also, to handle the temporally continuous data missing issue, we adopt a modified bi-directional recurrent neural network to capture traffic flows' temporal dependencies by considering both historical and future information, and employ a temporal decay mechanism to control the irregular information transfer between adjacent time slices. Moreover, a spatio-temporal knowledge integration module is devised to comprehensively integrate multi-resolution spatiotemporal knowledge for traffic flow imputation. Finally, extensive experiments on the real-world dataset demonstrate that the performance of MACRO outperforms several state-of-the-art baselines with respect to traffic flow imputation.**

***Index Terms*—Spatio-temporal Model, Traffic Data Imputation, Graph Neural Network, Recurrent Neural Network**

I. INTRODUCTION

Spatio-temporal traffic data collected from various sensor devices (e.g., loop detectors and cameras) is the basis for a wide range of important applications in Intelligent Transportation Systems (ITS) [1], [2]. But in reality, many unpredictable systematic failures, such as equipment damage and communication error, usually interrupt data collection of the sensors, which results in severe data missing issues and significantly destroys the validity of collected data [3]–[5]. To this end, traffic flow imputation is proposed to impute the missing traffic values based on the uncompleted observed traffic data, which is a fundamental ability towards building effective ITS, and indeed benefits various downstream applications, such as traffic flow forecasting and control.

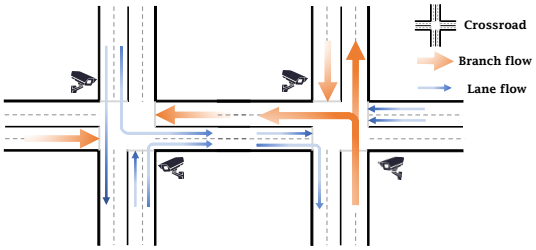


Fig. 1. An illustrative example of road network.

Traffic flow imputation problem has attracted extensive attention from both academia and industry [6], [7]. The early studies mainly rely on the statistical methods, such as linear interpolation [4] and ARIMA [8]. Since traffic data is usually collected by multiple fixed sensors at fixed time points, some studies formulate them as the format of matrix or tensor, and introduce the low-rank matrix/tensor completion methods for imputation [9]–[11]. Recently, studies widely leverage deep learning methods to capture more complex spatio-temporal dependencies within data [12]–[14]. For example, Graph Neural Networks (GNNs) have shown a strong capability to model spatial non-Euclidean dependencies between multiple sensors located in the irregular road network [15]–[17], and Recurrent Neural Networks (RNN) are widely used to capture the temporal dependencies within data [18]–[20]. However, existing studies mainly investigate the data imputation for branch roads' traffic flows. In reality, as shown in Fig. 1, a crossroad can contain multiple branches, and each branch consists of multiple lanes. To achieve a meticulous awareness for dynamic traffic situations, it is essential to consider the real-time traffic state of each lane, which is a more finer-grained and complicated scenario that has been rarely studied before.

In this paper, we investigate a new *Lane-Level Traffic Flow Imputation* (LTFI) problem, which aims to impute traffic flow for each lane involving a more finer-grained and complicated dynamic traffic scenario. However, it is a non-trivial task, which faces the following technical challenges. The first chal-

* denotes the corresponding author
Code available at https://github.com/Jingci/MACRO

lenge comes from the *diversified spatial correlations within lane-level traffic flows*, which can be induced by physical traffic connectivity, geographical proximity and traffic pattern's similarity between multiple lanes. These correlations are non-Euclidean and cause integrated spatial dependencies of different lanes, which are challenging to capture. The second challenge is the *severe spatio-temporal data sparsity*. Different sensors' data may irregularly miss at different time, and even the data missing issue could appear in a particular whole area at some time. Besides, since many systematic failures of sensors cannot be recovered immediately, which causes temporally continuous data missing. The severe spatio-temporal data sparsity leads to spatio-temporal dependencies extremely irregular, which is intractable to model. Finally, it is challenging for *spatio-temporal knowledge integration*. There are substantial knowledge from both spatial and temporal domains. However, since the spatio-temporal factors deeply interplay with each other in different resolution, it's hard to comprehensively integrate the spatio-temporal knowledge.

To tackle above challenges, in this work, we propose a Multi-grAph Convolutional Recurrent netwOrk (MACRO) for LTFI problem. Specifically, MACRO consists of three components. First is a *Spatial Dependency Module*. We construct the multi-relation graphs to consider the diversified correlations between lanes from perspectives of connectivity, adjacency, and similarity. Then a multi-graph convolution neural network is proposed to model the integrated spatial dependencies of lane-level traffic flows, in which a diffusion graph convolution layer is adopted to aggregate the observed traffic information from different-order neighbors for mitigating the severe data sparsity problem in spatial domain. Then an attentive multi-graph fusion layer is further devised to comprehensively integrate traffic knowledge of lanes from multiple graph perspectives. Second is a *Temporal Dependency Module*. We introduce the Bi-directional Long Short-Term Memory (Bi-LSTM) to capture temporal dependencies with consideration of both historical and future information. Furthermore, to handle the temporally continuous data missing issue, a temporal decay mechanism is employed to control the irregular information transfer between adjacent time slices. Finally, a *Spatio-Temporal Knowledge Integration Module* is devised to comprehensively integrate multi-resolution spatio-temporal knowledge for traffic flow imputation. Specifically, we first incorporate the learned representation from both spatial and temporal domains at each time slice to derive the micro knowledge representation. Then a tailored-designed autoencoder module is proposed to capture the distribution of traffic data from a global spatio-temporal perspective, and serves as an initialization of the temporal dependency module to provide a macro guidance for imputation. In summary, our major contributions are summarized as:

- We investigate a new fine-grained lane-level traffic flow imputation problem, which involves a more fine-grained and complicated dynamic traffic scenario. To best of our knowledge, we are among the first one to study lane-level traffic flow imputation problem.
- We design a multi-graph convolutional recurrent network for lane-level traffic flow imputation, where a multi-graph convolution neural network is designed to model the diversified spatial correlations within lane-level traffic flows and mitigate data sparsity problem from spatial domain, then a modified bi-directional recurrent neural network with temporal decay mechanism is applied to handle the temporally continuous data missing issue, lastly a spatio-temporal knowledge integration module is proposed to comprehensively integrate the multi-resolution spatio-temporal knowledge for imputation.
- We conduct extensive experiments on the real-world dataset, and the results demonstrate that our approach achieves the best performance against several competitive baselines in lane-level traffic flow imputation problem.

II. RELATED WORK

Previous studies on traffic data imputation can be mainly categorized into three classes, namely statistical methods, tensor-based methods and deep learning methods.

Statistical methods impute the traffic flow values by considering the proximity of temporal and spatial domains. For example, [4] fill the missing data based on temporal and spatial interpolation. [8], [21] study traffic periodical characteristics and estimate the missing values by the ARIMA model. [22] impute erroneous data by exploiting the relationship between detector flows using linear regression. [23], [24] compare the temporal correlations in traffic flow series, and aggregate K-Nearest Neighbours'(KNN) values to estimate the missing data. However, these statistical methods impute traffic data mainly by exploiting simple and explicit proximity, which fails to capture the latent patterns and complicated dependencies within real traffic data.

Tensor-based methods formulate traffic data imputation task as a tensor completion problem, then introduce typical matrix or tensor based approach for imputation with capturing the latent patterns in traffic flow. For instance, [25], [26] utilize Principle Component Analysis (PCA) algorithm to study the relationship between observed data and latent variables, then estimate the missing values by a probabilistic model. [9] propose a tensor decomposition method to capture the high dimensional temporal inherent correlations within traffic flow data. [11] incorporate generic forms of domain knowledge from transportation systems to discover traffic flow pattern. [27] characterize temporal dependency on neighboring fragments of traffic flow by using a state transition matrix with low-rank regularization. [10] employ a low rank auto-regressive tensor completion method to capture local and global temporal consistency of the data. Although these methods attempt to capture latent patterns within the traffic flows, they are not incapable of capturing the spatial knowledge within traffic data.

Recently, deep learning methods are widely developed for the missing data imputation task to capture more complicated temporal and spatial non-linear dependencies. Recurrent

Neural Networks (RNN) have been proved their capability to capture the temporal dependency for time series data [15], [28]. To apply RNN into time series imputation, [18], [19] propose variant RNN models by considering temporal decay coefficient to overcome the continuous missing value problem. [29] propose a stacked multiple auto-encoder layers mechanism to learn the spatio-temporal dependency represented in low-dimensional latent space, then the auto-encoder models impute the missing values by transforming the sparse observed data into latent space and reconstructing the completed data. Moreover, some works introduce Generative Adversarial Networks (GAN) [30] into missing data imputation by adopting a discriminator to reduce the error between estimated value and ground truth. For example, [31] adopt a generator to imputes the missing components based on observed data, and employ a discriminator to discriminate the observed and imputed components. [32] further introduce a new loss function to balance the adversarial and facilitated relations between discriminator and constructor. Particularly, some recent studies successfully apply graph representation learning techniques into traffic forecasting [15], [17], [33], traffic volume inference [34] and time series imputation [16] tasks to model the complicated spatial dependencies within data. [34] adopt multi-view graph embedding to encode the multi-hop correlations between road segments into real-valued vectors with modeling trajectories data for the citywide traffic volume inference. [16] introduce a graph neural network architecture to reconstruct the missing data in the different channels of a multivariate time series by learning spatio-temporal representations through message passing. However, existing deep learning based studies mainly focus on general time series or branch-level traffic data imputation, whereas we investigate the lane-level traffic data imputation problem in a more fine-grained scenario.

III. Preliminary

In this section, we first introduce the real-world dataset used in our study, and then conduct several preliminary analysis towards the collected data. Finally, we formally define the problem of traffic flow imputation.

A. Data Description

We collected real-world traffic data from Baoding, China, which contains two types of information, i.e., road network data and sensor-collected traffic flow data.

The basic elements in road network are crossroad, branch and lane. And road network data contains the detailed information of these basic elements, including *coordinate* and *direction*. In addition, some relations between different elements are included. The first one is the *belonging relation*. Generally, each lane belongs to a specific branch, and each branch belongs to a specific crossroad. The second one is the *connectivity relation*, which indicates the reachability between different crossroads or branches. According to the traffic direction, the branch road can be divided into two types: the in-branch entering the crossroad and the out-branch leaving the crossroad. In a similar way, lanes can also be divided into two types, namely in-lanes and out-lanes respectively.

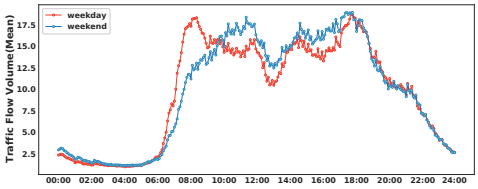
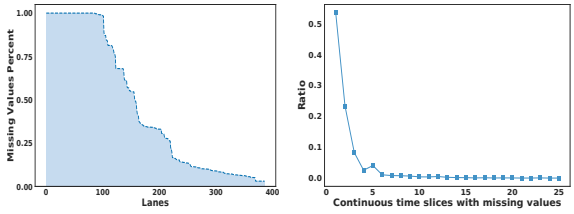


Fig. 2. Hourly traffic flow distribution on weekdays and weekends.



(a) (b)

Fig. 3. The characteristics of our data, (a) the missing ratio of different lanes, (b) the distribution of time slice lengths with continuous missing values.

Traffic flow data is collected by the distributed sensors. In our scenario, only in-lanes are equipped with sensors to monitor traffic flow. Specifically, the sensor records the total amount of traffic flow in a fixed time interval, such as 5 minutes, and then sends the record with the corresponding timestamp to the central server. We can align these traffic flow records of different in-lanes according to the timestamp. Due to the unpredictable systematic failures, such as equipment damage and communication error, the data collection would be interrupted, which results in severe data missing issues.

B. Data Exploration

In general, urban traffic conditions are extremely complex, which highly correlate with people's mobility and activities. We explore the traffic pattern from daily perspective. Figure 2 presents the daily traffic pattern of weekday and weekend. Obviously, there is a big difference between the daily patterns on weekdays and weekends. For example, morning peak hours on weekdays are in the range 7:00 $\sim$ 9:00, while morning peak hours on weekends are in the range 10:00 $\sim$ 12:00. As a result, it's very challenging to model such diversified correlation patterns within severely sparse traffic flow data.

Next, we analyze the characteristics of our data. After data preprocessing, we found the real-world dataset is pretty sparse, whose sparsity ratio is about 54%. Fig. 3(a) shows the missing ratio of different lanes. About 40% of lanes have more than 50% missing rates. Even 20% of the lanes are with 100% missing ratio. Apart from that, continuous missing values issue commonly exists in our data. Fig. 3(b) shows the distribution of time slice lengths with continuous missing values. The proportion of consecutive missing cases is about 45%. And the time slice length of continuous missing values is mainly concentrated between 2 to 5. These data sparsity issues pose great challenges for imputation.

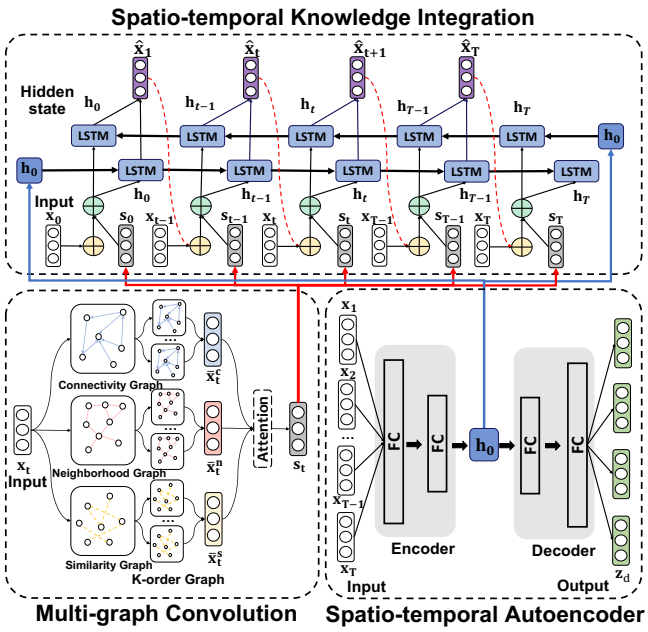


Fig. 4. The diagrammatic sketch of the proposed MACRO framework for lane-level traffic flow imputation.

C. Problem Formulation

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$ denote the traffic flow volume of different lanes perceived by the sensors, where $\mathbf{x}_t \in \mathbb{R}^N$ presents the traffic flow of all lanes at time slice t. T and N denote the number of time slices and lanes respectively. Without loss of generality, we set the interval between different time slices as the same. As mentioned before, due to many unpredictable systematic failures, $\mathbf{X}$ is partially filled. Correspondingly, we can construct a mask matrix $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_T\} \in \mathbb{R}^{N \times T}$, where $\mathbf{m}_{ti} = 1$ if $\mathbf{x}_{ti}$ is observed, and $\mathbf{m}_{ti} = 0$ if $\mathbf{x}_{ti}$ is missing. The LTFI problem is defined as follows:

Given the partially observed traffic flow matrix of lanes $\mathbf{X}$ *with the mask matrix* $\mathbf{M}$*, and the information of road network (e.g., the belonging relations between lanes and branches, the connectivity between branches, and the geographic information of lanes), our target is to construct the completed traffic volume matrix* $\hat{\mathbf{X}}$ *by imputing missing values.*

IV. METHODOLOGY

In this section, we first present the framework overview of our proposed Multi-grAph Convolutional Recurrent netwOrk (MACRO); then we introduce its spatial dependency module, the temporal dependency module, and spatio-temporal knowledge integration module in more details.

A. Framework Overview

Fig. 4 shows an architecture overview of our proposed MACRO framework. The MACRO framework takes the partially observed traffic flows as input and imputes the missing values with three different modules: (i) the *Spatial Dependency Module* (the lower left part) is designed to encode diversified spatial correlations among lanes and capture the integrated spatial dependencies of traffic flows; (ii) the *Temporal Dependency Module* (the upper part) leverages the modified Bi-directional Long Short Term Memory network (Bi-LSTM) to model the temporal dependencies of traffic flows and handle the temporally continuous data missing issue; (iii) the *Spatio-temporal knowledge integration module* (the lower right part) is devised to comprehensively integrate multi-resolution spatiotemporal knowledge for traffic flow imputation, where a tailored-designed autoencoder is applied to simultaneously capture the distribution of traffic data from a global spatio-temporal perspective.

B. Spatial Dependency Module

In this part, we will introduce how to encode diversified correlations among lanes from the perspective of constructing multi-relation graphs, and then we propose a multi-graph convolution neural network to model the integrated spatial dependencies of lane-level traffic flows.

1) Spatial graph construction: To encode the diversified correlations among lanes, we construct three types of graph from different views. The first one is traffic connectivity graph $\mathcal{G}_c$, which encodes the physical traffic mobility between different lanes. The second one is neighborhood graph $\mathcal{G}_n$, which encodes the geographical proximity between lanes. And the last one is similarity graph $\mathcal{G}_s$, which encodes the traffic pattern's similarity of lanes. These graphs are represented respectively by adjacency matrices as $\mathbf{A}^c, \mathbf{A}^n, \mathbf{A}^s \in (0, 1)^{N \times N}$, where N is the total number of lanes.

a) Traffic connectivity graph: Intuitively, the reachable lanes may have the similar traffic volume. In this regard, in terms of the reachable relation of lanes at the same crossroad, it is obvious that the lanes belonging to the same branch are mutually reachable. Next, we consider the reachable relationship between different crossroads. As mentioned before, a crossroad contains several branches, and different crossroads are connected based on the their corresponding branches. If two crossroad c_i and c_j are connected, there exists an out-branch of c_i connected to an in-branch of c_j, and vice verse. To extract the reachable relations of in-lanes between different crossroads, we can take advantage of out/in-branch connections between different crossroads. In particular, for an in-lane l_i belonging to a specific crossroad, it can reach one of the out-branch at the same crossroad according to its direction. Furthermore, l_i can reach to the in-branch connected by the its reachable out-branch.Thus we can constructed the connection for l_i and all lanes of the reachable in-branch. Finally, the traffic connectivity graph is defined as follows:

$$\mathbf{A}^c_{ij} = \begin{cases} 1, & l_i \text{ can reach } l_j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

b) Neighborhood graph: Generally, lanes that are relatively close may exhibit similar traffic volume patterns, e.g., lanes belonging to crossroads within the same region may have similar fluctuating trends over time. To this end, we propose the neighborhood graph to capture the distance

correlation between lanes. Specifically, for any lanes l_i and l_j, we calculate the distance between them as $dist(\cdot)$ with their coordinates, and we set a threshold ϵ for the distance to decide whether connect the lanes together:

$$\mathbf{A}^n_{ij} = \begin{cases} 1, & dist(l_i, l_j) \leq \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

c) Similarity graph: When imputing the missing value for the traffic flow of a lane, it is intuitive to take advantage of the other similar lanes for imputation. Thus we derive a similarity graph to facilitate the spatial dependency learning. The edge is defined according to the similarity of lane's traffic pattern (i.e., the observed traffic flow volume). Specifically,

$$\mathbf{A}^s_{ij} = \begin{cases} 1, & l_i \text{ is top-}k \text{ similar to } l_j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where k is the prior hyperparameter to control number of edges, the similarity is calculated by the cosine function.

2) Multi-graph convolution neural network: Based on the constructed multi-relation graphs, we model the integrated spatial dependencies of lane-level traffic flows. Due to graph neural network (GNN) has achieved great success on general graph learning problems, we extend GNN to model the spatial information. Specifically, we first introduce the diffusion convolution layer to propagate the observed traffic information to different-order neighbors on a single graph, and then an attentive multi-graph fusion layer is further devised to comprehensively integrate traffic knowledge of lanes from multiple graph perspectives.

a) Diffusion Convolution Layer: Denote the observed traffic flow of all lanes at time slice t as a graph signal $\mathbf{x}_t \in \mathbb{R}^N$. The core of GNN is to represent node with neighborhood information. However, the absence of node features is common in the imputation scenarios. Conducting the convolution operation on immediate neighbor may limit the aggregation of effective information. To this end, we propose to construct multiple high-order graphs to generate more neighbors at different levels for a target node. By aggregating information from these more neighbors can prevent the invalid aggregation to some extent. Specifically, let $\mathbf{A}$ denote the original adjacency matrix, then $\mathbf{A}^{(k)}$ is the k-th power of matrix $\mathbf{A}$, in which the edges indicate the k-order relationships between nodes. Without loss of generality, we construct K matrices, i.e., $\{\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, ..., \mathbf{A}^{(K)}\}$, to capture different-order neighbors for each node. Then, we aggregate neighboring information to generate node representation. Formally, the diffusion convolution layer is defined as:

$$\bar{\mathbf{x}}_t = \sigma(\sum_{k=1}^{K} \mathbf{A}^{(k)} \mathbf{W}_{\mathbf{A}^{(k)}} \mathbf{x}_t), \tag{4}$$

where $\mathbf{W}_{\mathbf{A}^{(k)}} \in \mathbb{R}^{N \times N}$ is a learnable weighted matrix towards k-th order adjacency matrix, and σ is the activation function.

b) Multi-graph Fusion Layer: By conducting the diffusion convolution operation on three graphs at time slice t, we can obtain the corresponding node embedding, namely $\bar{\mathbf{x}}^c_t$, $\bar{\mathbf{x}}^n_t$, $\bar{\mathbf{x}}^s_t$ respectively, which contains traffic information from different views. Next, we fuse all these embeddings learned from three graphs to generate the overall representation for each node. Obviously, three graphs capture different types of spatial knowledge, which play different roles in the traffic flow imputation task. To this end, we leverage the attention mechanism to automatically capture the spatial dependencies for the downstream application. At first, we concatenate the representations from these three graphs as follows:

$$\bar{\mathbf{x}}^{cns}_t = [\bar{\mathbf{x}}^c_t, \bar{\mathbf{x}}^n_t, \bar{\mathbf{x}}^s_t]. \tag{5}$$

Afterwards, we add an attention layer by acquiring the attention weights towards the representations of three graphs:

$$\alpha_t = \bar{\mathbf{x}}^{cns}_t \cdot \mathbf{W}_\alpha, \tag{6}$$

where α_t is the attention weights at time splice t, $\mathbf{W}_\alpha \in \mathbb{R}^{N \times 1}$ are learnable parameter. Accordingly, we calculate the weighted sum to get the final spatial representations:

$$\mathbf{s}_t = \alpha_t \bar{\mathbf{x}}^{cns}_t = \alpha_{t,1} \bar{\mathbf{x}}^c_t + \alpha_{t,2} \bar{\mathbf{x}}^n_t + \alpha_{t,3} \bar{\mathbf{x}}^s_t \tag{7}$$

In each time slice t, we can obtain the overall spatial representation $\mathbf{s}_t \in \mathbb{R}^N$, which preserves the integrated spatial dependencies and serves for temporal imputation.

C. Temporal Dependency Module

In addition to the spatial dependency among different lanes, traffic flow volume also has temporal dependency for each single lane in a time period. In this part, we will introduce the temporal dependency module.

As we all know, recurrent neural networks (RNN) have been widely used in the temporal-related tasks. We thus choose a special RNN model to capture the temporal dependency of the traffic flow sequence, namely Bi-directional Long Short Term Memory network (Bi-LSTM). Specifically, Bi-LSTM contains two LSTM modules that are able to capture both of the past and future information simultaneously when estimating the current state. Formally, the standard LSTM model is formulated as follows:

$$\begin{aligned} &\mathbf{i}_t = \sigma\left(\mathbf{W}_1 \left[\mathbf{x}_t, \mathbf{h}_{t-1}\right] + \mathbf{b}_1\right), \mathbf{f}_t = \sigma\left(\mathbf{W}_2 \left[\mathbf{x}_t, \mathbf{h}_{t-1}\right] + \mathbf{b}_2\right), \\ &\widetilde{\mathbf{C}}_t = \sigma_1\left(\mathbf{W}_3 \left[\mathbf{x}_t, \mathbf{h}_{t-1}\right] + \mathbf{b}_3\right), \mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \widetilde{\mathbf{C}}_t, \\ &\mathbf{o}_t = \sigma\left(\mathbf{W}_4 \left[\mathbf{x}_t, \mathbf{h}_{t-1}\right] + \mathbf{b}_4\right), \mathbf{h}_t = \mathbf{o}_t \odot \sigma_1\left(\mathbf{C}_t\right), \end{aligned} \tag{8}$$

where $\mathbf{x}_t$ is the input vector at time slice t, and $\mathbf{W}_*, \mathbf{b}_*$ are the parameters as weight matrices and bias, $\odot$ is element-wise multiplication, σ and σ_1 denote sigmoid and tanh activation function. $\mathbf{h}_t$ represents the hidden state. For simplicity, the above formulas can be represented in short as:

$$\mathbf{h}_t = \mathrm{LSTM}\left(\mathbf{x}_t, \mathbf{h}_{t-1}\right). \tag{9}$$

And Bi-LSTM uses the input sequential data and its reverse to learn the hidden states $\mathbf{h}_t$, which can be expressed as:

$$\begin{aligned} \overrightarrow{\mathbf{h}_t} = \mathrm{LSTM}\left(\mathbf{x}_t, \overrightarrow{\mathbf{h}_{t-1}}\right), \overleftarrow{\mathbf{h}_t} = \mathrm{LSTM}\left(\mathbf{x}_t, \overleftarrow{\mathbf{h}_{t+1}}\right), \\ \mathbf{h}_t = \left[\overrightarrow{\mathbf{h}_t}, \overleftarrow{\mathbf{h}_t}\right]. \end{aligned} \tag{10}$$

By this way, $\overrightarrow{\mathbf{h}_t}$ can capture the temporal trend from historical data, and $\overleftarrow{\mathbf{h}_t}$ can learn the temporal characteristics from future data. As a result, $\mathbf{h}_t$ can leverage both the past and future temporal information.

In this work, we modify the Bi-LSTM for traffic data imputation task. Considering that the forward and backward LSTM modules have exactly the same mechanism, we take the modification of the forward LSTM as an example. Firstly, as for the traffic flow imputation task, the input is the partially observed traffic flows in a time period T denoted by $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T)$, and each of $\mathbf{x}_t$ denotes traffic flow volume at time slice t. If there exists missing value in the vector $\mathbf{x}_t$ at time slice t, we utilize the corresponding value in the estimated vector $\hat{\mathbf{x}}_t$ dependent on hidden state at time slice $t-1$ to replace the missing value at time slice t. This process can be represented by:

$$\hat{\mathbf{x}}_t = \mathbf{W}_x \mathbf{h}_{t-1} + \mathbf{b}_x \tag{11}$$

$$\mathbf{x}_t^C = \mathbf{m}_t \odot \mathbf{x}_t + (1 - \mathbf{m}_t) \odot \hat{\mathbf{x}}_t \tag{12}$$

where equation (11) is a simple linear component that yields the estimated $\hat{\mathbf{x}_t}$ based on the previous hidden state $\mathbf{h}_{t-1}$; equation (12) is used to integrate the estimated values with the ground truth values where the parameter $\mathbf{m}_t$ is the missing position vector for $\mathbf{x}_t$.

Secondly, as mentioned before, continuous data missing issue is common existed in traffic data. Intuitively, different time intervals with continuous missing values before the current time slice may bring different influence on estimation of current time slice. Therefore, we follow [19] to adopt the temporal decay parameter γ_t to control the information transferred from the previous time slice. Then the LSTM updating equation can be rewritten as follows:

$$\mathbf{h}_t = \mathrm{LSTM}\left(\mathbf{x}_t^C, \mathbf{h}_{t-1} \odot \gamma_t\right), \tag{13}$$

where γ_t is able to decrease the influence from the missing values of long previous steps, which is calculated by:

$$\gamma_t = \exp\left\{-\max\left(0, \mathbf{W}_\gamma \delta_t + \mathbf{b}_\gamma\right)\right\}, \tag{14}$$

where δ_t indicates the length of time slices with continuous missing value before the current step.

Similarly, the backward LSTM has similar operations. Finally, we can combine the forward and backward estimations together to predict current traffic flow in an intuitive way as:

$$\hat{\mathbf{x}}_t^* = \frac{\overrightarrow{\hat{\mathbf{x}}_t} + \overleftarrow{\hat{\mathbf{x}}_t}}{2}, \tag{15}$$

where $\overrightarrow{\hat{\mathbf{x}}_t}$ and $\overleftarrow{\hat{\mathbf{x}}_t}$ represent the forward estimation and backward estimation respectively; $\hat{\mathbf{x}}_t^*$ is the combined final output.

D. Spatio-temporal Knowledge Integration

Barely considering either the spatial information or temporal information is not enough for traffic flow imputation task, we try to aggregate them together in this block.

First, we incorporate the learned representation from both spatial and temporal domain at each time slice to derive the ***micro*** knowledge representation. Specifically, as we obtain the completed temporal estimated vector $\mathbf{x}_t^C$ in the temporal based model, we input it into the spatial based estimation equation (5) and output a new spatial estimated vector $\hat{\mathbf{s}}_t$. Then we further combine the temporal based estimation $\mathbf{x}_t^C$ with the spatial based estimation $\hat{\mathbf{s}}_t$. We denote the combined vector as $\hat{\mathbf{c}}_t$, and the aggregation process is defined as:

$$\hat{\mathbf{c}}_t = \beta_\mathbf{t} \odot \hat{\mathbf{s}}_t + (1 - \beta_\mathbf{t}) \odot \mathbf{x}_t^C. \tag{16}$$

$$\mathbf{c}_t^C = \mathbf{m}_t \odot \mathbf{x}_t + (1 - \mathbf{m}_t) \odot \hat{\mathbf{c}}_t. \tag{17}$$

Here we let β_t denote the weight of combining the temporal-based estimation $\hat{\mathbf{x}}_t^*$ and spatial-based estimation $\hat{\mathbf{s}}_t$, which is learnable. Finally, we replace the the missing values with the corresponding values in $\hat{\mathbf{c}}_t$, and the obtained vector is the final completed estimation which is then fed back into the temporal dependency module to compute the memory of $\mathbf{h}_t$. We rewrite the hidden state updating process in LSTM as:

$$\mathbf{h}_t = \mathrm{LSTM}\left(\mathbf{c}_t^C, \mathbf{h}_{t-1} \odot \gamma_t\right). \tag{18}$$

Further, we propose a tailored-designed autoencoder module to simultaneously capture the distribution of traffic data from ***global*** spatial and temporal perspectives. The global spatiotemporal embedding is used to capture the global latent space distribution of the traffic flow volume over all in-lanes in a fixed time period as the input traffic flow sequence, which can cooperate with the micro-view embeddings from a global view.

Specifically, we first construct multiple hidden layers as an encoder to generate a hidden state vector of the inputs. The encoding process can be interpreted as a non-linear transformation by the following form:

$$\mathbf{z}_1^e = \sigma(\mathbf{W}_{e,1}\mathbf{X} + \mathbf{b}_{e,1}), \tag{19}$$

$$\mathbf{z}^e = \sigma(\mathbf{W}_{e,2}\mathbf{z}_1^e + \mathbf{b}_{e,2}), \tag{20}$$

where $\mathbf{X} \in \mathbb{R}^{N \times T}$ is the input matrix of traffic flow volume over N lanes in a range of T time slices; the $\mathbf{W}_{e,1}, \mathbf{W}_{e,2}$ and $\mathbf{b}_{e,1}, \mathbf{b}_{e,2}$ respectively denote the weight matrices and bias parameters; σ is the activation function; $\mathbf{z}^e$ is the learned hidden states by the encoder.

To learn the global spatiotemporal embedding, we utilize a decoder composed by hidden layers to reconstruct the matrix $\mathbf{X} \in \mathbf{R}^{N \times T}$ from the latent hidden state. The decoding process can be represented as follows:

$$\mathbf{z}_1^d = \sigma(\mathbf{W}_{d,1}\mathbf{z}^e + \mathbf{b}_{d,1}), \tag{21}$$

$$\mathbf{z}^d = \sigma(\mathbf{W}_{d,2}\mathbf{z}_1^d + \mathbf{b}_{d,2}), \tag{22}$$

where the $\mathbf{W}_{d,1}, \mathbf{W}_{d,2}$ and $\mathbf{b}_{d,1}, \mathbf{b}_{d,2}$ respectively denote the learnable weight matrices and bias parameters for the decoder; $\mathbf{z}^d$ is the output of the decoder.

The learning process of global spatiotemporal embedding can be formulated as minimizing the L1 error between the input matrix $\mathbf{X}$ and the output matrix $\mathbf{z}^d$. After obtaining the global spatiotemporal embedding $\mathbf{z}^e \in \mathbb{R}^{d_k \times T}$ from the

autoencoder, we transform it into a new vector as $\mathbf{h}^e \in R^{d_k}$ by a function g to get the final embedding:

$$\mathbf{h}^e = g(\mathbf{z}^e), \tag{23}$$

where in our real-world application we use a **mean** function to achieve better performances. Finally, with the learned global temporal distribution embeddings, we propose to feed the hidden embeddings into Bi-LSTM model for initialization, to help the overall model training. We initialize the initial hidden state h_0 of backward and forward LSTM in the Equation (11). As a result, this global embedding is able to improve the convergence efficiency and estimation performance.

E. Model Training

In this part, we will introduce the training process for our model step by step. Firstly, we pre-train our global temporal embedding module to learn the global temporal distribution by minimizing the error between input $\mathbf{X}$ and reconstructed value $\mathbf{X}_r = \mathbf{z}^d$ denoted by:

$$\mathcal{L}_r = \frac{\|(\mathbf{X} - \mathbf{X}_r) \odot \mathbf{M}\|_1}{\|\mathbf{M}\|_1}. \tag{24}$$

And then, in the spatio-temporal dependency model, we first introduce the learning process of unidirectional LSTM model, in order to converge efficiently we combine three types of loss respectively corresponding to between input with each of temporal estimation, spatial estimation and completed estimation. The total loss at time step t is denoted by:

$$\mathcal{L}_t = \mathcal{L}_1\left(\mathbf{x}_t, \hat{\mathbf{x}}_t\right) + \mathcal{L}_1\left(\mathbf{x}_t, \hat{\mathbf{s}}_t\right) + \mathcal{L}_1\left(\mathbf{x}_t, \hat{\mathbf{c}}_t\right), \tag{25}$$

where $\mathcal{L}_1$ is the standard L1 loss if given $\mathbf{x}_t, \mathbf{y}_t$ as the input and $\mathbf{m}_t$ as the missing value mask, by:

$$\mathcal{L}_1(\mathbf{x}_t, \mathbf{y}_t) = \frac{\|(\mathbf{x}_t - \mathbf{y}_t) \odot \mathbf{m}_t\|_1}{\|\mathbf{m}_t\|_1}. \tag{26}$$

The forward estimation and backward estimation loss are respectively denoted by $\mathcal{L}_t^f$ and $\mathcal{L}_t^b$. Moreover, to enforce the estimation in each step to be consistent from both directions, we follow [19] to introduce the a consistency loss at each time slice t as follows:

$$\mathcal{L}_t^{cons} = \mathcal{L}_1\left(\overrightarrow{\hat{\mathbf{x}}_t}, \overleftarrow{\hat{\mathbf{x}}_t}\right) \tag{27}$$

As a final combination of the all losses in the spatio-temporal knowledge integration module is represented by:

$$\mathcal{L}_t^{overall} = \mathcal{L}_t^{cons} + \mathcal{L}_t^f + \mathcal{L}_t^b \tag{28}$$

The overall optimization objective is to minimize the error of $\mathcal{L}_t^{overall}$.

V. EXPERIMENTS

In this section, we will introduce the experimental details conducted on the real-world dataset for evaluating the performance of the proposed model.

TABLE I
STATISTICS OF THE REAL-WORLD DATASET.

Description	Value
# of traffic flow records	9,585,216
# of missing values	4,376,090
record period	2021-12-01 $\sim$ 2022-02-25
# of the time slices	24,768
average traffic flow	9.64
max traffic flow	212

A. Experimental setup

1) Dataset: In our dataset, the traffic flow volume are recorded every 5 minutes over a period of 86 days (from December 1, 2021 to February 25, 2022). And the road network consists of 29 crossroads, 114 branches and 387 inlanes. The detailed statistics are summarized in Table I.

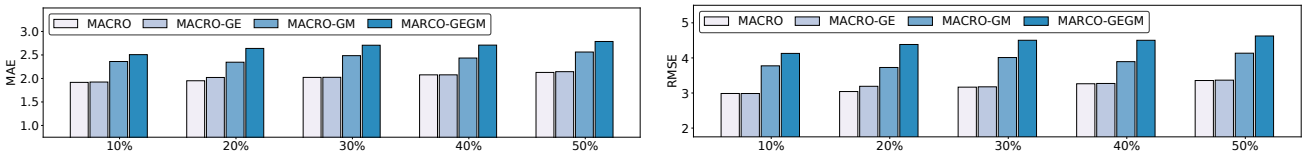
To evaluate the performance, we constructed five datasets by randomly masking the raw observed data with different ratios from 10% to 50% in increments of 10%. In each dataset, the masked records were considered as test set. And the rest records were treated as the training and validation set, in which the proportion of the validation set was a fixed ratio of 10%. All of models were trained at the training set, and their parameters were tuned in the validation set. Finally, all of models were evaluated at the test set.

2) Baselines: We compared MACRO with some state-of-art methods. Specifically, the statistical methods contain Mean, ARIMA [8] and KNN [24]. The tensor-based methods contain LRDMD [27] and LATC [10]. And the deep learning methods contain GRIN [16] and BRITS [19]. The details of these baselines are introduced as follows:

- **Mean** is a heuristic method, which imputes missing value with the mean of all observed values.
- **ARIMA** [8] is a classic time series prediction model. which estimates the missing value at current time slice by the linear combination of the historical data after elementary differencing.
- **KNN** [24] is a clustering method, which leverages the mean values of several the most similar neighboring sensors for imputation.
- **LRDMD** [27] is a matrix completion-based model, which uses a state transition matrix to capture the temporal dependency in traffic state data with low rank regularization.
- **LATC** [10] is a state-of-the-art tensor factorization-based model for traffic data imputation task, which transforms the tensor of traffic state into matrices with lower dimensions and estimates missing values by a modified matrix completion method, which is able to capture global and local temporal relationships in these matrices.
- **GRIN** [16] is one of the most advanced deep learning model for time series imputation task. It learns spatio-temporal representation of multivariate time series by introducing GNN and RNN models to respectively capture the spatial and temporal dependencies within data.
- **BRITS** [19] is a deep learning-based time series im-

TABLE II
THE OVERALL PERFORMANCE OF TRAFFIC FLOW IMPUTATION WITH DIFFERENT PROPORTION OF MASK VALUES.

Methods	10%		20%		30%		40%		50%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
MEAN	6.8411	9.3579	6.8449	9.3476	6.8615	9.4089	6.8878	9.4984	6.9247	9.6042
ARIMA	3.5034	5.6990	4.0275	6.5648	4.6670	7.5742	5.3763	8.6227	6.1505	9.6928
KNN	6.1422	9.5876	6.2199	9.7199	6.4032	10.0172	6.5120	10.1600	6.7523	10.6075
LRDMD	2.4620	3.7339	2.5754	3.9722	2.7215	4.0785	2.7382	4.20267	2.9035	4.4300
LATC	2.3458	3.5702	2.3722	3.6080	2.4101	3.6777	2.4406	3.7313	2.4782	3.7911
BRITS	2.0090	3.1667	2.0731	3.2787	2.0935	3.2953	2.1431	3.3699	2.2155	3.5021
GRIN	2.1851	3.4363	2.1994	3.4513	2.2254	3.5012	2.2654	3.5749	2.2622	3.5779
MACRO	**1.9169**	**2.9861**	**1.9790**	**3.1213**	**2.0227**	**3.1788**	**2.0768**	**3.2735**	**2.1284**	**3.3557**



(a) MAE (b) RMSE

Fig. 5. The performance of MARCO and its variants under different mask ratio.

putation model, it uses a bi-directional RNN to impute missing values in multivariate time series by regarding missing values as variables of the RNN graph.

3) Implementation details: All models were implemented with Tesla V100 16GB GPU. In terms of the hyperparameter settings for MACRO, we respectively set the input time steps T as 24 and the size of output hidden state in autoencoder module as 512. And we choose the distance threshold $\epsilon = 1.5km$ and the order of adjacency matrix $K = 2$. For a fair comparison, all baselines parameters were carefully tuned based on recommended settings.

4) Evaluation metrics: To evaluate the performance of all models, we utilize the widely used *Mean Absolute Error* (MAE) and *Rooted Mean Square Error* (RMSE) as the evaluation metrics, which are widely used for time series imputation problem [19]. And the smaller values of these metrics means better performance.

B. Overall Performance

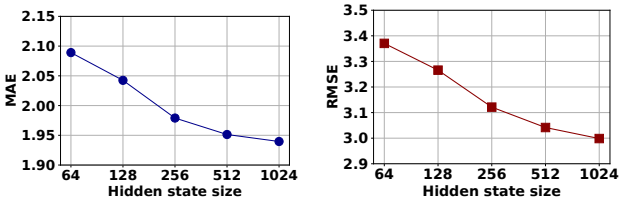
The overall results of different models are summarized in Table II. Obviously, MACRO outperforms all baselines in MAE and RMSE metrics on five datasets. Particularly, MACRO improves the performance by approximately average 5% and 10% respectively to the state-of-the-art approaches of BRITS and GRIN under different mask ratios. Looking further into the results, we can have the following observations. First, the statistical models, i.e., MEAN, ARIMA and KNN, always get the worst performance. Because they impute missing value by exploiting only simple and explicit proximity, which cannot handle the complicated dependencies within traffic data. Second, comparing to the tensor-based models, i.e., LRDMD and LATC, the deep learning based models, i.e., GRIN and BRITS, can achieve better performance. This is due to tensor-based models rely on the low-rank assumption, which may not be perfectly suitable for the traffic data. In addition, the deep learning based models have stronger representational capabilities and can capture non-linear dependencies in the data. Third, although GRIN and BRITS are the most comparable baselines, our MACRO still gets better performance. Indeed, they are designed for the general time series imputation, while MACRO is tailored-designed for our LGTFI problem, which can simultaneously model the diversified spatial correlations and handle the temporally continuous data missing issue within lane-level traffic flows. Finally, comparing different datasets, the performance of MACRO shows a gradually decreasing trend as the mask ratio increases. Nevertheless, it still achieves the best performance, which demonstrates the robustness of MACRO with respect to data sparsity.

C. Ablation Study

To further investigate the effectiveness of each component of MACRO, we conducted experiments to compare the performance of MACRO with its three variants:

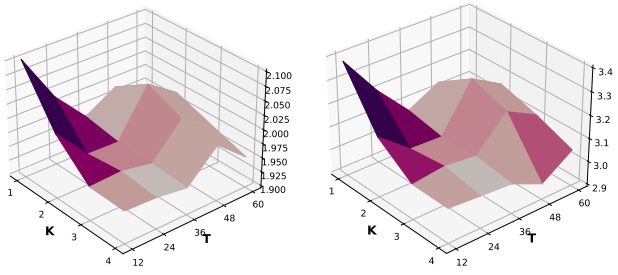
- **MACRO-GE**: It replaces the initial hidden states in temporal dependency module with all-zero vectors.
- **MACRO-MG**: It drops the spatial dependency module in spatio-temporal knowledge integration of MACRO.
- **MACRO-GEMG**: It only reserves a temporal dependency module of MACRO.

The experimental setting is the same as the overall experiment. As shown in Fig. 5, MARCO-MG and MACRO-GEMG perform significantly lower than MACRO and MACRO-GE, which clearly demonstrates that capturing the complicated spatial dependency of lanes plays an important role for LGTFI task. By comparing MACRO with MRCRO-GE, we find that the absence of global spatio-temporal information slightly impairs the imputation accuracy. Finally, from the comparison between MACRO and MACRO-GEMG, the great improvement substantiates the great significance of spatio-temporal knowledge integration for traffic flow imputation.



(a) MAE (b) RMSE

Fig. 6. Parameter sensitive test on hidden state size



(a) MAE (b) RMSE

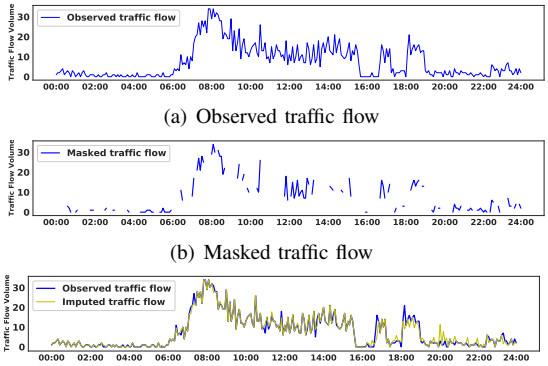
Fig. 7. Parameter sensitive test on k-order graph and time steps T

D. Parameter Sensitivity

In order to test the impact of different hidden state size of global embedding module, K-order nodes in graph convolution, and input time steps, we conducted three experiments by respectively varying these parameters and setting other parameters as default. First, we vary the hidden state size of global embedding module from 64 to 1024. The results are summarized in Fig. 6. In general, with increasing hidden state size, the performance of MACRO is also improving, since higher dimension of hidden state is able to better represent the spatio-temporal nonlinear correlations of traffic flow. However, in the matter of computation efficiency, higher dimension also leads to a higher computation cost. To trade off the performance and computation cost, we set our hidden state size with a balanced value of 512.

Then, to test the impact of different order neighboring nodes in graph convolution on model performance, we vary the order values K from 1 to 4, the results are summarized in Fig. 7. It is observed that the imputation error presents a decreasing trend, which could be reasonable since more spatial information is conducive to improving performance. However, improvement is not significant when K is bigger than 2, which is possible because $K = 2$ has been able to aggregate adequate spatial knowledge for imputation.

Finally, to test the impact of the input time step T, we vary its value from 12 to 60, the result is also reported in Fig. 7. Obviously, MACRO performance fluctuates in a range and it achieves least errors when $T = 24$. It was reasonable that an extremely short input cannot provide enough temporal information to model, whereas long input introduced noise temporal information into the model.

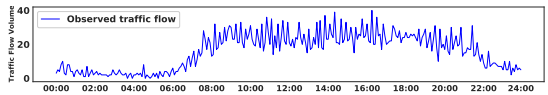


(a) Observed traffic flow

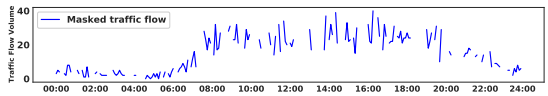
(b) Masked traffic flow

(c) Imputed traffic flow

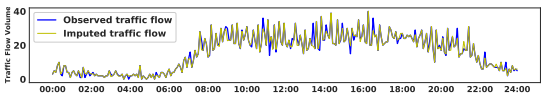
Fig. 8. Case study on weekday.



(a) Observed traffic flow



(b) Masked traffic flow



(c) Imputed traffic flow

Fig. 9. Case study on weekend.

E. Case Study

We further examined the effectiveness of MACRO with two specific experimental cases derived from real traffic flow of a randomly selected lane l_i separately in weekday and weekend. We take the weekday traffic flow as an example. Fig. 8(a) presents the observed traffic flow of l_i in a weekday. In the training process, some values of the observed traffic flow volume were randomly masked to train model. The data distribution of l_i used for training is shown in Fig. 8(b), and the gaps between Fig. 8(a) and Fig. 8(b) represent the masked observation of l_i. After the imputation process, we filled the gaps with estimated traffic flow volume by MACRO, and compared the imputed traffic flow distribution with observed distribution, the result is shown in Fig. 8(c). As can be seen, the imputed traffic flow distribution is similar to the original observed distribution, which clearly demonstrates the effectiveness of MACRO. On the other hand, the imputation result on weekend traffic flow is reported in Fig. 9(c), and it is observed that the distribution of observed traffic flow volume and imputed distribution are significantly similar at most time slices. It further demonstrates that MACRO can capture diversified patterns of traffic flow.

VI. Conclusion

In this paper, we investigated a new lane-level traffic flow imputation problem, and proposed a Multi-grAph Convolutional Recurrent netwOrk (MACRO) framework to solve it.

To capture the diversified spatial correlations within lane-level traffic flows, we designed a spatial dependency module, where the multi-relation graphs were constructed to consider different kinds of correlations from various perspective, and a multi-graph convolution neural network was proposed to model the integrated spatial dependencies between lanes. Then, we adopted a modified bi-directional recurrent neural network to model the temporal dependency of each lane, and employed a temporal decay mechanism to control the irregular information transfer between adjacent time slices for handling the temporally continuous data missing issue. After that, we devised a spatio-temporal knowledge integration module to comprehensively integrate multi-resolution spatiotemporal knowledge for traffic flow imputation. Finally, extensive experiments on the real-world dataset demonstrated the imputation performance of MACRO comparing with several state-of-the-art baselines.

VII. Acknowledgement

This work is supported in part by Foshan HKUST Projects (FSUST21-FYTRI01A, FSUST21-FYTRI02A).

References

[1] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 630–638, 2010.

[2] N. Zhang, F.-Y. Wang, F. Zhu, D. Zhao, and S. Tang, "Dynacas: Computational experiments and decision support for its," *IEEE Intelligent Systems*, vol. 23, no. 6, pp. 19–23, 2008.

[3] S. Turner, L. Albert, B. Gajewski, and W. Eisele, "Archived intelligent transportation system data quality: preliminary analyses of san antonio transguide data," *Transportation Research Record*, vol. 1719, no. 1, pp. 77–84, 2000.

[4] C. Chen, J. Kwon, J. Rice, A. Skabardonis, and P. Varaiya, "Detecting errors and imputing missing data for single-loop surveillance systems," *Transportation Research Record*, vol. 1855, no. 1, pp. 160–167, 2003.

[5] Y. Li, Z. Li, and L. Li, "Missing traffic data: comparison of imputation methods," *IET Intelligent Transport Systems*, vol. 8, no. 1, pp. 51–57, 2014.

[6] Y. Duan, Y. Lv, W. Kang, and Y. Zhao, "A deep learning based approach for traffic data imputation," in *17th International IEEE conference on intelligent transportation systems (ITSC)*. IEEE, 2014, pp. 912–917.

[7] G. Chang, Y. Zhang, and D. Yao, "Missing data imputation for traffic flow based on improved local least squares," *Tsinghua Science and Technology*, vol. 17, no. 3, pp. 304–309, 2012.

[8] N. L. Nihan, "Aid to determining freeway metering rates and detecting loop errors," *Journal of Transportation Engineering*, vol. 123, no. 6, pp. 454–458, 1997.

[9] H. Tan, G. Feng, J. Feng, W. Wang, Y.-J. Zhang, and F. Li, "A tensor-based method for missing traffic data completion," *Transportation Research Part C: Emerging Technologies*, vol. 28, pp. 15–27, 2013.

[10] X. Chen, M. Lei, N. Saunier, and L. Sun, "Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[11] X. Chen, Z. He, Y. Chen, Y. Lu, and J. Wang, "Missing traffic data imputation and pattern discovery with a bayesian augmented tensor factorization model," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 66–77, 2019.

[12] W. Zhang, H. Liu, J. Han, Y. Ge, and H. Xiong, "Multi-agent graph convolutional reinforcement learning for dynamic electric vehicle charging pricing," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2471–2481.

[13] L. Zhang, D. Zhou, H. Zhu, T. Xu, R. Zha, E. Chen, and H. Xiong, "Attentive heterogeneous graph embedding for job mobility prediction," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2192–2201.

[14] Z. Guo, H. Liu, L. Zhang, Q. Zhang, H. Zhu, and H. Xiong, "Talent demand-supply joint prediction with dynamic heterogeneous graph enhanced meta-learning," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2957–2967.

[15] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.

[16] A. Cini, I. Marisca, and C. Alippi, "Multivariate time series imputation by graph neural networks," 2022.

[17] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640.

[18] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.

[19] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "Brits: Bidirectional recurrent imputation for time series," *Advances in neural information processing systems*, vol. 31, 2018.

[20] W. Fan, S. Zheng, X. Yi, W. Cao, Y. Fu, J. Bian, and T.-Y. Liu, "Depts: Deep expansion learning for periodic time series forecasting," *arXiv preprint arXiv:2203.07681*, 2022.

[21] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[22] W. Yin, P. Murray-Tuite, and H. Rakha, "Imputing erroneous data of single-station loop detectors for nonincident conditions: Comparison between temporal and spatial methods," *Journal of Intelligent Transportation Systems*, vol. 16, no. 3, pp. 159–176, 2012.

[23] Z. Liu, S. Sharma, and S. Datla, "Imputation of missing traffic data during holiday periods," *Transportation Planning and Technology*, vol. 31, no. 5, pp. 525–544, 2008.

[24] B. Sun, L. Ma, W. Cheng, W. Wen, P. Goswami, and G. Bai, "An improved k-nearest neighbours method for traffic time series imputation," in *2017 Chinese Automation Congress (CAC)*. IEEE, 2017, pp. 7346–7351.

[25] L. Qu, L. Li, Y. Zhang, and J. Hu, "Ppca-based missing data imputation for traffic flow volume: A systematical approach," *IEEE Transactions on intelligent transportation systems*, vol. 10, no. 3, pp. 512–522, 2009.

[26] J.-M. Chiou, Y.-C. Zhang, W.-H. Chen, and C.-W. Chang, "A functional data approach to missing value imputation and outlier detection for traffic flow data," *Transportmetrica B: Transport Dynamics*, vol. 2, no. 2, pp. 106–129, 2014.

[27] Y. Yu, Y. Zhang, S. Qian, S. Wang, Y. Hu, and B. Yin, "A low rank dynamic mode decomposition model for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6547–6560, 2020.

[28] W. Zhang, H. Liu, Y. Liu, J. Zhou, and H. Xiong, "Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 1186–1193.

[29] Y. Duan, Y. Lv, Y.-L. Liu, and F.-Y. Wang, "An efficient realization of deep learning for traffic data imputation," *Transportation research part C: emerging technologies*, vol. 72, pp. 168–181, 2016.

[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[31] J. Yoon, J. Jordon, and M. Schaar, "Gain: Missing data imputation using generative adversarial nets," in *International conference on machine learning*. PMLR, 2018, pp. 5689–5698.

[32] H. Qin, X. Zhan, Y. Li, X. Yang, and Y. Zheng, "Network-wide traffic states imputation using self-interested coalitional learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1370–1378.

[33] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *International Joint Conference on Artificial Intelligence 2019*, 2019, pp. 1907–1913.

[34] X. Tang, B. Gong, Y. Yu, H. Yao, Y. Li, H. Xie, and X. Wang, "Joint modeling of dense and incomplete trajectories for citywide traffic volume inference," in *The World Wide Web Conference*, 2019, pp. 1806–1817.