Transportation Research Part C 156 (2023) 104331

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

# A macro–micro spatio-temporal neural network for traffic prediction

Siyuan Feng [a,b,c], Shuqing Wei [a,f], Junbo Zhang [b,c,*,1], Yexin Li [b,c], Jintao Ke [d], Gaode Chen [e], Yu Zheng [b,c], Hai Yang [a,f]

[a] *Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China*
[b] *JD iCity, JD Technology, Beijing, China*
[c] *JD Intelligent Cities Research, China*
[d] *Department of Civil Engineering, The University of Hong Kong, Hong Kong, China*
[e] *Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
[f] *Intelligent Transportation Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong, China*

ARTICLE INFO

*Keywords:*
Traffic prediction
Graph convolution
Attention mechanism
Urban computing

## ABSTRACT

Accurate traffic prediction is crucial for planning, management and control of intelligent transportation systems. Most state-of-the-art methods for traffic prediction effectively capture complex traffic patterns (e.g. spatial and temporal correlations of traffic data) by employing spatio-temporal neural networks as prediction models, together with graph convolution networks to learn spatial correlations of prediction objects (*e.g.* traffic states of road segments, as in this study). Such spatial correlations can be regarded as micro correlations. However, there are also macro correlations between regions, each of which is composed of multiple road segments or artificially partitioned areas. Macro correlations represent another type of interaction within road segments, and should be carefully considered when predicting traffic. The diversity of micro spatial correlations and corresponding macro spatial correlations (*e.g.* correlations based on physical proximity or traffic pattern similarity) further increases the complexity of traffic prediction. We overcome these challenges by developing a macro–micro spatio-temporal neural network model, denoted 'MMSTNet'. MMSTNet captures spatio-temporal patterns by (a) utilizing a graph convolution network and a spatial attention network to capture micro and macro spatial correlations, respectively; (b) employing a temporal convolution network and a temporal attention network to learn temporal patterns; and (c) integrating hierarchically learned representations based on designed attention mechanisms. We perform evaluations on two real-world datasets and thereby demonstrate that MMSTNet outperforms state-of-the-art models in traffic prediction tasks.

## 1. Introduction

Intelligent transportation systems (ITSs) are important components of future smart cities. Rapid improvements in data collection, communication and storage methods mean that ITSs have wide applications in modern traffic management systems. A key task for ITS implementation is high-accuracy traffic prediction, which entails forecasting traffic speed, flow or density within real

[*] Corresponding author at: JD iCity, JD Technology, Beijing, China.
*E-mail address:* msjunbozhang@outlook.com (J. Zhang).

[1] The research was done when the first author was an intern at JD iCity, JD Technology, China and JD Intelligent Cities Research under the supervision of Junbo Zhang. The code for the paper is open-sourced at https://github.com/sfengag/MMSTNet.

https://doi.org/10.1016/j.trc.2023.104331
Received 26 February 2023; Received in revised form 3 September 2023; Accepted 4 September 2023
Available online 18 September 2023

infrastructure (*e.g.* road segment) or artificially partitioned areas based on previously collected data. Traffic prediction can support multiple downstream tasks in ITSs. For example, the prediction of road-wise traffic speeds can be utilized for congestion prediction, allowing road management agencies to make real-time adjustments to their pricing strategies for road service, and thus control propagation of congestion in advance. In addition, predicted traffic speeds can be used to determine the expected travel time on each road segment in a network, and the travel time can be employed to plan routes for traveling in the road network. Other applications for traffic prediction include crowd flow prediction (Zhang et al., 2017, 2018) and ride-sourcing demand prediction (Feng et al., 2021; Ke et al., 2021), and some downstream applications.

For traffic predictions, accuracy is the most important goal. The first-developed prediction models include historical average (HA) (Smith, 1995; Pan et al., 2012), autoregressive integrated moving average (ARIMA) (Kumar and Vanajakshi, 2015), and support-vector regression models (Wu et al., 2004). However, the performances of these models are usually limited by their capacity to capture complex patterns in traffic data. The emergence and popularization of graphics processing units, and the improved accessibility and quality of traffic data, have led to neural network (nn) models becoming important for traffic prediction. In particular, deep nn models have high capacities for nonlinear function approximation and thus are typically effective in performing complex prediction tasks.

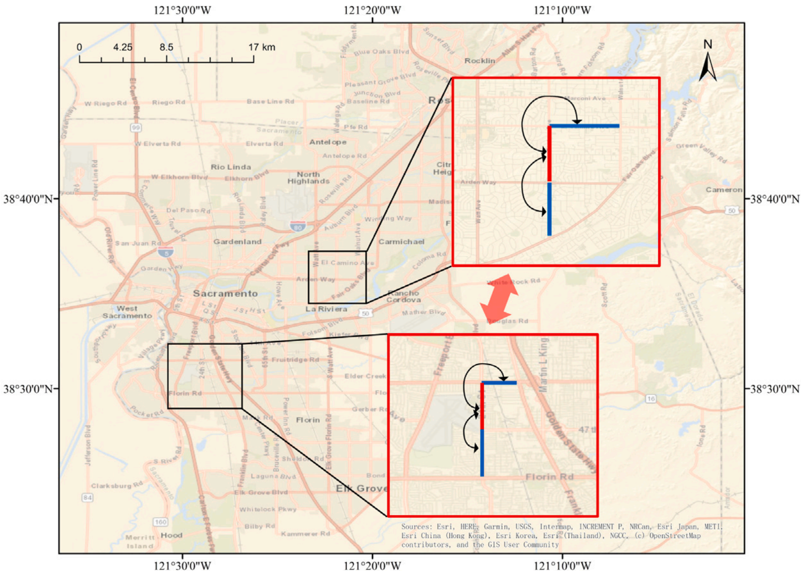
The general structure and training process of nn models are standardized but their designs must be tailored to a given scenario and data type. Complex patterns in traffic data are typically reflected in two properties, as described in Zheng et al. (2014), Zheng (2019). The first property of traffic data is the spatial correlation between prediction objects (*e.g.* speeds on road segments), which means that the traffic patterns of a target road segment are affected by those of its neighboring road segments in the road network, as shown by the road segments in the red boxes of Fig. 1. We denote a direct spatial correlation between prediction objects a 'micro spatial correlation', or 'micro correlation', for simplicity. Analogously, we denote a scenario in which a collection of micro prediction objects interact and affect each other's macro traffic patterns a 'macro spatial correlation', or 'macro correlation', for simplicity. The macro correlation of regions can be used as a different way to observe the spatial relationships within micro objects. For example, in Fig. 1, examining only the micro correlations reveals that the two red road segments are not neighbors of each other, since they are not directly adjacent with each other. This means that the relationship between their traffic patterns is ignored. In contrast, the two regions that contain these two red segments are both residential areas, and thus may share some similar traffic patterns, representing some macro spatial correlation with each other. Therefore, the traffic patterns of one red segment and its neighborhood may first affect the traffic situation of its own region, and then the other region via macro correlation. Since another red segment is included in the latter region, its traffic pattern may be also influenced.

The complexity of traffic prediction is increased by the diversity of micro correlations and their corresponding macro correlations. In a single road network, road segments can be spatially correlated for multiple reasons, such as their spatial proximity, public transit connections, or the similarity of their historical traffic patterns. Road segments with similar historical traffic patterns can be regarded as highly correlated even if they are physically distant from each other. In such a situation, considering only one type of correlation (e.g., spatial proximity) may mean that the intrinsic relationships between prediction objects are not fully captured, thereby degrading prediction accuracy. Moreover, the diversity of micro correlations means that there are different types of macro correlations; that is, neighborhoods built based on different micro correlations naturally form different regions, resulting in different corresponding macro correlations. For example, two regions' traffic patterns may be correlated if they are both residential areas, or because they are close to each other physically.

Many studies have considered micro correlations. For example, Feng et al. (2021) designed a multi-task prediction framework to forecast ride-sourcing demands that uses two different adjacency matrices for graph convolution. Li et al. (2017b) devised a graph convolution framework that considers a bi-directional diffusion process and employs several different types of adjacency for micro spatial learning. However, most studies have ignored the micro-macro hierarchy of spatial correlations (Feng et al., 2021; Yu et al., 2017; Seo et al., 2018) or the diversity of micro correlations and their corresponding macro correlations (Guo et al., 2021), and thus their models may learn incomplete spatial information. In addition, even if multiple spatial correlations are considered, it can be highly challenging to effectively integrate the representations they each generate. For example, regular integration may involve direct summation or concatenation and thus may fail to examine differences between the importances of representations during an entire learning process. This type of problem is encountered in pure micro spatial learning when multiple graphs are involved and can be more complicated when macro spatial learning is considered. To handle the challenge, a proper integration approach must be developed for micro-level learned representations and macro-level learned representations, respectively, and for micro-macro representations.

The second property of traffic data is the temporal correlation between traffic patterns. That is, the correlation between the traffic pattern at a certain time step and the patterns at previous time steps. The early models that were developed to consider temporal correlations for traffic prediction include long short-term memory (LSTM) models and gated recurrent unit (GRU) models, both of which can capture the sequential relationships between different time steps. LSTM and GRU models are often viewed as temporal learning modules and are integrated with graph convolutional networks (GCNs) to learn spatio-temporal correlations, instead of being utilized individually. However, despite the ability of LSTM and GRU models to capture relationships between a small range of temporal data (Cui et al., 2018; Agarap, 2018), they may not be able to directly capture correlations between a large range of time steps in an efficient manner. Therefore, the simultaneous integration of temporal correlations within both close and distant time steps should be carefully considered during the design of models.

To solve the above-described problems, we develop a macro–micro spatio-temporal nn model for traffic prediction. We first collect historical traffic data to serve as micro features. Next, we pool the micro features for each region as the macro features for that region. In addition, we consider both physical adjacency and semantic adjacency in traffic patterns, and construct multiple graphs



Sources: Esri, HERE, Garmin, USGS, Intermap, INCREMENT P, NRCan, Esri Japan, METI, Esri China (Hong Kong), Esri Korea, Esri (Thailand), NGCC, (c) OpenStreetMap contributors, and the GIS User Community

**Fig. 1.** Representations of different micro and macro correlations. The two red boxes represent two residential regions, each of which contain several road segments, including the red segments. The black arrows represent micro spatial correlations, and the red arrows represent macro spatial correlations.

to represent each of these adjacencies for micro learning. We also prepare for macro learning by carefully defining regions based on multi-graphs at the microscopic level and generating their corresponding spatial embeddings. Next, spatio-temporal patterns are captured by feeding micro and macro features, multi-graphs and region embedding, and temporal embedding into a stack of macro–micro learning modules. Each macro–micro learning module is composed of a macro learning block, a micro learning block and a macro–micro fusion module. The micro features are input into the micro learning blocks to generate representations for road segments, whereas the macro features are input into the macro learning blocks to determine the interactions between regions.

In a micro learning block, a combination of temporal convolution network (TCN) and temporal attention network is first used to capture local and global temporal correlations, respectively. Compared with recurrent nns (RNNs) and related temporal sequence learning modules (e.g., LSTM, GRU, and RNN modules), TCNs and a temporal attention networks are more efficient in computation. After temporal learning, the representations are fed into GCN to exploit the micro spatial correlations. GCNs are effective as a general tool for aggregating spatial information of graph data (*e.g.* traffic speed on road segments within a road network) (Zhou et al., 2020). The adjacency matrix is a core component of GCN and when it is relatively sparse, the GCN's computational complexity can be improved from $\mathcal{O}(N^2)$ to $\mathcal{O}(|E|)$, where $N$ and $|E|$ are the number of nodes and edges, respectively, of the studied network. As there can be a very large number of nodes in the infrastructure networks of some metropolitan areas, it is proper to utilize a GCN for micro spatial learning. Specifically, a multi-graph GCN module is employed for adaptation to the diversity of micro spatial correlations As mentioned, micro spatial correlations exist when there are relationships between the information or representations of different road segments. Such correlated road segments are regarded as neighbors. Thus, the new representation of a given road segment is composed of its original representation and those of its neighbors, and is determined by the GCN in the micro learning block. As there are various definitions of a neighborhood, there are various types of micro spatial correlations, and thus different ways to generate new representations during micro learning. Given that the representations learned from different graphs for neighborhoods may have different importances, we integrate these representations based on a designed attention mechanism. An adaptive context embedding is developed as one of components of the attention mechanism to increase the adaptability of integration to varying input data.

In a macro learning block, the temporal learning part is exactly the same as in a micro block. However, a spatial attention network is used for macro spatial learning. The complexity of spatial attention is $\mathcal{O}(N^2)$, which is too high for micro learning but acceptable for macro learning, as there are far fewer regions in macro learning than road segments in micro learning. Moreover, it is more difficult to define an adjacency matrix for regions in macro learning than for road segments in micro learning, but a spatial attention network can adaptively and automatically find the required macro correlations. Furthermore, recall that macro regions are generated based on different micro correlations, and thus the structural information of the latter is also important for the former. Accordingly, a fusion node embedding is used as the positional embedding for the spatial attention network. This embedding is generated based on different neighborhood structures from multi-graphs and thus introduces diversity into the process of macro learning. Thus, macro spatial learning simultaneously considers different types of micro structural information in one module rather than in multiple modules, thereby saving computational resources.

A macro–micro fusion module is also developed to integrate representations learned from micro and macro learning blocks, respectively. This module is also based on the attention mechanism and can therefore dynamically and adaptively adjust the importance of representations from micro and macro learning. Moreover, matrix factorization technology is incorporated into the module to ensure that the heterogeneity and homogeneity of different regions are sufficiently considered. The final multi-step prediction results are output from a stack of fully connected layers. The above-described method is evaluated on two real-world datasets and exhibits performance superior to that of the baseline methods.

Overall, the work described in this paper makes the following contributions to the literature.

- We devise a macro–micro spatiotemporal nn model for traffic prediction. The model is capable of capturing multiple types of micro and macro spatial correlations, and both local and global temporal correlations.
- The model employs a combination of a GCN and a spatial attention network to accommodate the characteristics of micro and macro spatial learning. In addition, the model integrates a TCN and a temporal attention network to learn local and global temporal correlations simultaneously.
- Different fusion layers are developed to integrate learned representations for multiple micro graphs, and micro and macro blocks based on the attention mechanism and matrix factorization.
- Extensive experiments are implemented on real-world traffic datasets. The results demonstrate that the model outperforms state-of-the-art baselines.

The remainder of the paper is organized as follows. In Section 2, we provide a literature review on classic and recent approaches to traffic prediction. In Section 3, we provide some preliminaries for our work. In Sections 4 and 5 we present the details of the model framework and numerical experiments, respectively. In Section 6, we provide our conclusions and describe directions for future research.

## 2. Literature review

In this section, we first briefly cover the history of traffic prediction, and then review the mechanism and application of graph convolution and attention approaches to traffic prediction.

### 2.1. Traffic prediction

As mentioned, the goal of traffic prediction is to forecast traffic states based on historical traffic data. The traffic states can be traffic flow, density or speed for different modes of transportation, such as vehicles (Li et al., 2017b; Guo et al., 2021), train Yu et al. (2022, 2020), subway (Li et al., 2017a; Chen et al., 2019), etc. The types of traffic prediction methods to be first developed, such as HA (Pan et al., 2012) and ARIMA (Kumar and Vanajakshi, 2015; Williams and Hoel, 2003), focus on the temporal relationship between historical data and make predictions for a single prediction object (*e.g.* a single road segment). These types of methods utilize simple functions (*e.g.* linear functions) to delineate this relationship. Subsequently, the development of deep nn models enabled simple functions to be replaced by more complex functions, which enhanced these models' representation capabilities. For example, LSTM and GRU models are widely applied for time-series prediction tasks, such as traffic forecasting (Fu et al., 2016; Zhao et al., 2017; Agarap, 2018). LSTM and GRU models perform the same basic task: they generate a prediction based on the previous value in a sequential manner and thereby forecast the values of a future time series. However, these types of models examine temporal correlations but ignore spatial correlations between nodes.

Accordingly, spatio-temporal frameworks have been developed, which contain an integrated convolution network-based module to perform spatial correlation learning. A classic framework is convolutional neural network (CNN) + LSTM/GRU (Zhang et al., 2017, 2018), which partitions a study area into standard grid cells and then applies a CNN to capture correlations of traffic patterns between these grid cells. However, as discussed in Feng et al. (2021), a CNN may not be well applicable to a graph data scenario, such as making traffic predictions for a road network. This problem was solved by replacing a CNN with a GCN, such that the convolution operation is transformed from the Euclidean domain to the non-Euclidean domain. For example, for traffic prediction (Yu et al., 2017) adopted the GCN developed by Defferrard et al. (2016) to capture spatial correlations, and a combination of one-dimensional (1D) convolution and gated linear units as the temporal learning module. A similar type of GCN was employed in Sun et al. (2020) to predict traffic flows, with the major focus being to fetch multiple temporal correlations of different time ranges. In Li et al. (2017b), a diffusion convolution layer was devised to consider the bi-directional diffusion of traffic in a directional network, such as a road network, and a novel GCN was integrated into an encoder–decoder framework to make traffic predictions. In addition to pure neural network based approaches, other technology is also widely explored for traffic prediction, such as multi-streaming learning (Yu et al., 2021, 2022), federated learning (Liu et al., 2020), transfer learning (Huang et al., 2021), etc. In this research, we still focus on the optimization of neural network based application.

However, most neural network based studies have not considered the existence of macro correlations nor the diversity of micro correlations and their corresponding macro correlations. This dependence on a single type of correlation (*e.g.* physical adjacency in Yu et al. (2017), Sun et al. (2020), Li et al. (2017b)) may mean that other important spatial relationships are not detected and thus negatively affect model performance. This problem is solved in the current study by using a macro learning block in addition to a micro learning block and by considering different correlations.

### 2.2. GCN

Standard convolutional networks have been widely applied in fields such as computer vision and speech recognition, and others that involve prediction or classification tasks. However, as mentioned above, in a graph data scenario, it is better to adopt a GCN than a standard convolution network to aggregate information in the non-Euclidean domain. GCNs have been developed for over 7 years and have become a common tool in various applications, such as protein structure prediction and design (Fout et al., 2017; Strokach et al., 2020), recommendation systems (He et al., 2020; Wei et al., 2019) and traffic prediction (Yu et al., 2017; Li et al., 2017b; Guo et al., 2021). As discussed in Wu et al. (2020), GCNs can be divided into two groups: spectral-type GCNs (Defferrard et al., 2016; Kipf and Welling, 2016; Henaff et al., 2015) and spatial-based GCNs (Hamilton et al., 2017; Monti et al., 2017). However, both types of GCNs require a clear definition of an adjacency matrix for nodes, as this reveals the exact neighborhood for each node. It is relatively simple to formulate the adjacency for micro correlation. For example, if two road segments are directly connected to each other, they can be regarded as neighbors. In contrast, it can be more difficult to determine the macro adjacency between regions. For example, consider a scenario in which two regions are formulated by clustering road segments (as is described in Section 4). The two regions are physically close to each other but do not share any short connecting paths in the road network. Thus, it is risky to regard them as neighbors. Moreover, even when two regions share several such paths, the traffic volume traversing these paths may be very low, such that the regions remain uncorrelated. This scenario may be even more complicated when different types of macro spatial correlations are integrated. Therefore, we employ a combination of a GCN and an attention mechanism, where the latter manages macro learning. Details on attention approaches are provided in the next subsection.

### 2.3. Attention mechanism

The basis of an attention mechanism is simple. First, the importance of different representations is formulated via a softmax operation. Next, a weighted summation is conducted, with representations as values and calculated importances as weights, to generate a new representation. This completes the general process of the attention approach. As the new representation is obtained using dynamic weights, the attention mechanism enhances the capability of an nn to capture the changing patterns of data. Similar to GCNs, attention mechanisms have been broadly utilized in various fields of research and industry, such as natural language processing (Vaswani et al., 2017; Hu, 2019) and computer vision (Bello et al., 2019; Ramachandran et al., 2019). The most prominent attention mechanism structure devised in recent years is the Transformer model (Vaswani et al., 2017). However, direct application of such attention mechanism approaches to traffic prediction may generate a slow model, as these approaches' computational complexity is $\mathcal{O}(N^2)$, where $N$ is the number of nodes. This value can be rather large for a real road network, as discussed in Zheng et al. (2020). Therefore, in the current study, we apply an attention network only in macro spatial learning. This means that the adjacency definition problem for macro correlation learning can be solved, as unlike a GCN, an attention network does not require an explicit adjacency matrix. Moreover, the number of regions is much smaller than the number of road segments, and thus the computational difficulty associated with the attention network is decreased. In addition to applying the attention mechanism to learn spatial correlations, we also utilize it to learn temporal correlations, following the attention-based series prediction approach (Vaswani et al., 2017; Zheng et al., 2020) and with temporal embedding as positional embedding. However, in our approach, a TCN is integrated with temporal attention to strengthen the learning of local temporal correlations. Moreover, we apply the attention mechanism to adaptively and dynamically combine different representations learned from different spatial structures.

## 3. Preliminaries

### 3.1. Basic definitions

We first provide some fundamental definitions for the formulation of traffic prediction in road networks.

**Definition 1.** (Graph) The graph we use is defined as $G(V, E, A)$, where $V$ is the set of road segments, $N = |V|$ is the size of $V$, $E$ is the set of edges with different weights, and $A \in \mathbb{R}^{N\times N}$ denotes the adjacency matrices. Multiple graphs can be constructed using different settings for edge weights and adjacency matrices.

**Definition 2.** (Traffic data vectors) We use $\boldsymbol{x}_t^v \in \mathbb{R}^C$ to represent the traffic data in road segment $v$ at time $t$, where $C$ is the size of the feature dimensions. Correspondingly, $\boldsymbol{X}_t = [\boldsymbol{x}_t^1, \dots, \boldsymbol{x}_t^N]^T \in \mathbb{R}^{N\times C}$ is the feature data vector of the whole network at time $t$.

### 3.2. Research problem

Given $T1$ historical traffic data points for a road network, denoted by $\{\boldsymbol{X}_1, \dots, \boldsymbol{X}_{T1}\} \in \mathbb{R}^{T1\times N\times C}$, predict the next T2 traffic states $\{\hat{\boldsymbol{X}}_{T1+1}, \dots, \hat{\boldsymbol{X}}_{T1+T2}\} \in \mathbb{R}^{T2\times N\times C}$. The true value for prediction is $\{\boldsymbol{X}_{T1+1}, \dots, \boldsymbol{X}_{T1+T2}\} \in \mathbb{R}^{T2\times N\times C}$.

## 4. Model framework

In this section, we first introduce the process of data preparation, in which we generate regions and their spatial embeddings for macro learning, and multi-graphs for micro spatial learning. We then provide an overview of the model framework, followed by details on micro and macro learning blocks, and their integration layers. Finally, we specify the forecasting blocks and loss function.

### 4.1. Data preparation

The entire process of data preparation is shown in Fig. 2. The major goal of spatial data preparation is to generate multi-graphs to enable the GCN to learn micro correlations and regions, together with their embeddings, to enable the spatial attention network to learn macro correlations. The process starts with multi-graph generation, in which we utilize different graphs to represent the physical relationship of road segments and their intrinsic pattern correlations. We build two graphs that have the same nodes but different edge weights, namely, different adjacent matrices. The first graph represents the physical proximity of road segments. To construct this graph, we first compute the physical road network distances $dist(v_i, v_j)$ between each pair of vertices $v_i$ and $v_j$, and then construct the adjacent matrix using a thresholded Gaussian kernel-weighting function (Shuman et al., 2013). The edge weight between $v_i$ and $v_j$ can be calculated by $[\boldsymbol{A}_p]_{i,j} = exp(-\frac{dist(v_i,v_j)^2}{\sigma^2})$ if $dist(v_i, v_j) < \kappa_p$, otherwise 0. In this formula, $\kappa_p$ is the threshold of distances, and $\sigma$ is their standard deviation. The second graph reflects the intrinsic pattern correlations between road segments and is constructed based on historical data similarity. That is, $[\boldsymbol{A}_h]_{i,j} = \frac{Cov(S_i,S_j)}{\sqrt{var(S_i)var(S_j)}}$. There is also a threshold $\kappa_h$, and the edge weight that is less than the threshold is set to 0. This process is scalable to settings with more graphs. Subsequently, node embeddings are generated for each graph based on the adjacency matrices of each graph via the node2vec approach (Grover and Leskovec, 2016). This method can capture both local and global structural information of a node in a given graph. We then concatenate embeddings from different graphs for each node to obtain fusion node embeddings, which integrate different structural information of multiple graphs. Next, the k-means method is implemented on the fusion node embeddings, and the regions are obtained based on multiple micro graphs rather than on a single micro graph. Finally, the node embeddings for nodes within the same region are summed to generate the regional embedding.

The goal of temporal data preparation is to generate a temporal embedding for each time step in a temporal attention network. First, two properties of a given time step, namely the time of day and day of the week, are represented in one-hot vectors. Next, these two vectors are concatenated for each time step to form temporal embedding.

### 4.2. Overview

Our model is shown in Fig. 3. The input comprises three parts: micro and macro-level input data, fusion region embedding and multi-graphs, and temporal embedding. The latter two parts are constructed as described in the previous sub-section. We calculate the mean, minimum and standard error of the micro-level input traffic data within a given region, and use these three metrics as the macro-level input features for the region. Next, the micro and macro input data are each fed into a fully connected (FC) layer, and then input into a macro–micro learning module composed of a micro learning block, a macro learning block and a macro–micro fusion module. In the micro block, the hidden representations are sequentially input into temporal attention networks for global temporal correlation learning (with temporal embedding as the positional embedding) and into the TCN for strengthened local correlation learning, as the patterns in close time steps generally have the strongest correlations with each other. The learned representations are then input into a multi-graph GCN, which is responsible for learning different types of micro spatial correlations. Next, the micro fusion module integrates the results learned from multi-graphs based on the attention mechanism and an adaptive context vector. The same process is implemented for the macro learning block, except that the spatial learning part is replaced by a spatial attention network. Specifically, we process the spatial embedding via an FC layer and use the products as positional embeddings for spatial attention. Thus, the spatial attention network can utilize structural information from different micro graphs. The representations output from the micro and macro blocks interact with each other in the macro–micro fusion module, which is constructed based on attention mechanism and matrix factorization approaches. High-dimension spatio-temporal patterns can be extracted from a stack of macro–micro learning modules connected sequentially. The final predictions are output via a stack of FC layers that serve as a forecasting block. The details of each module are provided in the following sub-sections.

### 4.3. Micro learning block

As shown in Fig. 3, a micro learning block successively comprises a temporal attention network, a TCN, a multi-graph GCN and a micro fusion module. In this subsection, we describe each module.

(1) Temporal attention network. Based on the ideas in Vaswani et al. (2017) and Zheng et al. (2020), we first adopt an attention mechanism to model the correlations between different time steps. As temporal correlations are influenced by both traffic patterns and the corresponding time steps, the attention score is computed based on a combination of hidden representation and temporal embedding in a multi-head manner. The specific process is shown below.

$$u^w_{t_j,t} = \frac{\langle f^w_{tem,1}(\boldsymbol{h}^{v_i}_{t_j} \parallel \boldsymbol{em}_{tem,t_j}), f^w_{tem,2}(\boldsymbol{h}^{v_i}_{t} \parallel \boldsymbol{em}_{tem,t})\rangle}{\sqrt{d}} \tag{1}$$

$$\alpha^w_{t_j,t} = \frac{exp(u^w_{t_j,t})}{\sum_{t_a \in N_{tem}(t_j)} exp(u^w_{t_j,t_a})} \tag{2}$$

$$\boldsymbol{h}'^{v_i}_{t_j} = \parallel^{W}_{w=1}\{\sum_{t \in N_{tem}(t_j)} \alpha^w_{t_j,t} f^w_{tem,3}(\boldsymbol{h}^{v_i}_{t})\} \tag{3}$$

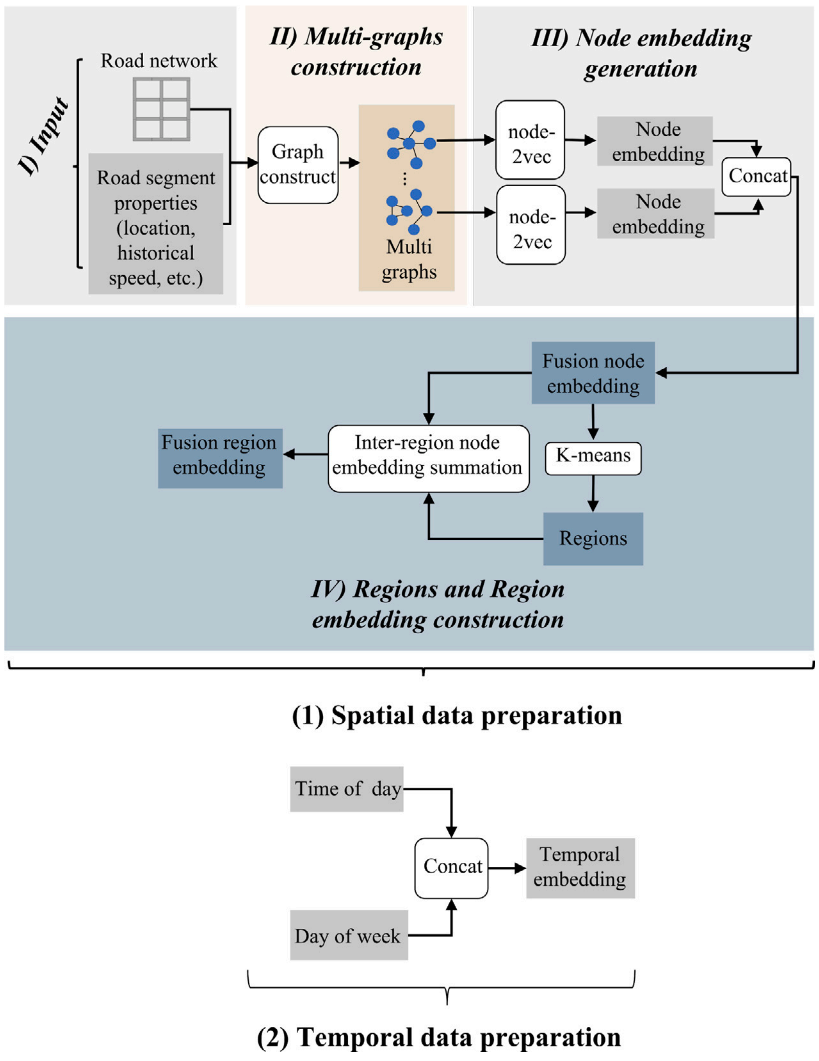

I) Input
Road network
Road segment properties (location, historical speed, etc.)
II) Multi-graphs construction
Graph construct
Multi graphs
III) Node embedding generation
node-2vec
Node embedding
node-2vec
Node embedding
Concat
Fusion node embedding
K-means
Regions
Inter-region node embedding summation
Fusion region embedding
IV) Regions and Region embedding construction
(1) Spatial data preparation
Time of day
Day of week
Concat
Temporal embedding
(2) Temporal data preparation

**Fig. 2.** Data preparation process.

where $\boldsymbol{h}_{t_j}^{v_i}$ is the hidden representation to be processed for time step $t_j$ and road segment $v_i$; $\boldsymbol{em}_{tem,t_j}$ is the temporal embedding vector for $t_j$ generated as per Fig. 2 after being processed by the FC layer; $d$ is a scaling factor; $W$ is the number of attention heads; $f_{tem,1}^{w}$, $f_{tem,2}^{w}$ and $f_{tem,3}^{w}$ are the three non-linear transformation functions (*e.g.* FC) used for each head $w$; $N_{tem}(t_j)$ is the set of time steps before $t_j$; $\boldsymbol{h}_{t_j}^{\prime v_i}$ is the new hidden representation; $\|$ represents the concatenation operation; and $\langle\cdot\rangle$ is the inner product operation. New representations generated in such a manner are based on the relationships between the current time step and all of the previous time steps, which assists the model to capture global temporal correlations.

2) TCN. It is natural to presume that close time steps may be more strongly correlated than distant time steps in traffic patterns. However, during representation generation, temporal attention—despite its ability to consider correlations with all time steps—may place insufficient attention on time steps that are closer and thus possibly more correlated than those that are distant. Accordingly, we add the dilated causal TCN (Yu and Koltun, 2015) after the temporal attention network to ensure that we capture local temporal correlation. Mathematically, the temporal convolution operation can be defined as follows.

$$\boldsymbol{h}_{c_m}^{v_i} \star \boldsymbol{f}_{TC} = \sum_{p=0}^{K_{TC}-1} \boldsymbol{f}_{TC}[p]\boldsymbol{h}_{c_m}^{v_i}[t_j - d_{TC} \times p] \tag{4}$$

where $\boldsymbol{h}_{c_m}^{v_i} \in \mathbb{R}^{T}$ is the representation vector for feature channel $c_m$ and road segment $v_i$, $\boldsymbol{f}_{TC} \in \mathbb{R}^{K_{TC}}$ is the temporal convolution filter with size $K_{TC}$, $d_{TC}$ is the dilation factor, and $t_j$ is a time step for temporal convolution. An operation similar to that defined by Eq. (4) is implemented for all of the time steps, road segments and feature channels to achieve information aggregation within a relatively short time range.

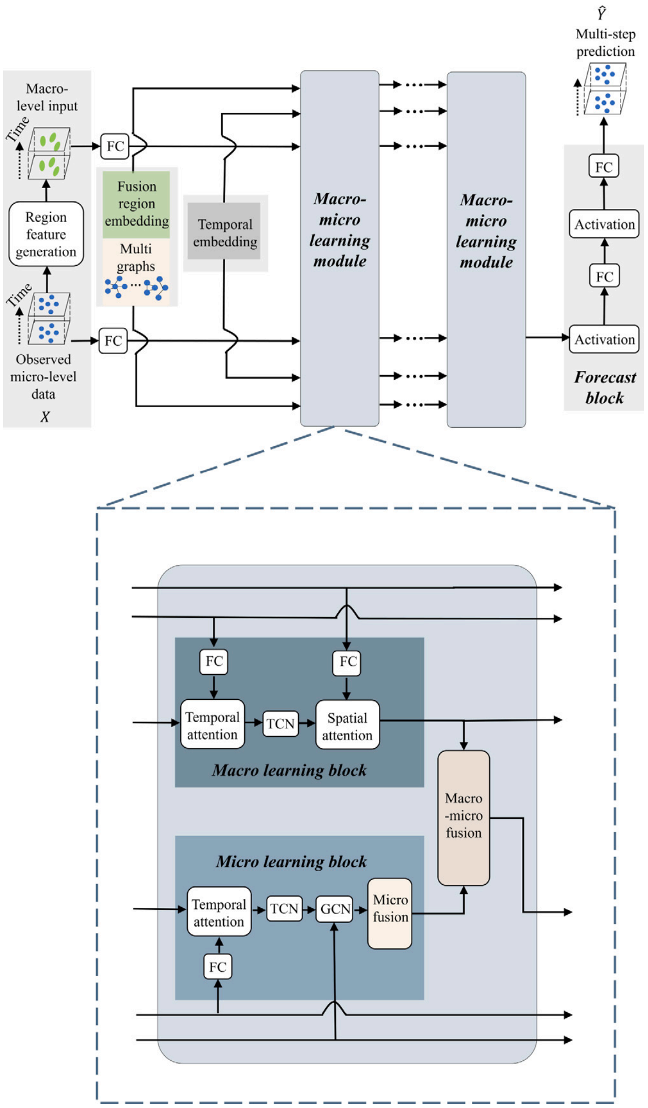


**Fig. 3.** Model framework.

(3) Multi-GCN Graph convolution aggregates representations of a neighborhood of a certain node to capture their spatial correlations and generate a new representation for the node. A classic GCN with wide application was developed by Kipf and Welling (2016) and has the following form:

$$\boldsymbol{H}'_{t_j} = \tilde{\boldsymbol{A}}\boldsymbol{H}_{t_j}\boldsymbol{W}_{GC} \tag{5}$$

where $\boldsymbol{H}'_{t_j}$ and $\boldsymbol{H}_{t_j} \in \mathbb{R}^{N\times C}$, $N$ is the number of road segments, $C$ is the number of channels, $\tilde{\boldsymbol{A}} \in \mathbb{R}^{N\times N}$ is the normalized adjacency matrix with self-loops, $\boldsymbol{W}_{GC} \in \mathbb{R}^{C\times C'}$ is the linear transformation matrix and $C'$ is the channel of the new representation. However, the graphs considered in the current study can be directional, that is, the adjacency matrices $\boldsymbol{A}_p$ and $\boldsymbol{A}_h$ in sub-section A can be asymmetrical. For this reason, we use another type of GCN, based on the approach of Li et al. (2017b). This GCN models the spatial aggregation of representations as the diffusion of graph signals in a given graph and has the following form:

$$\boldsymbol{H}'_{t_j} = \sum_{k=0}^{K_{GC}-1} \boldsymbol{P}_f^k\boldsymbol{H}_{t_j}\boldsymbol{W}_{GC,1} + \boldsymbol{P}_b^k\boldsymbol{H}_{t_j}\boldsymbol{W}_{GC,2} \tag{6}$$

where $K_{GC}$ represents the steps of one-round diffusion of the representation in the graph, $\boldsymbol{P}_f = \boldsymbol{A}/rowsum(\boldsymbol{A})$ is the transition matrix of the forward direction of diffusion, $\boldsymbol{P}_b = \boldsymbol{A}^T/rowsum(\boldsymbol{A}^T)$ is the transition matrix of the backward direction and $P^k$ is the power series of the transition matrix. This operation is implemented for the two adjacency matrices, respectively, with different sets of parameters used to capture multiple micro correlations.

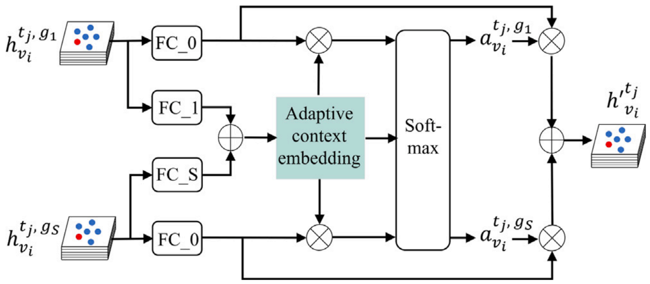


**Fig. 4.** Micro fusion layer.

(4) Micro fusion layer. Efficient integration of representations learned from different micro graphs is important for achieving good model performance. The most straightforward approach is to sum the representations. However, this may overlook their potential differences that may affect predictions. For example, when a neighborhood of a road segment becomes congested, there is a high probability that the road segment itself may soon become congested, too, even if the other neighborhoods of the road segment defined on the other graphs do not show congestion temporarily. In this situation, more attention should be paid to the representation in which congestion is observed than to the other representations. To consider such differences adaptively, we adopt an attention mechanism to fuse representations generated from various graphs, as described in the previous sub-section. As the goal is to generate a single output representation and there is not a direct context embedding for this representation, a natural idea is to use the potential output as context embedding for the attention calculation. The mathematics of this process is shown below:

$$\boldsymbol{h}'^{t_j}_{v_i} = \sum_{s=1}^{S} \alpha^{t_j,g_s}_{v_i} \boldsymbol{W}_{in} \boldsymbol{h}^{t_j,g_s}_{v_i} \tag{7}$$

$$\alpha^{t_j,g_s}_{v_i} = \frac{exp((\boldsymbol{W}_{out}\boldsymbol{h}'^{t_j}_{v_i})^T(\boldsymbol{W}_{in}\boldsymbol{h}^{t_j,g_s}_{v_i}))}{\sum_{s_a=1}^{S} exp((\boldsymbol{W}_{out}\boldsymbol{h}'^{t_j}_{v_i})^T(\boldsymbol{W}_{in}\boldsymbol{h}^{t_j,g_{s_a}}_{v_i}))}, s = 1, \dots, S \tag{8}$$

where $\boldsymbol{h}'^{t_j}_{v_i}$ is the output representation of the fusion layer for road segment $v_i$ and time step $t_j$; $\boldsymbol{h}^{t_j,g_s}_{v_i}$ is the input representation generated by the GCN applied to graph $g_s$; $S$ is the number of graphs; and $\boldsymbol{W}_{in}$ and $\boldsymbol{W}_{out}$ are transformation matrices for input and output representations, respectively. The solutions of Eqs. 7 and 8 can provide the output representation that we require. However, the solution-finding process of such a non-linear system can be complex during nn training. Accordingly, we adopt some approximations and simplifications of the equations. Observe that $\boldsymbol{h}'^{t_j}_{v_i} = \boldsymbol{F}_{\boldsymbol{w}}(\boldsymbol{h}^{t_j,g_1}_{v_i}, \dots, \boldsymbol{h}^{t_j,g_S}_{v_i})$ in the system in Eqs. 7 and 8, where $\boldsymbol{F}_{\boldsymbol{w}}$ is some non-linear function. We thus approximate $\boldsymbol{W}_{out}\boldsymbol{h}'^{t_j}_{v_i} = \boldsymbol{W}_{out}\boldsymbol{F}_{\boldsymbol{w}}(\boldsymbol{h}^{t_j,g_1}_{v_i}, \dots, \boldsymbol{h}^{t_j,g_S}_{v_i})$ as the following form:

$$\boldsymbol{u}'^{t_j}_{v_i} = \sum_{s=1}^{S}(\boldsymbol{W}_{u,g_s}\boldsymbol{h}^{t_j,g_s}_{v_i}) + \boldsymbol{b}_u \tag{9}$$

Eq. (8) can be formulated as follows:

$$\alpha^{t_j,g_s}_{v_i} = \frac{exp((\boldsymbol{u}'^{t_j}_{v_i})^T(\boldsymbol{W}_{in}\boldsymbol{h}^{t_j,g_s}_{v_i}))}{\sum_{s_a=1}^{S} exp((\boldsymbol{u}'^{t_j}_{v_i})^T(\boldsymbol{W}_{in}\boldsymbol{h}^{t_j,g_{s_a}}_{v_i}))}, s = 1, \dots, S \tag{10}$$

The fusion representation can then be obtained by calculating Eqs. 9, 10, 7 sequentially, without the need to solve any non-linear equation systems. A graphical illustration of the process is provided in Fig. 4, where $FC_0$ represents $\boldsymbol{W}_{in}$, and $FC_1$ to $FC_S$ represents $\boldsymbol{W}_{u,g_s}, s = 1, \dots, S$.

### 4.4. Macro learning block

As shown in Fig. 3, a macro learning block is composed of a temporal attention network, a TCN and a spatial attention network. The mechanism of the temporal learning part is exactly the same as that in the micro learning block, except for the input size. Therefore, for simplicity, we detail only the spatial attention network in this sub-section.

The principle of the spatial attention network is similar to that of the temporal attention network. The attention mechanism is applied to describe the spatial correlations between regions. As the spatial correlation is affected by both the traffic patterns and the network structure inside and around regions, we concatenate hidden representations and regional embeddings to generate attention scores. As mentioned in Section 4.1, the regional embedding is obtained via a combination of different micro node embeddings based on multiple micro correlations and thus can capture the diversity of correlations for macro learning. Details on the multi-head attention network are provided below.

$$u^{w}_{r_i,r} = \frac{\langle f^{w}_{sp,1}(\boldsymbol{h}^{r_i}_{t_j} \parallel \boldsymbol{em}_{sp,r_i}), f^{w}_{sp,2}(\boldsymbol{h}^{r}_{t_j} \parallel \boldsymbol{em}_{sp,r})\rangle}{\sqrt{d}} \tag{11}$$

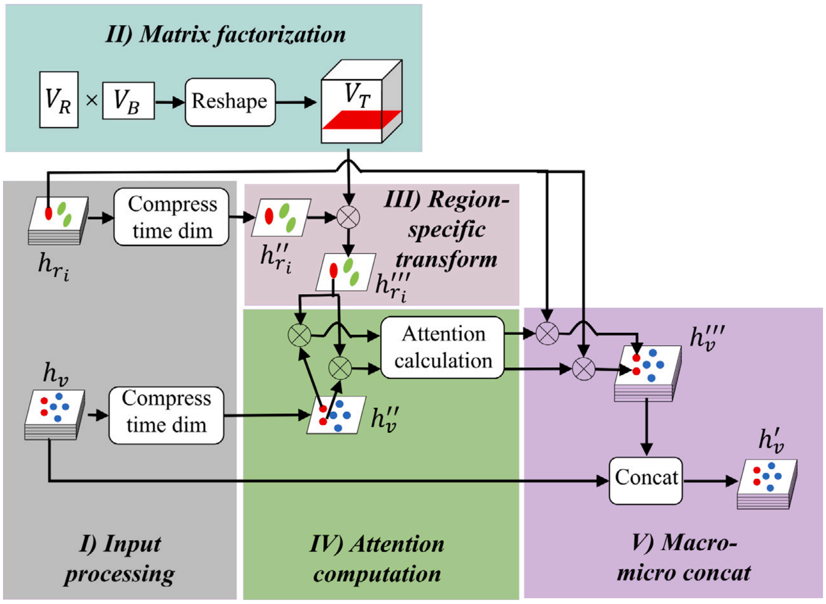


**Fig. 5.** Macro–micro fusion process.

$$\alpha^{w}_{r_i,r} = \frac{exp(u^{w}_{r_i,r})}{\sum_{r_a \in Re} exp(u^{w}_{r_i,r_a})} \tag{12}$$

$$\boldsymbol{h}'^{r_i}_{t_j} = \|_{w=1}^{W} \{ \sum_{r \in Re} \alpha^{w}_{r_i,r} f^{w}_{sp,3}(\boldsymbol{h}^{r}_{t_j}) \} \tag{13}$$

where $\boldsymbol{h}^{r_i}_{t_j}$ and $\boldsymbol{h}'^{r_i}_{t_j}$ are original and new hidden representations for region $r_i$, respectively; $\boldsymbol{em}_{sp,r_i}$ is the spatial embedding vector for region $r_i$ generated as per Fig. 2 after processing by the FC layer; $f^{w}_{sp,1}$, $f^{w}_{sp,2}$ and $f^{w}_{sp,3}$ are three non-linear transformation functions for each head $w$; $\boldsymbol{Re}$ is the set of regions; and the other symbols are the same as in temporal attention. As the attention scores are automatically and adaptively generated, there is no need for an explicit definition of the adjacency matrix here, which satisfies the requirement of macro spatial learning.

### 4.5. Macro–micro fusion

Macro representation, after being learned in the macro blocks, must be integrated with micro representations to make predictions. Therefore, an efficient fusion module is required. A natural approach is to concatenate representations of a given region and the road segments that belong to this region. However, even road segments in the same region may have different correlations with representations of the region. Therefore, a fusion process is devised to sufficiently and adaptively consider the region-wise and road-segment-wise differences in a macro–micro interaction, as shown in Fig. 5.

Specifically, we use input representations $\boldsymbol{h}_{r_i} \in \mathbb{R}^{T' \times C_R}$ for region $r_i$ and $\boldsymbol{h}_v \in \mathbb{R}^{T' \times C_v}$ for all of the road segments that belong to $r_i$, where $T'$ represents the time dimension for the current input; and $C_R$ and $C_v$ represent the feature dimensions for macro and micro representations, respectively. In the first step, we compress the time dimension of $\boldsymbol{h}_{r_i}$ and $\boldsymbol{h}_v$ based on the FC layers to generate $\boldsymbol{h}''_{r_i} \in \mathbb{R}^{C_R}$ and $\boldsymbol{h}''_v \in \mathbb{R}^{C_v}$. The compressed region representations are then transformed into representations with the same feature dimensions as the micro representations. As the interaction between a certain region and its road segments may have some uniqueness, a matrix factorization process is utilized to perform a region-specific transformation, given by

$$\boldsymbol{V}_T = reshape(\boldsymbol{V}_R \boldsymbol{V}_B) \tag{14}$$

$$\boldsymbol{h}'''_{r_i} = \boldsymbol{V}_{T,i} \boldsymbol{h}''_{r_i} \tag{15}$$

where $\boldsymbol{V}_T \in \mathbb{R}^{N_R \times C_v \times C_R}$ is the region-specific transformation tensor, $N_R = |Re|$ is the number of regions, $\boldsymbol{V}_R \in \mathbb{R}^{N_R \times k_B}$ and $\boldsymbol{V}_B \in \mathbb{R}^{k_B \times (C_R \cdot C_v)}$ are two matrices to be learned, $reshape(\cdot)$ is a function that transforms a matrix into a tensor, $\boldsymbol{V}_{T,i} \in \mathbb{R}^{C_v \times C_R}$ is a slice of $\boldsymbol{V}_T$, and $\boldsymbol{h}'''_{r_i}$ is the transformed macro representation with similar dimensions to the micro representation. The adoption of tensor $\boldsymbol{V}_T$ enables every region to possess a unique transformation. The matrix factorization decomposes the large tensor $\boldsymbol{V}_T$ into two smaller matrices to learn, which reduces the number of parameters. Moreover, some similarity within regions can still be captured by the share of base $\boldsymbol{V}_B$ in region-wise transformation matrix $\boldsymbol{V}_{T,i}$. Let $N_m(r_i)$ denote the road segments in region $r_i$. Then, the interaction between $r_i$ and a road segment $v \in N_m(r_i)$ can be considered via the attention mechanism, as follows:

$$\beta_{r_i,v_a} = exp(\langle \boldsymbol{h}'''_{r_i}, \boldsymbol{h}''_{v_a} \rangle), \forall v_a \in N_m(r_i) \tag{16}$$

$$Avg(r_i) = Mean_{v_a \in N_m(r_i)}(\beta_{r_i,v_a}) \tag{17}$$

$$\alpha_{r_i,v} = exp(\beta_{r_i,v} - Avg(r_i)) \tag{18}$$

$$\boldsymbol{h}'''_v = \alpha_{r_i,v} \times \boldsymbol{h}_{r_i} \tag{19}$$

$$\boldsymbol{h}'_v = Concat(\boldsymbol{h}'''_v, \boldsymbol{h}_v) \tag{20}$$

where $Concat$ is a concatenation operation performed on the feature dimension, and $\boldsymbol{h}'_v$ is the output of the entire fusion module. Repeating the above process for all of the regions effectively integrates the macro and micro representations.

### 4.6. Forecasting block and loss function

As shown in Fig. 3, the forecasting block is composed of several FC layers and an activation function, which is a regular setting for a prediction task. In the first FC layer, a 1D convolution is added to compress the time dimension into one dimension. In the second FC layer, the feature dimension is transformed into the dimension of the number of time steps for prediction, thereby generating the multi-step prediction results. We use the mean absolute error (MAE) as the loss function, which is calculated as follows:

$$\mathcal{L} = \frac{\sum_{t=T1+1}^{T1+T2} \sum_{v=1}^{N} |x_t^v - \hat{x}_t^v|}{T2 \times N} \tag{21}$$

## 5. Experimental results

### 5.1. Data and models

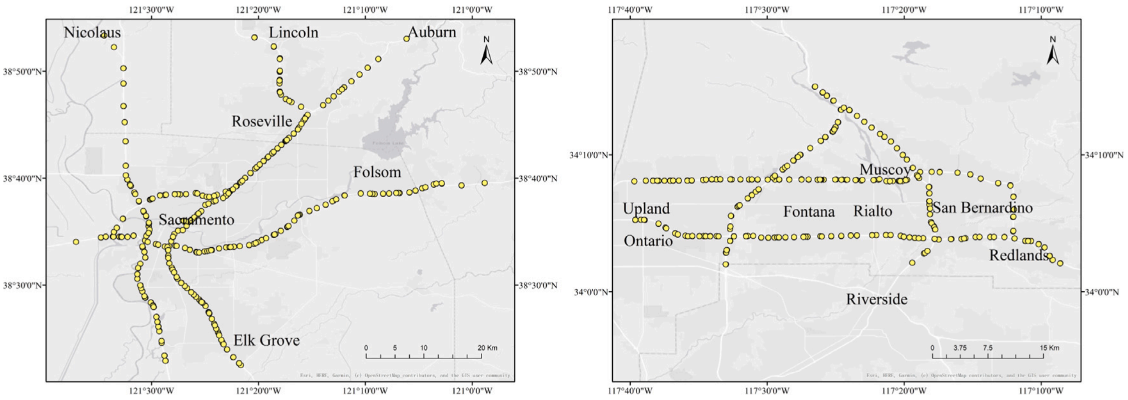
The experiments are implemented on two speed datasets, both of which are formulated based on the data from the California Department of Transportation (Caltrans) Performance Measurement System (PeMS)[2]. PeMS is a consolidated database that provides access to real-time and historical traffic data collected in California, USA. The first dataset is called PEMS03; it contains data from 617 sensors in California's 3rd congressional district for 5 months, i.e., from Feb 1, 2019, to Jun 30, 2019. The data are aggregated in 5-minute windows, and there are a total of 43,200 speed records for each road segment. The sensors are distributed along the main road of California's 3rd congressional district at longitudes 120°58′W to 121°36′W and latitudes 38°22′N to 38°53′N. The second dataset is called PEMS08; it contains data from 474 sensors in California's 8th congressional district for 6 months, from Jan 1, 2019, to Jun 30, 2019. These data are also aggregated in 5-minute windows and there are a total of 52,128 speed records for each road segment. The dataset covers longitudes 117°09′W to 117°39′W and latitudes 34°02′N to 34°15′N. The sensor distributions of both datasets are visualized in Fig. 6, and each node represents a road segment.

The data are subjected to Z-score normalization in preprocessing. Both datasets are split: 70% are used for training, 10% are used for validation and 20% are used for testing. We construct the physical proximity graph by using an open-source router application programming interface known as Open Source Routing Machine, which is supported by OpenStreetMap, to retrieve the route distance between each road pair. We construct the correlation graph by using all of the available historical speed data in the training set to calculate the correlation between each pair of road segments. Specifically, $\boldsymbol{S}_i$ is now a vector containing all of the speed data points in the training set for road segment $i$.

Our macro–micro spatio-temporal nn model is denoted 'MMSTNet' in the following experiments. We demonstrate its effectiveness by comparing it with the following state-of-the-art machine learning approaches:

- **HA** (Smith, 1995): This model is based on the historical average of speed data on several previous time steps.
- **MLP**: Multi-layer perceptron, which comprises a stack of multiple FC layers and is a classic artificial nn.
- **LSTM** (Cui et al., 2018): An RNN with fully connected LSTM hidden units.
- **GRU** (Agarap, 2018): A graph GRU network that is based on the same principle as an LSTM model but has a simpler gated unit.
- **STGCN** (Yu et al., 2017): Spatio-temporal graph convolution network, which integrates 1D convolution and a graph convolution.
- **GCRN** (Seo et al., 2018): Spatiotemporal graph convolution network, which integrates recurrent units and a graph convolution network.
- **HGCN** (Guo et al., 2021): Spatio-temporal nn that captures spatial correlations with pure graph convolutions.

[2] The data are available at https://pems.dot.ca.gov/



(a) Sensor distribution on PEMS03 (b) Sensor distribution on PEMS08

**Fig. 6.** Sensor distribution.

In addition to the above-listed baseline methods, we also test the following eight variants of MMSTNet as ablation studies for major modules:

- **MMSTNet-V1**: This variant has no macro learning blocks or integration layers, and thus considers only micro spatial correlations.
- **MMSTNet-V2 and -V3**: This variant considers one type of micro correlation (physical proximity/speed similarity) and its corresponding macro correlation.
- **MMSTNet-V4**: This variant has no modules that are responsible for representation integration (i.e., no micro fusion module or macro–micro fusion module).
- **MMSTNet-V5**: This variant uses a GCN for both macro and micro spatial correlation learning.

Regarding hyperparameters, the learning rate is selected from 0.001 and 0.01. The size of micro and macro blocks is selected from 32 and 64, and the numbers of micro and macro blocks are both 2. The number of layers and their sizes for FC layers in the forecasting blocks are selected from [128, 256] and [256, 512]. The dimension for matrix factorization is selected from 4 and 8. The order for the GCN is 3, as suggested in Li et al. (2017b). The kernel sizes of the temporal and spatial attention networks are both chosen from 4 and 8. The batch size for training is 64 and the maximum number of epochs is 50. PyTorch is utilized to implement the experiments with Adam as the optimizer. A server with 32 GB of RAM and two NVIDIA GeForce RTX 3090 24 GB Blower (CUDA 10,496) is employed for implementation of the experiments.

### 5.2. Comparison with baselines

The experimental results are respectively summarized in Tables 1 to 4, where Tables 1 and 2 provides the prediction accuracy comparison for different approaches, and Tables 3 and 4 show the comparison between MMSTNet and its variants. In brief, MMSTNet can outperform the baselines on most of the metrics for different time-range predictions. The advantage of MMSTNet is obvious for long-range prediction, especially. Moreover, the ablation studies also show the effectiveness of different modules in MMSTNet. For more straightforward visualization of the results, readers can also refer to sub-Section 5.4. In this sub-section, we focus on the analysis of the comparison with baseline methods.

We compare the methods in terms of their predictions for the next 15, 30 and 60 min, as these three time steps correspond to short-range, mid-range and long-range predictions, respectively. The metrics used are the MAE, the root-mean-square error (RMSE) and the mean absolute percentage error (MAPE). On PEMS03, our method outperforms the best of the baselines in terms of the MAE and RMSE by 3.0% and 3.3%, respectively, for 15 min; by 4.4% and 5.0%, respectively, for 30 min; and by 6.0% and 5.9%, respectively, for 60 min; and reduces the MAPE from 2.66% to 2.53%, from 3.43% to 3.22%, and from 3.28% to 3.09%, respectively, for the three time steps. On PEMS08, our method outperforms the best of the baselines in terms of the MAE and RMSE by 3.6% and 3.6%, respectively, for 15 min; by 4.2% and 2.8%, respectively, for 30 min; and by 6.0% and 4.4%, respectively, for 60 min; and reduces the MAPE from 2.68% to 2.60%, from 3.48% to 3.36%, and from 4.24% to 4.01%, respectively, for the three time steps.

We also analyze the superiority of our method more deeply. Compared with the simplest models, namely the HA and MLP models, MMSTNet performs much better on all of the time steps on the two datasets. This is expected, as the HA and MLP models do not consider any spatial correlations and only examine the simplest temporal relationships. In comparison, the LSTM and GRU models more extensively depict the sequential relationships between different time steps, and thus are far superior to the HA and MLP models. However, the LSTM and GRU models overlook spatial correlations as they lack a spatial learning module; thus, their performances are limited. In contrast the STGCN and GCRN models consider multiple micro spatial correlations via a GCN,

**Table 1**
Comparison of performances of our model and baselines on PEMS03.

| Model | MAE | RMSE | MAPE |
|---|---|---|---|
| 15 min | | | |
| HA | 2.04 | 4.38 | 4.52% |
| MLP | 1.47 | 3.08 | 2.82% |
| LSTM | 1.42 | 2.99 | 2.78% |
| GRU | 1.42 | 3.00 | 2.78% |
| STGCN | 1.39 | 2.85 | 2.74% |
| GCRN | 1.39 | 2.87 | 2.82% |
| HGCN | 1.33 | 2.75 | 2.66% |
| **MMSTNet (ours)** | **1.29** | **2.66** | **2.53%** |
| 30 min | | | |
| HA | 2.04 | 4.38 | 4.52% |
| MLP | 1.85 | 4.12 | 3.83% |
| LSTM | 1.78 | 4.03 | 3.79% |
| GRU | 1.78 | 4.03 | 3.80% |
| STGCN | 1.70 | 3.72 | 3.63% |
| GCRN | 1.66 | 3.71 | 3.60% |
| HGCN | 1.59 | 3.57 | 3.43% |
| **MMSTNet (ours)** | **1.52** | **3.39** | **3.22%** |
| 60 min | | | |
| HA | 2.04 | 4.38 | 4.52% |
| MLP | 2.34 | 5.18 | 5.37% |
| LSTM | 2.25 | 5.14 | 5.20% |
| GRU | 2.25 | 5.13 | 5.21% |
| STGCN | 2.06 | 4.62 | 4.72% |
| GCRN | 1.95 | 4.42 | 4.42% |
| HGCN | 1.83 | 4.23 | 4.07% |
| **MMSTNet (ours)** | **1.72** | **3.98** | **3.81%** |

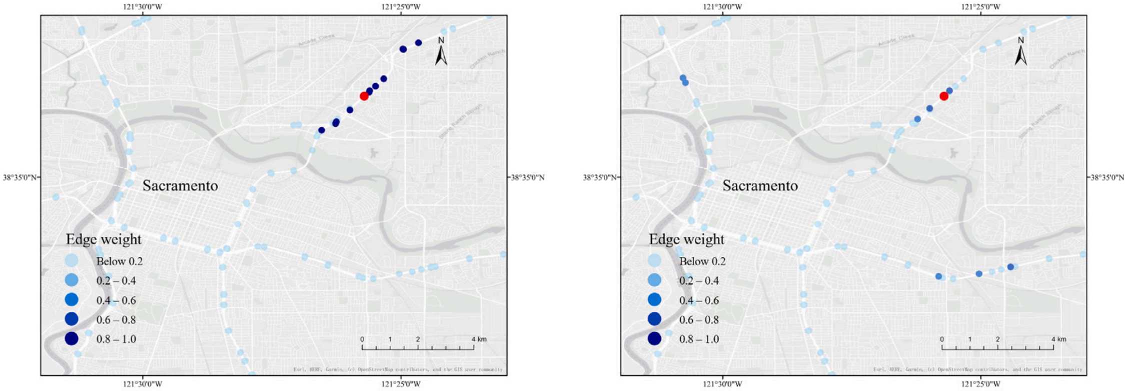
together with temporal correlations. The new representations in these models are formulated based on different neighborhoods, instead of based on a single road segment, which improves their overall prediction accuracy. Nevertheless, the STGCN and GCRN models ignore macro spatial correlations and thus cannot detect pattern changes emerging in distant regions; thus, they perform worse in longer-temporal-range prediction than in shorter-temporal-range prediction. In contrast, MMSTNet perfectly manages the problem by integrating macro learning modules. Moreover, it considers diverse macro correlations, together with corresponding fusion approaches, to effectively integrate micro and macro correlations, respectively. The performance of MMSTNet is obviously superior to that of the other models in mid-range and long-range prediction, as its learning is based on different macro correlations, which enables it to perceive traffic interactions on a larger scale than the other models. In addition, the combined use of the GCN and spatial attention by MMSTNet enables it to well adapt to the properties of macro and micro spatial learning, unlike the other baselines with complicated structures that use only the GCN. All of these designs improve the comprehensive performance of our method for predictions on different time steps. A more detailed ablation study is described in the following sub-section to demonstrate the effect of each module.

The superiority of our model is validated for real-time forecasting tasks via experiments. However, more understanding can be extracted from the model itself. Although deep learning-based models are often viewed as ‘black boxes’ and thus difficult to understand, we can nevertheless attempt to analyze some of our model’s parameters, middle representations or utilized data to obtain some deep insights for transportation operators. Most of the representations or modules are dynamically adjusted during prediction and thus would be rather challenging to analyze. Thus, as an example, we consider only relatively fixed adjacency matrices in the model. In Fig. 7, we provide two different neighborhoods for a given road segment, which follow adjacent matrices $A_p$ and $A_h$, respectively. Operators would easily recognize that the figure shows that there are different micro correlations defined on separate adjacencies. As a result, operators would know they should carefully consider multiple relationships between spatial objects when designing a prediction model or other theoretic models for such an example.

Furthermore, even prediction results themselves can provide some practical insights for operators. A key feature of the model is that it can make co-predictions for multiple future time steps. Thus, we use this feature in this example. By comparing the prediction accuracy of the model on different future time steps, we find that the model performs better on shorter-range time steps than on longer-range time steps. This is logical because compared with the latter, the former have closer temporal relationships with input data. As a result, operators who employ similar models should rely more on shorter-range predictions than on longer-range predictions, and the latter should be carefully tested and validated before they are utilized. This is also one of the reasons we consider diversified macro learning, i.e., to improve the performance on longer-range prediction, as analyzed previously.

**Table 2**
Comparison of performances of our model and baselines on PEMS08.

| 15 min | | | |
|---|---|---|---|
| Model | MAE | RMSE | MAPE |
| HA | 2.51 | 5.02 | 5.21% |
| MLP | 1.50 | 3.06 | 2.84% |
| LSTM | 1.46 | 2.99 | 2.80% |
| GRU | 1.46 | 2.98 | 2.82% |
| STGCN | 1.43 | 2.81 | 2.77% |
| GCRN | 1.43 | 2.82 | 2.82% |
| HGCN | 1.38 | 2.75 | 2.68% |
| **MMSTNet (ours)** | **1.33** | **2.65** | **2.60%** |
| 30 min | | | |
| Model | MAE | RMSE | MAPE |
| HA | 2.51 | 5.02 | 5.21% |
| MLP | 1.91 | 4.16 | 3.85% |
| LSTM | 1.85 | 4.06 | 3.80% |
| GRU | 1.86 | 4.05 | 3.81% |
| STGCN | 1.78 | 3.76 | 3.69% |
| GCRN | 1.76 | 3.75 | 3.64% |
| HGCN | 1.68 | 3.61 | 3.48% |
| **MMSTNet (ours)** | **1.61** | **3.51** | **3.36%** |
| 60 min | | | |
| Model | MAE | RMSE | MAPE |
| HA | 2.51 | 5.02 | 5.21% |
| MLP | 2.43 | 5.29 | 5.24% |
| LSTM | 2.34 | 5.16 | 5.12% |
| GRU | 2.34 | 5.15 | 5.13% |
| STGCN | 2.19 | 4.75 | 4.87% |
| GCRN | 2.15 | 4.65 | 4.58% |
| HGCN | 1.99 | 4.35 | 4.24% |
| **MMSTNet (ours)** | **1.87** | **4.16** | **4.01%** |



a) distance correlation

b) speed correlation

**Fig. 7.** Physical and semantic neighborhood for a road segment (marked by red color) on PEMS03. Edge weight represents the degree of correlation between the red segment and other road segments.

### *5.3. Comparison with variants*

As an ablation study for all of the major modules, we test the performances of variants V1 to V5, which are shown in Tables 3 and 4. By comparing the performance of V1 with that of our model, we verify the necessity of considering macro correlations. The comparison of the performances of V2 and V3 demonstrates the importance of integrating multiple types of correlations, instead of a single type of correlation. By comparing the performance of V4 with the proposed model, we verify the effects of the representation fusion modules. Finally, the significance of the combination of the GCN and spatial attention is shown by the comparison of the performances of V5 and our model.

As mentioned, the introduction of macro correlations into a model can allow the model to perceive pattern interactions between a road segment and more distant regions than the model without macro correlations. Therefore, the former model can perform

**Table 3**
Comparison of performances of variants on PEMS03.

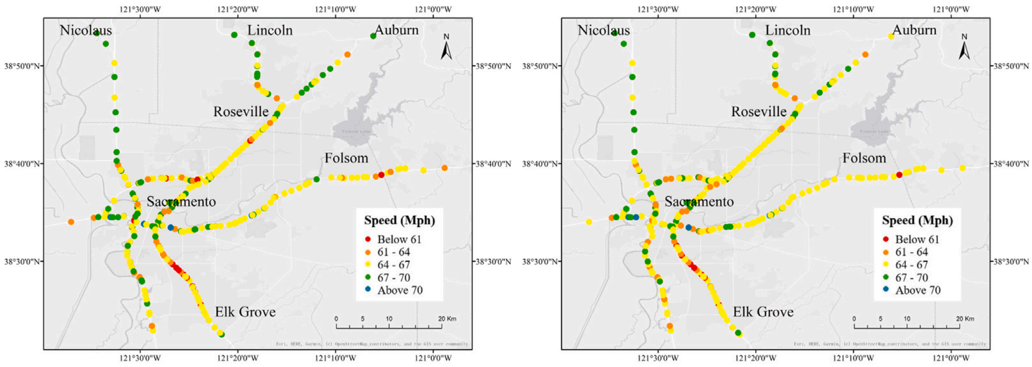
| 15 min | | | |
|---|---|---|---|
| Model | MAE | RMSE | MAPE |
| **MMSTNet (ours)** | **1.29** | **2.66** | **2.53%** |
| MMSTNet-V1 | 1.31 | 2.71 | 2.61% |
| MMSTNet-V2 | 1.30 | 2.69 | 2.56% |
| MMSTNet-V3 | 1.30 | 2.69 | 2.55% |
| MMSTNet-V4 | 1.30 | 2.67 | 2.56% |
| MMSTNet-V5 | 1.31 | 2.72 | 2.61% |
| 30 min | | | |
| Model | MAE | RMSE | MAPE |
| **MMSTNet (ours)** | **1.52** | **3.39** | **3.22%** |
| MMSTNet-V1 | 1.55 | 3.47 | 3.32% |
| MMSTNet-V2 | 1.54 | 3.48 | 3.30% |
| MMSTNet-V3 | 1.54 | 3.47 | 3.24% |
| MMSTNet-V4 | 1.54 | 3.42 | 3.27% |
| MMSTNet-V5 | 1.54 | 3.47 | 3.30% |
| 60 min | | | |
| Model | MAE | RMSE | MAPE |
| **MMSTNet (ours)** | **1.72** | **3.98** | **3.81%** |
| MMSTNet-V1 | 1.79 | 4.11 | 4.02% |
| MMSTNet-V2 | 1.77 | 4.10 | 3.94% |
| MMSTNet-V3 | 1.76 | 4.06 | 3.83% |
| MMSTNet-V4 | 1.77 | 4.04 | 3.94% |
| MMSTNet-V5 | 1.76 | 4.09 | 3.92% |

**Table 4**
Comparison of performances of variants on PEMS08.

| 15 min | | | |
|---|---|---|---|
| Model | MAE | RMSE | MAPE |
| **MMSTNet (ours)** | **1.33** | **2.65** | 2.60% |
| MMSTNet-V1 | 1.35 | 2.68 | 2.65% |
| MMSTNet-V2 | 1.34 | 2.68 | 2.60% |
| MMSTNet-V3 | 1.35 | 2.70 | 2.68% |
| MMSTNet-V4 | 1.34 | 2.67 | 2.62% |
| MMSTNet-V5 | **1.33** | **2.65** | **2.57%** |
| 30 min | | | |
| Model | MAE | RMSE | MAPE |
| **MMSTNet (ours)** | **1.61** | **3.51** | **3.36%** |
| MMSTNet-V1 | 1.63 | 3.53 | 3.42% |
| MMSTNet-V2 | 1.64 | 3.56 | 3.38% |
| MMSTNet-V3 | 1.64 | 3.52 | 3.44% |
| MMSTNet-V4 | 1.65 | 3.57 | 3.39% |
| MMSTNet-V5 | 1.62 | 3.53 | 3.38% |
| 60 min | | | |
| Model | MAE | RMSE | MAPE |
| **MMSTNet (ours)** | **1.87** | **4.16** | **4.01%** |
| MMSTNet-V1 | 1.92 | 4.25 | 4.17% |
| MMSTNet-V2 | 1.93 | 4.28 | 4.10% |
| MMSTNet-V3 | 1.90 | 4.21 | 4.12% |
| MMSTNet-V4 | 1.93 | 4.29 | 4.13% |
| MMSTNet-V5 | 1.90 | 4.27 | 4.16% |

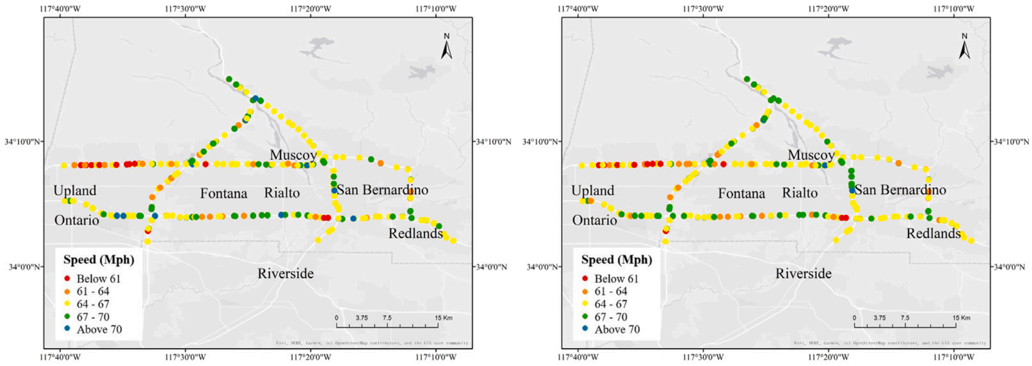
better than the latter model in making long-range predictions. The results verify this point; our model outperforms V1 in terms of the MAE and RMSE by 3.9% and 3.2%, respectively, for a 60-minute prediction on PEMS03; and by 2.6% and 2.1%, respectively, for 60-minute prediction on PEMS08. In addition, the MAPE is reduced from 4.02% to 3.81%, and from 4.17% to 4.01% on the datasets, respectively, after employing macro correlations.

The diversities of micro and macro correlations are also important for prediction accuracy. Comparing the testing results of V2 and V3 reveals that V2 outperforms V2 in some metrics, whereas the reverse is true for the other metrics. This indicates that each type of correlation has a unique advantage for predictions for some metrics and some time ranges. Thus, integrating these type of correlations within an efficient model framework combines their advantages, as shown in the results generated by MMSTNet.



(a) Ground truth. (b) 30-minute prediction

**Fig. 8.** Comparison between ground truth and 30-minute predictions in the morning peak hour of PEMS03.



(a) Ground truth. (b) 30-minute prediction

**Fig. 9.** Comparison between ground truth and 30-minute predictions in the evening peak hour for PEMS08.

For V4, the MAE is increased by 3.0% and 3.2% for a 60-minute prediction on PEMS03 and PEMS08, respectively, and the MAPE changes from 3.81% to 3.94%, and from 4.01% to 4.13% for the two datasets, compared with our model. These results verify the importance of the specially designed integration modules for adaptively combining representations that are learned hierarchically. Compared with V5 (which uses a pure GCN as a spatial learning tool), our model performs better in terms of nearly all of the metrics for the predictions on all three time steps, especially for 60 min. This is because it is difficult to obtain a clear definition of the adjacency matrix for macro correlation, and an inaccurate adjacency matrix may negatively influence the performance of GCN. The adoption of spatial attention in the macro layer effectively eliminates this uncertainty, by allowing the model to find region-wise relationships on its own in a data-driven manner.

The advantage of our model and the effectiveness of each of its modules are thus experimentally validated. Before we describe visualizations, we provide some further discussions on the limitations of our model for the benefit of potential users. First, the limitation faced by all of the current predictive models is their assumption that the traffic patterns of a certain area remain nearly unchanged for a given period, that is, a limited historical training dataset is sufficient to represent most of the traffic patterns. However, this may not be the case in reality (such as during sudden road closures or other emergencies, which may trigger traffic patterns that a model has never seen, thereby affecting its predictions). There are two solutions for the above problem that can be implemented in practical applications of the models. (1) A large-sized model can be used for training, fed with sufficient data, and then allowed to learn sufficient traffic patterns to adapt to various possible situations. A classic example is Chat Generative Pre-trained Transformer (developed by OpenAI), which can perform many tasks simultaneously. Although researchers may not be able to directly train such a large model, in practical applications, artificial intelligence companies or platforms can follow the framework devised by researchers to sufficiently expand the scale of a model to solve the given problem. (2) A given database can be kept updated in real time and repeated training can be performed on the model. This is necessary because no matter how large a model is and how many historical data there are, there are always limits and the possibility that novel patterns will appear. Therefore, only by constantly updating a database and using it to repeatedly adjust a model can the model adapt to various changes in transportation networks.

Second, another limitation of a deep learning-based model is that data are assumed to be always available and abundant. However, this may sometimes not be true, due to laws, privacy protections, limited sensing capabilities or other reasons. Therefore,

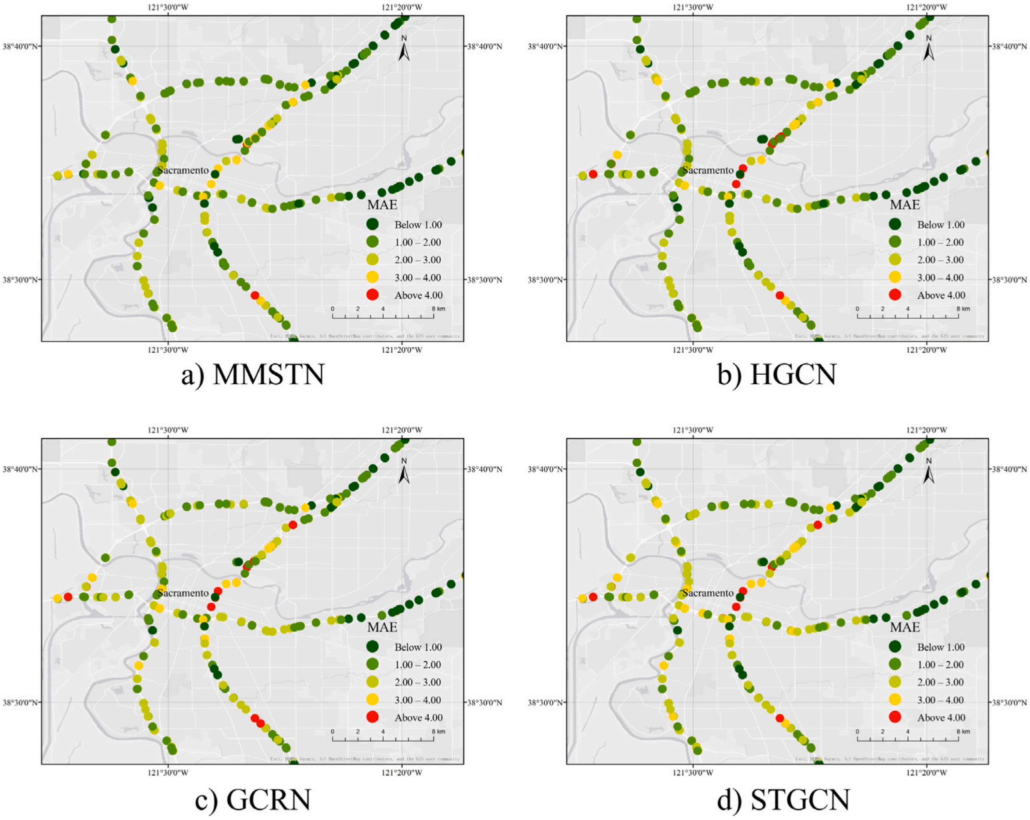


**Fig. 10.** Comparison of MAEs for 30-minute predictions generated by MMSTNet and by baselines for PEMS03.

users may need to adjust a model's framework or training method. For example, the method of transfer learning could be used. That is, first, data from some similar other areas could be used to train some pre-models. Second, some modules of these pre-models could be further trained with data from the studied area and used in final predictions. This is tantamount to using other datasets to compensate for a lack of local data.

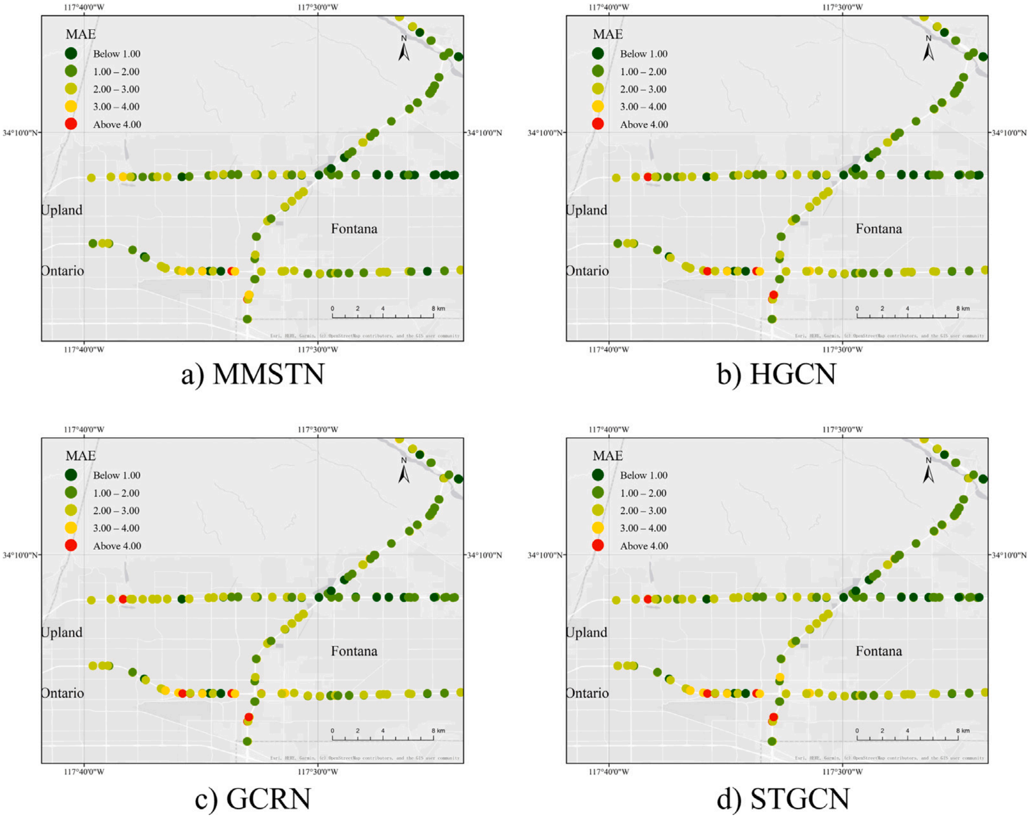
### *5.4. Visualizations*

In this sub-section, we provide two different types of visualization to directly demonstrate the capability of our model. First, we select snapshots of morning peak-hour traffic and evening peak-hour traffic from PEMS03 and PEMS08. Fig. 8 shows a visualization of ground truth and the predicted traffic speed for the next 30 min in the morning peak hour of PEMS03, whereas Fig. 9 shows the same visualization in the evening peak hour of PEMS08. As in Fig. 6, each node represents a road segment. The color of each node represents the traffic situation, and colors more similar to green indicate faster traffic speeds and better traffic conditions than colors less similar to green. The figures show that the true values and predictions are very close.

In addition, we visualize the MAE of MMSTNET and some of the most important baselines for 30-minute predictions on road networks of PEMS03 and PEMS08, respectively, as shown in Figs. 10 and 11. For clarity, we show the core areas for the two datasets, where a deep red color represents a large MAE and a deep green color represents a small MAE. The two figures reveal that from a global view, the MAE of our method is smaller than that of the other baselines. Moreover, there are fewer red dots for MMSTNet than for the other baselines, showing that MMSTNet is more capable of dealing with road segments for which it is difficult to make predictions.

## 6. Conclusion

This paper studies the traffic prediction problem based on different temporal and spatial correlations. To fully capture the diversity of micro spatial correlations and corresponding macro spatial correlations, separate learning modules are built, with a multi-graph GCN used for micro learning and a spatial attention network used for macro learning. A combination of a TCN and a temporal attention network is employed for both local and global temporal correlation. The hierarchically learned representations are adaptively integrated via micro fusion layers and macro–micro fusion modules. The resulting end-to-end model is called MMSTNet.



a) MMSTN

b) HGCN

c) GCRN

d) STGCN

**Fig. 11.** Comparison of MAEs for 30-minute predictions generated by MMSTNet and by baselines for PEMS08.

Evaluations on two real-world traffic speed datasets reveal that MMSTNet outperforms baseline approaches significantly. The necessity of each module in MMSTNet is validated by an ablation study.

In the future, MMSTNet could be further explored from the following perspectives. 1) Other types of GCNs and attention network tools could be incorporated into MMSTNet to further improve its performance. 2) Currently, MMSTNet employs a sophisticated method to integrate representations generated by spatial learning, such as an attention-based micro learning fusion approach. In future, similar methods could also be applied in temporal learning, such as the integration of representations generated via locally and globally temporal learning. 3) MMSTNet is designed to predict only one type of forecasting object. However, in practice, simultaneous predictions for multiple traffic-related objects are commonly made (e.g. traffic flows of different transportation modes). Thus, the corresponding extension should be applied to MMSTNet to allow its application to multi-task learning scenarios. 4) MMSTNet requires a sufficient dataset for training but sometimes data can be difficult to obtain. In such a scenario, transfer learning technology could be employed to modify MMSTNet to enable it to utilize other datasets as a data supplement. 5) The setting of macro and micro learning blocks could also be used in other prediction frameworks, such as encoder–decoder models or transformer structures. 6) The principle of MMSTNet could be applied in other scenarios, such as taxi demand prediction, transit origin–destination flow prediction or logistic forecasting, or other types of prediction tasks beyond the transportation industry.

## CRediT authorship contribution statement

**Siyuan Feng:** Methodology, Software, Data curation, Investigation, Writing – original draft. **Shuqing Wei:** Methodology, Data curation, Writing – original draft. **Junbo Zhang:** Conceptualization, Methodology, Investigation, Funding acquisition, Writing – review & editing. **Yexin Li:** Conceptualization, Investigation, Writing – review & editing. **Jintao Ke:** Conceptualization, Investigation, Writing – review & editing. **Gaode Chen:** Conceptualization, Investigation, Writing – review & editing. **Yu Zheng:** Conceptualization, Supervision, Funding acquisition, Writing – review & editing. **Hai Yang:** Conceptualization, Supervision, Funding acquisition, Writing – review & editing.

## Acknowledgments

The work described in this paper was supported by the National Key R&D Program of China (2019YFB2103201), the Beijing Nova Program (Z201100006820053), the National Natural Science Foundation of China (62172034, 72242106), the Beijing Natural

Science Foundation (4212021), a grant from the Research Grants Council (RGC) Theme-based Research Scheme T41-603/20R and a research grant (project N_HKUST627/18) from the Hong Kong Research Grants Council under the National Natural Science Foundation of China/RGC Joint Research Scheme.

## References

Agarap, A.F.M., 2018. A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data. In: Proceedings of the 2018 10th International Conference on Machine Learning and Computing. pp. 26–30.

Bello, I., Zoph, B., Vaswani, A., Shlens, J., Le, Q.V., 2019. Attention augmented convolutional networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3286–3295.

Chen, E., Ye, Z., Wang, C., Xu, M., 2019. Subway passenger flow prediction for special events using smart card data. IEEE Trans. Intell. Transp. Syst. 21 (3), 1109–1120.

Cui, Z., Ke, R., Pu, Z., Wang, Y., 2018. Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. arXiv preprint arXiv:1801.02143.

Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. Adv. Neural Inf. Process. Syst. 29.

Feng, S., Ke, J., Yang, H., Ye, J., 2021. A multi-task matrix factorized graph neural network for co-prediction of zone-based and OD-based ride-hailing demand. IEEE Trans. Intell. Transp. Syst.

Fout, A., Byrd, J., Shariat, B., Ben-Hur, A., 2017. Protein interface prediction using graph convolutional networks. Adv. Neural Inf. Process. Syst. 30.

Fu, R., Zhang, Z., Li, L., 2016. Using LSTM and GRU neural network methods for traffic flow prediction. In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation. YAC, IEEE, pp. 324–328.

Grover, A., Leskovec, J., 2016. Node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 855–864.

Guo, K., Hu, Y., Sun, Y., Qian, S., Gao, J., Yin, B., 2021. Hierarchical graph convolution networks for traffic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, No. 1. pp. 151–159.

Hamilton, W., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs. Adv. Neural Inf. Process. Syst. 30.

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M., 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 639–648.

Henaff, M., Bruna, J., LeCun, Y., 2015. Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163.

Hu, D., 2019. An introductory survey on attention mechanisms in NLP problems. In: Proceedings of SAI Intelligent Systems Conference. Springer, pp. 432–448.

Huang, Y., Song, X., Zhang, S., James, J., 2021. Transfer learning in traffic prediction with graph neural networks. In: 2021 IEEE International Intelligent Transportation Systems Conference. ITSC, IEEE, pp. 3732–3737.

Ke, J., Feng, S., Zhu, Z., Yang, H., Ye, J., 2021. Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach. Transp. Res. C 127, 103063.

Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Kumar, S.V., Vanajakshi, L., 2015. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. Eur. Transp. Res. Rev. 7 (3), 1–9.

Li, Y., Wang, X., Sun, S., Ma, X., Lu, G., 2017a. Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks. Transp. Res. C 77, 306–328.

Li, Y., Yu, R., Shahabi, C., Liu, Y., 2017b. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926.

Liu, Y., James, J., Kang, J., Niyato, D., Zhang, S., 2020. Privacy-preserving traffic flow prediction: A federated learning approach. IEEE Internet Things J. 7 (8), 7751–7763.

Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M., 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5115–5124.

Pan, B., Demiryurek, U., Shahabi, C., 2012. Utilizing real-world transportation data for accurate traffic prediction. In: 2012 IEEE 12th International Conference on Data Mining. IEEE, pp. 595–604.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J., 2019. Stand-alone self-attention in vision models. Adv. Neural Inf. Process. Syst. 32.

Seo, Y., Defferrard, M., Vandergheynst, P., Bresson, X., 2018. Structured sequence modeling with graph convolutional recurrent networks. In: International Conference on Neural Information Processing. Springer, pp. 362–373.

Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P., 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process. Mag. 30 (3), 83–98.

Smith, B.L., 1995. Forecasting Freeway Traffic Flow for Intelligent Transportation Systems Application. University of Virginia.

Strokach, A., Becerra, D., Corbi-Verge, C., Perez-Riba, A., Kim, P.M., 2020. Fast and flexible protein design using deep graph neural networks. Cell Syst. 11 (4), 402–411.

Sun, J., Zhang, J., Li, Q., Yi, X., Liang, Y., Zheng, Y., 2020. Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks. IEEE Trans. Knowl. Data Eng.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Wei, Y., Wang, X., Nie, L., He, X., Hong, R., Chua, T.-S., 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 1437–1445.

Williams, B.M., Hoel, L.A., 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. J. Transp. Eng. 129 (6), 664–672.

Wu, C.-H., Ho, J.-M., Lee, D.-T., 2004. Travel-time prediction with support vector regression. IEEE Trans. Intell. Transp. Syst. 5 (4), 276–281.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y., 2020. A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. 32 (1), 4–24.

Yu, F., Koltun, V., 2015. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

Yu, H., Liu, A., Wang, B., Li, R., Zhang, G., Lu, J., 2020. Real-time decision making for train carriage load prediction via multi-stream learning. In: AI 2020: Advances in Artificial Intelligence: 33rd Australasian Joint Conference, AI 2020, Canberra, ACT, Australia, November 29–30, 2020, Proceedings 33. Springer, pp. 29–41.

Yu, H., Lu, J., Liu, A., Wang, B., Li, R., Zhang, G., 2022. Real-time prediction system of train carriage load based on multi-stream fuzzy learning. IEEE Trans. Intell. Transp. Syst. 23 (9), 15155–15165.

Yu, H., Lu, J., Zhang, G., 2021. Morstreaming: a multioutput regression system for streaming data. IEEE Trans. Syst. Man Cybern. A 52 (8), 4862–4874.

Yu, B., Yin, H., Zhu, Z., 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875.

Zhang, J., Zheng, Y., Qi, D., 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Thirty-First AAAI Conference on Artificial Intelligence.

Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X., Li, T., 2018. Predicting citywide crowd flows using deep spatio-temporal residual networks. Artificial Intelligence 259, 147–166.

Zhao, Z., Chen, W., Wu, X., Chen, P.C., Liu, J., 2017. LSTM network: a deep learning approach for short-term traffic forecast. IET Intell. Transp. Syst. 11 (2), 68–75.

Zheng, Y., 2019. Urban Computing. MIT Press.

Zheng, Y., Capra, L., Wolfson, O., Yang, H., 2014. Urban computing: concepts, methodologies, and applications. ACM Trans. Intell. Syst. Technol. 5 (3), 1–55.

Zheng, C., Fan, X., Wang, C., Qi, J., 2020. Gman: A graph multi-attention network for traffic prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, No. 01. pp. 1234–1241.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M., 2020. Graph neural networks: A review of methods and applications. AI Open 1, 57–81.