Dynamic Multi-View Graph Neural Networks for Citywide Traffic Inference

SHAOJIE DAI and JINSHUAI WANG, College of Computer Science and Technology, Ocean University of China
CHAO HUANG, Department of Computer Science, The University of Hong Kong
YANWEI YU and JUNYU DONG, College of Computer Science and Technology, Ocean University of China

Accurate citywide traffic inference is critical for improving intelligent transportation systems with smart city applications. However, this task is very challenging given the limited training data, due to the high cost of sensor installment and maintenance across the entire urban space. A more practical scenario to study the citywide traffic inference is effectively modeling the spatial and temporal traffic patterns with limited historical traffic observations. In this work, we propose a dynamic multi-view graph neural network for citywide traffic inference with the method CTVI+. Specifically, for the temporal dimension, we propose a temporal self-attention mechanism that is capable of learning the dynamics of traffic data with the time-evolving traffic volume variations. For spatial dimension, we build a multi-view graph neural network, employing the road-wise message passing scheme to capture the region dependencies. With the designed spatial-temporal learning paradigms, we enable our traffic inference model to encode the dynamism from both spatial and temporal traffic patterns, which is reflective of intra- and inter-road traffic correlations. In our evaluation, CTVI+ achieves consistent better performance compared with different baselines on real-world traffic volume datasets. Further ablation study validates the effectiveness of key components in CTVI+. We release the model implementation at https://github.com/dsj96/TKDD.

CCS Concepts: • **Information systems** → **Spatial-temporal systems**; • **Computing methodologies** → **Learning latent representations**; • **Mathematics of computing** → *Graph algorithms;*

Additional Key Words and Phrases: Traffic volume inference, spatio-temporal dependence modeling, intelligent transportation system

ACM Reference format:
Shaojie Dai, Jinshuai Wang, Chao Huang, Yanwei Yu, and Junyu Dong. 2023. Dynamic Multi-View Graph Neural Networks for Citywide Traffic Inference. *ACM Trans. Knowl. Discov. Data.* 17, 4, Article 53 (February 2023), 22 pages.
https://doi.org/10.1145/3564754

This work is partially supported by the National Natural Science Foundation of China under grant Nos. 62176243, 61773331, U1706218, and 41927805, the Fundamental Research Funds for the Central Universities under grant No. 201964022, and the National Key Research and Development Program of China under grant No. 2018AAA0100602.
Authors' addresses: S. Dai, J. Wang, Y. Yu (corresponding author), and J. Dong, College of Computer Science and Technology, Ocean University of China, Songling RD 238, Qingdao 266100, Shandong, China; emails: {daishaojie, wangjinshuai}@stu.ouc.edu.cn, {yuyanwei, dongjunyu}@ouc.edu.cn; C. Huang, Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong, China; email: chaohuang75@gmail.com.

1556-4681/2023/02-ART53 $15.00
https://doi.org/10.1145/3564754

1 INTRODUCTION

Real-time traffic monitoring has attracted much attention in smart cities because of various urban sensing applications can benefit from it, including intelligent transportation system [2], user mobility trace analysis [10], location-based recommendation [11, 20], and public safety [14]. For instance, accurate citywide traffic inference can benefit the decision maker to generate appropriate traffic volume management strategies, which reduces traffic congestion and contributes to a more efficient transportation system [50]. Additionally, to improve the efficiency of ride-hailing service for public transit, reliable traffic volume inference can be beneficial for offering better on-demand ride-sharing service [22].

However, accurately inferring citywide traffic volume is non-trivial with several key challenges. First, ***Arbitrary Missing Values:*** Because the device communication failure or sensor errors can happen at any time, data incompleteness is ubiquitous in the sensed traffic volume data arbitrary missing values. Such traffic data incompleteness brings the challenge of modeling spatial-temporal dependencies for inferring the citywide traffic volume. Second, ***Limited Sensed Traffic Data:*** Due to the high cost of sensor deployment for monitoring traffic information with large geographical coverage, the collected traffic volume data is very limited. In the city of Jinan, only 2% of road segments in the entire urban space are covered by the deployed surveillance cameras for real-time traffic volume monitoring [45]. It is worth noting that different from the problem of traffic volume forecast [35] based on the historical data, there is no any historical data available for the unmonitored roads. Therefore, it is necessary to design effective traffic inference model to learn quality representations with insufficient traffic data. Third, ***Complex Spatial-Temporal Dependencies:*** To capture complex traffic patterns, dynamic spatial and temporal dependency modeling plays an important role in inferring traffic volume of different road segments in a city. Moreover, various road context features, such as speed limitation and the number of lanes, also affect traffic volume distribution as they reveal the characteristics of a road.

Inspired by the spatial-temporal data analytical solutions and deep learning techniques, several methods have been proposed to infer the citywide traffic volume based on partial past observations. In particular, ST-SSL [25] constructs a spatio-temporal affinity graph based on travel speed patterns and spatial correlations extracted from loop detectors and taxi trajectories. Then, it infers the citywide volume by applying graph-based semi-supervised learning on the affinity graph. References [32, 48] extend ST-SSL by incorporating with simulation module (i.e., SUMO [16]) to recover full routers from incomplete trajectories. In addition, [32] leverages reinforcement learning to improve the route recovery, and jointly models road segment similarities using graph-based embedding on both dense and incomplete trajectories. Despite their effectiveness, these works suffer from two limitations: (i) most existing models either require dense GPS trajectories or aim to recover full trajectories based on the designed traffic simulator; (ii) transition probability based on biased dense trajectories or uncertain recovered trajectories may not accurately model the complex traffic patterns between adjacent segments. Yi et al. [45] propose CT-Gen based on key-value memory neural network for traffic volume inference, which consists of a candidate selection module and a key-value attention network. Particularly, the former component selects related road segments with volume information as candidates and the latter network learns the extrinsic dependencies among road segments. However, due to the high dynamics and complexity of urban traffic, road segments with similar road contexts are not necessarily guaranteed to have similar traffic volume.

In light of the aforementioned challenges, we proposed a spatial-temporal learning framework CTVI+ which effectively captures both spatial and temporal dependencies across space and time, to achieve accurate traffic inference results. In particular, to inject the spatial context into our

representation model, dynamic spatial and feature affinity graphs are generated between road segments with respect to their geographical and road characteristics (e.g., speed limit, road type). To capture road segment-wise spatial relationships, we develop a multi-view graph convolution network to perform message passing over the constructed spatial affinity graph for road segment representation. To encode the traffic dependency with the time dimension, a temporal self-attention mechanism is designed to consider time-evolving traffic patterns with different resolutions (e.g., daily, weekly). We validate the performance improvement of our proposed CTVI+ method against state-of-the-art methods on several real-life traffic datasets.

We highlight key contributions of this work as follows:

— To tackle the citywide traffic inference task, we develop a graph neural network-based model CTVI+ to capture spatial and temporal dependencies among different road segments in a dynamic environment.
— We integrate multi-view graph convolution network with temporal self-attention network to model spatial and temporal correlations. To enhance the representation learning over road segments, a semi-supervised spatial-temporal constraint is introduced with the random walk in our CTVI+ framework.
— Experiments on real-world datasets verify the performance superiority of our proposed traffic inference model.

While this work is based on a conference article [9], the scope of the proposed work has been significantly extended. The differences between this work and the conference paper are summarized as follows:

— We extend our CTVI+ by taking the periodic traffic volume patterns into consideration to optimize the semi-supervised objective.
— We conduct additional experiments to demonstrate the effectiveness of the optimized model on two new real-world traffic datasets. Experimental results show the optimized model is significantly better than the model in the conference version. We also re-perform the experiments for parameter sensitivity of our optimized model.
— We add the ablation study to verify the effectiveness of each component in our CTVI+. We also visualize the temporal self-attention weights to understand the impact of different historical volume information on the current volume. Experimental results are shown in Table 5 and Figure 8.
— We perform the time efficiency evaluation experiment to demonstrate the effect of parallel optimization for our model. Experimental results are shown in Table 6.
— With more examples and explanations, we elaborate on each component and the benefits of our model, making the method easier to understand. In addition, we also provide the pseudocode of CTVI+ and time complexity analysis in Algorithm 1 and Section 4.6, respectively.
— We also add and discuss the recent related work for traffic volume inference in the related work section.

2 RELATED WORK

Traffic Volume Forecast. Forecasting traffic volume is a critical issue in the field of transportation. Inspired by the effectiveness of **graph neural networks** (**GNNs**), T-GCN [53] utilizes the **graph convolutional network** (**GCN**) to learn complex topological structures for capturing spatial dependence, and exploits the **gated recurrent unit** (**GRU**) to learn dynamic changes of traffic data for capturing temporal dependence. Graph attention mechanism is adopted by ST-GDN [51] for encoding the global region dependencies. GPTE [23] represents the road network as a property

graph and performs traffic estimation via neural network modeling and iterative information propagation. ASTGCN [12] adopts the spatial-temporal attention mechanism and the spatial-temporal convolution module to capture the dynamic spatial-temporal correlations and spatial patterns in traffic data. STGNN [35] combines the positional graph neural layer, recurrent neural network layer, and transformer layer to model the spatial and temporal relations between road segments for traffic volume prediction. Based on meta and transfer learning, MetaST [44] uses a local CNN and an LSTM to jointly capture spatial-temporal features and correlations, and leverages information from multiple cities to increase the stability of transfer. However, traffic volume forecast is different from traffic volume inference, because there has been no historical data available for unmonitored road segments.

Recovering Missing Spatio-temporal Data. There are varieties of research try to fill missing spatio-temporal data based on **Principal Component Analysis** (**PCA**) [17, 28, 29] or extended **Collaborative Filtering** (**CF**) [1, 38, 46]. Yi et al. [46] propose ST-MVL model, which fills missing values in geo-sensory time series using CF from multiple spatial and temporal perspectives. Wang et al. [38] propose a three-dimensional tensor factorization method to estimate the missing travel time for drivers on road segments. Tayyabasif et al. [1] propose matrix and tensor based methods to estimate these missing values by extracting common traffic patterns in large road networks. Ruan et al. [30] propose a robust low-rank tensor completion method, which utilizes the potential spatial-temporal structure and sparse noise characteristics to recover missing data. Xiang et al. [43] propose an edge computing-empowered system, GTR, for large-scale traffic data recovery with low-rank theory. GTR regards the data recovery problem as a low-rank minimization problem, then utilize the fixed-point continuation iterative scheme to model spatio-temporal correlations for accurate traffic recovery. However, these approaches rely on historical data heavily when filling in missing data. Hence they are not suitable for traffic volume inference for unmonitored road segments.

Citywide Traffic Volume Inference. Semi-supervised learning becomes the effective solution in inferring the missing traffic data. For example, Zhan et al. [49] propose a Bayesian-based method to estimate citywide traffic volume using probe taxi trajectories. They need to estimate travel speeds for volume inference using full taxi trajectories. Wang et al. [37] propose a real-time traffic volume inference model based on sparse surveillance cameras. They first learn the transition probability from the third-party GPS dataset to model the entire road network traffic. Then, they estimate the unobserved traffic patterns using a multivariate normal distribution model with the transition probabilities. However, these methods require full GPS trajectories, which are not available from actual transportation systems.

Deep neural networks have emerged as promising techniques in representation learning of complex spatial-temporal dependencies for traffic volume inference. For instance, memory-augmented neural network is designed in [45] to model traffic patterns based on the key-value attentive mechanism and the identification of relevant road segments for spatial relation learning. In the study [32], the proposed JMDI method utilizes the reinforcement learning method to impute vehicle mobility trace. In JMDI, the traffic simulator SUMO [16] is adopted. A multi-view graph embedding method is designed to capture the dependencies among different road segments. Zhang et al. [52] propose TGMC-S model, which constructs a spatial affinity graph employing the correlation coefficients of speed data to characterize the similarities among roads. Then, the spatial affinity graph and temporal continuity constraint are incorporated into the geometric matrix completion framework for network-wide traffic volume estimation.

Different from JMDI [32] constructs a spatio-temporal graph based on recovered trajectory and sets multi-hop edges among different time intervals, CTVI+ designs spatial and feature affinity

graphs on each time interval based on road network and features at each time interval, respectively, and captures temporal correlation through a temporal self-attention mechanism. Different from constructing spatial or temporal affinity graphs in our CTVI+, CT-Gen [45] aggregates the traffic volume through key-value attention network [26] from related road segments according to adjacent roads and road characteristics directly. TGMC-S [52] is a matrix completion framework, which incorporates spatial affinity correlations formed through crowdsourcing floating car data and temporal continuity characteristics into a geometric matrix factorization model, and utilizes the **alternating direction method of multipliers** (**ADMM**) algorithm to solve it. While CTVI+ obtains the representation of road segments through multi-view graph convolution, and leverages the correlation of learned representations to infer the traffic volume. Furthermore, compared with the previous works, the biggest difference is that we also propose a well-designed joint learning objective function, which combines unsupervised topological structure and semi-supervised traffic volume constraint.

Graph Neural Networks for Spatial-Temporal Data. Recent years has witnessed the success of GNNs in various domains, such as social network analysis [3, 31], recommendation [4, 13], and healthcare [8, 21]. The key idea of GNNs is to capture the structural dependence of graph data through the message passing schemes for information aggregation [34, 42, 47]. Due to the strong relation learning ability of GNNs, many GNN-based spatial-temporal learning methods have been proposed to tackle the challenges in spatial-temporal data, such as location-based recommendation [39], traffic speed prediction [24], interactive behavior forecasting [19], crime prediction [18], and region representation learning [41]. Motivated by the effectiveness of GNNs, our proposed CTVI+ is designed with graph-based message passing for spatial context learning.

3 PROBLEM DEFINITION

In this section, important definitions with notations are introduced. Then, the studied task is formally presented.

Definition 1 (Road Segment). For a city, each road segment serves as the spatial unit for traffic volume inference. We define $R = \{r_1, r_2, \ldots, r_n\}$ to represent the set of citywide road segments.

In our problem, various geographical contextual features $\mathbf{x}_i = \{x_i^1, x_i^2, \ldots, x_i^f\}$ are considered in our model to enhance road segment representations, such as road length, road levels, the number of lanes, starting/ending positions, as well as speed limitation. $\mathbf{X}$ is defined to denote the feature matrix of all road segments.

Definition 2 (Time Interval). For temporal dimension, the entire time period is partitioned into m time intervals with equal time length, i.e., $T = \{t_1, t_2, \ldots, t_m\}$.

To measure the traffic volume of each road segment at a certain time interval, sensing devices (e.g., loop detectors, and surveillance cameras) are adopted as the traffic information monitor. However, due to expensive installation and maintenance costs, it is far from sufficient in terms of acquiring the city scale volume information [25]. We define $\mathcal{M}$ and $\mathcal{U}$ to represent the road segments with and without the monitored traffic volume, respectively.

Definition 3 (Traffic Volume). For each road segment r_i, we define y_i^j to represent its traffic volume at the jth time interval which is the total number of corresponding traversing vehicles.

Problem 1 (Problem Statement). Given the above definitions, our studied problem of citywide traffic volume inference is to predict the unobserved traffic volume of the road segments ($r_i \in \mathcal{U}$) without the monitored traffic information at the target time intervals.

Key notations used in this article are summarized in Table 1.

Table 1. Main Notations and their Definitions

Notation	Definition
n, m	the number of road segments and time intervals
r_i	the road segment r_i
$\mathbf{x}_i$	the feature vector of road segment r_i
$\mathbf{X}$	the feature matrix of all road segments
$y_i^j / \hat{y}_i^j$	the ground truth/referred traffic volume of r_i during t_j
$\mathcal{M}/\mathcal{U}$	the monitored/unmonitored road segment set
$\mathcal{G}_s^j / \mathcal{G}_f^j$	the spatial/feature affinity graph at time interval t_j
$\mathbf{A}_s / \mathbf{A}_f$	the adjacent matrix of affinity/feature affinity graph
$\mathcal{G}^j$	the set of spatial and feature affinity graphs at time interval t_j
$\mathbb{G}$	the affinity graph set at all time intervals
$\mathbf{H}_s, \mathbf{H}_f, \mathbf{H}_c$	the hidden representation in spatial/feature/common space
$\mathbb{H}_s, \mathbb{H}_f, \mathbb{H}_c$	the hidden representation matrix at all time intervals
$\mathbb{H}$	the fused hidden representation matrix at all time intervals
t_c, t_r, t_d, t_w	the currently/recently/daily/weekly time intervals

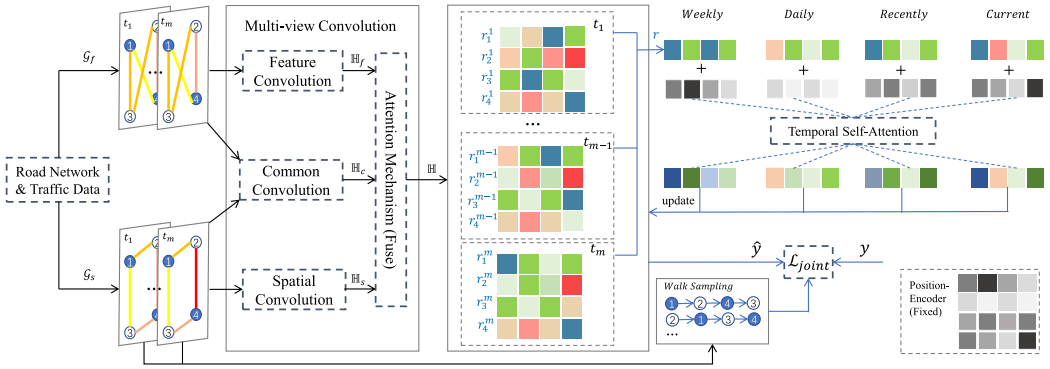


Fig. 1. The model architecture of our proposed CTVI+.

4 METHODOLOGY

This section describes the details of our CTVI+ method which is composed of four key components: (i) *affinity graph construction*, (ii) *multi-view graph convolution network*, (iii) *temporal self-attention mechanism*, and (iv) *joint learning optimization*. *First*, we construct spatial/feature affinity graphs on each time interval to model road constraints and feature similarities, respectively. *Second*, we present a multi-view graph convolution network on the affinity graphs to adaptively capture the deep correlations of road segment representation in both spatial structures and road contexts. *Third*, we employ a temporal self-attention mechanism to learn the different temporal dependencies of road segments in the embedding space. *Finally*, we design a joint learning objective function to guide the learning of final road segment representations for citywide traffic volume inference. The overall model architecture is illustrated in Figure 1.

4.1 Generating the Affinity Graph

The key challenge of inferring citywide traffic volume lies in the accurately encoding of spatial-temporal dependencies. Traffic patterns of different road segments are inter-dependent in a

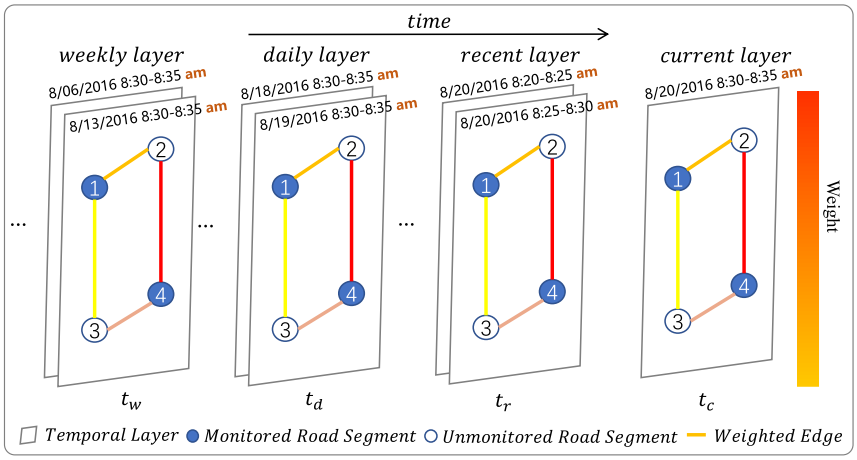


Fig. 2. An illustration of spatio-temporal affinity graph.

time-evolving scenario across different time intervals. Hence, it is important to capture the dynamic spatial dependencies between different regions. To tackle the challenge of learning region-wise spatial dynamics, our method proposes to generate spatio-temporal affinity graph illustrated in Figure 2.

The constructed spatial affinity graphs vary by different time intervals. Each individual spatial affinity graph $\mathcal{G}_s^i$ will be differentiated with weights. In our affinity graph, nodes and edges represent the road segments and their connections. In particular, the connection edge will be added between two road segments if the end intersection of one road segment is the same as the start intersection of another road segment. Thus, road network constraints are modeled by the spatial affinity graph.

By considering that the traffic volume similarity between regions may be affected by the number of lanes at an intersection, the edge weight e_{ij} in our graph is defined as follows:

$$w_{ij} = \sigma\left(liner\left(\frac{\min(lane_i, lane_j)}{\max(lane_i, lane_j)}\right)\right), \tag{1}$$

where $lane_i$ denotes the number of lanes on road segment r_i, $liner$ is a linear function, and σ is the sigmoid function to compression weight to the range (0, 1). Notice that all spatial affinity graphs are the same since the road network structure is generally unchanged.

Furthermore, we extract road contextual features from the road network. In particular, various types of road features can be considered in our model, e.g., road type, the number of lanes, speed limitation as well as the starting/ending locations. We also consider the traffic volume value as an additional road segment feature for each time interval. We initialize the unobserved volume with traffic volume averaged from its spatially k-nearest road segments. Then, we generate the feature affinity graph $\mathcal{G}_f^i$ based on kNN method with our extracted time interval-specific feature matrix $\mathbf{X}$.

Specifically, we first calculate the feature similarity matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ among n road segments. In this article, we employ cosine similarity (Equation (2)), which is a popular way to obtain the similarity between two vectors.

$$\mathbf{S}_{ij} = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{|\mathbf{x}_i||\mathbf{x}_j|}, \tag{2}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are feature vectors of road segments i and j. Afterward, we select top k similar road segments for each road segment to build edges and finally we get the adjacency matrix $\mathbf{A}_f$.

Specifically, we generate a spatial affinity graph $\mathcal{G}_s^j$ based on the road network connections in urban space. Furthermore, we incorporate road contextual features into the spatial dependency modeling by constructing the feature affinity graph $\mathcal{G}_f^j$. The set of our generated affinity graphs of different time intervals is denoted as $\mathbb{G} = \{\mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^m\}$. Here, $\mathbb{G} = \{\mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^m\}$.

4.2 Multi-View Graph Convolution Network

Graph convolutional neural networks (GCNs) have been widely used in representation learning to capture graph structure by aggregating neighbor features and have achieved great success [6, 40, 54]. Motivated by [36], we propose a multi-view GNN over the generated spatial and feature affinity graphs for road segment representations in each time interval.

4.2.1 Spatial Convolution Module. To capture the spatial dependency across regions and aggregate spatial contextual signals from neighboring road segments, we design a convolutional layer over our spatial affinity graph $\mathcal{G}_s$ based on the spectral graph theory [15].

Following the learning paradigm in [15], we define our multi-layer spatial convolution based on the following propagation scheme:

$$\mathbf{H}_s^{(l+1)} = ReLU(\tilde{\mathbf{D}}_s^{-\frac{1}{2}}\tilde{\mathbf{A}}_s\tilde{\mathbf{D}}_s^{-\frac{1}{2}}\mathbf{H}_s^{(l)}\mathbf{W}_s^{(l)}), \tag{3}$$

where $\mathbf{W}_s^{(l)}$ represents the learnable projection layer, $\tilde{\mathbf{A}}_s = \mathbf{A}_s + \mathbf{I}$ and $\tilde{\mathbf{D}}_{s,ii} = \sum_j \tilde{\mathbf{A}}_{s,ij}$. $\mathbf{H}_s^{(0)} = \mathbf{X} \in \mathbb{R}^{n \times f}$, where $\mathbf{X}$ denotes the feature matrix of all road segments. Here, f denotes the feature dimension. Furthermore, $\mathbf{H}_s^{(l)} \in \mathbb{R}^{n \times d}$ represents the output of the lth layer. The hidden state dimensionality is denoted by d for latent representations of all road segments.

4.2.2 Feature Convolution Module. Nevertheless, our developed spatial graph convolutional operations may not be able to encode the complex dependencies with respect to the graph topological structures and corresponding node features [36]. Specifically, when just spatial graph convolution is performed, it may not distinguish the importance of road constrains and road features.

Intuitively, the more similar the road features are, the more similar the traffic volume is. Therefore we perform feature convolution with $\mathbf{A}_f$ and $\mathbf{X}$ as input:

$$\mathbf{H}_f^{(l+1)} = ReLU(\tilde{\mathbf{D}}_f^{-\frac{1}{2}}\tilde{\mathbf{A}}_f\tilde{\mathbf{D}}_f^{-\frac{1}{2}}\mathbf{H}_f^{(l)}\mathbf{W}_f^{(l)}), \tag{4}$$

where $\mathbf{W}_f^{(l)}$ is a trainable neural layer for embedding transformation. By doing so, we can generate road segment feature representation $\mathbf{H}_f^{(l)}$.

4.2.3 Common Convolution Module. In reality, the spatial and feature spaces are not completely irrelevant. Thus, we not only need to extract the road segment-specific embedding in these two spaces, but also to extract the common information shared by these two spaces. After that, we design a common GCN to perform convolution operations with the parameter sharing strategy. We formally define the propagation scheme with the following operations:

$$\mathbf{H}_{cs}^{(l+1)} = ReLU(\tilde{\mathbf{D}}_s^{-\frac{1}{2}}\tilde{\mathbf{A}}_s\tilde{\mathbf{D}}_s^{-\frac{1}{2}}\mathbf{H}_{cs}^{(l)}\mathbf{W}_c^{(l)}), \tag{5}$$

$$\mathbf{H}_{cf}^{(l+1)} = ReLU(\tilde{\mathbf{D}}_f^{-\frac{1}{2}}\tilde{\mathbf{A}}_f\tilde{\mathbf{D}}_f^{-\frac{1}{2}}\mathbf{H}_{cf}^{(l)}\mathbf{W}_c^{(l)}). \tag{6}$$

Given our generated spatial graph $\mathcal{G}_s$ and feature graph $\mathcal{G}_f$, we can obtain two representations $\mathbf{H}_{cs}$ and $\mathbf{H}_{cf}$. We define a common embedding $\mathbf{H}_c$ in the *spatial and feature space* as follows:

$$\mathbf{H}_c^{(l)} = \frac{\mathbf{H}_{cs}^{(l)} + \mathbf{H}_{cf}^{(l)}}{2}. \tag{7}$$

4.2.4 Multi-View Fusion. During the fusion phase, attentive aggregation mechanism $att(\mathbf{H}_s, \mathbf{H}_f, \mathbf{H}_c)$ is introduced:

$$(\mathbf{a}_s, \mathbf{a}_f, \mathbf{a}_c) = att(\mathbf{H}_s, \mathbf{H}_f, \mathbf{H}_c), \tag{8}$$

where $\mathbf{a}_s, \mathbf{a}_f, \mathbf{a}_c \in \mathbb{R}^{n\times 1}$ denotes the attention weight of n road segments w.r.t. $\mathbf{H}_s, \mathbf{H}_f$, and $\mathbf{H}_c$, respectively. Take one road segment representation $\mathbf{h}_s^i \in \mathbb{R}^{1\times d}$ in spatial space $\mathbf{H}_s$ for instance. We first transform the representation through a nonlinear transformation, and then use one shared attention vector $\mathbf{q} \in \mathbb{R}^{d\times 1}$ to get the attention weight ω_s^i as follows:

$$\omega_s^i = \mathbf{q}^\top \cdot \mathrm{Tanh}(\mathbf{W} \cdot (\mathbf{h}_s^i)^\top + \mathbf{b}), \tag{9}$$

where $\mathbf{W} \in \mathbb{R}^{d\times d}$ denotes the trainable matrix, and $\mathbf{b} \in \mathbb{R}^{d\times 1}$ is the bias. Similarly, we can obtain the attention weight ω_f^i and ω_c^i for road segments r_i, respectively. Afterward, we perform $softmax$ function to normalize the attention weight as follows:

$$a_s^i = softmax(\omega_s^i) = \frac{\exp(\omega_s^i)}{\exp(\omega_s^i) + \exp(\omega_f^i) + \exp(\omega_c^i)}. \tag{10}$$

Similarly, $a_f^i = softmax(\omega_f^i)$ and $a_c^i = softmax(\omega_c^i)$. We generalize this definition to all roads and have the learned attention weight $\mathbf{a}_S = diag(\mathbf{a}_s)$, $\mathbf{a}_F = diag(\mathbf{a}_f)$, and $\mathbf{a}_C = diag(\mathbf{a}_c)$. The multi-view representations are aggregated with the following operations:

$$\mathbf{H} = \mathbf{a}_S \cdot \mathbf{H}_s + \mathbf{a}_F \cdot \mathbf{H}_f + \mathbf{a}_C \cdot \mathbf{H}_c. \tag{11}$$

4.2.5 Multi-View Graph Learning across Different Time Intervals. Finally, we apply our multi-view graph encoder on the constructed affinity and feature graphs of each time interval. Due to the multi-view convolution operation on each time interval does not influence each other, this operation has high parallelism and efficiency.

Specifically, we take the spatial affinity graph and feature affinity graph as the input of multi-view convolution networks, and the forward-propagation output of multi-view convolution networks on $\mathbb{G} = \{\mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^m\}$ is the representation $\mathbb{H} \in \mathbb{R}^{m\times n\times d}$ of all road segments at all time intervals in a d-dimensional space.

4.3 Temporal Self-Attention Mechanism

In real-world urban sensing scenarios, traffic patterns may exhibit multi-grained transitional regularities, such as daily, weekly, or even seasonally temporal dependencies [25, 45, 48]. As shown in Figure 3, the traffic volume of a road segment has strong recent, daily, and weekly patterns during a period of half a month with 5-min time intervals.

To effectively consider such periodic traffic patterns, we propose to endow our temporal encoder with the consideration of recent, daily, and weekly traffic transitional patterns.

To encode the temporal traffic patterns, our model considers four types of time intervals in our temporal encoder (as illustrated in Figure 4): (i) the current time interval t_c, (ii) the recent time intervals t_r, (iii) time intervals with day resolution, and (iv) time intervals t_w with week resolution. Then, the learned embeddings with different types of time intervals are fed into our temporal self-attention module. Formally, we define the relation learning function in our temporal self-attention

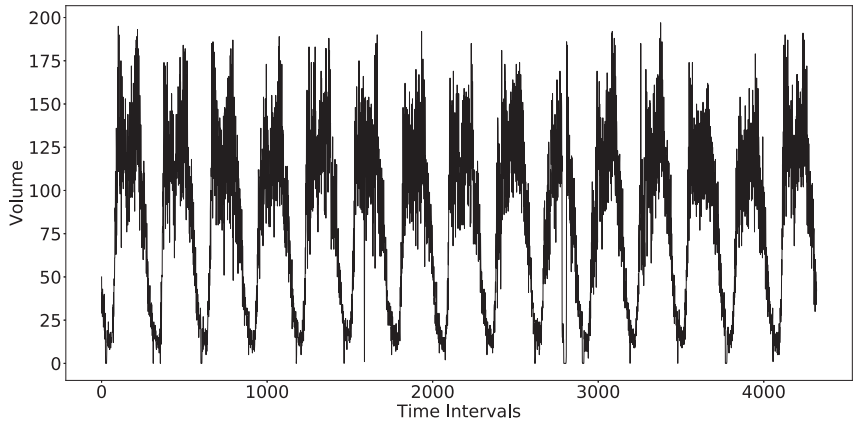


Fig. 3. The traffic volume of a road during a 15-day interval.

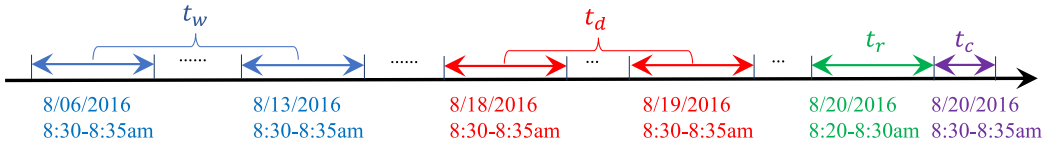


Fig. 4. An illustration of the input of temporal self-attention.

as follows:

$$\mathbf{S}_i = (\mathbf{H}_i + \mathbf{P})\mathbf{W}^Q((\mathbf{H}_i + \mathbf{P})\mathbf{W}^K)^\mathsf{T}(i = \{1, 2, \ldots, n\}), \tag{12}$$

time interval-specific representations of road segments are concatenated as $\mathbf{H}_i = \{\mathbf{h}_i^{t_w}, \mathbf{h}_i^{t_d}, \mathbf{h}_i^{t_r}, \mathbf{h}_i^{t_c}\}$. Here, $\mathbf{W}^Q \in \mathbb{R}^{d\times d}$ and $\mathbf{W}^K \in \mathbb{R}^{d\times d}$ represents the transformation weight matrices over the node embedding $\mathbf{H}_i$. We further inject the positional embedding $\mathbf{P}$ into the node representation so as to discriminate the temporally ordered information of the traffic sequence.

We define our pre-defined representation $\mathbf{P}$ as follows:

$$\mathbf{P}_{i,j} = \begin{cases} \sin(i/10000^{j/d}) & if\ i\%2 = 0, \\ \cos(i/10000^{j-1/d}) & else. \end{cases} \tag{13}$$

Then temporal self-attention score matrix $\mathbf{S}_i$ is divided by the scalar $\sqrt{d}$. The reason behind this is that the softmax function is sensitive to very large inputs. This will kill the gradient, slow down or even stop learning. If we use $\sqrt{d}$ to scale the input vector, we can prevent it from entering the saturation region or making the gradient too small. The node representations which preserve the historical information can be presented as follows:

$$\mathbf{Z}_i = softmax\left(\frac{\mathbf{S}_i}{\sqrt{d}}\right)(\mathbf{H}_i + \mathbf{P})\mathbf{W}^V(i = \{1, 2, \ldots, n\}), \tag{14}$$

where $\mathbf{W}^V \in \mathbb{R}^{d\times d}$ denotes the learnable transformation matrix for embedding projection.

4.4 Multi-Head Temporal Self-Attention

To encode the temporal dependency across different time intervals, we design a temporal self-attention as the encoder to aggregate information along the time dimension. In practice, traffic volume patterns during different time intervals may be dependent on different aspects. Hence, we

endow our temporal self-attentive network to model cross-time relation learning under multiple head representation spaces [33]. Multiple temporal self-attention heads (or facets) are computed over historical time intervals to compute final road segment representations.

$$\mathbf{Z}_i = FC(concat(\mathbf{Z}_i^{(1)}, \mathbf{Z}_i^{(2)}, \ldots, \mathbf{Z}_i^{(\#h)})). \tag{15}$$

Here, $\#h$ represents the number of attention heads in our temporal encoder. We further design a fully connected network FC to aggregate head-specific representations.

4.5 Joint Learning Optimization

After learning representations for road segments that preserve spatio-temporal dynamics based on the spatial/feature affinity graph and temporal self-attention, a joint learning objective is introduced for road segment representations. Spatial and temporal traffic patterns are effectively preserved in the encoded embeddings.

Below we formally define the joint learning objective function to obtain the results of global optimization.

We first propose to encode the dynamic spatial and temporal context with an unsupervised objective function for road segment representation. In our model, we use the dynamic representations of a node v_i at time interval t, $\mathbf{Z}_i^t$ to capture the local spatial information of v based on the spatial affinity graph representation. Specifically, we use a binary cross-entropy loss function at each time interval to encourage nodes co-occurring in fixed-length random walks, to have similar representations:

$$\mathcal{L}_{walk} = \sum_{t \in T} \sum_{v_i \in \mathcal{V}} \left(\sum_{v_j \in \mathcal{N}_{walk}^t(v_i)} -\log\left(\sigma\left(s_{ij}^t\right)\right) - \sum_{v_k \in Neg^t(v_i)} \log\left(1 - \sigma\left(s_{ik}^t\right)\right) \right), \tag{16}$$

s_{ij}^t denotes the representation similarity between the road segments r_i and r_j (i.e., similarity between $\mathbf{Z}_i^t$ and $\mathbf{Z}_j^t$), which can be any vector similarity measure function (e.g., inner product operation). σ is the sigmoid function, $\mathcal{N}_{walk}^t(v_i)$ represents the set of nodes sampled with the v_i during the process of random walks. We define the negative edge sampling set as $Neg^t(v_i)$ for node v_i at time interval t.

Second, we propose to reach an agreement between the target road segment and its top-k most similar road segments with respect to their traffic patterns under the representation space. In addition to considering the current time interval, we also consider the periodicity of the traffic volume, that is, the traffic volume of each road segment should be close to its inferred historical volume. Specifically, we consider four types of traffic volume patterns, i.e., current pattern, recent pattern, daily pattern, and weekly pattern, to be integrated into our objective. We formally present the optimized loss objective with the semi-supervised learning paradigm below:

$$\mathcal{L}_{volume} = \beta_{t_c}\mathcal{L}_{vol}^{t_c} + \beta_{t_r}\mathcal{L}_{vol}^{t_r} + \beta_{t_d}\mathcal{L}_{vol}^{t_d} + \beta_{t_w}\mathcal{L}_{vol}^{t_w}, \tag{17}$$

we define hyperparameters $\beta_{t_c}, \beta_{t_r}, \beta_{t_d}, \beta_{t_w}$ to control the importance of the current, recent, daily, and weekly traffic patterns for the traffic volume inference over the target time interval.

$$\mathcal{L}_{vol}^{t_c} = \sum_{t \in T} \sum_{r_i \in \mathcal{M}} \left| y_i^t - \frac{\sum_j^k s_{ij}^t y_j^t}{\sum_j^k s_{ij}^t} \right|, \tag{18}$$

$$\mathcal{L}_{vol}^{t_r} = \sum_{t \in T} \sum_{r_i \in \mathcal{M}} \left| y_i^t - \frac{\sum_j^k s_{ij}^{t-t_r} y_j^{t-t_r}}{\sum_j^k s_{ij}^{t-t_r}} \right|, \tag{19}$$

$$\mathcal{L}_{vol}^{t_d} = \sum_{t \in T} \sum_{r_i \in \mathcal{M}} \left| y_i^t - \frac{\sum_j^k s_{ij}^{t-t_d} y_j^{t-t_d}}{\sum_j^k s_{ij}^{t-t_d}} \right|, \tag{20}$$

$$\mathcal{L}_{vol}^{t_w} = \sum_{t \in T} \sum_{r_i \in \mathcal{M}} \left| y_i^t - \frac{\sum_j^k s_{ij}^{t-t_w} y_j^{t-t_w}}{\sum_j^k s_{ij}^{t-t_w}} \right|, \tag{21}$$

y_i^t denotes the ground truth traffic volume of road r_i during the time interval t. $\frac{\sum_j^k s_{ij}^t y_j^t}{\sum_j^k s_{ij}^t}$ aims to infer the traffic volume at the road segment r_i based on top-k most similar road segments of this target road segment in the spatial-temporal representation space at the time interval of t.

Finally, we integrate $\mathcal{L}_{walk}$ and $\mathcal{L}_{volume}$ into a joint learning framework through hyperparameters $\alpha, \beta_{t_c}, \beta_{t_r}, \beta_{t_d}$, and β_{t_w}, which are used to balance the importance of spatial structural proximities and spatio-temporal volume patterns and can be optimized during the training process. By minimizing the joint objective $\mathcal{L}_{joint}$, we can learn the hyperparameters of our framework.

$$\mathcal{L}_{joint} = \alpha \mathcal{L}_{walk} + \mathcal{L}_{volume} + \frac{\lambda}{2} ||\Theta||^2, \tag{22}$$

where λ represents the hyperparameter for regularization. Here, we define Θ to denote the model parameters. With the Equation (22), our CTVI+ method is able to well preserve the spatial and temporal dynamic patterns for traffic volume.

Taking all the aforementioned factors into consideration, we can infer the traffic volume for unmonitored road segments according to the final learned road segment representations:

$$\hat{y}_i^t = \frac{\sum_j^k s_{ij}^t y_j^t}{\sum_j^k s_{ij}^t}. \tag{23}$$

Algorithm 1 shows the pseudo-code of our CTVI+ model. First, we feed the spatial adjacency matrices and the feature adjacency matrices in historical traffic volume data into our model. Then, according to the joint objective function, the multi-view GNN and multi-head temporal self-attention network are trained at the same time. As shown in lines 3–9, we can obtain the road segment representations by optimizing Equations (3), (4), (7), (11), and (15) recursively. Finally, the traffic volume on unmonitored road segments can be inferred using the learned road segment representations by Equation (23).

4.6 Time Complexity Analysis

We now analyze the time complexity of our proposed CTVI+ for citywide traffic volume inference. CTVI+ is mainly composed of three modules: affinity graph construction, multi-view graph convolution, and temporal self-attention module. First, the time complexity of affinity graph construction is $max(O(mnd_{max}), O(mn^2 f))$, where m is the number of time intervals, n is the number of road segments, f denotes the dimension of road segment features, and d_{max} is the maximum degree of nodes in the spatial affinity graph. We should notice that $d_{max} << n$. Therefore, the overall time complexity of affinity graph construction is $O(mn^2 f)$. This part also can be included in the preprocessing. Second, for the multi-view graph convolution module, we perform spatial, feature, and common convolution separately to aggregate neighbors' features. Therefore, the computational complexity is $O(m|\mathcal{E}|df)$, where $|\mathcal{E}|$ is the number of edges in the graph, and d is embedding dimension. Third, the computational complexity of the temporal self-attention mechanism is $O(mnd^2 \# h)$, where $\# h$ represents the number of attention heads in our temporal self-attention mechanism. Therefore, the total time complexity of CTVI+ is $O(mn^2 f + m|\mathcal{E}|df + mnd\# h)$. Since multi-view graph construction and convolution operations on each time interval are independent

ALGORITHM 1: The Learning Process of CTVI+

Input: Affinity graphs at all time intervals $\mathbb{G}$, road segment feature matrix $\mathbf{X}$, embedding dimension d, number of attention heads $\#h$, observed traffic volume $\{y_i^t | r_i \in \mathcal{M}, t = 1, 2, \ldots, m\}$.
Output: Inferred traffic volume $\{\hat{y}_i^t | r_i \in \mathcal{U}, t = 1, 2, \ldots, m\}$

```
Calculate spatial adjacency matrix 𝔸_s = {A_s^1, A_s^2, ..., A_s^m} by Equation (1)
Calculate feature adjacency matrix 𝔸_f = {A_f^1, A_f^2, ..., A_f^m} by Equation (2)
while L_joint not converge do
    ℍ_s = {H_s^1, H_s^2, ..., H_s^m} ← perform spatial convolution by Equation (3)
    ℍ_f = {H_f^1, H_f^2, ..., H_f^m} ← perform feature convolution by Equation (4)
    ℍ_c = {H_c^1, H_c^2, ..., H_c^m} ← perform common convolution by Equation (7)
    ℍ = {H^1, H^2, ..., H^m} ← perform attention mechanism by Equation (11)
    ℤ = {Z_1, Z_2, ..., Z_n} ← perform multi-head temporal self-attention by Equation (15)
end while
Infer traffic volume ŷ_i^t by Equation (23)
```

of each other, that is, they can be performed in parallel, so the total time complexity of our CTVI+ can be reduced to $O(n^2 f + |\mathcal{E}|df + mnd^2\#h)$.

5 EXPERIMENT

In this section, we first introduce the details of four evaluation datasets and the competitor algorithms. We study the effectiveness of our method in inferring citywide traffic volume on four datasets compared to state-of-the-art baselines. We then focus on the ablation study to verify the effect of each component of our proposed model on four datasets. Finally, the parameter sensitivity of our model w.r.t. the important parameters and model efficiency are investigated.

5.1 Datasets

We evaluate the performance of our CTVI+ method on two collected traffic volume datasets from the cities of Hangzhou and Jinan in China. Furthermore, we also conduct extensive experiments on two public real-world datasets collected from California in USA by the Caltrans **Performance Measurement System** (**PeMS**) [5, 12]. Particularly, the traffic volume of road segments in Hangzhou is measured through the deployed traffic radars. In Jinan city, traffic surveillance cameras serve as traffic volume detector. The loop detectors, deployed on the highway, are utilized to monitor the traffic flow information in California. These four datasets are collected from 46 traffic radars in Yuhang district at Hangzhou city, 165 surveillance cameras in Jinan city, 307, and 170 loop detectors in major urban areas of California. Since the traffic volume during the morning rush hour has the greatest impact on urban traffic, our experiments are conducted only during the period between 7AM and 9AM on each dataset. Table 2 lists the detailed information of evaluation datasets.

5.2 Methods for Performance Comparison

In the performance comparison, the following methods are considered as baselines.

- **K-Nearest Neighbors (KNN)** - This baseline calculates the averaged volume from the top-k nearest road segments as the inference results.
- **Contextual Average (CA)** - CA estimates the traffic volume of the target road segment from the identified top-k similar road segments based on the generated features.

Table 2. Basic Statistics of Four Datasets

Dataset	Hangzhou City	Jinan City	PeMS04	PeMS08
Time Spans	2021/01/03-01/03	2016/08/01-08/31	2018/01/01-01/31	2016/07/01-07/31
#Road Segments	553	493	340	295
#Monitored Segments	46	165	307	170
#Features	8	7	3	3
Time Interval (minute)	5	5	5	5
Sensor Type	Traffic radar	Surveillance camera	Loop detector	Loop detector

— **MLP** - This baseline takes the flattened features as the input and incorporates them into the Multilayer perceptron for feature vector projection.
— **XGBoost** [7] - XGBoost is an efficient algorithm with the gradient boosted trees to perform regression over the traffic volume of each road segment. Each time interval is trained separately in XGBoost method.
— **ST-SSL** [25] - This approach is built on the semi-supervised learning framework to fusion data from different sources. In ST-SSL, the affinity graph is constructed to model spatial and temporal correlations across time intervals and road segments.
— **CityVolInf** [48] - CityVolInf combines SSL-based similarity module with traffic simulation module to model spatio-temporal correlations and transitions of traffic volume between adjacent road segments.
— **CT-Gen** [45] - CT-Gen proposes to consider similar traffic patterns among adjacent road segments by proposing a memory neural network.
— **JMDI** [32] - JMDI is a reinforcement learning method to learn vehicle mobility information from incomplete trajectories. This approach introduces a graph embedding component with the semi-supervised learning scheme to estimate the traffic volume information across the urban space.

For a fair comparison, we perform the model training on each time interval separately for baselines that cannot deal with the traffic volume on multiple time intervals.

5.3 Evaluation Metric

We adopt the widely used evaluation metrics: **Root Mean Square Error** (**RMSE**) [27] and **Mean Absolute Percentage Error** (**MAPE**) [45] to measure the accuracy of our inference results. We formally present those metrics as follows:

$$RMSE = \sqrt{\frac{1}{nm}\sum_{t=1}^{m}\sum_{i=1}^{n}(y_i^t - \hat{y}_i^t)^2}, \tag{24}$$

$$MAPE_t = \frac{100\%}{nm}\sum_{t=1}^{m}\sum_{i=1}^{n}\left|\frac{y_i^t - \hat{y}_i^t}{y_i^t}\right|, \tag{25}$$

$$MAPE_p = \frac{100\%}{nm}\sum_{t=1}^{m}\sum_{i=1}^{n}\left|\frac{y_i^t - \hat{y}_i^t}{\hat{y}_i^t}\right|. \tag{26}$$

Here, m and n denote the number of time intervals and test samples, respectively. $y_i^t \backslash \hat{y}_i^t$ represents the operation of the ground truth\inferred traffic volume information of road segment r_i during the time interval t.

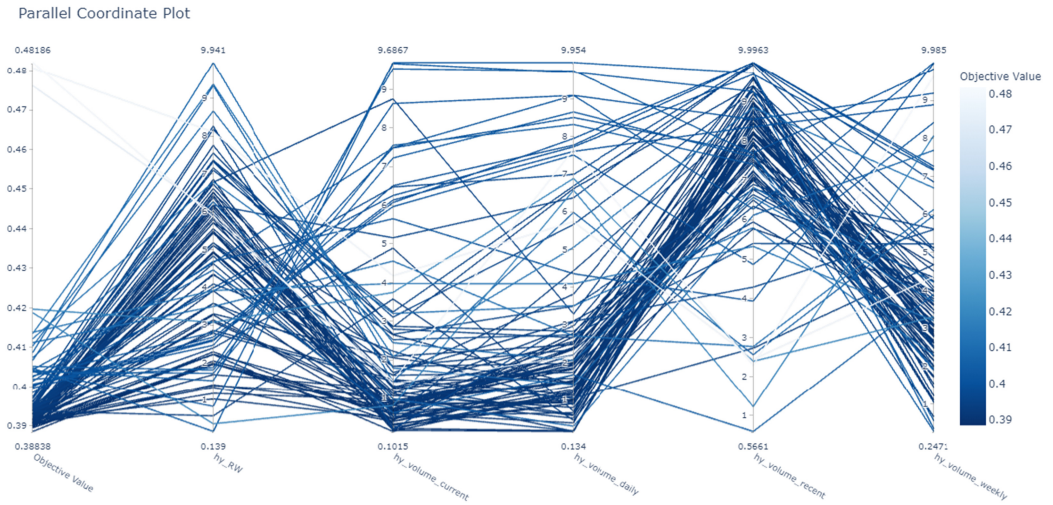


Fig. 5. Illustration of searching parameters on Jinan data.

Notice that RMSE focuses more on larger values, while MAPE receives more punishments from smaller values. Therefore, the combination of two metrics evaluates the performance of inference methods more comprehensively.

5.4 Parameter Setting

In our experiments, the length of the time interval is set as 5 minutes. We randomly partition road segments with the traffic volume information into two sets for training (80%) and testing (20%). We further set 20% of our training data as the validation set for parameter tuning. The learning rate is set as 0.005 for model optimization. The depth of our graph neural architecture is searched from the range of {1, 2, 3}. We tune the embedding dimensionality from $\{2^4, 2^5, 2^6, 2^7, 2^8\}$. λ is set to $5e^{-3}$. The number of similar road segments (k) selected in the feature space is set as 5. In our multi-head attention, the number of representation heads is set to 3. We set the number of negative samples to 5. We perform optuna,[1] a Bayesian hyperparameter optimization tool, for 100 rounds to tune the hyperparameters, and the search ranges for weights are set to $[0.1, 10]$. Specifically, when $\alpha = 7.8635$, $\beta_{t_c} = 0.8568$, $\beta_{t_r} = 1.5996$ in Hangzhou dataset, $\alpha = 2.0253$, $\beta_{t_c} = 0.6178$, $\beta_{t_r} = 8.9403$, $\beta_{t_d} = 0.9342$, $\beta_{t_w} = 3.6822$ in Jinan dataset, $\alpha = 1.6481$, $\beta_{t_c} = 6.7405$, $\beta_{t_r} = 0.9033$, $\beta_{t_d} = 2.3741$, $\beta_{t_w} = 3.6643$ in PeMS04 dataset, and $\alpha = 3.4267$, $\beta_{t_c} = 1.3315$, $\beta_{t_r} = 4.2419$, $\beta_{t_d} = 0.5515$, $\beta_{t_w} = 0.5048$ in PeMS08 dataset, our model CTVI+ obtains the best performance. For all methods, experiments are repeated with 10 runs and the averaged performance are reported. The detailed searching of optimal hyperparameters on Jinan dataset is shown in Figure 5.

The parameters setting for other baselines are listed as follows. For KNN and CA, we set $k = 5$. For MLP, we set the learning rate to 0.05, the dropout rate to 0.1, the number of layers to 3, and the hidden dimension size to 128. For XGBoost, we set the learning rate to 0.05, the maximum depth to 5, the minimum sum of instance weight needed in a child to 1, and L2 regularization term on weights to 0.01.[2] For ST-SSL, we set the learning rate to 0.001, the number of spatial neighbors to 30, and spatio-temporal factor $\alpha = 1$, $\beta = 1$. For CityVolInf, we set the maximum iteration number

[1] https://github.com/pfnet/optuna.
[2] https://github.com/dmlc/xgboost.

Table 3. Performance Comparison of Different Baselines on Hangzhou and Jinan Datasets

Dataset	Hangzhou City			Jinan City		
Methods	$MAPE_t$	$MAPE_p$	$RMSE$	$MAPE_t$	$MAPE_p$	$RMSE$
KNN (k = 5)	0.6636	0.7139	63.1035	0.6446	0.6306	60.3842
CA (k = 5)	0.6879	0.7325	65.4562	0.6568	0.6423	61.2357
MLP	0.6029	0.6561	56.4201	0.8180	0.6808	69.3974
XGBoost	0.4689	0.5243	53.9832	1.5811	0.5917	93.3649
ST-SSL	0.5638	0.5983	44.2793	0.7052	0.6883	59.0377
CityVolInf	0.4891	0.5047	37.4397	0.6526	0.4985	57.8793
CT-Gen	0.3602	0.4622	31.9691	0.6727	0.4760	57.4482
JMDI	\	\	\	0.4655	0.5574	42.0020
CTVI+	0.2420	0.3368	28.8727	0.3884	0.3780	30.2166

Table 4. Performance Comparison of Different Baselines on PeMS04 and PeMS08 Datasets

Dataset	PeMS04			PeMS08		
Methods	$MAPE_t$	$MAPE_p$	$RMSE$	$MAPE_t$	$MAPE_p$	$RMSE$
KNN (k = 5)	0.7923	0.8234	87.4532	0.6734	0.7125	83.4526
CA (k = 5)	0.8121	0.8033	88.2539	0.6811	0.7301	84.2223
MLP	0.7464	0.6498	86.8951	0.6275	0.6628	79.2848
XGBoost	0.8491	0.9433	90.1311	0.7728	0.8807	83.0742
ST-SSL	0.6522	0.6308	84.1763	0.6425	0.7581	77.9416
CityVolInf	0.5572	0.5053	72.7174	0.5915	0.5394	75.8549
CT-Gen	0.4625	0.5323	68.3129	0.4433	0.5125	66.2835
JMDI	\	\	\	\	\	\
CTVI+	0.3994	0.4573	63.1672	0.3792	0.4247	59.7413

ψ = 1000, coefficient parameters α = 4.6, β = 8.3, η = 25, respectively, and use the default setting of SUMO in the experiment. For CT-Gen, we set the candidate number to 15, the dimension of volume key embedding to 10, volume value embedding to 10, and the dimension of road context embedding to 5. For JMDI, we set discount factor γ to 0.8, mini-batch to 128, learning rate to 0.001 for deep reinforcement learning framework, speed limits to 1 m/s and 40 m/s in the simulating environment, and window size to 10 in Skip-gram.

5.5 Performance Validation

We show the evaluation results of all compared methods on four real-world traffic datasets in Tables 3 and 4. Due to the unavailability of vehicle trajectories in Hangzhou and PeMS datasets, JDMI method cannot be applied on Hangzhou, PeMS04, and PeMS08 for performance comparison.

From the evaluation results, we can observe that our new CTVI+ framework achieves the best inference results as compared to alternative solutions. In particular, the relative performance improvement of our CTVI+ over the best-performed baseline CT-Gen is 25.80%, 19.74%, and 18.62% in terms of $MAPE_t$, $MAPE_p$, and $RMSE$ (on average across four experimented datasets). This observation sheds light on the limitation of CT-Gen method in encoding complex spatial dependencies among road segments. In CT-Gen, the region-wise correlations are considered merely based on the handcrafted road features. Instead, our CTVI+ designs the multi-view graph convolution layer to learn the time-aware spatial relationships and then proposes a temporal self-attention to aggregate relevant context across all historical time slots for inference. By doing so, the spatial and temporal dynamics can be well preserved in our traffic data representation paradigm. In addition,

Table 5. Ablation Experiment Results

Dataset	Hangzhou City			Jinan City		
Methods	$MAPE_t$	$MAPE_p$	$RMSE$	$MAPE_t$	$MAPE_p$	$RMSE$
CTVI-TA	0.3078	0.4483	34.6892	0.4431	0.4259	33.0753
CTVI-PE	0.2888	0.4197	31.5093	0.4728	0.5052	35.8841
CTVI-RW	0.3286	0.5055	34.1412	0.4434	0.4256	33.0653
CTVI-VL	0.3957	0.4683	38.0991	0.4814	0.4936	36.0105
CTVI-C	0.3480	0.4136	34.3503	0.4389	0.4275	33.4542
CTVI-R	0.3086	0.3946	32.4999	0.4463	0.4280	33.1873
CTVI-D	\	\	\	0.4412	0.4281	33.1672
CTVI-W	\	\	\	0.4451	0.4298	33.2166
CTVI+	0.2420	0.3368	28.8727	0.3884	0.3780	30.2166

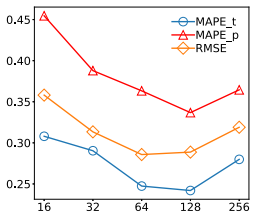
CTVI+ performs better than the baseline JMDI by 16.56%, 32.19%, and 28.06% in terms of $MAPE_t$, $MAPE_p$, and $RMSE$ on Hangzhou dataset, respectively.

The performance gain of our CTVI+ can be attributed to the joint consideration of spatial-temporal context with the geographical proximity and time-evolving dependencies. In addition, feature similarities are incorporated into our model to enhance the spatial-temporal dependency modeling in citywide traffic volume inference. As compared to representative conventional baselines, including CA, KNN, XGboost, and MLP, CTVI+ achieves better performance. We attribute this performance improvement to the advantage of our CTVI+ method in capturing traffic dynamics from both spatial and temporal dimensions. Additionally, the inference performance superiority can be observed in our CTVI+ approach compared with all competitive methods, which validates the effectiveness of our traffic inference framework with the integration of the multi-view graph convolutional module and temporal self-attention scheme.

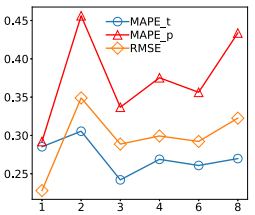
5.6 Ablation Study

To verify each component of CTVI+, we further conduct the ablation study. We compare our model with eight carefully designed variants. Despite the changed part(s), all variations have the same framework structure and parameter settings. The performance of all variations on Hangzhou and Jinan datasets are shown in Table 5.

- **CTVI-TA** - This variation removes the temporal self-attention mechanism, and directly uses the representations learned from spatial and feature affinity graphs for traffic volume inference.
- **CTVI-PE** - This variation removes the position encoding structure and ignores the historical sequence information to verify its necessity.
- **CTVI-RW** - This variation does not take unsupervised loss $\mathcal{L}_{walk}$, for augmenting the final objective function, into consideration. Specifically, we set α to 0, and the other components remain the same.
- **CTVI-VL** - This variant does not take the traffic volume loss $\mathcal{L}_{volume}$ into consideration, which aims to verify the necessity of traffic volume patterns and constraints. Specifically, we set β_{t_c}, β_{t_r}, β_{t_d}, and β_{t_w} to 0.
- **CTVI-C** - This variant does not take the current traffic volume constraint into consideration by setting β_{t_c} to 0.
- **CTVI-R** - This variant does not take the recent traffic volume constraint into consideration by setting β_{t_r} to 0.

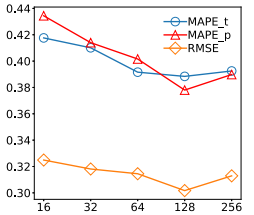


(a) *w.r.t. d*

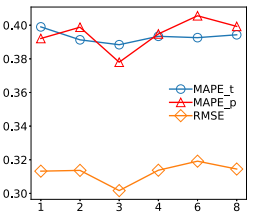


(b) *w.r.t. h*

Fig. 6. Parameter sensitivity w.r.t. d and $\#h$ on Hangzhou.



(a) *w.r.t. d*



(b) *w.r.t. h*

Fig. 7. Parameter sensitivity w.r.t. d and $\#h$ on Jinan.

— **CTVI-D** - This variant does not take the daily traffic volume pattern into consideration by setting β_{t_d} to 0.
— **CTVI-W** - This variant does not take the weekly traffic volume pattern into consideration by setting β_{t_w} to 0.

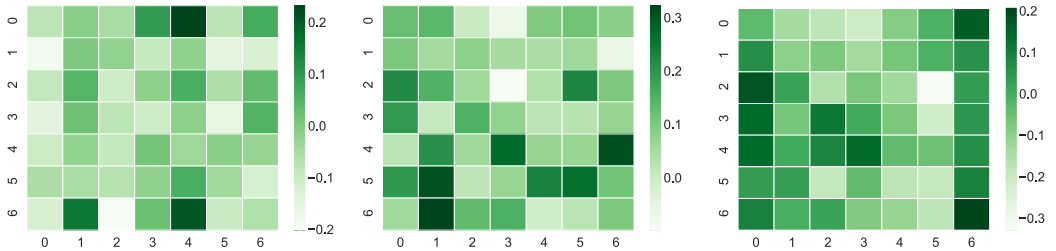
CTVI-TA and CTVI-PE mainly aim to verify the structure of the proposed framework. CTVI-RW, CTVI-VL, CTVI-C, CTVI-R, CTVI-D, and CTVI-W pay more attention on the setting of the joint leaning objective function, which reflects random walk enhancement and current\recent\daily\weekly traffic volume patterns, respectively. As we can see in Table 5, all variants significantly perform worse than CTVI+, which fully validates the effectiveness of all components of our model.

The comparisons between CTVI-TA, CTVI-RW, and CTVI+ highlight the effectiveness of the temporal self-attention structure and unsupervised random walk enhancement, respectively. The temporal self-attention mechanism attempts to capture the correlation dependence of road segment representations in the temporal dimension, which is crucial in urban traffic prediction and inference. The performance comparison verifies that the designed temporal self-attention module effectively implements the expected function. More specifically, on the long-term Jinan dataset, CTVI-PE performs worse than CTVI-TA, which further illustrates that in the temporal self-attention module, the sequence information of historical data is more important, and thus the sequence position coding is also necessary.

From Table 5, we can observe that CTVI-VL performs the worst among all variants for inferring traffic in Hangzhou and Jinan city. This indicates the effectiveness of spatio-temporal traffic volume pattern constraints in inferring traffic volume. In addition, the comparisons between CTVI-C, CTVI-R, CTVI-D, CTVI-W, and CTVI+ reflect the importance of four types of traffic volume patterns. Specifically, the current traffic volume pattern plays a more important role compared to the recent on Hangzhou dataset. In addition, current\recent\daily\weekly traffic volume patterns appear similar importance in Jinan dataset. Finally, our CTVI+ significantly outperforms all variants on both datasets. The reason behind this is that the joint embedding in CTVI+ arguments the representation capacity obtained from spatiotemporal correlations and traffic volume pattern constraints.

5.7 Parameter Sensitivity

In this subsection, we study the influence of hyperparameters on the model inference accuracy, i.e., the embedding dimensionality d and the number of heads for attentive representation. Figures 6 and 7 present the model performance in terms of $MAPE_t$, $MAPE_p$, and $RMSE$ with the configurations of different hyperparameters on Hangzhou and Jinan datasets. To keep different metrics with the same scale, we multiply RMSE score by 0.01 and show results in terms of three metrics in Figures 6 and 7.



(a) head 0 (b) head 1 (c) head 2

Fig. 8. Temporal self-attention weight.

Table 6. Runtime Comparison Analysis (Second)

Cost \ Dataset / Method	Hangzhou	Jinan	PeMS04	PeMS08
CTVI+ (w/o parallel)	460.33	130894.50	114323.76	93638.35
CTVI+ (parallel)	287.04	3376.18	4930.64	5263.68
Speedup	1.60	38.77	23.19	17.79

From the evaluation results shown in Figures 6(a) and 7(a), we can observe that the best inference performance can be achieved with the embedding dimensionality d of 128 on two experimented datasets. With the increase of hidden state dimensionality d, the model tends to be overfitting. To investigate the effect of the number of heads in our attention mechanism, we show the evaluation results with the settings of different head numbers in Figures 6(b) and 7(b). As we can see, CTVI+ is more sensitive to the number of heads in our attentional temporal aggregation on Hangzhou dataset than that on Jinan dataset. This overfitting phenomenon is caused by the sparsity of Hangzhou dataset.

Furthermore, to understand the impact of different historical volume information on the current volume, e.g., daily\weekly traffic patterns, we visualize the temporal self-attention weight S_i with $\#h = 3$ of a road segment on Jinan dataset in Figure 8. Each column denotes a type of attention score, e.g., the first and second columns denote the weekly pattern weight, and the third and fourth columns denote the daily pattern weight. Similarly, each row denotes the combination of historical sequence scores. For example, the last row denotes the effect of the historical volume sequence on the current volume. We can see that the importance of current\recent\daily\weekly traffic patterns on different heads is significantly different, and the importance of historical volume at different time intervals in the same attention score matrix is also different. From Figure 8(a), we can observe that the current volume and recent\weekly volume have a more strong relation. Additionally, in Figure 8(b), the current volume pays more attention to the daily\weekly volume patterns. Therefore, our CTVI+ can capture the complex temporal relationships from multiple perspectives by incorporating the multi-head temporal self-attention into the spatial-temporal learning architecture.

5.8 Model Efficiency Study

Finally, we further evaluate the optimized efficiency of our model by reporting the running time on four datasets in Table 6. We evaluate the time efficiency of all methods on a machine with an 8-core 1.70 GHz Intel Xeon E5-2609 CPU, 32G RAM, and 2 × GeForce RTX 2080 (8G).

As shown in Table 6, our model with parallel optimization is significantly faster than the model without parallel optimization. Specifically, CTVI+ (parallel) achieves 1.60×, 38.77×, 23.19×, and

17.79× faster than CTVI+ (w/o parallel) on Hangzhou, Jinan, PeMS04, and PeMS08 datasets, respectively. This is consistent with our previous analysis of time complexity (CTVI+ (w/o parallel) is $O(mn^2f + m|\mathcal{E}|df + mnd\#h)$ and CTVI+ (parallel) is $O(n^2f + |\mathcal{E}|df + mnd\#h)$). Since parallel optimization only works on the multi-view GCNs in different time intervals, not on the temporal self-attention module, the speedup ratio is much smaller than the number of the time intervals. Additionally, the speedup ratio on Hangzhou dataset is the smallest. The reason is that the time span of Hangzhou dataset (only one day) is much smaller than other datasets, and thus the number of time intervals is small, so the effect of parallel optimization is not significant.

6 CONCLUSION

In this article, we present a novel multi-view graph neural architecture CTVI+ which performs information aggregation over the spatial and feature affinity graphs, so as to capture the spatial and feature dependence. Additionally, CTVI+ designs a temporal self-attention mechanism to discriminate dependencies across different historical time slots. In our CTVI+ framework, a joint learning objective function is introduced to guide the representation learning of road segments for accurate traffic volume inference by incorporating both spatial and temporal traffic patterns. We perform comprehensive experiments on four real-world datasets to demonstrate the model superiority of our new proposed CTVI+ framework as compared to various state-of-the-art baselines. Moreover, we further evaluate the rationality of our designed sub-modules for improving our inference performance.

REFERENCES

[1] Muhammad Tayyab Asif, Nikola Mitrovic, Justin Dauwels, and Patrick Jaillet. 2016. Matrix and tensor based methods for missing data estimation in large traffic networks. *IEEE Transactions on Intelligent Transportation Systems* 17, 7 (2016), 1816–1825.

[2] Richard Barnes, Senaka Buthpitiya, James Cook, Alex Fabrikant, Andrew Tomkins, and Fangzhou Xu. 2020. BusTr: Predicting bus travel times from real-time traffic. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3243–3251.

[3] Qi Cao, Huawei Shen, Jinhua Gao, Bingzheng Wei, and Xueqi Cheng. 2020. Popularity prediction on social platforms with coupled graph neural networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 70–78.

[4] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 378–387.

[5] Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. 2001. Freeway performance measurement system: Mining loop detector data. *Transportation Research Record* 1748, 1 (2001), 96–102.

[6] Jie Chen, Tengfei Ma, and Cao Xiao. 2018. Fastgcn: Fast learning with graph convolutional networks via importance sampling. In *Proceedings of the International Conference on Learning Representations*. (2018).

[7] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 785–794.

[8] Limeng Cui, Haeseung Seo, Maryam Tabar, Fenglong Ma, Suhang Wang, and Dongwon Lee. 2020. Deterrent: Knowledge guided graph attention network for detecting healthcare misinformation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 492–502.

[9] Shaojie Dai, Jinshuai Wang, Chao Huang, Yanwei Yu, and Junyu Dong. 2021. Temporal multi-view graph convolutional networks for citywide traffic volume inference. In *Proceedings of the 2021 IEEE International Conference on Data Mining*. 1048–1053.

[10] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 World Wide Web Conference*. 1459–1468.

[11] Shanshan Feng, Lucas Vinh Tran, Gao Cong, Lisi Chen, Jing Li, and Fan Li. 2020. HME: A hyperbolic metric embedding approach for next-POI recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1429–1438.

[12] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, 922–929.

[13] Chao Huang, Huance Xu, Yong Xu, Peng Dai, Lianghao Xia, Mengyin Lu, Liefeng Bo, Hao Xing, Xiaoping Lai, and Yanfang Ye. 2021. Knowledge-aware coupled graph neural network for social recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4115–4122.

[14] Chao Huang, Junbo Zhang, Yu Zheng, and Nitesh V. Chawla. 2018. DeepCrime: Attentive hierarchical recurrent networks for crime prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1423–1432.

[15] Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*.

[16] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. 2012. Recent development and applications of SUMO-simulation of Urban MObility. *International Journal on Advances in Systems and Measurements* 5, 3&4 (2012), 128–138.

[17] Li Li, Yuebiao Li, and Zhiheng Li. 2013. Efficient missing data imputing for traffic flow by considering temporal and spatial dependence. *Transportation Research Part C: Emerging Technologies* 34 (2013), 108–120.

[18] Zhonghang Li, Chao Huang, Lianghao Xia, Yong Xu, and Jian Pei. 2022. Spatial-temporal hypergraph self-supervised learning for crime prediction. In *Proceedings of the 38th IEEE International Conference on Data Engineering*. 2984–2996.

[19] Zirui Li, Chao Lu, Yangtian Yi, and Jianwei Gong. 2021. A hierarchical framework for interactive behaviour prediction of heterogeneous traffic participants based on graph neural network. In *Proceedings of the IEEE Transactions on Intelligent Transportation Systems*. IEEE.

[20] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-aware sequential location recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2009–2019.

[21] Xuan Lin, Zhe Quan, Zhi-Jie Wang, Tengfei Ma, and Xiangxiang Zeng. 2020. KGNN: Knowledge graph neural network for drug-drug interaction prediction. In *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence*, Vol. 380. 2739–2745.

[22] Lingbo Liu, Zhilin Qiu, Guanbin Li, Qing Wang, Wanli Ouyang, and Liang Lin. 2019. Contextualized spatial–temporal network for taxi origin-destination demand prediction. *Transactions on Intelligent Transportation Systems* 20, 10 (2019), 3875–3887.

[23] Zhidan Liu, Pengfei Zhou, Zhenjiang Li, and Mo Li. 2018. Think like a graph: Real-time traffic estimation at city-scale. *IEEE Transactions on Mobile Computing* 18, 10 (2018), 2446–2459.

[24] Zhilong Lu, Weifeng Lv, Zhipu Xie, Bowen Du, Guixi Xiong, Leilei Sun, and Haiquan Wang. 2022. Graph sequence neural network with an attention mechanism for traffic speed prediction. *Transactions on Intelligent Systems and Technology* 13, 2 (2022), 1–24.

[25] Chuishi Meng, Xiuwen Yi, Lu Su, Jing Gao, and Yu Zheng. 2017. City-wide traffic volume inference with loop detector data and taxi trajectories. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–10.

[26] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1400–1409.

[27] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. 2019. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1720–1730.

[28] Li Qu, Li Li, Yi Zhang, and Jianming Hu. 2009. PPCA-based missing data imputation for traffic flow volume: A systematical approach. *IEEE Transactions on Intelligent Transportation Systems* 10, 3 (2009), 512–522.

[29] Li Qu, Yi Zhang, Jianming Hu, Liyan Jia, and Li Li. 2008. A BPCA based missing value imputing method for traffic flow volume data. In *Proceedings of the 2008 IEEE Intelligent Vehicles Symposium*. IEEE, 985–990.

[30] Wenjie Ruan, Peipei Xu, Quan Z. Sheng, Nickolas J. G. Falkner, Xue Li, and Wei Emma Zhang. 2017. Recovering missing values from corrupted spatio-temporal sensory data via robust low-rank tensor completion. In *Proceedings of the International Conference on Database Systems for Advanced Applications*. Springer, 607–622.

[31] Aravind Sankar, Yozen Liu, Jun Yu, and Neil Shah. 2021. Graph neural networks for friend ranking in large-scale social platforms. In *Proceedings of the Web Conference*. 2535–2546.

[32] Xianfeng Tang, Boqing Gong, Yanwei Yu, Huaxiu Yao, Yandong Li, Haiyong Xie, and Xiaoyu Wang. 2019. Joint modeling of dense and incomplete trajectories for citywide traffic volume inference. In *Proceedings of the World Wide Web Conference*. 1806–1817.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017).

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the International Conference on Learning Representations*.

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

[36] Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, and Jian Pei. 2020. AM-GCN: Adaptive multi-channel graph convolutional networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1243–1253.

[37] Yang Wang, Yiwei Xiao, Xike Xie, Ruoyu Chen, and Hengchang Liu. 2018. Real-time traffic pattern analysis and inference with sparse video surveillance information. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3571–3577.

[38] Yilun Wang, Yu Zheng, and Yexiang Xue. 2014. Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 25–34.

[39] Zhaobo Wang, Yanmin Zhu, Qiaomei Zhang, Haobing Liu, Chunyang Wang, and Tong Liu. 2022. Graph-enhanced spatial-temporal network for next POI recommendation. *Transactions on Knowledge Discovery from Data* 16, 6 (2022), 1–21.

[40] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *Proceedings of the International Conference on Machine Learning*. 6861–6871.

[41] Ning Wu, Xin Wayne Zhao, Jingyuan Wang, and Dayan Pan. 2020. Learning effective road network representation with hierarchical graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 6–14.

[42] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S. Yu Philip. 2020. A comprehensive survey on graph neural networks. *Transactions on Neural Networks and Learning Systems* 32, 1 (2020), 4–24.

[43] Chaocan Xiang, Zhao Zhang, Yuben Qu, Dongyu Lu, Xiaochen Fan, Panlong Yang, and Fan Wu. 2020. Edge computing-empowered large-scale traffic data recovery leveraging low-rank theory. *IEEE Transactions on Network Science and Engineering* 7, 4 (2020), 2205–2218.

[44] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. 2019. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *Proceedings of the World Wide Web Conference*. 2181–2191.

[45] Xiuwen Yi, Zhewen Duan, Ting Li, Tianrui Li, Junbo Zhang, and Yu Zheng. 2019. Citytraffic: Modeling citywide traffic via neural memorization and generalization approach. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2665–2671.

[46] Xiuwen Yi, Yu Zheng, Junbo Zhang, and Tianrui Li. 2016. ST-MVL: Filling missing values in geo-sensory time series data. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*.

[47] Pengyang Yu, Chaofan Fu, Yanwei Yu, Chao Huang, Zhongying Zhao, and Junyu Dong. 2022. Multiplex heterogeneous graph convolutional network. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2377–2387.

[48] Yanwei Yu, Xianfeng Tang, Huaxiu Yao, Xiuwen Yi, and Zhenhui Li. 2021. Citywide traffic volume inference with surveillance camera records. *IEEE Transactions on Big Data* 7, 6 (2021), 900–912.

[49] Xianyuan Zhan, Yu Zheng, Xiuwen Yi, and Satish V. Ukkusuri. 2016. Citywide traffic volume estimation using trajectory data. *IEEE Transactions on Knowledge and Data Engineering* 29, 2 (2016), 272–285.

[50] Xiyue Zhang, Chao Huang, Yong Xu, and Lianghao Xia. 2020. Spatial-temporal convolutional graph attention networks for citywide traffic flow forecasting. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1853–1862.

[51] Xiyue Zhang, Chao Huang, Yong Xu, Lianghao Xia, Peng Dai, Liefeng Bo, Junbo Zhang, and Yu Zheng. 2021. Traffic flow forecasting with spatial-temporal graph diffusion network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 15008–15015.

[52] Zhengchao Zhang, Meng Li, Xi Lin, and Yinhai Wang. 2020. Network-wide traffic flow estimation with insufficient volume detection and crowdsourcing data. *Transportation Research Part C: Emerging Technologies* 121 (2020), 102870.

[53] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* 21, 9 (2019), 3848–3858.

[54] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81.

Received 3 May 2022; revised 3 August 2022; accepted 4 September 2022