One Size Fits All: A Unified Traffic Predictor for Capturing the Essential Spatial–Temporal Dependency

Guiyang Luo, Hui Zhang, *Member, IEEE*, Quan Yuan, *Member, IEEE*, Jinglin Li, *Member, IEEE*, Wendong Wang, *Senior Member, IEEE*, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*— Traffic prediction is a keystone for building smart cities in the new era and has found wide applications in traffic scheduling and management, environment policy making, public safety, and so on. Instead of creating a traffic predictor for each city, this article focuses on designing a unified network model that could be directly applied for traffic prediction in any city, by learning the essential spatial–temporal dependencies, i.e., the mutual relationship between traffic and the corresponding fine-grained road network. To achieve this goal, this article proposes a joint knowledge- and data-driven mechanism that novelly divides dependencies into three kinds of correlations, i.e., road segment, intra-intersection, and inter-intersection correlation, which capture the microcosmic, middle, and macroscopic dependencies between traffic and the road network, respectively. Specifically, we first construct traffic datasets that could cover all road segments from real-world trajectory datasets, which makes it possible to model the whole road network as a graph, with the help of fine-grained road topology. Then, we propose meta road segment learner, connection-aware spatial–temporal graph convolutional network (GCN), and multiscale residual networks for capturing the microcosmic, middle, and macroscopic dependencies, respectively. Our experiments on three real-world datasets demonstrate that our proposed method could: 1) achieve better prediction accuracy compared with several approaches and 2) capture the mutual relationship between traffic and the fine-grained road network since our model trained only using data from the source city achieves good performance when it is directly applied for traffic prediction in the target city, without any fine-tuning. The codes will be made publicly available.**

***Index Terms*— Graph convolutional network (GCN), spatial–temporal dependencies (STDs), spatial–temporal GCN, traffic prediction.**

Manuscript received 12 April 2022; revised 12 November 2022 and 3 February 2023; accepted 15 March 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB1600401 and Grant 2022YFB4300403; in part by the National Natural Science Foundation of China under Grant 62102041, Grant 62203040, and Grant 62272053; and in part by the Young Elite Scientists Sponsorship Program by China Association for Science and Technology (CAST) under Grant 2022QNRC001. *(Corresponding author: Hui Zhang.)*

Guiyang Luo, Jinglin Li, and Wendong Wang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: luoguiyang@bupt.edu.cn; jlli@bupt.edu.cn; wdwang@bupt.edu.cn).

Hui Zhang is with the School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China (e-mail: huizhang1@bjtu.edu.cn).

Quan Yuan is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: yuanquan@bupt.edu.cn).

Fei-Yue Wang is with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: feiyue@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2023.3259045.

Digital Object Identifier 10.1109/TNNLS.2023.3259045

I. Background

TRAFFIC prediction focuses on capturing complex spatial correlations and dynamic temporal dependencies, thus predicting the future traffic states using historical observations [1], [2], [3]. It is one of the core components in intelligent transportation systems and plays a remarkable role in a wide range of real-world applications [4], [5]. For example, citizens benefit substantially from traffic prediction by bypassing the crowded path and keeping away from rush hours when scheduling a trip; transportation administrators enjoy the benefits of traffic prediction by preallocating transportation resources and intelligently scheduling traffic signals [6], [7]. Consequently, traffic prediction has attracted significant attention from both academia and industry and has been successfully applied in intelligent route planning, dynamic traffic management, smart location-based applications, and so on [8], [9], [10], [11], [12].

Recent years have witnessed the prosperity and popularity of data-driven solutions to traffic predictions [13], [14]. Apart from traditional methods, such as autoregressive integrated moving average (ARIMA) [15], deep learning techniques have been widely applied in traffic prediction, achieving satisfactory performance. Convolutional neural networks (CNNs) [16], [17] are utilized to model spatial dependency by decomposing the traffic network as grids [18], which are coarse-grained since each grid covers many road segments and are not able to consider the fine-grained road network. Many traffic networks are graph-structured in nature. In order to utilize such spatial information fully, it is more appropriate to formulate traffic networks as graphs mathematically [19]. Consequently, graph convolutional network (GCN), which is a variant of neural networks and suits well for nonregular data structures, has quickly gained prominence and popularity for the task of traffic forecasting since graph structure arises naturally in the traffic network [20], [21]. GCN has been demonstrated effectively in learning complex topological structures for capturing

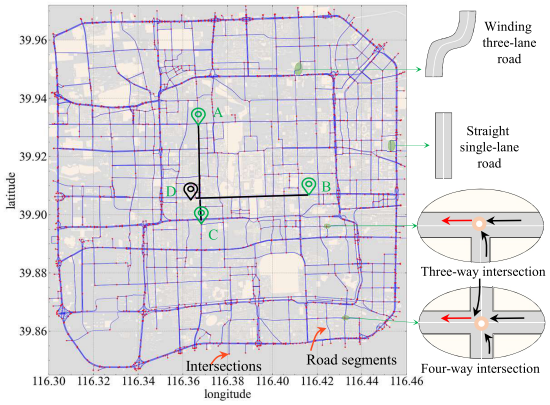


Fig. 1. Road network of Beijing, China, where the red dot means intersections and the blue line stands for road segments.

spatial dependence, integrating with recurrent neural networks (RNNs) or CNN that have been proved advantageous in learning temporal correlations. Following this framework, several GCN models have been proposed, e.g., T-GCN [22], SLC [23], DCRNN [24], dynamic spatial–temporal GCNN [25], Graph WaveNet [26], GMAN [27], and AGCRN [28].

The data-driven approaches can well learn the complex and nonlinear correlation that emerged in big data. However, the learned model can only perform well on the targeted city where the training data come from, i.e., creating a traffic predictor for each city. To address this problem, this article is focused on designing a unified prediction model that could be directly applied for traffic prediction in any city, by learning the essential spatial–temporal dependencies (STDs) through knowledge graph representation learning [29].

A. Motivation: Essential STDs

The road network of the Fourth Ring Road of Beijing, China, is shown in Fig. 1, which produces the following observations.

1) *Microcosmic Level:* Each road segment has its distinguishable attributes, e.g., the number of lanes, length, and speed limit, as shown in Fig. 1, which have an influence on the traffic.
2) *Middle Level:* The traffic of road segments within different kinds of intersections manifests different patterns. For example, for a three-way intersection, the road segment (red line) is directly affected by only two connected road segments. However, it can be directly affected by three connected road segments in a four-way intersection.
3) *Macroscopic Level:* The traffic of a road segment can be affected by multiple distant road segments, e.g., as shown in Fig. 1, the traffic of road segment D can be affected by road segments A, B, and C.

Consequently, the current traffic of a road segment can be affected by: 1) historical traffic of the road segment as well as the road segment attributes; 2) historical traffic on adjacent road segments; and 3) historical traffic of distant road segments. Based on the above observations, the essential STDs of a node depend on the current and historical traffic information of adjacent nodes, the road topology, as well as fine-grained road attributes. Existing data-driven approaches predict traffic from a macroscopic level, using only part of the knowledge. For example, these methods capture the dependencies of two sparse sensors only through the correlations in traffic time series of these two sensors. Therefore, microcosmic and middle-level knowledge are ignored.

B. Solutions

We adopt a heterogeneous semantic network, i.e., a knowledge graph to represent the essential STDs. We construct a dynamic transportation knowledge graph (DTKG) to capture the relationship among traffic information, road topology, and fine-grained road attributes. We further convert the traffic prediction problem into a knowledge graph representation learning problem to derive embeddings encoded with knowledge of correlations. To this end, we propose a joint knowledge- and data-driven mechanism to capture the essential STDs.

We first divide STD into three kinds of correlations, i.e., road segment, intra-intersection, and inter-intersection correlation, which capture the microcosmic, middle, and macroscopic dependencies between traffic and the road network, respectively. We then subtly design network models to learn each kind of correlation as follows.

1) *Meta Road Segment Learner:* Meta-learning is applied to learn the mutual influence between road segment attributes and historical traffic. A meta-knowledge learner is proposed to learn attribute-related embedding, in which the input is the road segment attributes and the corresponding output is the embedding (vector representation). The extracted embedding is then used for generating the weights of another network to learn how the traffic is affected by the historical traffic as well as the road segment attributes.
2) *Connection-Aware Spatial–Temporal GCN:* The traffic of a road segment is directly influenced by that of the connected road segments and this influence is related to the connection of intersections, i.e., the number of directly connected road segments. We propose a connection-aware spatial–temporal GCN (CSTGCN), which applies a 3-D convolutional kernel to aggregate information from the connected road segments for each type of connection, rather than a weighted average for all road segments using the same parameters. This module makes it possible to capture the essential dependencies between the road segments in an intersection.
3) *Multiscale Residual Networks:* The traffic of a road segment can be influenced by multiple distant road segments through multiple paths. We adopt multiscale GCN for capturing this influence. Furthermore, residual networks are applied to build deeper networks as well as capture long-range dependencies between road segments.

Powered with the above modules, it is possible to capture how the traffic is affected by the road network, i.e., how the traffic is affected by road segments, intersections, and nearby

road topology. In order to present a comprehensive evaluation of our proposed model, a traffic dataset that embodies all knowledge is required, i.e., containing the traffic information of all road segments. Existing publicly available traffic datasets, such as METR-LA [24], PeMS series [30], and NYC [25], which are collected from sparse and discrete sensors, hardly cover all road segments. Consequently, using these datasets cannot construct a complete DTKG. To this end, we construct three traffic datasets that could cover all road segments from the trajectory datasets. The construction consists of several procedures, e.g., road network collection and refinement, trajectory process, and map matching. To the best of our knowledge, it is the first time to construct a traffic dataset that involves all road segments. Exploiting such datasets, we first conduct experiments to evaluate the prediction accuracy of our proposed model, which achieves the best accuracy compared with nine baselines. We then exploit a model trained by data from a city to predict the traffic of another city, without further training or fine-tuning. Our proposed model achieves the best performance compared with several baselines, demonstrating that our proposed model has successfully captured the essential relationship between traffic and the road network.

C. Contributions

The contributions are summarized as follows.

1) This article incorporates knowledge graph into data-driven traffic prediction approaches, proposing a DTKG to represent the relationship among traffic information, road topology, and fine-grained road attributes, which helps to capture the essential STDs.
2) This article attempts to unify the traffic prediction model, i.e., a unified network model that could be directly applied for traffic prediction in any city instead of creating a traffic predictor for each city. This article converts the traffic prediction problem into a knowledge graph representation learning problem to derive embeddings encoded with knowledge of correlations. Furthermore, a joint knowledge- and data-driven mechanism is proposed to extract the embeddings.
3) Existing publicly available traffic datasets, which are collected from sparse and discrete sensors, hardly cover all road segments. This article presents an approach to construct a dataset that could cover all road segments. Exploiting the constructed dataset, extensive experiments on three datasets demonstrate that our proposed method achieves the best accuracy compared with existing methods. Furthermore, it is validated that our model trained only using data from the source city achieves good performance when it is directly applied for traffic prediction in the target city, without any fine-tuning.

The remainder of this article is organized as follows. Section II reviews related works. Section III provides a new insight into STDs. In Section IV, we introduce our proposed CSTGCN. The datasets that cover all road segments are introduced in Section V. The experimental evaluation and results are discussed in Section VI. Finally, Section VII concludes this work.

II. Related Works

Traffic prediction is a classic and challenging problem that has gained a lot of attention in the research community, due to the high nonlinearity and complexity of traffic data [9]. Traffic prediction approaches can be divided into two categories, i.e., model-driven and data-driven.

Recent years have witnessed unprecedented prosperity and development of data-driven approaches, e.g., machine learning and deep learning methods [31], [32], [33]. However, machine learning models suffer from stationary assumptions on the data and fail to account for highly nonlinear temporal dependency [34]. Deep learning models deliver new promises for time series forecasting problems. For example, Lai et al. [35] proposed LSTNet that uses CNN and RNN to extract short-term local dependency patterns among variables and to discover long-term patterns for time series trends. Lin et al. [36] exploited CNN for predicting inflow and outflow, jointly considering point of interests (POI) distributions and time factors that introduce prior knowledge of the crowd movements. Recently, GCN has been successfully incorporated into the traffic prediction task. Li et al. [24] modeled the traffic flow as a diffusion process on a directed graph, which captures the spatial dependency using bidirectional random walks on the graph and the temporal dependency using the encoder–decoder architecture with scheduled sampling. Chen et al. [37] exploited GCN for capturing spatial correlation and RNN for learning temporal dependencies. However, RNN-based approaches suffer from gradient explosion/vanishing and computational inefficiency for capturing long-range temporal dependencies [38]. Therefore, RNN is replaced with temporal convolution network [10], [22], [26], [39], gated CNN [34], [40], and pseudo three dimensional convolution network (P3D) [23] for capturing temporal correlation. Li et al. [9] studied neural architecture search for spatial–temporal prediction and proposed an efficient spatial–temporal neural architecture search method. However, in many application scenarios, spatial dependencies change over time, and the use of a fixed adjacency matrix cannot capture the change. Therefore, Diao et al. [25] proposed a dynamic Laplacian matrix estimator to follow the change of spatial dependencies. Song et al. [30] proposed a novel model STSGCN to capture spatial–temporal correlation simultaneously. Similar works can also be found in [41] and [42]. Furthermore, attention mechanism is adopted in [8], [27], and [43] to help capturing spatial–temporal correlations. Lan et al. [44] proposed a dynamic spatial–temporal aware graph based on a data-driven strategy, without the need of predefined static graph. Liu et al. [45] considered multistep dependency relations in traffic forecasting and proposed a new variant of RNN. Xu et al. [46] investigated spatial–temporal heterogeneity on traffic evolution.

The above approaches model the spatial correlation purely based on geographical proximity between nodes. They oversimplify the complex influence since other factors, such as the number of lanes, intersections, and speed limit, are ignored.

To settle this problem down, Zhang et al. [47] utilized the attention mechanism to model the complex relations between the road segments. Chen et al. [48] proposed a bidirectional spatial–temporal adaptive transformer (Bi-STAT) for accurate traffic forecasting. Deng et al. [49] proposed a spatiotemporal graph convolutional adversarial network to detect traffic anomalies. Gao et al. [50] presented an interacting multiple model (IMM) for short-term prediction and long-term trajectory prediction. Wang et al. [51] learned a positional representation to capture these factors for each road. Furthermore, transferring ability has been considered. Yao et al. [52] leveraged information from multiple cities to increase the stability of transfer by extracting region-level representation for transferring knowledge from a source city to a target one. Wang et al. [53] learned an intercity region matching function to match each target city region to a similar source city region, thus effectively transferring knowledge from a data-rich source city to a data-scarce target city. Wei et al. [54] proposed FLORAL to transfer knowledge from a city where there exist sufficient multimodal data and labels to similar kinds of cities to fully alleviate the problems of label scarcity and data insufficiency. Furthermore, attributes, such as weather and POI, would also affect the traffic prediction model [55], [56]. Zhu et al. [57] considered the impacts of various external factors on traffic, where the factors include weather conditions, the presence of transportation stations, emergency events, holidays, the distribution of nearby POIs, and so on. Ma et al. [58] proposed a spatial–temporal graph attentional long short-term memory neural network framework to predict short-run bike-sharing demand at a station level using multisource datasets, including historical bike-sharing trip data, historical weather data, users' personal information, and land-use data. In this article, we investigate how the traffic flow is influenced by road topology, and fine-grained road attributes. Other attributes will be considered in our future works.

In this article, we propose a joint knowledge- and data-driven mechanism to capture the essential relationship between the traffic and the road network. Specifically, we first divide STD into three kinds of correlations, i.e., road segment, intra-intersection, and inter-intersection correlation, which capture the microcosmic, middle, and macroscopic dependencies between the traffic and the road network, respectively. We then subtly propose meta road segment learner (MRSL), CSTGCN, and multiscale residual graph neural networks for implementing these three kinds of correlations.

III. Dynamic Transportation Knowledge Graph

In this section, we incorporate knowledge graph [59], [60] into the transportation area and propose a DTKG to represent a transportation road network.

Definition 1 (Road Segment): A road segment is the specific representation of a portion of a road with uniform characteristics. A road segment has its distinguishable attributes, e.g., the number of lanes, length, type (e.g., motorway, avenue way, and residential way), speed limits, and shape (e.g., curvature and slope).

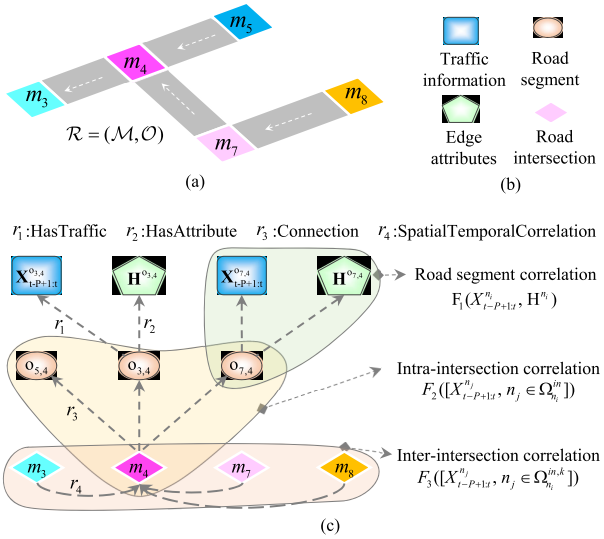


Fig. 2. DTKG is constructed to capture the essential relationship between traffic information, road network topology, and fine-grained road attributes. DTKG converts the spatial–temporal forecasting problem to a knowledge graph embedding problem, which learns a representation of DTKG's entities and relations while preserving their semantic meaning. However, the relations are dynamic and complex, and we split them into road segment correlation, intra-intersection correlation, and inter-intersection correlation. (a) Road network. (b) Legend. (c) DTKG for road intersection m_4.

Definition 2 (Road Intersection): An intersection is an at-grade junction where two or more road segments converge, diverge, meet, or cross.

Road intersections can be distinguished by the number of road segments that are involved, e.g., three- and four-way intersections. Road intersections and road segments are the basic components of any road network, which is defined as follows.

Definition 3 (Road Network): The road network of any city can be modeled by a directed graph as $\mathcal{R} = (\mathcal{M}, \mathcal{E})$, where $\mathcal{M} \triangleq \{m_1, m_2, \ldots, m_{|\mathcal{M}|}\}$ denotes the set of road intersections and $\mathcal{E} \triangleq \{e_{i,j}\}$ is the set of edges. An directed edge $e_{i,j} \triangleq m_i \rightarrow m_j \in \mathcal{E}$ represents a road segment from $m_i \in \mathcal{M}$ to $m_j \in \mathcal{M}$, as shown in Fig. 2(a).

Definition 4 (Traffic Information): For each road segment $o_{i,j} \in \mathcal{E}$ in the road network $\mathcal{R} = (\mathcal{M}, \mathcal{E})$, the traffic information (speed, volume, in-out flow, and so on) at the tth time step records the state of $o_{i,j}$, which can be denoted by $\boldsymbol{X}_t^{o_{i,j}}$.

The traffic information of all road segments $\mathcal{E}$ at time step t can be represented as graph information $\boldsymbol{X}_t \in \mathbb{R}^{|\mathcal{E}| \times C_s}$, where C_s is the number of traffic features of interest (e.g., traffic speed, volume, and in-out flow). For a road segment $o_{i,j} \in \mathcal{E}$, the traffic information at the tth time step is denoted as $\boldsymbol{X}_t^{o_{i,j}}$ and the fine-grained edge attributes are denoted as $\boldsymbol{H}^{o_{i,j}}$.

A. Traffic Prediction Formulation

Given the historical P time steps $\boldsymbol{X}_{t-P+1:t} \triangleq [\boldsymbol{X}_{t-P+1}, \boldsymbol{X}_{t-P+2}, \ldots, \boldsymbol{X}_t]$, the purpose of traffic prediction is to learn a function mapping f_θ to predict its successive Q steps' graph information $\boldsymbol{X}_{t+1:t+Q}$. The mapping function can be

represented as

$$[\boldsymbol{X}_{t-P+1:t}, \mathcal{R}, \boldsymbol{H}] \xrightarrow{f_\theta} \boldsymbol{X}_{t+1:t+Q}. \tag{1}$$

We organize the transportation road network in the form of knowledge graph $\mathcal{G}$ [60], which is a directed graph composed of subject–property–object triple facts. Formally, it is presented as $\{(\varphi, r, \psi)\}$, where each triplet describes that there is a relationship r from head entity φ to tail entity ψ [61].

Definition 5 (DTKG): Given the road network $\mathcal{R} = (\mathcal{M}, \mathcal{E})$, fine-grained road attributes $\boldsymbol{H}$, and real-time traffic information $\boldsymbol{X}_{t-P+1:t}$, a DTKG $\mathcal{G}_t$ at the tth time step can be constructed to integrate external information and spatial–temporal information, as shown in Fig. 2. The constructed DTKG possesses the following class of relationships.

1) *HasTraffic:* A road segment has the historical traffic information, e.g., $o_{3,4}$ has the traffic information $\boldsymbol{X}^{o_{3,4}}_{t-P+1:t}$.
2) *HasAttribute:* A road segment has fine-grained road attributes, e.g., $o_{3,4}$ has the road attributes $\boldsymbol{H}^{o_{3,4}}$.
3) *Connection:* A road segment has a connection with a road intersection, e.g., $o_{3,4}$ has a connection with road intersection m_4.
4) *Spatial Temporal Correlation:* Spatial–temporal correlations exist among road intersections, e.g., m_3 is correlated with m_4.

The DTKG of road intersection m_4 is shown in Fig. 2. Built upon DTKG, it is possible to capture the essential relationship between the traffic and the road network, by exploiting the fine-grained road topology when the state and information of all road segments are available. On this basis, a novel insight into STD is presented.

B. New Insights to STDs

DTKG represents the essential relationship between traffic information, road network topology, and fine-grained road attributes. As shown in Fig. 2, DTKG expresses how the traffic information of road intersection m_4 is affected by historical traffic information, road attributes, adjacent road segments, and other road intersections. DTKG can represent the knowledge of road intersections in any city. Therefore, a prediction model trained using DTKG should perform well in any city. Exploiting DTKG, a spatial–temporal forecasting problem can be treated as a knowledge graph embedding problem, which learns a representation of DTKG's entities and relations while preserving their semantic meaning. However, the relations are dynamic and complex, and we split it into road segment correlation, intra-intersection correlation, and inter-intersection correlation, which are defined as follows.

Definition 6 (Road Segment Correlation): For a road segment $o_{i,j} \in \mathcal{E}$, the traffic information at different time steps is correlated, e.g., $\boldsymbol{X}^{o_{i,j}}_t$ and $\boldsymbol{X}^{o_{i,j}}_{t-1}$ are nonindependent. Meanwhile, this correlation is related to the edge attributes $\boldsymbol{H}^{o_{i,j}}$. Therefore, the future traffic information $\dot{\boldsymbol{X}}^{o_{i,j}}_{t+1}$ is a transformation of $\boldsymbol{H}^{o_{i,j}}$ and historical traffic $\boldsymbol{X}^{o_{i,j}}_{t-P+1:t} \triangleq [\boldsymbol{X}^{o_{i,j}}_{t-P+1}, \boldsymbol{X}^{o_{i,j}}_{t-P+2}, \ldots, \boldsymbol{X}^{o_{i,j}}_{t}]$, which can be denoted as

$$\dot{\boldsymbol{X}}^{o_{i,j}}_{t+1} = F_1\big(\boldsymbol{X}^{o_{i,j}}_{t-P+1:t}, H^{o_{i,j}}\big). \tag{2}$$

For the road network $\mathcal{R} = (\mathcal{M}, \mathcal{E})$, an edge $o_{h,l} \in \mathcal{E}$ is an in-neighbor edge of $o_{i,j} \in \mathcal{E}$ if $l = i$. The set of in-neighbor edges of $o_{i,j}$ is denoted as

$$\Omega^{\text{in}}_{o_{i,j}} = \{o_{h,i} \mid o_{h,i} \in \mathcal{E}\}. \tag{3}$$

$o_{i,j}$ and $\Omega^{\text{in}}_{o_{i,j}}$ are connected by a road intersection. As shown in Fig. 2(a) (only the road segments on a direction are shown for simplicity.), $\Omega^{\text{in}}_{o_{3,2}} = \{o_{1,3}, o_{4,3}, o_{6,3}\}$, where $\Omega^{\text{in}}_{o_{3,2}}$ and $o_{3,2}$ are connected by road intersection m_3. Therefore, the traffic information $\boldsymbol{X}^{o_{3,2}}_{t+1}$ is directly influenced by traffic information of road segments $\Omega^{\text{in}}_{o_{3,2}}$, which are defined as intra-intersection correlation.

Definition 7 (Intra-Intersection Correlation): For each road segment $o_{i,j}$, its traffic information is correlated with traffic information in $\Omega^{\text{in}}_{o_{i,j}}$. Therefore, the future traffic information $\ddot{\boldsymbol{X}}^{o_{i,j}}_{t+1}$ is a transformation of $\dot{\boldsymbol{X}}^{\Omega^{\text{in}}_{o_{i,j}}}$, which can be denoted as

$$\ddot{\boldsymbol{X}}^{o_{i,j}}_{t+1} = F_2\Big(\Big[\dot{\boldsymbol{X}}^{o_{h,l}}, o_{h,l} \in \Omega^{\text{in}}_{o_{i,j}}\Big]\Big). \tag{4}$$

Meanwhile, the correlation could diffuse to distant edges. Therefore, for road segment $o_{i,j} \in \mathcal{E}$, based on in-neighbor edges $\Omega^{\text{in}}_{o_{i,j}}$, we define k-hop in-neighbor edges as $\Omega^{\text{in},k}_{o_{i,j}}$.

Definition 8 (Inter-Intersection Correlation): For each road segment $o_{i,j}$, its traffic information is correlated with traffic information in $\Omega^{\text{in},k}_{o_{i,j}}, k \geq 2$. Therefore, the future traffic information $\dddot{\boldsymbol{X}}^{o_{i,j}}_{t+1}$ is a transformation of $\ddot{\boldsymbol{X}}^{\Omega^{\text{in},k}_{o_{i,j}}}$, which can be denoted as

$$\dddot{\boldsymbol{X}}^{o_{i,j}}_{t+1} = F_3\Big(\Big[\ddot{\boldsymbol{X}}^{o_{h,l}}, o_{h,l} \in \Omega^{\text{in},k}_{o_{i,j}}\Big]\Big). \tag{5}$$

C. Traffic Prediction Reformulation

Based on the above definitions of a road segment, intra-intersection, and inter-intersection correlation, the predicted traffic $\hat{\boldsymbol{X}}_{t+1:t+Q}$ can be reformulated as

$$\hat{\boldsymbol{X}}_{t+1:t+Q} = F_3(F_2(F_1(\boldsymbol{X}_{t-P+1:t}, H))). \tag{6}$$

The purpose of prediction is to learn the set of parameters $(\theta_1, \theta_2, \theta_3)$ with the following objective:

$$\min_{\theta_1,\theta_2,\theta_3} \sum ||\hat{\boldsymbol{X}}_{t+1:t+Q} - \boldsymbol{X}_{t+1:t+Q}||_2. \tag{7}$$

Exploiting the above analysis, we subtly design the network architecture for F_1, F_2, and F_3. Then, the stochastic gradient descent algorithm is applied to solve (7).

IV. Methodology

In this section, we propose MRSL, CSTGCN, and multi-scale residual graph neural networks for implementing F_1, F_2, and F_3, respectively.

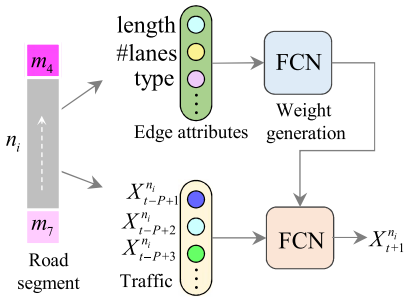


Fig. 3. MRSL for learning road segment correlation $F_1(\boldsymbol{X}^{o_{i,j}}_{t-P+1:t}, H^{o_{i,j}})$.

A. Meta Road Segment Learner

We propose MRSL to capture road segment correlation $F_1(\boldsymbol{X}^{o_{i,j}}_{t-P+1:t}, H^{o_{i,j}})$, which has to take the edge attributes into consideration. Road attributes, e.g., the number of lanes and length, would lay a distinct impact on traffic. Inspired by meta-learning [43], for a road segment $o_{i,j}$, a fully connected network (FCN) is applied to extract meta-knowledge from edge attributes $\boldsymbol{H}^{o_{i,j}}$. Then, the learned meta-knowledge is further used to learn the weights of another network for handling the time series $\boldsymbol{X}^{o_{i,j}}_{t-P+1:t}$.

For each edge $o_{i,j} \in \mathcal{E}$ (represents a road segment), its fine-grained edge attributes are denoted as a $1 \times C_e$ vector $\boldsymbol{H}^{o_{i,j}}$, by concatenating the length, the number of lanes, type, speed limit, direction, and curvature of the road segment $o_{i,j}$, where the type could be motorway, trunk way, primary way, and secondary way.[1] A meta-knowledge learner is proposed to learn edge attribute-related embedding, i.e., which is referred to as MRSL. As shown in Fig. 3, MRSL consists of several layers of FCN, in which the input is edge attributes of a road segment and the corresponding output is the embedding (vector representation) for that road segment. The extracted embedding is then used for generating the weights of another FCN to learn road segment correlation F_1. In this way, the MRSL could consider both the historical traffic data as well as edge attributes.

B. Connection-Aware Spatial–Temporal GCN

Intersections with a distinct number of road segments exert different influences on traffic. For example, as shown in Fig. 4(b), m_3 and m_4 are two significantly different intersections: one is a four-way intersection and the other is three-way. According to DTKG, the traffic of different road segments is correlated. We refer to the spatiotemporal correlation and dynamics among traffic between road segments as the mutual relationship. Therefore, the mutual relationship among traffic between road segments $o_{1,3}$, $o_{6,3}$, $o_{4,3}$, and $o_{3,2}$ is different from that between $o_{7,4}$, $o_{5,4}$, and $o_{4,3}$. Hence, the information of in-neighbor road segments should not be evenly aggregated, as done by most existing GCN approaches [22], [23], [24], [25], [26], [27]. These approaches first aggregate the information and then balance it with node degree. To capture the essence of intra-intersection correlation, we take road segment connection in intersections into consideration, i.e., road segments with a different number of in-neighbor edges are distinguished. To this end, a convolutional kernel is applied to aggregate information for each type of connection, rather than a weighted average for all edges using the same parameters.

For a road segment $o_{i,j} \in \mathcal{E}$, let D_i stand for its in-neighbor road segments $\Omega^{\text{in}}_{o_{i,j}}$ as well as $o_{i,j}$, i.e.,

$$D_i \triangleq \left\{\Omega^{\text{in}}_{o_{i,j}} \cup o_{i,j}\right\}. \tag{8}$$

The traffic of edges in D_i has a direct influence on the traffic of edge $o_{i,j}$, which is also known as spatial correlation or intra-intersection correlation, as shown in Fig. 4(a), and this influence is heavily corrected with the in-neighbor edges of $o_{i,j}$. To learn this distinguished influence, we propose a connection-aware graph convolutional layer (CGCN), which designs a convolutional kernel for the edges with the same number of in-neighbor edges. Therefore, for edge $o_{i,j}$, the convolution operation can be defined as

$$\ddot{\boldsymbol{X}}[o_{i,j}, :] \triangleq \frac{1}{|D_i|}\sum_{l=0}^{|D_i|} \dot{\boldsymbol{X}}_t^{D_i[l]} \boldsymbol{W}_{2D}^{|D_i|}[l] \tag{9}$$

where $\ddot{\boldsymbol{X}} \in \mathbb{R}^{N\times C_{\text{out}}}$, $D_i[l]$ is the lth value of the set D_i that sorts the road segments in clockwise order, $\dot{\boldsymbol{X}}_t^{D_i[l]} \in \mathbb{R}^{1\times C_s}$ is the output features of MRSL module, C_s is the feature dimension, and $\boldsymbol{W}_{2D}^{|D_i|} \in \mathbb{R}^{|D_i|\times C_s\times C_{\text{out}}}$ is the convolutional kernel for the set of edges $\{o_{h,l}||D_j| = |D_i|, o_{h,l} \in \mathcal{E}\}$.

The edges in D_i also show spatial, temporal, and spatial–temporal correlations, as shown in Fig. 4(a). The adjacent edges in a time step reveal spatial correlation, which can be efficiently captured by CGCN because a kernel is able to learn mutual correlation among road segments in D_i (intra-intersection correlation). The temporal and spatial–temporal correlations emerge among road segments with different time steps. The former is aroused by the same road segment with different time steps, while the latter is induced by adjacent road segments across different time steps. To simultaneously capture spatial, temporal, and spatial–temporal correlations, inspired by 3-D convolution [62], we propose a CSTGCN, which is referred to as CSTGCN. CSTGCN is based on CGCN and extends the set of 2-D convolution kernels $\{\boldsymbol{W}_{2D}^{|D_i|}, o_{i,j} \in \mathcal{E}\}$ to 3-D, as shown in Fig. 4(b).

Suppose that there exist K class of convolutional kernels in CSTGCN, $K \triangleq |\mathcal{K}|$, and $\mathcal{K} \triangleq \{|D_i| \mid o_{i,j} \in \mathcal{E}\}$. In each time step, CGCN applies a set of 2-D convolutional kernels $\{W_{2D}^{|\mathcal{K}[1]|}, W_{2D}^{|\mathcal{K}[2]|}, \ldots, W_{2D}^{|\mathcal{K}[K]|}\}$ to learn the spatial correlation. In CSTGCN, with P historical time steps, the corresponding set of 3-D kernel $\{W_{3D}^{|\mathcal{K}[1]|}, W_{3D}^{|\mathcal{K}[2]|}, \ldots, W_{3D}^{|\mathcal{K}[K]|}\}$ can be applied to capture the spatial, temporal, and spatial–temporal correlations simultaneously, as shown in Fig. 4(b). The convolution operation of CSTGCN can be denoted as

$$\ddot{\boldsymbol{X}}[o_{i,j}, :] \triangleq \frac{1}{P|D_i|}\sum_{t'=t-P+1}^{t}\sum_{l=1}^{|D_i|} \dot{\boldsymbol{X}}_{t'}^{D_i[l]} \boldsymbol{W}_{3D}^{|D_i|}[t', l] \tag{10}$$

where $\boldsymbol{W}_{3D}^{|D_i|} \in \mathbb{R}^{P\times|D_i|\times C_s\times C_{\text{out}}}$.

[1]The fine-grained edge attributes have been defined by OpenStreetMap, and please refer to https://wiki.openstreetmap.org/wiki/Map_Features

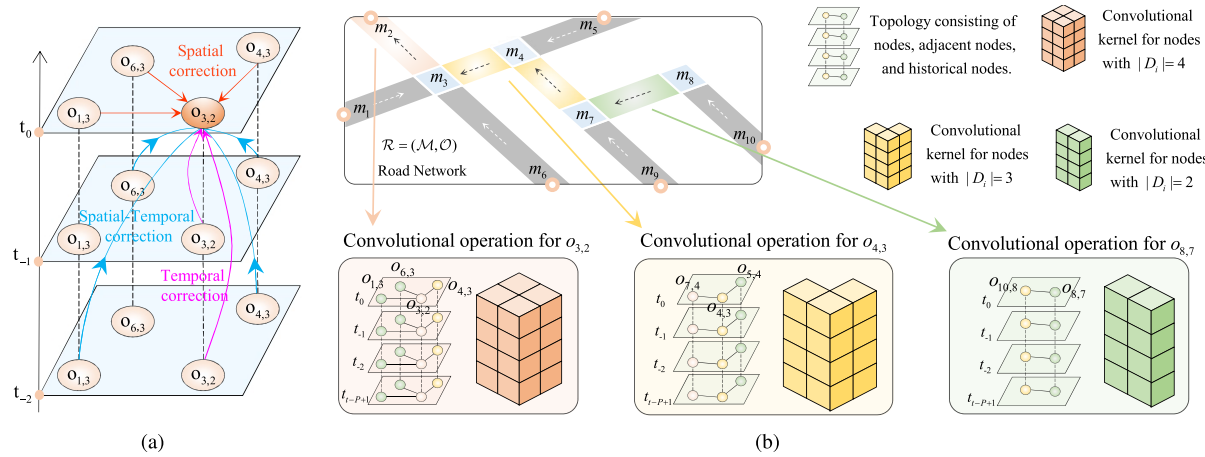


Fig. 4. (a) Correlations among edges within different time steps, i.e., spatial, temporal, and spatial–temporal correlations. (b) Example of the convolution layer for CSTGCN, which is connection-aware, i.e., a different convolution kernel is designed for each class of edges that has the same number of in-neighbor edges. CSTGCN can simultaneously capture the spatial, temporal, and spatial–temporal correlations and learn the mutual influence of road segments within an intersection.

CSTGCN can be expressed in a matrix form with efficient implementation. We first define an edge connection matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ as follows:

$$\boldsymbol{A}[o_{i,j}, o_{h,l}] = \begin{cases} 1, & \text{if } j = h \\ 0, & \text{Otherwise.} \end{cases} \tag{11}$$

Given the edge connection matrix $\boldsymbol{A}$ of a graph, we could obtain $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix. We first divide $\tilde{\boldsymbol{A}}$ into K set of matrices by using $|D_i|$ as follows, with the edges in each set sharing a convolutional kernel:

$$\tilde{\boldsymbol{A}} \to \{(\tilde{\boldsymbol{U}}^1, \tilde{\boldsymbol{V}}^1), (\tilde{\boldsymbol{U}}^2, \tilde{\boldsymbol{V}}^2), \ldots, (\tilde{\boldsymbol{U}}^K, \tilde{\boldsymbol{V}}^K)\} \tag{12}$$

where $\tilde{\boldsymbol{U}}^l \in \mathbb{R}^{N\times\mathcal{K}[l]}$, $1 \le l \le K$, is defined as

$$\tilde{\boldsymbol{U}}^l[o_{i,j}, :] = \begin{cases} D_i, & |D_i| = \mathcal{K}[l] \\ \boldsymbol{0}_{1\times\mathcal{K}[l]}, & \text{Otherwise} \end{cases} \tag{13}$$

and $\tilde{\boldsymbol{V}}^l \in \mathbb{R}^{N\times\mathcal{K}[l]}$, $1 \le l \le K$, is defined as

$$\tilde{\boldsymbol{V}}^l[o_{i,j}, :] = \begin{cases} \boldsymbol{1}_{1\times\mathcal{K}[l]}, & |D_i| = \mathcal{K}[l] \\ \boldsymbol{0}_{1\times\mathcal{K}[l]}, & \text{Otherwise.} \end{cases} \tag{14}$$

The above operations are not related to graph information and can be done during initialization.

Given the graph information $\dot{\boldsymbol{X}}_{t-P+1:t} \in \mathbb{R}^{P\times N\times C_s}$, $\tilde{\boldsymbol{U}}^l$ can be used as the index to gather the corresponding values, i.e.,

$$\boldsymbol{Z}[o_{i,j}, :, :, :] = \dot{\boldsymbol{X}}_{t-P+1:t}[:, \tilde{\boldsymbol{U}}^l[o_{i,j}, :], :] \cdot \tilde{\boldsymbol{V}}^l[o_{i,j}, :] \tag{15}$$

where $\boldsymbol{Z} \in \mathbb{R}^{N\times P\times\mathcal{K}[l]\times C_s}$. After gathering, a clockwise order is applied to sort the road segments.

With the lth kernel as $\boldsymbol{W}^{\mathcal{K}[l]} \in \mathbb{R}^{P\times\mathcal{K}[l]\times C_s\times C_{\text{out}}}$, the operation of convolution on edge $o_{i,j}$ can be denoted as

$$\ddot{\boldsymbol{X}}[o_{i,j}, :] = \frac{1}{P|D_i|} \sum_{t'=t-P+1}^{t} \sum_{o_{h,l}\in D_i} \boldsymbol{Z}[o_{i,j}, t', o_{h,l}, :] \cdot \boldsymbol{W}_{3D}^{\mathcal{K}[l]}[o_{h,l}, t', :, :] \tag{16}$$

where $|D_i|$ is equal to $\mathcal{K}[l]$, and $\ddot{\boldsymbol{X}} \in \mathbb{R}^{Q\times N\times C_{\text{out}}}$.

To this end, the convolutional operation of CSTGCN for the edges $\{o_{i,j} || D_i| = \mathcal{K}_l, o_{i,j} \in \mathcal{E}\}$ can be expressed as

$$\ddot{\boldsymbol{X}} = \frac{\text{diag}(\hat{\boldsymbol{A}} \cdot \boldsymbol{1}_{N\times N})^{-1}}{P} \left(\dot{\boldsymbol{X}}[\tilde{\boldsymbol{U}}^l]\tilde{\boldsymbol{V}}^l * \boldsymbol{W}_{3D}^{\mathcal{K}[l]}\right) \tag{17}$$

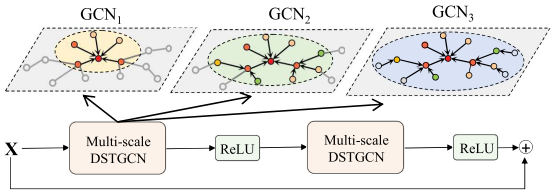


Fig. 5. Multiscale residual CSTGCN unit.

where $\boldsymbol{1}$ is the $N \times N$ all-one matrix and diag$\{\boldsymbol{A}\}$ means the diagonal matrix of $\boldsymbol{A}$.

C. Multiscale Residual Graph Neural Netowrks

We propose multiscale residual graph neural networks based on CSTGCN for implementing F_3, which is denoted as multiscale residual CSTGCN. It consists of a multiscale graph neural network for capturing inter-intersection correlation and a residual network module for building deeper networks.

We apply GCN proposed in [63] for efficient multiscale feature extraction, and the propagation rule is denoted as follows:

$$\boldsymbol{Y} = \sigma\left(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{W}\right) \tag{18}$$

where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are the input and output graph signals, respectively, $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ is the adjacency matrix with added self-connections, $\tilde{\boldsymbol{D}} = \text{diag}(\tilde{\boldsymbol{A}}\boldsymbol{1})$ is the degree matrix of $\tilde{\boldsymbol{A}}$, $\boldsymbol{W}$ is the trainable weight matrix, and σ is the activation function.

Suppose that the maximum range of road segments considered is M hops. The feature extraction network for the mth hop range can be denoted as GCN$_m$, $0 \le m \le M$. Therefore, if $m = 0$, only features of individual road segment are considered, and GCN$_0$ can be denoted as

$$\text{GCN}_0(\ddot{\boldsymbol{X}}) = \sigma(\boldsymbol{I}\ddot{\boldsymbol{X}}\boldsymbol{W}). \tag{19}$$

Inspired by random walk graph embeddings [64] and N-GCN [65], if $m > 0$, GCN$_m$ can be denoted as

$$\text{GCN}_m(\ddot{\boldsymbol{X}}) = \sigma\left(\left(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\right)^m \ddot{\boldsymbol{X}}\boldsymbol{W}_m\right). \tag{20}$$

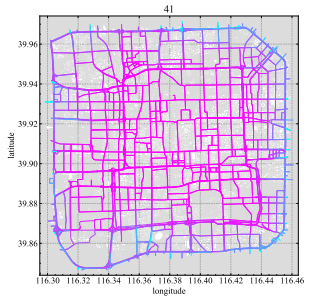


Fig. 6. $\tilde{\boldsymbol{M}}$ for road segments in Beijing, with $k = 50$.

Therefore, given the edge connection matrix $\boldsymbol{A}$ and graph signal $\ddot{\boldsymbol{X}}$ of the road network $\mathcal{R}$, the extracted multiscale static features are denoted as $\{\mathrm{GCN}_0(\ddot{\boldsymbol{X}}), \mathrm{GCN}_1(\ddot{\boldsymbol{X}}), \ldots, \mathrm{GCN}_K(\ddot{\boldsymbol{X}})\}$, with $\mathrm{GCN}_m(\ddot{\boldsymbol{X}})$ defined as

$$\mathrm{GCN}_m(\ddot{\boldsymbol{X}}) = \begin{cases} \sigma(\boldsymbol{I}\ddot{\boldsymbol{X}}\boldsymbol{W}_0), & m = 0 \\ \sigma\left(\left(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\right)^m \ddot{\boldsymbol{X}}\boldsymbol{W}_m\right), & \text{Otherwise.} \end{cases} \tag{21}$$

The multiscale graph neural network is built upon the CSTGCN module and is applied to learn inter-intersection correlation. F_3 can be expressed as

$$\dddot{\boldsymbol{X}} = F_3(\ddot{\boldsymbol{X}}_{t-P+1:t}) = \sum_{m=0}^{M-1} \mathrm{GCN}_m(\ddot{\boldsymbol{X}}). \tag{22}$$

From the perspective of representative learning [66], by going deeper, it is beneficial to extract compact and semantic-rich features that are beneficial to predict the complex and nonlinear correlations. Inspired by residual networks [67], we stack a multiscale graph neural network module and a ReLU activation layer into a residual CSTGCN unit, as shown in Fig. 5, which can be denoted as follows:

$$\dddot{\boldsymbol{X}}^{(l+1)} = \mathrm{ReLU}(F(\mathrm{ReLU}(F(\ddot{\boldsymbol{X}}^{(l)})))) + \ddot{\boldsymbol{X}}^{(l)} \tag{23}$$

where $\ddot{\boldsymbol{X}}^{(l)}$ is the graph signal on the lth layer and F means the multiscale CSTGCN function. By applying residual connection, it is possible to stack multiple layers without gradient explosion and performance degradation problems and learn citywide spatial–temporal correlations.

D. Marginal Loss

Considering the restricted area of the road network, correlations of marginal road segments are truncated and destroyed. Fig. 6 shows the road network in Beijing. In the constructed DTKG, road segments in the middle have a complete knowledge graph. However, road segments in the margin have only a partial knowledge graph, and its prediction is comparatively difficult. Therefore, different weights should be assigned to road segments, i.e., higher weight in the middle and lower weight in the margin. Inspired by the diffusion process on a directed graph [24], we exploit a similar approach to measure the marginalization of a road segment.

Given the edge connection matrix $\boldsymbol{A}$ of a graph, the number of paths of length k between edges can be denoted by $\boldsymbol{A}^k$. Furthermore, a vector $\boldsymbol{M} \in \{0, 1\}^{1\times N}$ denotes the marginalization situations of all nodes, which is defined as

$$\boldsymbol{M}[i] = \begin{cases} 0, & \text{if } |D_i| = 1 \\ 1, & \text{Otherwise.} \end{cases} \tag{24}$$

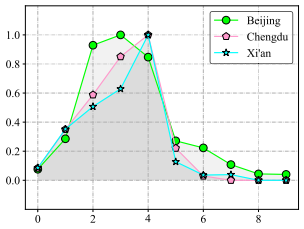


Fig. 7. Distribution of $|D_i|$.

To this end, the marginalization of edges $\mathcal{E}$ can be denoted as

$$\tilde{\boldsymbol{M}} = \boldsymbol{M}\sum_{i=1}^{k}\left(\left(\mathrm{diag}\{\boldsymbol{A}\cdot\boldsymbol{1}_{N\times N}\}\right)^{-1}\boldsymbol{A}\right)^i \tag{25}$$

where $\boldsymbol{1}$ is the $N \times N$ all-one matrix and $\mathrm{diag}\{\boldsymbol{A}\}$ means the diagonal matrix of $\boldsymbol{A}$.

CSTGCN can be trained end-to-end via backpropagation by minimizing the mean absolute error (MAE) between predicted values and observations, taking the marginalization into account, which is denoted as

$$\mathcal{L} = \frac{1}{P}\sum_{o_{i,j}\in\mathcal{E}} \left\|\hat{\boldsymbol{X}}^{o_{i,j}}_{t+1:t+Q} - \boldsymbol{X}^{o_{i,j}}_{t+1:t+Q}\right\|_2 \tilde{\boldsymbol{M}}[i] \tag{26}$$

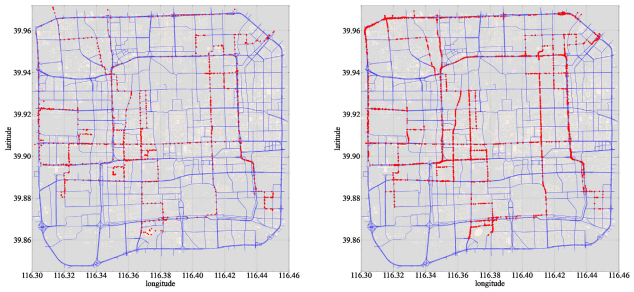
where $\boldsymbol{X}^{o_{i,j}}_{t+1:t+Q}$ is the ground truth and $\hat{\boldsymbol{X}}^{o_{i,j}}_{t+1:t+Q}$ is the predicted value for road segment $o_{i,j}$.

E. Scalability

For a real-world road network, each road segment has a direct connection with only a few road segments. For example, the maximum in-neighbor road segments in Beijing, Xi'an, and Chengdu, China, is 9, i.e., $\max_{o_{i,j}\in\mathcal{E}} |D_i| = 10$, as shown in Fig. 7. Thus, the maximum number of convolution kernels is 10 and there exist only ten kernels in each CSTGCN unit. Consequently, our proposed CSTGCN module is applicable and practical for possible applications in traffic areas.

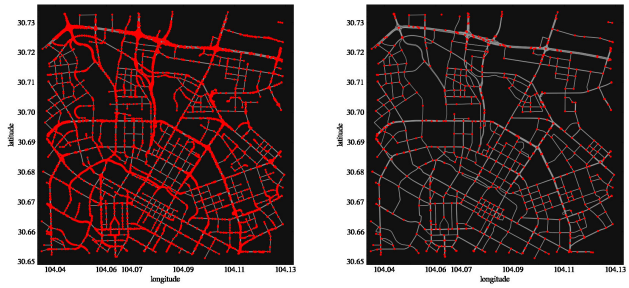
The transportation network of any city can be decomposed into road intersections and road segments, which can be expressed as a road network $\mathcal{R} = (\mathcal{M}, \mathcal{E})$. Based on $\mathcal{R}$, the STDs are divided into road segment, intra-intersection, and inter-intersection correlation, i.e., F_1, F_2, and F_3. We subtly design MRSL, CSTGCN, and multiscale residual networks for implementing F_1, F_2, and F_3, respectively.

Due to the above considerations and implementations, it is possible to unearth the essential relationship between traffic and fine-grained road topology, i.e., how the traffic is affected by road segments, intersections, and nearby road topology. Trained with data from a city, the model learns how the traffic of a road segment is affected by the traffic of adjacent road segments, by edge attributes, and by the fine-grained road topology, i.e., learning the essential dependencies between the traffic and the road topology. Therefore, when encountering with a new city without training data, it is convenient to obtain fine-grained road topology from map providers such as Google, Baidu, or OpenStreetMap. Therefore, the well-trained model can be directly applied to predict the traffic without the need for training or fine-tuning. Therefore, CSTGCN makes it possible to predict the traffic of any city with a unified network model.



(a) (b)

Fig. 8. Map matching. (a) Original trajectory of a taxi in a day in Beijing, China. (b) Corresponding trajectory after preprocessing and map matching.



(a) (b)

Fig. 9. Road network consolidation and simplification. (a) Original road network downloaded from OpenStreetMap in Chengdu, China. (b) Road networks after consolidation and simplification.

V. Datasets

Three real-world trajectory datasets are adopted to demonstrate the feasibility and advantages of DTSGCN, which are introduced as follows.

1) *TaxiBJ*: This dataset consists of GPS trajectories of more than 20 000 taxis of Beijing, China, from September 1 to November 30, 2016.
2) *DiDiXA and DiDiCD*[2]: These two datasets consist of the floating-car data collected by DiDi from October 1 to November 30, 2018, which belong to Xi'an and Chengdu, China.

Fig. 8(a) shows the trajectory of a taxi in TaxiBj. As mentioned above, we need to obtain a dataset that could record the traffic of all road segments based on a trajectory dataset, which mainly consists of three steps.

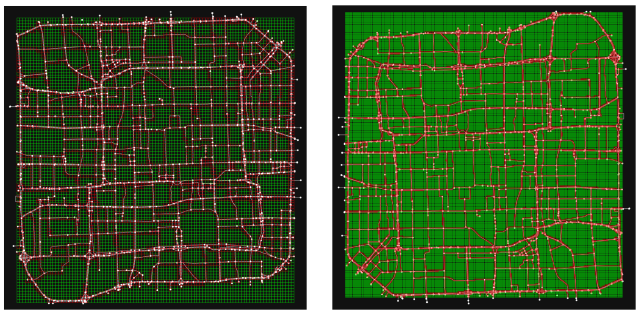
1) *Road Network Construction:* For each dataset, the corresponding road network is collected from OpenStreetMap.[3] Road networks are described by end (middle) points, road segments (which are connections between points), and fine-grained attributes of road segments. The original downloaded road network contains rich features describing complex intersections and traffic circles, resulting in a cluster of graph nodes where there is just one true intersection as we would think of it in transportation or urban design, as shown in Fig. 9(a). We consolidate nearby intersections (with distance tolerance 50 m), simplify the connections, and rebuild the graph's topology to reconnect edges to the newly consolidated nodes, as shown in Fig. 9(b). Taking the road network in Beijing, China, as an example, the number of nodes for the original downloaded road network is 65 327 nodes. After consolidation and simplification, the number of nodes is 3246.
2) *Trajectory Process:* The original downloaded road network contains all types of road segments, including residential ways, which could not be considered, due to data sparsity. Therefore, for each trajectory, as shown in Fig. 8(a), the following preprocessing procedures are conducted: 1) remove the points that locate on the residential way; 2) delete the abnormal points; and 3) remove the trajectory that is less than ten points, making sure that the minimum number of points for matching is at least ten continuous points.
3) *Map Matching:* For each trajectory, after data cleansing and preprocessing, we exploit a map-matching algorithm presented in [68] to map it to each road segment, minimizing the GPS positioning error. To this end, we have successfully obtained a dataset that could cover the traffic of all road segments. The matched trajectory for the original GPS points shown in Fig. 8(a) is shown in Fig. 8(b).

However, during the trajectory process, the points that locate in the residential way (referred to as residential points) shall be removed, which is quite time-consuming. Taking TaxiBJ as an example, there exist 900 000 000 points per day. For each point, it requires computing the distance to the nearest road segment and applies this distance to determine whether the point should be kept. Denote the set of road segments on the road network $\mathcal{R}$ is $\mathcal{E}$. Therefore, the complexity of removing residential points is with $O(N_{\text{total}}|\mathcal{E}|)$, where N_{total} is the total number of GPS points. However, since N_{total} is extremely large, and thus, it is quite time-consuming. Therefore, we propose a grid-based point search (GBPS) algorithm to improve the computation performance.

GBPS first partitions the considered area into $N_{\text{lat}} \times N_{\text{lon}}$ grids based on the latitude and longitude, where a grid (i, j) that lies at the ith row and the jth column denotes a region, $0 \le i < N_{\text{lat}}$, and $0 \le j < N_{\text{lon}}$. For the considered area, denote the position of the bottom-left corner as $P^{\min} = (P^{\min}_{\text{lat}}, P^{\min}_{\text{lon}})$ and the position of the top-right corner as $P^{\max} = (P^{\max}_{\text{lat}}, P^{\max}_{\text{lon}})$, where $P^{\min}_{\text{lat}}$ and $P^{\max}_{\text{lat}}$ are the latitude and $P^{\min}_{\text{lon}}$ and $P^{\max}_{\text{lon}}$ are the longitude. Therefore, for a grid (i, j), the corresponding area is $(P^{\min}_{\text{lat}} + (((P^{\max}_{\text{lat}} - P^{\min}_{\text{lat}}))/N_{\text{lat}}) * i, P^{\min}_{\text{lon}} + (((P^{\max}_{\text{lon}} - P^{\min}_{\text{lon}}))/N_{\text{lon}}) * j)$ to $(P^{\min}_{\text{lat}} + (((P^{\max}_{\text{lat}} - P^{\min}_{\text{lat}}))/N_{\text{lat}}) * (i + 1), P^{\min}_{\text{lon}} + (((P^{\max}_{\text{lon}} - P^{\min}_{\text{lon}}))/N_{\text{lon}}) * (j + 1))$. Meanwhile, for each grid (i, j), a Boolean value $B(i, j)$ is used to denote its state, where $B(i, j)$ is 1 if this grid contains a residential way, otherwise 0. For each point $p = (P_{\text{lat}}, P_{\text{lon}})$, it will be removed if it lies in the area of a grid whose $B(i, j)$ is 1.

The pseudocode of the GBPS algorithm is shown in Algorithm 1, which consists of the initialization phase and execution phase. The first phase constructs $B(i, j)$, which is executed only once for each dataset. The second phase is executed for each GPS point. GBPS does not require finding the nearest road segments for each point, thus with

[2]Data are open-sourced at https://gaia.didichuxing.com
[3]https://www.openstreetmap.org/



(a) (b)

Fig. 10. Visualization of $B(i, j)$. The grid is green if $B(i, j) = 1$, otherwise red. (a) $B(i, j)$ with $N_{\text{lat}} = 100$ and $N_{\text{lon}} = 100$ (100 × 100 grids). (b) $B(i, j)$ with $N_{\text{lat}} = 250$ and $N_{\text{lon}} = 250$ (250 × 250 grids).

Algorithm 1 GBPS Algorithm

function 1: INITIALIZATION PHASE
  This algorithm is executed only once for each dataset
  for $i = 0; i < N_{\text{lat}}; i++$ **do**
    for $j = 0; j < N_{lon}; j++$ **do**
      if grid (i,j) contains a residential way **then**
        $B(i, j) \leftarrow 1$
      else
        $B(i, j) \leftarrow 0$
      end if
    end for
  end for
end function
function 2: EXECUTION PHASE
  This algorithm is executed for each point
  for each point $p = (P_{lat}, P_{lon})$ **do**
    $i = \lfloor \frac{P_{lat} N_{lat}}{(P_{lat}^{\max} - P_{lat}^{\min})} \rfloor$, $j = \lfloor \frac{P_{lon} N_{lon}}{(P_{lon}^{\max} - P_{lon}^{\min})} \rfloor$
    if $B(i, j) == 1$ **then**
      Remove point p
    else
      Keep point p
    end if
  end for
end function

TABLE I
COMPARISON OF THREE REAL-WORLD DATASETS

Datasets	TaxiBJ	DiDiXA	DiDiCD
Data type	Taix	Ride-hailing car	Ride-hailing car
Location	Beijing	Xi'an	Chengdu
#Road segments	3246	1002	1795
Total road length (km)	1513.36	513.88	729.86
#Intersections	1250	422	748
Area (km^2)	156.92	63.63	64.00
# GPS points	6.67×10^8	6.34×10^8	8.68×10^8
Road attributes	(Length, type, speed limit, direction, lanes)		
Time interval	∼30 Sec	∼10 Sec	∼10 Sec

the complexity of $O(N_{\text{total}})$. $B(i, j)$ is visualized in Fig. 10, where the grid is green if $B(i, j) = 1$, otherwise red. It can be concluded from these figures that a smaller grid would introduce a higher accuracy. Besides, the GBPS algorithm can be applied to remove abnormal points, which lie outside of the road segments.

Detailed information and comparisons between these three datasets are shown in Table I. The missing values in the datasets are filled by linear interpolation. As for multistep traffic prediction, we adopt 1-h historical data to predict the next hour's data. In each dataset, 70% are treated as training sets, 10% are treated as validation sets, and 20% are treated as test sets.

VI. EXPERIMENT

In this section, we conduct extensive experiments based on three real-world datasets from different cities, focused on the following three questions.

1) *Accuracy:* Does our proposed CSTGCN outperform existing approaches in traffic prediction?
2) *Essential STDs:* Can CSTGCN learn the actual relationship between traffic and the road network?
3) *Effects of Hyperparameters:* How do the hyperparameters of DTSGCN affect the performance of the prediction task?

A. Baselines and Metrics

To validate the advantages and effectiveness, DTSGCN is compared with other benchmark models, including the traditional time-series analysis approaches (i.e., vector autoregressive (VAR) and ARIMA), deep learning [i.e., gated recurrent unit (GRU), fully connected long short-term memory network (LSTM) (FC-LSTM), and stacked autoencoder (SAE)], and GCN models (i.e., DCRNN, Graph WaveNet, STSGCN, and AGCRN), which are introduced as follows.

1) *VAR [69]:* It captures the pairwise relationships among all traffic speeds of all road segments, which is computation-extensive due to a large number of parameters.
2) *ARIMA [15]:* It is a well-known model for understanding and predicting future values in a time series.
3) *GRU [70]:* Network consists of multiple GRUs.
4) *FC-LSTM [71]:* RNN with FC-LSTM hidden units.
5) *SAE [72]:* It is trained by a layerwise greedy fashion.
6) *DCRNN [24]:* It is a GCN-based approach that models the traffic flow as a diffusion process.
7) *Graph WaveNet [26]:* It exploits a self-adaptive adjacency matrix to preserve hidden spatial dependencies and stacked dilated casual convolutions to capture temporal dependencies.
8) *STSGCN [30]:* It captures spatial and temporal correlations simultaneously.
9) *AGCRN [28]:* It learns node-specific patterns for traffic forecasting without the need for a predefined graph.

Note that a well-trained CSTGCN (trained with only a source dataset) can directly predict the traffic of any city (without any training using data from the target dataset). However, most existing transferring learning works must fine-tune their model in the target dataset. To present a fair comparison, we extract three days' data from the target dataset as the fine-tuning dataset. As for our CSTGCN, it is not fine-tuned over the target datasets (only trained with source

data). CSTGCN is further compared with GRU, FC-LSTM, RegionTrans, and AGCRN to demonstrate that it could learn the actual relationship between traffic and the fine-grained road topology, which are introduced as follows.

1) *GRU:* Train it on the source city data and fine-tune it with data from the target city.
2) *FC-LSTM:* Train it on the source city data and fine-tune it with data from the target city.
3) *RegionTrans [53]:* It transfers knowledge from one city to another city for traffic flow prediction. Since we do not have auxiliary data, we compare the S-Match of RegionTrans. For each region in the target city, RegionTrans uses short-period data to calculate the linear similarity value with each region in the source city.
4) *AGCRN (FT):* Train AGCRN on data from the source city and then fine-tune with data from the target city.

We adopt three commonly used metrics in traffic forecasting to evaluate the performance, i.e., MAE, root-mean-square error (RMSE), and mean absolute percentage error (MAPE).

B. Implementation Details

CSTGCN stacks several residual CSTGCN units to capture the actual relationship between traffic and fine-grained road network. We believe that different STDs are generated by different traffic patterns and the road network would also affect traffic. We adopt a multistep learning rate, i.e., the learning rate is decayed by 0.2 once the epoch reaches one of [10, 30, 50, 70], with an initial learning rate of 0.02. For all the deep learning models, we optimize them with the Adam optimizer for 100 epochs.

All the deep-learning-based models, including our CSTGCN, are implemented in Python with Pytorch 1.2.0 and executed on a server with one GeForce RTX 2080 Ti GPU card. The best parameters for all deep learning models are chosen through a carefully parameter-tuning process on the validation set.

As shown in Fig. 7, there exist a few edges whose in-neighbor road segments exceed 7, leading to insufficient training data for these edges. To handle this problem, we apply a fixed number of kernels, i.e., $K = 7$. As for the edges whose in-neighbor edges are greater than 7, traditional GCN approaches are adopted to enhance scalability and efficiency, which is a modification over CSTGCN. Consequently, (10) can be modified as

$$\ddot{\boldsymbol{X}}[o_{i,j},:] \triangleq \frac{1}{P|D_i|}\sum_{t'=t-P+1}^{t}\left(\sum_{l=0}^{|D_i|}\hat{\boldsymbol{X}}^{S+E}_{t',D_i[l]}\right)\boldsymbol{W}_{3D}[t']. \quad (27)$$

Therefore, all edges whose in-neighbor road segments are greater than 7 share a common convolution kernel $\boldsymbol{W}_{3D}$, which could also capture the spatial and temporal correlations simultaneously.

As for the marginal loss, we adopt $k = 50$, and the visualization over the edges in the three datasets is shown in Fig. 6.

C. Accuracy

Table II shows the performance of CSTGCN on TaxiBJ of 15-min, 30-min, 45-min, and 1-h ahead forecasting. Table III shows the performance of CSTGCN of 15-min, 30-min, and 1-h ahead forecasting on DiDiXA and DiDiCD. After a careful analysis of the results shown in Tables II and III, we can make the following observations.

1) Traditional methods achieve poor prediction performance since they do not consider the traffic conditions of the immediate past and the spatial–temporal correlations. Therefore, they are not sufficient to handle the dynamic changes or spatial dependencies of traffic.
2) Deep learning-based models (GRU, FC-LSTM, and SAE) perform better than traditional methods since they can model more complex and nonlinear traffic. SAE needs to mess up the spatial–temporal data to form a long series, which breaks the STDs and makes it harder to capture the correlations. Consequently, the temporal neural network-based methods FC-LSTM and GRU achieve a comparatively better performance than SAE.
3) GCN-based models (DCRNN, STSGCN, Graph WaveNet, AGCRN, and our proposed CSTGCN) perform better than traditional methods and deep learning methods since they model the sensors as a graph and could handle the nonregular data structures. Among the GCN-based models, STSGCN performs poorly. STSGCN connects individual spatial graphs of adjacent time steps into one graph. Taking the TaxiBJ as an example, it has 3246 edges, as shown in Table I, which means that the adjacency matrix of the spatial graph for each time step is 3246×3246. If multiple spatial graphs of different time steps are concatenated, a larger adjacency matrix is produced, and therefore, it suffers from long-sequence temporal dependencies, which are hard to capture and approximate.
4) Our proposed CSTGCN outperforms all the baseline models in all settings. CSTGCN proposes a new insight into STD and divides STD into three kinds of correlations, i.e., road segment, intra-intersection, and inter-intersection correlation, which captures the microcosmic, middle, and macroscopic dependencies between the traffic and the road network, respectively. We then propose MRSL, CSTGCN, and multiscale residual graph neural networks for implementing the three correlations, capturing the essential relationship between the traffic and the road network. Therefore, it achieves the best performance, demonstrating the effectiveness and advantages of our proposed approach.

D. Essential STDs

To demonstrate that our proposed CSTGCN could learn the essential STDs, we apply the model trained on one dataset for prediction on another dataset, without any fine-tuning. Using this experiment, if our proposed CSTGCN has successfully learned the actual relationship between traffic and the road network, the model trained only on the source data could achieve good prediction performance on the task of

TABLE II

PERFORMANCE OF DIFFERENT MODELS ON TAXIBJ, WHERE A SMALLER VALUE MEANS BETTER PERFORMANCE. RESULTS IN BOLD ARE THE BEST RESULT OF EXISTING MODELS. CSTGCN ACHIEVES THE BEST PERFORMANCE WITH ALL THREE METRICS FOR ALL FORECASTING HORIZONS

Model		TaxiBJ (15 / 30 / 45 / 60 min)		
		MAE	RMSE	MAPE(%)
Traditional Approaches	VAR [70]	2.71	3.80	38.02
	ARIMA [15]	1.92 / 2.06 / 2.15 / 2.22	2.71 / 2.86 / 2.96 / 3.04	42.15 / 38.25 / 39.92 / 41.24
Deep learning methods	GRU [71]	1.06 / 1.81 / 1.97 / 2.00	1.43 / 2.34 / 2.52 / 2.56	16.33 / 30.68 / 33.79 / 34.19
	FC-LSTM [72]	1.09 / 1.79 / 1.95 / 2.08	1.46 / 2.31 / 2.50 / 2.63	17.28 / 30.16 / 33.49 / 35.85
	SAE [73]	1.17 / 1.92 / 2.11 / 2.25	1.53 / 2.43 / 2.64 / 2.78	18.53 / 32.54 / 36.34 / 38.98
GCN models	DCRNN [24]	0.85 / 1.32 / 1.41 / 1.47	1.22 / 1.87 / 2.00 / 2.08	11.46 / 18.64 / 20.25 / 21.27
	STSGCN [30]	1.06 / 1.27 / 1.36 / 1.42	1.70 / 1.94 / 2.05 / 2.12	11.47 / 17.88 / 18.90 / 19.63
	Graph WaveNet [26]	0.88 / 1.39 / 1.50 / 1.58	1.26 / 1.97 / 2.11 / 2.22	11.76 / 19.43 / 21.24 / 22.37
	AGCRN [28]	0.83 / 1.28 / 1.33 / 1.41	1.18 / 1.75 / 1.88 / 1.92	11.95 / 17.81 / 19.61 / 20.06
Our proposed approach	**CSTGCN**	**0.78 / 1.16 / 1.25 / 1.34**	**1.16 / 1.58 / 1.77 / 1.81**	**11.32 / 13.19 / 18.67/ 19.45**

TABLE III

PERFORMANCE OF DIFFERENT MODELS ON DIDIXA AND DIDICD, RESULTS IN BOLD ARE THE BEST PERFORMANCE ACHIEVED BY BASELINES. A SMALLER VALUE MEANS BETTER PERFORMANCE. OUR PROPOSED CSTGCN ACHIEVES THE BEST PERFORMANCE

Model	DiDiXA (15 / 30 / 60 min)			DiDiCD (15 / 30 / 60 min)		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
VAR [70]	2.01	2.78	43.02	1.68	2.51	39.49
ARIMA [15]	1.63 / 1.84 / 1.99	2.01 / 2.31 / 2.40	32.00 / 37.45 / 39.02	1.42 / 1.68 / 1.77	1.92 / 2.13 / 2.25	27.00 / 32.45 / 36.02
GRU [71]	0.90 / 1.57 / 1.98	1.27 / 2.04 / 2.43	13.89 / 25.59 / 32.71	0.82 / 1.07 / 1.31	1.25 / 1.55 / 1.88	11.83 / 17.82 / 22.79
FC-LSTM [72]	0.93 / 1.52 / 1.91	1.29 / 2.00 / 2.38	12.56 / 20.03 / 25.17	0.82 / 1.16 / 1.32	1.25 / 1.61 / 1.88	12.60 / 17.55 / 19.90
SAE [73]	1.07 / 1.70 / 2.18	1.40 / 2.11 / 2.60	14.25 / 24.66 / 31.45	0.85 / 1.26 / 1.43	1.33 / 1.79 / 1.91	13.24 / 20.13 / 23.29
DCRNN [24]	0.87 / 1.21 / 1.23	1.29 / 1.71 / 1.89	11.63 / 16.33 / 17.46	0.71 / 1.03 / 1.17	1.10 / 1.55 / 1.63	9.30 / 13.22 / 14.28
STSGCN [30]	1.04 / 1.52 / 1.73	1.43 / 1.90 / 2.03	12.01 / 17.33 / 19.13	0.76 / 1.22 / 1.29	1.20 / 1.68 / 1.83	10.60 / 17.31 / 19.43
Graph WaveNet [26]	0.88 / 1.23 / 1.34	1.31 / 1.83 / 1.89	11.53 / 16.45 / 17.59	0.73 / 1.12 / 1.27	1.10 / 1.65 / 1.83	9.60 / 15.22 / 17.28
AGCRN [28]	0.87 / 1.17 / 1.26	1.26 / 1.73 / 1.83	11.55 / 16.01 / 17.19	0.69 / 0.95 / 1.01	1.02 / 1.43 / 1.50	8.91 / 12.98 / 13.97
Our approach	**0.74 / 1.01 / 1.09**	**1.15 / 1.45 / 1.59**	**10.46 / 14.13 / 14.85**	**0.65 / 0.89 / 0.94**	**0.98 / 1.38 / 1.46**	**8.65 / 12.15 / 12.66**

TABLE IV

CAPABILITY OF CSTGCN TO LEARN THE ACTUAL RELATIONSHIP BETWEEN TRAFFIC AND THE FINE-GRAINED ROAD NETWORK. THE RESULTS ARE AVERAGED OVER ALL HORIZONS. OUR PROPOSED CSTGCN ACHIEVES THE BEST PERFORMANCE OVERALL METRICS IN ALL HORIZONS

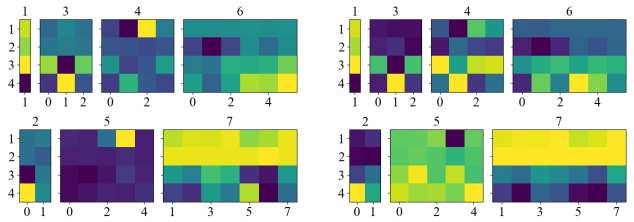
Model	TaxiBJ → DiDiXA			TaxiBJ → DiDiCD			DiDiCD → DiDiXA			DiDiCD → TaxiBJ		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
GRU	1.92	2.34	41.32	1.63	2.03	39.23	1.41	1.89	32.52	1.72	2.41	38.32
FC-LSTM	1.89	2.14	40.23	1.59	1.97	38.37	1.49	2.19	34.58	1.76	2.31	37.23
Graphtrans	1.56	1.98	31.43	1.43	1.69	30.23	1.35	1.77	29.10	1.54	2.28	30.41
AGCRN (FT)	1.63	2.05	36.54	1.51	1.80	33.46	1.41	1.97	30.18	1.67	2.32	37.32
Our approach	**1.12**	**1.67**	**19.97**	**1.13**	**1.46**	**17.01**	**1.19**	**1.67**	**18.87**	**1.31**	**1.87**	**22.34**

prediction in the target data. Otherwise, it performs poorly in the target data. In this way, this experiment could reasonably demonstrate whether our proposed CSTGCN can capture the actual relationship between traffic and the road network.

Considering the number of road segments for each dataset, i.e., 3246, 1795, and 1002 for TaxiBj, DiDiCD, and DiDiXA, respectively, we exploit the models trained on TaxiBj and DiDiCD for prediction on the remaining datasets. Therefore, we get four combinations, as shown in Table IV. Due to the setting of such experiments, most GCN models are not applicable since the input size of these methods is not changeable. We select Graphtrans, which transfers knowledge from one city to another for traffic flow prediction, AGCRN (FT), which is trained and tested on the same number of edges but with different datasets, i.e., trained on 1002 edges in Beijing, China, and tested on 1002 edges in DiDiXA, GRU, and FC-LSTM as the baselines to conduct experiments. Furthermore, we extract data for three days from the target city to fine-tune Graphtrans, AGCRN (FT), GRU, and FC-LSTM. However, our proposed CSTGCN is not fine-tuned on the target data, i.e., trained with only the source data.

Table IV shows the performance of CSTGCN. The input of GRU and FC-LSTM contains only the historical observations of p steps, which ignores the spatial correlation and could only capture the short-term temporal correlations. Therefore, GRU and FC-LSTM achieve poor performance. Graphtrans and AGCRN (FT), which are trained on a source city and fine-tuned on the target city, achieve a better performance than GRU and FC-LSTM but a worse performance than CSTGCN. Our proposed CSTGCN learns the essential STDs and captures the actual relationship between the traffic and the road network. Therefore, a well-trained CSTGCN model exploits the learned knowledge to predict future traffic directly in the target city.

To further validate that edges with different in-neighbor edges exert different mutual relationships, we visualize convolutional kernels $\{\boldsymbol{W}^1_{3D}, \boldsymbol{W}^2_{3D}, \ldots, \boldsymbol{W}^7_{3D}\}$, as shown Fig. 11. Fig. 11(a) shows the weight activations for seven kernels, and



(a) (b)

Fig. 11. Visualization of convolutional kernels $\{\boldsymbol{W}_{3D}^1, \boldsymbol{W}_{3D}^2, \ldots, \boldsymbol{W}_{3D}^7\}$. Different kernels are activated differently, verifying the rationality of intra-intersection correlation and different intersections (e.g., four-way and three-way) should be treated differently. (a) $t+1$ time step. (b) $t+3$ time step.

each one corresponds to a kind of edge. We can observe that the following conditions hold.

1) It verifies that traffic at different intersections needs to be considered separately, i.e., the three- and four-way intersections shall be treated separately, demonstrating the rationality of the connection-aware design of CST-GCN. In traditional GCN, the feature of each edge is a weighted average of the features of in-neighbor edges, which ignores the differences between intersections. However, in CSTGCN, edges are treated differently, i.e., edges with a different number of in-neighbor edges are distinguished and a convolution kernel is applied to aggregate features for each kind of edge.
2) The activations of different time steps are also different, demonstrating that the influence of intersections on future traffic is nonlinear and complex.

E. Effects of Hyperparameters

Extensive ablation experiments are conducted to evaluate the performance of modules proposed in this article, which are purposed to solve the following questions.

1) Are the edge attributes helpful for the traffic prediction problem? What is the gain of the MRSL compared with other data fusion methods?
2) Is multiscale GCN helpful for capturing inter-intersection correlation? What is the optimal number of scales for traffic prediction?
3) Can the residual networks produce higher prediction accuracy? What is the optimal number of layers?
4) Is CSTGCN outperforms the weighted average GCN? What is the gain of it over the weighted average GCN?

In the following, we conduct extensive experiments on TaxiBJ to answer the above questions.

The graph signal $\boldsymbol{X}_t$ and edge attributes $\boldsymbol{H}$ should be fused to capture the essential STDs. To better evaluate the performance, we have adopted the following variants.

1) *CSTGCN:* CSTGCN model with fine-grained edge attributes, which applies MRSL to include both the traffic signals $\boldsymbol{X}_{t-P+1:t}$ and edge attributes $\boldsymbol{H}$.
2) *CSTGCN-No:* CSTGCN model without fine-grained edge attributes, which applies FCN to encode the traffic signals $\boldsymbol{X}_{t-P+1:t}$.
3) *CSTGCN-Concat:* CSTGCN model with fine-grained edge attributes. However, it eliminates MRSL and applies two FCN to extract features from the traffic signals $\boldsymbol{X}_{t-P+1:t}$ and edge attributes $\boldsymbol{H}$ and then concatenate these two features.
4) *CSTGCN-Plus:* It modifies CSTGCN-Concat by adding these two features rather than concatenation.

TABLE V

PERFORMANCE COMPARISON FOR DIFFERENT CSTGCN VARIANTS. THE RESULTS ARE AVERAGED OVER ALL HORIZONS

Methods	MAE	RMSE	MAPE(%)
CSTGCN	**1.18**	**1.59**	**14.97**
CSTGCN-No	1.73	2.11	28.54
CSTGCN-Concat	1.21	1.63	17.43
CSTGCN-Plus	1.22	1.68	18.13

TABLE VI

PERFORMANCE COMPARISON FOR CSTGCN WITH DIFFERENT SCALES OF GCN (M). THE RESULTS ARE AVERAGED OVER ALL HORIZONS

M	MAE	RMSE	MAPE(%)
1	1.32	1.78	19.13
2	1.26	1.71	16.32
3	**1.18**	**1.59**	**14.97**
4	1.24	1.65	15.64
5	1.36	1.83	20.32

TABLE VII

PERFORMANCE COMPARISON BETWEEN CSTGCN AND WEIGHTED AVERAGE GCN. THE RESULTS ARE AVERAGED OVER ALL HORIZONS

	MAE	RMSE	MAPE(%)
Connection-aware spatial-temporal GCN	**1.18**	**1.59**	**14.97**
Weighted average GCN	1.35	1.68	19.13

Table V shows the results of these CSTGCN variants. CSTGCN-No achieves the worse performance, which demonstrates that fine-grained edge attributes are beneficial for traffic prediction. The edge attributes are significantly helpful for traffic prediction, which improves MAE by 0.55. CSTGCN achieves a better performance than CSTGCN-Concat and CSTGCN-Plus, which improves 0.35 MAE over other data fusion methods. The results demonstrate that our proposed meta-learning-based feature extraction is more suitable for CSTGCN.

Next, extensive experiments are conducted to evaluate the performance of multiscale GCN, and the results are shown in Table VI. Multiscale GCN makes it possible to aggregate the different features that are extracted by different weights from distant road segments. Therefore, multiscale GCN would help capture inter-intersection correlation. CSTGCN achieves the best performance when $M = 3$.

Multiple residual CSTGCN units are stacked to capture the nonlinear and complex correlations. We evaluate CSTGCN under different layers, ranging from 2 to 20, with a step size of 2. We conduct ten experiments under each setting and the results are expressed by the boxplot shown in Fig. 12. CSTGCN achieves the best performance with 14 layers.

To evaluate the performance of CSTGCN, we have replaced it with weighted average GCN. The results are shown in Table VII, which indicates that CSTGCN achieves a better performance than weighted average GCN.

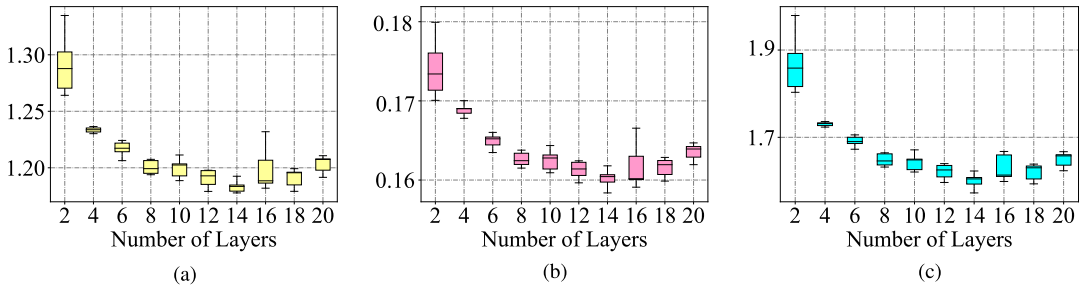


Fig. 12. Performance of CSTGCN with different layers, achieving the best with 14 layers. (a) MAE. (b) MAPE (%). (c) RMSE.

VII. Conclusion and Future Work

This article proposes a joint knowledge- and data-driven mechanism to capture the essential relationship between the traffic and the road network. A novel insight over STDs is first proposed, i.e., dividing it into three kinds of correlations, i.e., road segment, intra-intersection, and inter-intersection correlation, which capture the microcosmic, middle, and macroscopic dependencies between the traffic and the road network, respectively. Then, MRSL, CSTGCN, and multiscale residual graph neural networks are proposed for implementing these three kinds of correlations. Finally, datasets that could cover all the road segments are constructed from real-world datasets to demonstrate the performance and advantages of our proposed method. Extensive experiments on three real-world datasets demonstrate that CSTGCN achieves a better prediction accuracy and could capture the actual relationship between traffic and the road topology, compared with nine baselines. In the future, we will take more factors that affect traffic flows into account and extend our framework to a much broader set of urban computing tasks.

References

[1] Y. Guo, D. Zhou, P. Li, C. Li, and J. Cao, "Context-aware poly(A) signal prediction model via deep spatial–temporal neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 12, 2022, doi: 10.1109/TNNLS.2022.3226301.

[2] L. Lin, J. Li, F. Chen, J. Ye, and J. Huai, "Road traffic speed prediction: A probabilistic model fusing multi-source data," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1310–1323, Jul. 2017.

[3] Y. Zhang, Y. Li, X. Zhou, X. Kong, and J. Luo, "Curb-GAN: Conditional urban traffic estimation through spatio-temporal generative adversarial networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 842–852.

[4] G. Luo et al., "Software-defined cooperative data sharing in edge computing assisted 5G-VANET," *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 1212–1229, Mar. 2021.

[5] G. Luo et al., "Cooperative vehicular content distribution in edge computing assisted 5G-VANET," *China Commun.*, vol. 15, no. 7, pp. 1–17, 2018.

[6] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial–temporal graph data for traffic forecasting," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5415–5428, Nov. 2022.

[7] Z. Pan et al., "Spatio-temporal meta learning for urban traffic prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1462–1476, Mar. 2022.

[8] R. Huang, C. Huang, Y. Liu, G. Dai, and W. Kong, "LSGCN: Long short-term traffic prediction with graph convolutional networks," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 2355–2361.

[9] T. Li, J. Zhang, K. Bao, Y. Liang, Y. Li, and Y. Zheng, "AutoST: Efficient neural architecture search for spatio-temporal prediction," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 794–802.

[10] H. Hong et al., "HetETA: Heterogeneous information network embedding for estimating time of arrival," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 2444–2454.

[11] T. Anwar, C. Liu, H. L. Vu, M. S. Islam, and T. Sellis, "Capturing the spatiotemporal evolution in road traffic networks," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1426–1439, Aug. 2018.

[12] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 753–763.

[13] G. Luo, H. Zhang, Q. Yuan, J. Li, and F.-Y. Wang, "ClusterST: Clustering spatial–temporal network for traffic forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 706–717, Jan. 2023.

[14] G. Luo, H. Zhang, Q. Yuan, J. Li, and F.-Y. Wang, "ESTNet: Embedded spatial–temporal network for modeling traffic flow dynamics," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19201–19212, Oct. 2022.

[15] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: A general approach," *Transp. Res. B, Methodol.*, vol. 39, no. 2, pp. 141–167, 2005.

[16] H. Zhang, G. Luo, J. Li, and F.-Y. Wang, "C2FDA: Coarse-to-fine domain adaptation for traffic object detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12633–12647, Aug. 2022.

[17] J. Li et al., "An end-to-end load balancer based on deep learning for vehicular network traffic control," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 953–966, Feb. 2019.

[18] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[19] J. Ye, J. Zhao, K. Ye, and C. Xu, "How to build a graph-based deep learning architecture in traffic domain: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 3904–3924, May 2022.

[20] R. Barnes, S. Buthpitiya, J. Cook, A. Fabrikant, A. Tomkins, and F. Xu, "BusTr: Predicting bus travel times from real-time traffic," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 3243–3251.

[21] K. Pholsena, L. Pan, and Z. Zheng, "Mode decomposition based deep learning model for multi-section traffic prediction," *World Wide Web*, vol. 23, no. 4, pp. 2513–2527, Jul. 2020.

[22] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[23] Q. Zhang, J. Chang, G. Meng, S. Xiang, and C. Pan, "Spatio-temporal graph structure learning for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 1177–1185.

[24] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–14.

[25] Z. Diao, G. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial–temporal graph convolutional neural networks for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 890–897.

[26] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1907–1913.

[27] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI*, vol. 34, no. 1, 2020, pp. 1234–1241.

[28] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1–4.

[29] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, 2016, pp. 2659–2665.

[30] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial–temporal synchronous graph convolutional networks: A new framework for spatial–temporal network data forecasting," in *Proc. AAAI*, vol. 34, no. 1, 2020, pp. 914–921.

[31] H. Zhang, G. Luo, Y. Li, and F.-Y. Wang, "Parallel vision for intelligent transportation systems in metaverse: Challenges, solutions, and potential applications," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Dec. 20, 2022, doi: 10.1109/TSMC.2022.3228314.

[32] G. Luo, Q. Yuan, J. Li, S. Wang, and F. Yang, "Artificial intelligence powered mobile networks: From cognition to decision," *IEEE Netw.*, vol. 36, no. 3, pp. 136–144, May 2022.

[33] H. Zhang, Y. Tian, K. Wang, W. Zhang, and F.-Y. Wang, "Mask SSD: An effective single-stage approach to object instance segmentation," *IEEE Trans. Image Process.*, vol. 29, pp. 2078–2093, 2019.

[34] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3634–3640.

[35] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 95–104.

[36] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, "DeepSTN+: Context-aware spatial–temporal neural network for crowd flow prediction in metropolis," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 1020–1027.

[37] C. Chen, K. Li, S. G. Teo, X. Zou, and Z. Zeng, "Gated residual recurrent graph neural networks for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 485–492.

[38] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2018, pp. 362–373.

[39] S. Fang, Q. Zhang, G. Meng, S. Xiang, and C. Pan, "GSTNet: Global spatial–temporal network for traffic flow prediction," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2286–2293.

[40] R. Dai, S. Xu, Q. Gu, C. Ji, and K. Liu, "Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 3074–3082.

[41] Z. Wu, D. Zheng, S. Pan, Q. Gan, G. Long, and G. Karypis, "TraverseNet: Unifying space and time in message passing for traffic forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 14, 2022, doi: 10.1109/TNNLS.2022.3186103.

[42] T. Wang et al., "Synchronous spatiotemporal graph transformer: A new framework for traffic data prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 6, 2022, doi: 10.1109/TNNLS.2022.3169488.

[43] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1720–1730.

[44] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, and P. Li, "DSTAGNN: Dynamic spatial–temporal aware graph neural network for traffic flow forecasting," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 11906–11917.

[45] D. Liu, J. Wang, S. Shang, and P. Han, "MSDR: Multi-step dependency relation networks for spatial temporal forecasting," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 1042–1050.

[46] M. Xu et al., "Learning to effectively model spatial–temporal heterogeneity for traffic flow forecasting," *World Wide Web*, pp. 1–17, Mar. 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11280-022-01045-y

[47] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "GaAN: Gated attention networks for learning on large and spatiotemporal graphs," 2018, *arXiv:1803.07294*.

[48] C. Chen, Y. Liu, L. Chen, and C. Zhang, "Bidirectional spatial–temporal adaptive transformer for urban traffic flow forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 30, 2022, doi: 10.1109/TNNLS.2022.3183903.

[49] L. Deng, D. Lian, Z. Huang, and E. Chen, "Graph convolutional adversarial networks for spatiotemporal anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2416–2428, Jun. 2022.

[50] H. Gao, Y. Qin, C. Hu, Y. Liu, and K. Li, "An interacting multiple model for trajectory prediction of intelligent vehicles in typical road traffic scenario," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 31, 2021, doi: 10.1109/TNNLS.2021.3136866.

[51] X. Wang et al., "Traffic flow prediction via spatial temporal graph neural network," in *Proc. Web Conf.*, Apr. 2020, pp. 1082–1092.

[52] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities: A meta-learning approach for spatial–temporal prediction," in *Proc. World Wide Web Conf.*, May 2019, pp. 2181–2191.

[53] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang, "Cross-city transfer learning for deep spatio-temporal prediction," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1893–1899.

[54] Y. Wei, Y. Zheng, and Q. Yang, "Transfer knowledge between cities," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1905–1914.

[55] D. Das, "UApredictor: Urban anomaly prediction from spatial–temporal data using graph transformer neural network," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–8.

[56] S. Wang, J. Cao, H. Chen, H. Peng, and Z. Huang, "SeqST-GAN: Seq2Seq generative adversarial nets for multi-step urban crowd flow prediction," *ACM Trans. Spatial Algorithms Syst.*, vol. 6, no. 4, pp. 1–24, 2020.

[57] J. Zhu et al., "KST-GCN: A knowledge-driven spatial–temporal graph convolutional network for traffic forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15055–15065, Sep. 2022.

[58] X. Ma, Y. Yin, Y. Jin, M. He, and M. Zhu, "Short-term prediction of bike-sharing demand using multi-source data: A spatial–temporal graph attentional LSTM approach," *Appl. Sci.*, vol. 12, no. 3, p. 1161, Jan. 2022.

[59] D. Fensel et al., "Introduction: What is a knowledge graph," in *Knowledge Graphs*. Cham, Switzerland: Springer, 2020, pp. 1–10.

[60] Q. Wang, Z. Mao, and B. Wang, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, Dec. 2017.

[61] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 950–958.

[62] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[63] N. T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–14.

[64] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[65] S. Abu-El-Haija, A. Kapoor, B. Perozzi, and J. Lee, "N-GCN: Multi-scale graph convolution for semi-supervised node classification," in *Proc. Uncertainty Artif. Intell.*, 2020, pp. 841–851.

[66] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.

[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[68] C. Yang and G. Gidófalvi, "Fast map matching, an algorithm integrating hidden Markov model with precomputation," *Int. J. Geograph. Inf. Sci.*, vol. 32, no. 3, pp. 547–570, 2018.

[69] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, Jun. 2013.

[70] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, Nov. 2016, pp. 324–328.

[71] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[72] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.