Multi-relational Graph Attention Networks for Real-time Traffic Prediction

Jing Huang*, Kun Luo*, Longbing Cao, *Senior Member*, *IEEE*†, Yuanqiao Wen‡, Shuyuan Zhong*

***Abstract*—Accurate real-time traffic prediction is of great importance to ITS, yet it remains challenging due to the complex spatio-temporal dynamics of traffic systems. The continuous traffic signals from different channels and nodes in a traffic network are coupled with each other, including temporally over time points of each signal channel, spatially between traffic nodes, and in a spatio-temporal joint manner. These multi-aspect traffic signal couplings jointly reflect the conditions of a traffic system and evolve over traffic movement and system dynamics. The recent studies on formulating traffic prediction as a high-profile graph neural network-based modeling problem gains the state-of-the-art performance. However, they mainly focus on several well-known hidden relations captured by neural graph mechanisms with less on exploring the above multi-aspect interactions coupling diverse traffic signals. This work views a traffic system as a *coupled traffic network* and models the *multi-aspect traffic signal couplings* by a Multi-relational Synchronous Graph Attention Network (MS-GAT). Specifically, MS-GAT learns three separate embeddings to respectively represent traffic signal-based channel, temporal and spatial relations between nodes in a synchronous manner by specific graph attention networks, which are further adaptively coupled according to their respective importance to predictions. MS-GAT outperforms six state-of-the-art graph network-based traffic predictors tested on five real-world datasets, showing the value of its insights on not only capturing spatial and temporal relations over coupled traffic signals, but also the traffic signals-based channel relations and their couplings over the traffic evolution, where each relation contributes separately to the future traffic conditions of a node.**

***Index Terms*—Traffic prediction, multi-relational modeling, spatio-temporal graph modeling, graph attention networks, coupled traffic network, coupling learning, traffic signal coupling.**

I. Introduction

DATA-driven traffic prediction [1] forms a critical task of intelligent transport systems (ITS), valuable for various real-world applications such as efficient traffic management, dynamic route planning, and intelligent guidance service. Traffic prediction estimates the future traffic conditions based on historical observations of traffic systems such as physical road networks. Traffic conditions are reflected on temporal and spatial traffic signals such as channel or node-based traffic volume, density, speed, and others, captured by diverse sensors deployed at different geospatial locations in a traffic network. All aspects of traffic signals are coupled with each other temporally and spatially, forming a *coupled traffic network*, similar to many other complex systems where objects and their attributes interact and couple with each other [2], [3]. For example, the traffic conditions at one traffic node may be associated with those at its neighbouring and farther nodes before and after along the traffic network, forming *spatial relations* between traffic network nodes. Second, the traffic signals of one node may be temporally related to those of backward and forward nodes, forming *temporal relations*. Both spatial and temporal relations are node- and time-evolving, the same as their traffic conditions. They are further subject to the burst of traffic accidents and special events taken place on the way. Further, the spatial and temporal relations are interdependent, forming *spatio-temporal relations* in traffic systems. In addition, similar to the user-item interations in recommender systems [4], the spatio-temporal interactions in a traffic network and among its traffic conditions may be explicit and implicit, and hierarchical and heterogeneous [5]. These multi-aspect interactions and couplings make it difficult to precisely predict how the rise of vehicle flow at one node at a time point may cause the change of vehicle flows at other nodes at future time steps. It is thus essential yet challenging to effectively model the various spatio-temporal traffic signal couplings over channels and nodes, which could better capture the intrinsic characteristics and complexities in traffic networks, leading to better traffic modeling and prediction.

Extensive data-driven methods have been studied to analyze some relations in traffic systems in a road network, which can be categorized into time-series methods, shallow machine learning methods, and deep learning methods. First, in the early stage, traditional time-series methods such as historical average model [6], auto regressive integrated moving average (ARIMA) [7], Kalman filtering model [8] and canonical vector auto regressions (VAR) [9] were widely used in traffic forecasting. Then, shallow machine learners such as Support Vector Regression (SVR) [10], Bayesian model [11] and $k-$nearest neighbor method [12] succeeded. Lastly, deep neural networks (DNNs) has dominated today's literature showing the state-of-the-art performance of traffic prediction by capturing sequential relations in a large amount of traffic sequential data [13] and [14]. For example, temporal dependencies are modeled in [15] and [16] by recurrent neural networks (RNNs). In [17]–[19], spatial dependencies are cap-

Manuscript received xxx; revised xxx; accepted xxx.

This work was supported by the National Natural Science Foundation of China (No. 52072287) and Zhejiang Provincial Science and Technology Program (No. 2021C01010)

Jing Huang, Kun Luo and Shuyuan Zhong are with the School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan 430063, China (Corresponding auther: Jing Huang, e-mail: huangjing@whut.edu.cn).

Longbing Cao, IEEE Senior Member, is with the Advanced Analytics Institute, University of Technology Sydney, Sydney, NSW 2007, Australia.

Yuanqiao Wen is with the National Engineering Research Center for Water Transport Safety, Wuhan University of Technology, Wuhan 430063, China.

tured by convolutional neural networks (CNNs). Both RNNs and CNNs are insufficient in modeling spatio-temporal traffic interactions, which are more successfully characterized by the recent advances in graph neural networks (GNNs). GNNs present the interactions between entities as a graph and can capture spatial node interactions in a road network [20], [21]. Recent methods including STGCN [22], DCRNN [23], Graph WaveNet [24] and AGCRN [25] essentially formulate the traffic prediction as a spatio-temporal graph modeling problem and achieve more superior performance in traffic forecasting. GNN-based models focus on optimizing the representation of either spatial relations (e.g., [22] in urban roads by graph convolutional networks (GCN) and in [23] based on diffusion convolutional networks) or temporal relations (e.g., learning temporal patterns in [25] and [26] by gated recurrent units (GRU)).

Though DNNs including GNNs achieve the state-of-the-art traffic prediction performance by applying increasingly advanced network architectures and learning mechanisms and overparametering the networks, the current research focus is mainly on characterizing latent features and relations in a traffic network, e.g., the above spatial, temporal, and spatio-temporal features or relations. By taking a coupled traffic netowrk view with multi-aspect traffic signals coupled, there are still various issues and data complexities yet explored or insufficiently characterized. For example, sensors deployed at a node capture various aspects of traffic signals such as speed, volume and density, which are coupled in reality and thus their relations should be jointly represented to capture the multi-view node conditions. Further, the traffic conditions at one node or on one road may be related to that at other nodes or roads, i.e., a signal at a node and time point is essentially embedded in the entire traffic system and its dynamics. This coupled traffic network view focuses on modeling rich signal couplings and interactions in traffic systems, capable of addressing problems such that how an accident at one point may not only cause problems to its node and neighboring nodes and roads but also to other places affecting the evolving traffic network.

Specifically, building on the stronger capability of GNNs and addressing their significant gaps in modeling complex traffic systems, this work models the above multiple aspects of signal couplings and their evolution explicitly or implicitly embedded in complex traffic networks. We introduce a framework of modeling spatio-temporal traffic signal couplings (see Fig. 1), followed by a spatio-temporal graph network (see Fig. 2), named Multi-relational Synchronous Graph Attention Networks (MS-GAT). In MS-GAT, similar to the concept of image channel in computer vision, each traffic signal is viewed as a measurement channel at a traffic node. Here, the temporal signal movement over time forms the temporal channel representation; each node is often characterized by signals collected from multiple sensors deployed on the spot, with the signals interacting with each other to capture the multi-channel interdependencies; nodes in the traffic network are further connected to capture their spatial relations. Consequently, MS-GAT captures temporal relations in each signal over time, spatial relations over nodes, and multi-channel relations between signals. These are jointly modeled for traffic prediction. By viewing traffic systems as coupled traffic networks with coupled traffic signals over sensors, our contributions are as follows:

i) We recognize the significance of a kind of latent relation (called *channel relation* in this paper) and characterize this noteworthy relation in our proposed spatial-temporal graph model. To the best of our knowledge, this is the first work on explicitly modeling this channel relation in traffic prediction task.

ii) We propose a concrete spatial-temporal graph model based on multi-component fusion, called MS-GAT, which is accompanied by a flexible and effective data augmentation scheme for its supervised learning process. In this model, the newly-attended channel relation together with the other two familiar relations are picked abreast up by a core module called multi-relational embedding abreast module (MEAM). Furthermore, their contributions to the resulted traffic conditions in real-world forecasting scenarios are adaptively distinguished in a synchronous manner by a developed deep framework based on our devised multi-dimensional self-attention scheme.

iii) We develop a simple yet effective multi-dimensional self-attention scheme to improve the conveniences of applying and implementing self-attention mechanism for handling multi-dimensional input data, which is also interesting in ameliorating the influence of attention mechanism on model size while ensuring its prediction capability.

We conduct a series of experiments on five real-world traffic datasets. Experimental results demonstrate the superior performance of the proposed MS-GAT[1]. The remainder of this paper is organized as follows. Section II reviews the related work. The preliminaries are presented in Section III. Section IV details the proposed approach. Section V reports the experimental results, followed by the discussion in Section VI. Section VII concludes the paper.

II. Related Work

Here, we review three sets of related work: data-driven methods, coupling learning and attention-based relation modeling, and multi-component fusion for traffic prediction.

A. Data-driven traffic prediction

Compared to knowledge-driven methods using queuing theory such as in [27], data-driven methods have dominated the recent traffic forecasting research, benefiting from the convenient acquisition of a huge amount of traffic data and the vigorous development of machine learning models. The early work on data-driven traffic forecasting mostly focuses on time series analysis, such as ARIMA [7] and VAR [9], which typically rely on the stationarity assumption. To eliminate this assumption, the recent research shifts to deep-learning-based models. Various efforts, e.g., [15], [16], [25], [28], apply RNN and its variants such as LSTM [29] and GRU [30] to learn relations in traffic data automatically, taking advantage of the RNN's superior ability in modeling the temporal dynamics in

[1]The source code is publicly available from https://github.com/luokn/ms-gat

time series data [31]. However, they overlook other relations including spatial relations. Accordingly, other attempts based on deep learning for traffic forecasting (e.g., [17]–[19]) deploy CNN to pay more attention to spatial relations among the traffic series from different traffic nodes. However, subject to the fact that CNN is preferable to manipulate regular grid data (e.g., 2D images), those CNN methods force the spatial structure among different traffic series into a Euclidean space, which is violated by real-world traffic data. Considering the structural characteristics of traffic data, more recent work leverages the promising GNN (e.g., GCN [32] that is a special kind of CNN generalized for graph-structured data) to naturally model the spatial relations in traffic road networks. For example, STGCN [22] formulates the traffic prediction problem as graphs instead of applying regular recurrent and convolutional neural units in a deep learning architecture. Following the same manner, DCRNN [23], Graph WaveNet [24] and GSTNet [33] utilize various GCNs to capture the prominent spatial interactions among different traffic series. Their efforts gain the state-of-the-art performance, thus opening a new door for traffic prediction. In contrast, our work in this paper is also based on GCN ([34], [35]) to model a traffic system as a spatio-temporal graph, it pays attention not only to the different relations embedded in a graph structure but also their coupling strengths with each other to prediction. Our method uniquely models traffic signals-based channel relations, indispensable and hidden in a dynamic evolving systems like road networks. This channel relation modeling complements with the spatio-temporal prediction for capturing more comprehensive traffic conditions. Note that the traffic data we discuss in this paper refers to the readings of traffic signals acquired by a variety of geo-sensors in a traffic road network, e.g., traffic flow or speed.

B. Coupling learning and attention-based relation modeling

Complex systems like traffic networks are embedded with hierarchical and heterogneous coupling relationships and interactions within and between entities, subsystems and systems. Coupling learning captures them in various settings, such as couplings within and between heterogeneous data, attribute value-to-object couplings, multimodal data, coupled group behaviors, node couplings in social networks, user-item couplings in recommender systems, outlier detection, and cross-financial markets, etc. [2]–[5], [36], [37]. They involve various techniques including incorporating metric learning, representation learning, multikernel learning, coupled hidden markov models and deep neural networks into modeling complex couplings and interactions[2]. This paper expands coupling learning to learn multi-aspect traffic signal couplings by treating a traffic system as a coupled traffic network.

The attention mechanism is initially used in neural machine translation tasks [38]. Its core idea is to adaptively pick up features that are relatively critical to specific tasks by learning the relations hidden inside input data. In recent years, it has been widely applied in various sequence-to-sequence (seq2seq) issues, such as air quality forecasting [39] and sequential recommendation [37]. Typically, self-attention [40], [41] has been applied in traffic prediction to model dependencies, as shown in traffic prediction models like GeoMAN [42] using a multi-level attention-based recurrent neural network to predict the readings of a geo-sensor over several future hours and ASTGCN [43] implementing an attention-based spatial-temporal graph convolutional network for traffic flow forecasting. GMAN [44] proposes a graph multi-attention network in an encoder-decoder architecture to carry out long-term traffic prediction. All the attention-based prediction models (e.g., [42]–[48]) tend to improve the effect of capturing spatio-temporal dependencies by leveraging self-attention to cooperate with the related convolutional operations. However, considering that the input and output of a self-attention module are generally two sequences of vectors, existing prediction models either develop self-attention separately in the spatial dimension of traffic conditions or in its temporal dimension, and thus are not easily deployable to a multi-dimensional information space. Even if their self-attention processes are stubbornly implemented on each dimension of multi-dimensional input data in turn, the resultant model will fall into such dilemmas of a large parameter scale and the code-level redundancy. In contrast, we are interested in synchronously capturing several sorts of significant relations among multi-dimensional traffic data. Thus, a multi-dimensional self-attention scheme is proposed to universally and conveniently model different information dimensions of input data with less computational and development costs.

C. Multi-component fusion

The idea of multi-component fusion originates from ensemble learning [49]. In ITS, a few studies leverage the multi-component architectural style to deploy specific learning models for various tasks. For instance, [50] shows a novel deep learning model ST-MGCN for ride-hailing demand forecasting. For forecasting traffic flow in a road network, ASTGCN [43] enjoys a great success by adopting a gated mechanism based on time embedding to fuse multiple components. In contrast, our method MS-GAT also involves the multi-component fusion by using a time-gated mechanism for traffic prediction, it is more flexible in the preparation of training samples, which is crucial to the final performance and generalization capability of a supervised learning model. Specifically, our multi-component fusion design can offer richer and more diverse time-series graph data for the supervised learning according to the genuine importance of the input time-series graph data of each component to the same output sequence, instead of fixing the input sequence of each component as in [43] by intuitively judging the importance of the chosen input sequence (e.g., subjectively choosing time-series data with the daily- or weekly-period characteristics). In other words, our design is equipped with an effective data augmentation scheme, which can make full use of all the observed data along the time-axis to pick up their latent patterns sensitive to prediction.

[2]Interested readers may find more in https://datasciences.org/coupling-learning/.

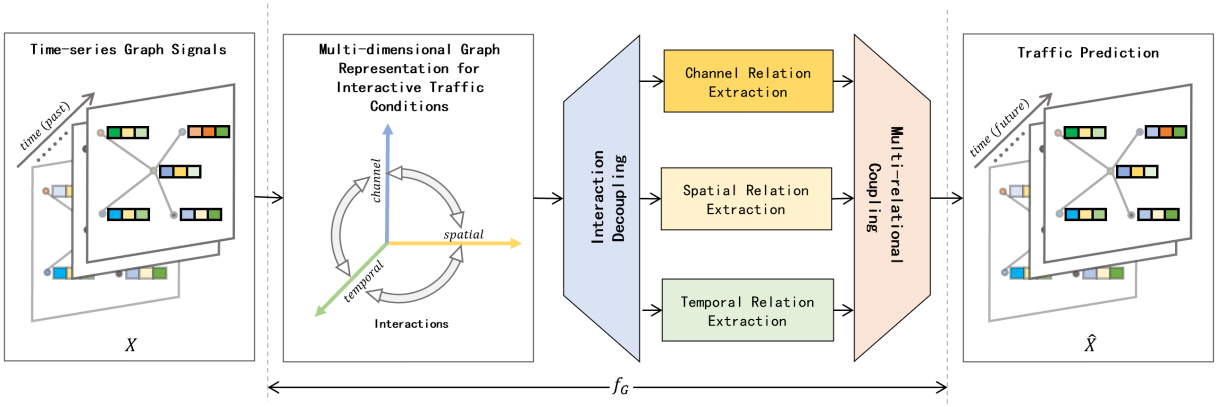

Fig. 1. The framework of modeling spatio-temporal traffic signal coupling relationships from multiple channels in a dynamic traffic system. It forms a three-dimensional graph representation of interactive traffic conditions and further provides a multi-relational view of traffic prediction, where the temporal, spatial and channel relations are explicitly and implicitly coupled in the entire evolving traffic network. Our method firstly decouple them for separate representations and then couple their representations forming a multi-relational traffic graph.

III. Problem Statement

The road network of a transport system is characterized as a directed graph $G = (V, E, A)$, where a set of $N = |V|$ vertices V denote the traffic nodes (e.g., sensors deployed in the traffic network); a set of edges E depict the connectivity among nodes; and the adjacency matrix $A \in \mathbb{R}^{N \times N}$ describes the connectivity in the network, where $A_{u,v} = 1$ refers to a connection between node u and node v, otherwise zero.

In a traffic network, all traffic conditions X describe the status of the network. Traffic conditions are multivariate, capturing various traffic signals such as traffic volume, density and speed. Suppose there are d_c types of available traffic signals (or indicators), each traffic signal is treated as a separate information channel of traffic nodes, similar to the concept of image channel in computer vision. We then have $X = \langle X(1), \ldots, X(d_c) \rangle$, where each term $X(l)$ (here, $1 \leq l \leq d_c$) is a random variable to depict a traffic condition, e.g., the observations (or measurement) of traffic flow. The multivariate observations at each traffic node capture the all-channel information of that node, which is called a node's *channel dimension*. The graph corresponding to the entire traffic network is represented by the interactions between all random variables (from $X(1)$ to $X(d_c)$), which is called the *channel relation* over all d_c channels in the network. In a road network, the traffic conditions may change over time, forming a dynamic graph. Assume there are t_p time steps, the traffic conditions X are composed of a sequence of t_p temporal random graph snapshots (each graph snapshot X^t corresponds to the traffic network profile at a time point t), i.e., $X = [X^{t_1}, \ldots, X^{t_p}]$. This time series describe the traffic conditions in terms of a *temporal dimension* of traffic nodes, and the interdependence between time-specific traffic conditions (i.e., different graph snapshots X^t and $X^{t'}$, here $t_1 \leq t, t' \leq t_p$) along the temporal dimension forms the *temporal correlation* in the network. Further, each node of the network is affiliated with multiple channels of signals, and each signal is temporal. Hence, the overall traffic conditions X can also be viewed as a set of N node-oriented random variables, i.e., $X = \{X_1, \ldots, X_N\}$, where each variable X_v denotes the traffic conditions of the node v, forming the *spatial dimension* of a node. The interdependence between node-specific variables captures the *spatial correlation* between nodes in the network.

Accordingly, we can model a traffic network in terms of channel, temporal and spatial dimensions, which offers a three-dimensional view of a traffic system. Further, any node v can be modeled by a three-dimensional vector $X_v^t(l)$ with its l^{th} signal at time point t. The three dimensions further capture various traffic conditions (signals), which directly reflect the observations and characteristics of the physical traffic world. In addition, the joint modeling of their temporal, spatial and spatio-temporal relations provide a deep quantification of a complex traffic network.

Consequently, the GNN representation of a traffic network maps a physical system with various traffic signals to a graph G with nodes $\{X_v^t(l)\}$ over t_p time steps. The problem of this GNN-based traffic prediction is to learn a mapping network f_G between the t_p steps of historical traffic conditions $X = [X^1, \ldots, X^{t_p}]$ and the next t_q steps of traffic conditions $\hat{X} = [X^{t_p+1}, \ldots, X^{t_p+t_q}]$ in the graph G.

$$[X^{t_p+1}, \ldots, X^{t_p+t_q}] = f_G(X^1, \ldots, X^{t_p}; \theta) \tag{1}$$

where θ stands for the learnable parameters of graph G.

Our insight. The above problem statement formulates a traffic system as a coupled traffic network where diverse traffic signales (e.g., from multiple sensors) are coupled with each, which differs from the existing assumptions and methods. Figure 1 illustrates the framework of modeling spatio-temporal traffic signal couplings in a dynamic coupled traffic network with various channels and nodes by a multi-relational graph. Our model extracts the channels of traffic signals, temporal signal development, and interactions between signals at each node, between nodes and over time. We further decouple them by representing their temporal relations, spatial relations, and channel relations. The multi-relational representations are then coupled to build the multi-relational view of a dynamic traffic

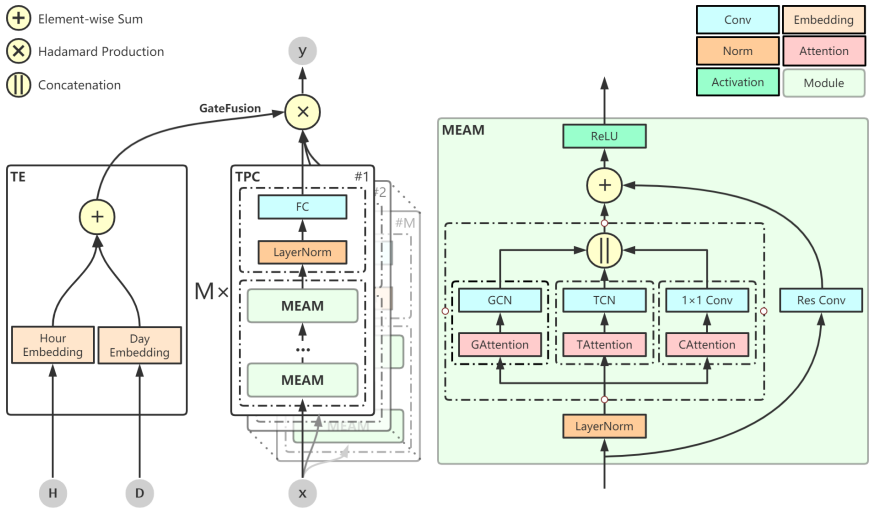

Fig. 2. An overview of the proposed MS-GAT to model multi-aspect traffic signal couplings. Left: Architecture of MS-GAT, which adopts the multi-component structure with time-gated fusion. Right: Core module of MS-GAT, i.e., multi-relational embedding abreast module (MEAM), which implements the interaction decoupling and multi-relational couplings respectively at its two ends by our multi-dimensional attention mechanism, while learning the embeddings of spatial relations, temporal relations and channel relations in a synchronously handling manner.

network. Consequently, the construction of multivariate time series of interactive traffic signals in a traffic system integrates various traffic signals and their multi-aspect relations. We expand GNNs to represent this multi-dimensional and multi-relational view, which shows a more powerful capability of capturing much richer multivariate observations and their hidden spatio-temporal relations in a traffic system than the existing methods.

IV. The MS-GAT Model

Following the problem statement and the framework of modeling traffic signal couplings in Section III, MS-GAT instantiates a concrete mapping network f_G for traffic prediction, as shown in Fig. 2. MS-GAT captures three-dimensional relations and their interactions on nodes of traffic network: i) the spatial and temporal relations among nodes; ii) the channel relations between traffic signal channels which influence the evolution of the whole traffic conditions on a road network; and iii) the distinct importance of each kind of relations to future traffic conditions of an individual node. Below, we introduce the network architecture, multi-dimensional self-attention, and multi-correlation embedding to model the multivariate and multi-relational traffic network, repsectively.

A. The Network Architecture

1) Overview: The MS-GAT model is composed of M same traffic prediction components (*TPC* for short) and adopts a multi-component structure based on the time-gated fusion. To achieve more accurate short-term prediction with a less parameter scale for efficient real-time prediction results, each *TPC* component is designed to share an identical network structure. The shared network structure follows the light-head structural style, which has proved very efficient in such computer vision tasks as target detection [51], rather than leveraging a heavy encoder-decoder framework.

As shown in the left part of Fig. 2, each *TPC* consists of multiple stacked multi-relational embedding abreast modules (MEAMs for short), and a light-head block associated with ultimate prediction. In a *TPC*, the MEAM stacking replicas are responsible for extracting the complex dynamic features related to various dependencies among traffic conditions from the input data to the current *TPC*. Subsequently, a light-head block is attached to its previous MEAMs to straightforwardly accomplish the mapping (regression) of the learned deep features to prediction results. Here, the light-head block is a sequential composition of two canonical operations, namely *Layer Normalization* and *Fully Connection*, for avoiding the heavy decoding overhead under the premise of ensuring the accuracy of short-term prediction. Formally, the input and output of the i^{th} $(1 \leq i \leq M)$ *TPC* are denoted as samples $x_i \in \mathbb{R}^{C\times N\times P_i}$ and output $\hat{y}_i \in \mathbb{R}^{C\times N\times Q_i}$ respectively, where C denotes the number of channels at a single traffic node, N denotes the number of traffic nodes, P_i and Q_i refer to the time steps respectively associated with the input samples and prediction results of the i^{th} *TPC*. Here, i refers to the identification number of a *TPC* in the proposed model adopting multi-component structure.

2) Multi-component structure based on time-gated fusion: According to the formal statement in Equation 1, the traffic prediction issue is naturally regarded as a seq2seq learning task. However, for a seq2seq model by supervised learning, we find: i) feeding it with longer historical sequential data at one time may lead to the rise of model overhead but does not necessarily improve the accuracy of forecasted sequence; ii) different historical segments with same sequence length but distinct offsets to the common '*now*' moment show separate importances to the identical future sequence. These show the sensitivity of the length and starting and ending points of a training sequence on the accuracy and efficiency of forecasting the output by a seq2seq model. To address these issues, MS-GAT adopts the ensemble learning and multi-component

architectural style to its design and training.

In MS-GAT, the i^{th} *TPC* acts as a separate sub-model that corresponds to a learnable mapping function f_i. Importantly, despite all *TPCs* take the same length of time-series graph signals as their input sequences, they deal with respective time slices that have different offsets relative to the same present moment. In detail, as shown in Fig. 3, we pick up *M* shorter time sequences from the past *P*-step graph signals according to their importance for the next *Q*-step traffic conditions, each of which behaves as the input samples of different *TPCs*. Formally, for the input sequence of the i^{th} *TPC*, $x_i = (x_i^1, x_i^2, \cdots, x_i^r) \in \mathbb{R}^{C\times N\times r}$, where r refers to the length of time steps of input samples (apparently, here r equals P_i), and each $x_i^\eta (1 \le \eta \le r)$ corresponds to a separate graph signal on the traffic network depicting the traffic condition at a specified moment. Moreover, the starting time point of x_i is denoted as $t(x_i^1)$, and $|t(x_i^1) - t_{now}|$ corresponds to the offset of x_i relative to the specified present moment denoted as t_{now}. Accordingly, the input samples of MS-GAT is as follows: $x = [x_1, \ldots, x_M] \in \mathbb{R}^{M\times C\times N\times r}$.

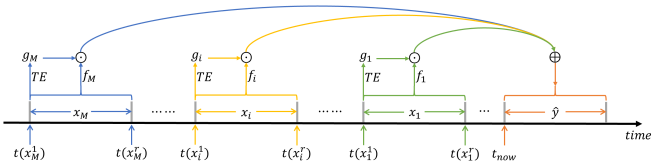

Fig. 3. Illustration of the sampling window that slides over the entire time-series graph signals

Meanwhile, when the specified present moment t_{now} moving over the whole observed historical graph signals along the time axis, the past P time steps together with the next Q time steps form a sliding window to generate training samples for our model. As a result, a rich number of distinct training samples can be flexibly yielded from a small historical time-series data in this manner, which evidently improves the capability of the regression model.

Time embedding for the gated fusion. It is observed that the traffic conditions at the past distinct time slices have different correlation strengths with that of the same future time slice. Accordingly, we propose to incorporate the time information of each input sequence x_i into its corresponding mapping function $f_i(x_i)$ by adding a timing embedding module (called *TE*) in each *TPC* (shown in Fig. 2), and further leverage a gated fusion mechanism to adaptively merge all sub-models (*TPCs*) to obtain an ultimate prediction model based on the multi-component structure, i.e., MS-GAT. To be specific, by using one-hot coding, we encode the day-of-week and time-of-day of the starting time point $t(x_i^1)$ of each input sequence x_i into the vectors $D_i \in R^7$ and $H_i \in R^{24}$, respectively. After that, by applying a vector embedding approach (e.g., word2vec [52]), the two vectors are transformed to two other vectors with the same dimension, respectively. They are then added together forming the time embedding of the input sequence x_i. Formally, the time embedding related to the input sequence x_i corresponds to a tensor denoted as g_i, which will be fused as a gate with the associated mapping function $f_i(x_i)$. Thus, the final prediction result after the fusion is:

$$\hat{y} = \sum_{i=1}^{M} g_i \odot f_i(x_i), \tag{2}$$

where

$$g_i = TE(D_i, H_i), \tag{3}$$

$\odot$ represents the element-wise product, the function $TE(\cdot)$ denotes the specific implementation of the above embedding process related with the timing embedding module called *TE*, both $f_i(\cdot)$ and function $TE(\cdot)$ contain learnable parameters, $g_i \in \mathbb{R}^{C\times N\times Q}$, and $f_i(x_i) \in \mathbb{R}^{C\times N\times Q}$.

B. Multi-dimensional self-attention scheme

We first briefly review an ordinary self-attention scheme, called scaled dot-product attention [40], which adopts ordinary dot-product between vectors as the core operation of whole self-attention process to obtain the correlation strength between input vectors, i.e., *attention score*.

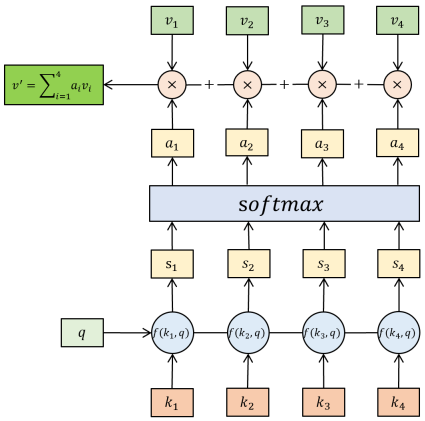

Fig. 4. The calculation process of self-attention in an exemplar sentence - *this is a cat*. Here, q refers to the *query* vector associated with the word 'this', 'is', 'a' or 'cat'; (k_1, v_1), (k_2, v_2), (k_3, v_3) and (k_4, v_4) represent the *key-value* pairs of the four separate words respectively; f denotes a type of attention function that adopts dot-product operation between two vectors (i.e., the above *key* and *query*), and s_1, s_2, s_3 and s_4 are its resulting attention scores respectively; the $softmax$ function is used to normalize these attention score to a_1, a_2, a_3 and a_4 respectively, which are all in the range of $[0, 1]$; v' is the result of current word after performing self-attention by its associated q, which contains the contexts of this sentence except for the information of current word itself.

Taking a linguistic sentence, e.g., *this is a cat*, as an example. Clearly, it is a sequence of words, and can be viewed as a sequential data just over temporal dimension. Importantly, this sentence also conceals the contextual dependencies in temporal dimension between these four separate words. To find out such hidden dependency relations from this sentence, valuable for the tasks of NLP (e.g., language translation), the ordinary attention scheme can be alone applied in a specified information dimension, i.e., temporal dimension of this sequential data. Concretely, each word of the sentence is initially embedded to a vector by a vectorization method (e.g.,

classic word2vec [52]), and further three same vectors (called query, key and value respectively) derived from the previous vector by linear transform are inputted simultaneously into the self-attention module. After each query of this sentence (i.e., corresponding to each word respectively) accomplishes self-attention operation with all of four key-value pairs, an output sequential data is obtained, which exposes those potential significant contexts into its each sequential element. Figure 4 shows the calculation process of this self-attention scheme in the exemplar sentence. Formally, given a query q, all keys (packed into matrices K) and values (packed into matrices V), the output value $value$ is weighted average over the input values as below.

$$value = softmax(\frac{qK^\top}{\sqrt{d_k}})V \tag{4}$$

where d_k is the dimension of K, the weights are the outputs of a *softmax* function, and the inputs of the *softmax* function are scaled dot products of the queries with all keys. Furthermore, when a sequence of queries is also combined in matrices form Q, the attention map, i.e., a resultant matrix including all attention scores, is calculated by Equation 5.

$$Attention(Q, K, V) = softmax(\frac{QK^\top}{\sqrt{d_k}})V \tag{5}$$

When only two aspects of information in the original input data, e.g., spatial-relevant and temporal-relevant information of normal spatio-temporal data, are attended to, the canonical attention scheme seems easy to carry out in a learning model. The existing traffic prediction models relying on attention mechanism (e.g., [43], [44]) mostly act in accordance with this scheme. However, if there are more aspects of information to be focused on, the ordinary scheme is not convenient to be deployed. Specifically, when the self-attention is separately paid to each aspect of information, every associated dimension in the input data needs to alternatively apply the above scheme, which would result in such dilemmas as the large parameter scale and the code-level redundancy, thus affecting the model performance. Therefore, we seek a better attention implementation way to capture those correlations when more aspects of information need to be considered from their associated multi-dimensional data.

In MS-GAT, we develop a simple yet effective multi-dimensional self-attention scheme to enable the convenient generalization of existing ordinary self-attention scheme to a multi-dimensional information space. Notably, the proposed scheme is formulated as a reusable process, which is good at being universally and conveniently carried out in various information dimension of a multi-dimensional data, even with less computational cost for the resultant model. It works as follows.

Taking a three-dimensional space including the $x-$, $y-$ and $z-$axes as an example, which represent three different aspects of information associated with the original data, respectively. As a matter of fact, x-axis, y-axis and z-axis can correspond to the spatial, temporal and channel dimensions of spatio-temporal traffic signal data respectively. For the

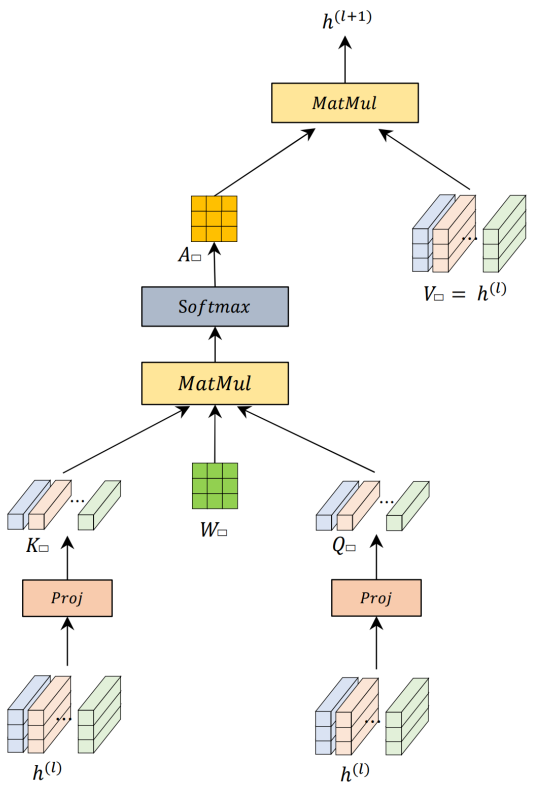

Fig. 5. Illustration of the calculation process of our multi-dimensional self-attention scheme. Here, *MatMul* refers to matrix multiplication; *Proj* denotes the operation of linear transformation; $Q_\square$, $K_\square$ and $V_\square$ are the query, key and value in the multi-dimensional self-attention respectively; $h^{(l)}$ and $h^{(l+1)}$ are the input and output of the l^{th} layer multi-dimensional self-attention.

ease of description, the exemplar data space is denoted as a discrete set $\{h_{x,y,z}\}$, each element $h_{x,y,z}$ is represented in a three-dimensional subscript form. Suppose that the potential correlation inside the information along the $x-$axis needs to be caught, the information separately along the other two dimensions (i.e., $y-$ and $z-$axes) will work together to achieve it. To be specific, our self-attention scheme sets query, key and value to the identical matrix denoted as $h_{x,:,:}$, where the subscript ':' denotes all elements in the corresponding dimension, rather than the vector in the prior ordinary self-attention scheme. Afterwards, both query and key are transformed to their associated vectors respectively by employing an efficient linear transformation without parameters for tensor operation (e.g., the general API function *torch.einsum()* can be used for the code implementation in its PyTorch version, or *tf.einsum()* for its TensorFlow version). For instance, the 1-D vector $h_{x,\tau,:}$ or $h_{x,:,\tau}$ acts as the vectorization of its associated 2-D matrix $h_{x,:,:}$, where τ refers to performing the above linear transformation in the second or third dimension. All the vectorized queries and keys are further packed into matrices $Q_\square$ and $K_\square$, respectively (here, the subscript $\square$ is a subscript placeholder for our subsequent formulas in the paper, e.g., Equation 7). Then, our attention map related to the current dimension (i.e., the correlation strength quantifying the interdependencies between information itself along the $x-$axis) is obtained through calculating all related attention

scores in parallel by Equation 6.

$$A_{\square} = Attention(Q_{\square}, K_{\square}) = softmax(Q_{\square} W_{\square} K_{\square}^{\top}) \quad (6)$$

where $W_{\square}$ is an $n_k \times n_k$ matrix that denotes all learnable parameters for applying attention mechanism in the $x-$dimension, and n_k is the number of current keys. Definitely, too large n_k will increase the computational cost of applying the attention, thus impacting on the performance and usage of the resultant model. Thereby, we adopt an effective trick to evade the computational difficulty caused by too much parameters in applying the attention. Specifically, the parameter matrix $W_{\square}$ is replaced with the multiplication of a small-size learnable parameter matrix and its transposed matrix, i.e., $W_{\square} = E_{\square} E_{\square}^{\top}$, where $E_{\square}$ refers to the new matrix with smaller parameter scale $n_k \times e_k$ (here, $e_k < n_k$). Lastly, all values on the current dimension (i.e., the $x-$axis) are renewed to their respective hidden states in parallel by calculating the corresponding weighted sums of all the values based on the current attention map $A_{\square}$. Figure 5 illustrates the calculation process of our multi-dimensional self-attention scheme. Furthermore, since the above self-attention process on the $x-$axis can be easily developed to a reusable code-level module, it can be also conveniently deployed to the $y-$ or $z-$axis, even to the other new information dimensions in a higher-dimensional data space. Thus, our self-attention scheme can save the development cost by reducing the code-level redundancy.

C. MEAM: multi-relational embedding abreast module

As the core module of *TPC*, the multi-relational embedding abreast module (MEAM) comprises three abreast embedding branches, which encode the spatial, temporal, and channel relations, respectively. The right part of Fig. 2 shows the MEAM architecture. The MEAM stacking replicas are dedicated to extracting the deep features related to the complicated spatial-temporal dynamics of traffic conditions. In each MEAM, the embeddings obtained from separate branches are concatenated into the next-layer MEAM, which attends to implicitly compute their respective importance to the predicted results by the self-attention mechanism of next-layer MEAM. Consequently, the respective contributions of the spatial, temporal and channel relations to the resultant traffic conditions for real-world forecasting can be adaptively distinguished in the whole deep regression model. Moreover, the residual structure [53] is likewise adopted in MEAM to train the model.

1) Spatial relation embedding: The left branch in MEAM serves for generating the embedding related to spatial relations (i.e., spatial relation extraction shown in Fig. 1), which consists of two important operations *GAttention* and *GCN*. The *GAttention* operation first deploys our proposed multi-dimensional self-attention scheme on the spatial dimension of the input data of each MEAM. Take the $j-$layer MEAM in the i^{th} *TPC* of MS-GAT as an example, its input is denoted as $h^{(j-1)} \in \mathbb{R}^{C_j \times N_j \times T_j}$, where $h^{(0)} = x_i$, and C_j, N_j, and T_j correspond to the channel, spatial and temporal dimensions of $h^{(j-1)}$, respectively. Here, j refers to the identification number of a layer in MEAM stacking replicas. The query, key and value of each spatial node are set to an identical matrix as described in Section IV-B. Then, all the queries and keys are encoded to their associated matrices denoted as $Q_s \in \mathbb{R}^{N_j \times T_j}$ and $K_s \in \mathbb{R}^{N_j \times T_j}$. According to Equation 6, a new matrix $A_s^{(j)} \in \mathbb{R}^{N_j \times N_j}$, reflecting the potential spatial relations, is generated. The *GAttention* operation is defined as follows:

$$\begin{aligned} GAttention(h^{(j-1)}) = A_s^{(j)} &= Attention(Q_s, K_s) \\ &= softmax(Q_s W_s K_s^{\top}) \end{aligned} \quad (7)$$

where $W_s \in \mathbb{R}^{T_j \times T_j}$ refers to a learnable parameter matrix for the attention on the spatial dimension.

Based on the generated $A_s^{(j)}$, we further apply the *GCN* operation to aggregate the hidden states of each node and its first-order neighbors in the MEAM of the previous layer $h^{(j-1)}$. The *GCN* operation is formulated as follows:

$$GCN(h^{(j-1)}) = \sigma(\hat{A}_s^{(j)} h^{(j-1)} W_G^{(j)} + b_G^{(j)}) \quad (8)$$

where $\hat{A}_s^{(j)} = A_s^{(j)} A \in \mathbb{R}^{N_j \times N_j}$ denotes an improved adjacency matrix for subsequent graph convolutional operation, $W_G^{(j)} \in \mathbb{R}^{(C_j \times T_j) \times (C_j \times T_j)}$ and $b_G^{(j)} \in \mathbb{R}^{(C_j \times T_j)}$ are the learnable parameters for first-order graph convolutional networks [32] in the $j-$layer MEAM, σ denotes the activation function, e.g., classic ReLU. After that, the embedding related to spatial relations $e_s^{(j)} = GCN(h^{(j-1)})$ is generated, which is then fused with the results of the other two embedding branches as the input of next-layer MEAM $h^{(j)}$. Note that, during the learning process of the current MEAM, the importance of the previous layer's spatial embedding $e_s^{(j-1)}$ to the prediction results has been synchronously learned in an implicit manner by performing self-attention on the channel and temporal dimensions of $h^{(j-1)}$.

2) Temporal relation embedding: In a real-world road network, there are typically explicit or implicit temporal relations between pair-wise traffic conditions at different timestamps. For example, the traffic speeds on an identical traffic node might present daily-periodic or weekly-periodic trends. Also, the current average vehicle speed on a node could be associated with the traffic volume at a past timestamp on another node. To this end, we build an independent branch in MEAM to focus on the embedding of such temporal relations (i.e., temporal relation extraction shown in Fig. 1).

The embedding branch also comprises two critical operations *TAttention* and *TCN*. The *TAttention* operation is similar to the aforementioned *GAttention* operation. Their difference is that *TAttention* realizes our proposed multi-dimensional self-attention scheme on the temporal dimension of the input data of each MEAM. Let us further take the $j-$layer MEAM in the i^{th} *TPC* of MS-GAT as an example, according to Equation 6, a temporal attention map $A_T^{(j)} \in \mathbb{R}^{T_j \times T_j}$ is obtained to semantically represent the pair-wise correlation strengths between the traffic conditions at different timestamps. The *TAttention* operation is defined as follows:

$$\begin{aligned} TAttention(h^{(j-1)}) = A_T^{(j)} &= Attention(Q_T, K_T) \\ &= softmax(Q_T W_T K_T^{\top}) \end{aligned} \quad (9)$$

where $Q_T \in \mathbb{R}^{T_j \times N_j}$ and $K_T \in \mathbb{R}^{T_j \times N_j}$ refer to the query and key matrices of temporal attention respectively, $W_T \in \mathbb{R}^{N_j \times N_j}$ is its learnable weight matrix. Considering that the number of traffic nodes straightly affects the parameter scale of the weight matrix of temporal attention, we adopt the trick mentioned in Section IV-B to balance the efficiency and accuracy of our attention-based model. Thus, the *TAttention* operation is further formulated as follows:

$$\begin{aligned} TAttention(h^{(j-1)}) &= A_T^{(j)} \\ &= Attention(Q_T, K_T) \\ &= softmax(Q_T E_T E_T^\top K_T^\top) \end{aligned} \tag{10}$$

where $E_T \in \mathbb{R}^{N_j \times d_E}$ and d_E denotes a number much smaller than N_j.

Subsequently, the input of this embedding branch (equivalent to the input of the current MEAM $h^{(j-1)}$) is taken as the value of the temporal attention, which is then renewed to $h_T^{(j-1)} \in \mathbb{R}^{C_j \times N_j \times T_j}$ by leveraging the obtained temporal attention map $A_T^{(j)}$. We further perform the *TCN* operation to deal with $h_T^{(j-1)}$ along its temporal dimension. *TCN* follows the idea of temporal convolutional networks [54] to preserve the chronological order of data (namely the outputs at the current time are only related to its historical data). To be specific, two levels of dilated causal convolutions are applied to represent $h_T^{(j-1)}$. A single dilated causal convolution operation F on element ξ of the time-series data $h_T^{(j-1)}(v)$ at a traffic node v takes the following form:

$$F(\xi) = \sum_{m=0}^{\pi-1} \phi_m \cdot [h_T^{(j-1)}(v)]_{\xi - \omega \cdot m} \tag{11}$$

where ω is the dilation factor, ϕ_m is our *TCN* filter, π is its size, and $\xi - \omega \cdot m$ accounts for the direction of the past. In practice, we utilize the zero padding strategy to keep the temporal length unchanged. At the end, the temporal embedding $e_T^{(j)}$ is obtained.

3) Channel relation embedding: Besides the two previous embedding branches, we build another separate branch in MEAM for the embedding of our proposed channel relations in traffic networks (i.e., channel relation extraction shown in Fig. 1). This embedding branch is primarily implemented by the *CAttention* operation to pick up the pairwise correlations between channels by a self-attention mechanism. Specifically, *CAttention* carries out our proposed multi-dimensional self-attention scheme on the channel dimension of the input data of each MEAM, while the information situated in the other two dimensions will jointly attend to the computation of the attention score between different traffic conditions along the channel dimension. To a certain extent, the attention score semantically represents the correlation strength between channels. Accordingly, also for the $j-$layer MEAM in the i^{th} *TPC* of MS-GAT, the channel attention map associated with the input of this branch, denoted as $A_c^{(j)} \in \mathbb{R}^{C_j \times C_j}$, is computed by:

$$\begin{aligned} CAttention(h^{(j-1)}) = A_c^{(j)} &= Attention(Q_c, K_c) \\ &= softmax(Q_c W_c K_c^\top) \end{aligned} \tag{12}$$

where $Q_c \in \mathbb{R}^{C_j \times T_j}$ and $K_c \in \mathbb{R}^{C_j \times T_j}$ are the query and key matrices of the channel attention respectively, $W_c \in \mathbb{R}^{T_j \times T_j}$ indicates its learnable weight matrix. Afterward, we set the input of this embedding branch (equal to the input of the current MEAM $h^{(j-1)}$ as the value of our channel attention, which can thus be renewed by applying the obtained channel attention map $A_c^{(j)}$. Finally, the channel embedding $e_c^{(j)}$ is generated after the necessary dimension transformation by the 1×1 convolution. Likewise, its importance weight to the prediction results will be implicitly evaluated in the MEAM of the next layer.

V. Experiments

In this section, extensive experiments are conducted to verify the effectiveness and superiority of the proposed MS-GAT[3].

A. Experiment setup

1) Data description: We evaluate the performance of MS-GAT on five public traffic network datasets: the PEMS-BAY dataset [23] and the PEMSD3, PEMSD4, PEMSD7 and PEMSD8 datasets [35]. They were all constructed from the Caltrans Performance Measurement System (PeMS) [55] and are commonly used in traffic prediction. Each of these five datasets is associated with a district of California, i.e., corresponding to a separate traffic road network, respectively. All time-series readings in these datasets are aggregated into 5-minute windows, which means there are 12 sampling points (i.e., timesteps) for every hour. Table I shows the related information of the five datasets. Note that, PEMSD4, PEMSD8 and PEMS-BAY contain three kinds of traffic signals (indicators) at each node: traffic flow, average speed and average occupancy, whereas PEMSD3 and PEMSD7 just involve the traffic flow.

TABLE I
THE DATA DESCRIPTION AND STATISTICS.

Datasets	#Nodes	#Edges	#TimeSteps	#Channels
PEMSD3	358	547	26208	1
PEMSD4	307	340	16992	3
PEMSD7	883	866	28224	1
PEMSD8	170	295	17856	3
PEMS-BAY	325	8033	52116	3

For each dataset, we derive a spatial adjacency matrix based on the connectivity among traffic nodes, and adopt z-score normalization to standardize all the data. Meanwhile, our data augmentation method (see details in Section IV-A) is applied to the five datasets separately. After that, we split the data from the five datasets respectively with the same ratio 6:2:2 into training, validation and test sets.

[3]The code is available at: https://github.com/luokn/ms-gat.

2) Baseline methods: Six state-of-the-art models are compared with MS-GAT for performance evaluation. As shown in [34], [35], the baseline methods based on deep neural networks (DNN) exhibit more superior traffic prediction results than traditional time-series methods such as HA [6], ARIMA [7], VAR [9] and SVR [10]. In particular, they achieve the impressive prediction results by exploiting the generalized DNN on graphs. Thus, our experiments select the following six representative graph networks as the baselines: DCRNN, STGCN, ASTGCN, Graph WaveNet, STSGCN and AGCRN, where STSGCN and AGCRN are the recent approaches.

DCRNN: A Diffusion Convolutional Recurrent Neural Network [23], which leverages the diffusion graph convolutional networks and the encoder-decoder recurrent neural networks to capture spatial dependencies and temporal dependencies in the traffic flow, respectively.

STGCN: A Spatio-Temporal Graph Convolutional Network [22], which combines graph convolution with 1-D convolution to model the spatial and temporal dynamics of the traffic flow.

ASTGCN: An Attention-based Spatial Temporal Graph Convolutional Network [43], which introduces attention mechanisms into the spatial-temporal modeling of traffic flow based on graph convolutional networks and also adopts multi-component structure.

Graph WaveNet: It refers to the method in [24], which develops an adaptive dependency matrix and combines it into graph convolution with dilated casual convolution to capture spatial-temporal dependencies.

STSGCN: A Spatial-Temporal Synchronous Graph Convolutional Network [35], which exploits a spatial-temporal graph convolutional module to synchronously capture the localized spatial-temporal correlations directly.

AGCRN: An Adaptive Graph Convolutional Recurrent Network [25], which captures fine-grained spatial and temporal correlations in traffic series automatically based on recurrent networks and two adaptive modules that enhance GCN with new capabilities.

3) Evaluation metrics: To make fair comparison, we deploy three commonly-used metrics to measure the difference between the ground truth and the predicted traffic flow, which are mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE), respectively. Suppose that $\mathcal{Y}_{m,n}$ represents the ground truth, $\hat{\mathcal{Y}}_{m,n}$ represents the predicted value (here, $1 \leq m \leq N$, $1 \leq n \leq Q$, N denotes the number of traffic nodes, and Q denotes the future time steps), and the three metrics are defined as follows:

$$MAE = \frac{1}{NQ} \sum_{m=1}^{N} \sum_{n=1}^{Q} \left| \mathcal{Y}_{m,n} - \hat{\mathcal{Y}}_{m,n} \right| \tag{13}$$

$$RMSE = \sqrt{\frac{1}{NQ} \sum_{m=1}^{N} \sum_{n=1}^{Q} \left(\mathcal{Y}_{m,n} - \hat{\mathcal{Y}}_{m,n} \right)^2} \tag{14}$$

$$MAPE = \frac{1}{NQ} \sum_{m=1}^{N} \sum_{n=1}^{Q} \left| \frac{\mathcal{Y}_{m,n} - \hat{\mathcal{Y}}_{m,n}}{\mathcal{Y}_{m,n}} \right| \times 100\% \tag{15}$$

4) The MS-GAT settings: We implement the MS-GAT model using PyTorch [56]. MS-GAT takes multi-segment historical data as input according to the number of its prediction components (i.e., *TPCs*). Each segment of the whole input samples of MS-GAT has fixed timesteps that is set to 12. Accordingly, every group of the spatial-temporal data of the past hour is assigned to a separate prediction component of MS-GAT, which could be any hour in the historically observed data, e.g., an hour, the second hour or the third hour before the forecasting time point, or one hour of the day before the forecasting time point, one hour of the week before the forecasting time point. In the experiments, we use MS-GAT to predict the traffic flow of one hour, half an hour and a quarter of an hour in the future. Meanwhile, we choose Huber loss [57] as the lose function, which is less sensitive to outliers than the squared error loss, which is defined below:

$$\mathcal{L}_\delta(\mathcal{Y}, \hat{\mathcal{Y}}) = \begin{cases} \frac{1}{2}\left(\mathcal{Y} - \hat{\mathcal{Y}}\right)^2 & if \ \left|\mathcal{Y} - \hat{\mathcal{Y}}\right| \leq \delta \\ \delta \left|\mathcal{Y} - \hat{\mathcal{Y}}\right| - \frac{1}{2}\delta^2 & otherwise \end{cases} \tag{16}$$

where $\mathcal{Y}$ denotes the ground truth, $\hat{\mathcal{Y}}$ denotes the predicted value, and δ is the hyperparameter to control the sensitivity of squared error loss that is set to 40 in our experiments. We train MS-GAT using the Adam optimizer with learning rate 0.001, and the training epoch is set to 100. MS-GAT is evaluated more than 8 times on each dataset. Moreover, all experiments are conducted on a 64-bit Ubuntu 18.04 computer with two CPUs (Intel Xeon Platinum 8176 @2.10 GHz, 312 GB memory) and eight GPUs (NVIDIA GeForce RTX 2080 TI, 11 GB memory).

B. Comparison results and analysis

Table II and Table III present the performance comparison between different models on the commonly-used datasets. Table II shows MS-GAT outperforms baselines consistently on the PEMSD3, PEMSD4, PEMSD7 and PEMSD8 datasets. For instance, MS-GAT achieves 7.0%, 1.4%, 7.8% and 12.6% improvement respectively over the state-of-the-art models on the metric of MAE. Meanwhile, Table III shows comparison with baselines, MS-GAT consistently achieves the best results for forecasting 15 minutes, 30 minutes and 1 hour ahead on the PEMS-BAY dataset. It suggests MS-GAT has an effective prediction capability in both short-range and long-range cases. By further comparison, we have the following observations.

On one hand, although the state-of-the-art models gain impressive results by developing different variants of GNN and RNN to deal with the spatial-temporal graph modeling issues for traffic prediction, their processes of capturing the spatial and temporal relations are carried out in an alternate manner. Instead, MS-GAT is aware of the couplings between spatial relations and temporal relations and their influence on future traffic conditions, which thus adopts a synchronously handling manner to capture diverse relations hidden in traffic time-series readings. Relying on this synchronous manner for modeling relations, MS-GAT can dynamically assign the importance weight of the spatial, temporal and channel relations by

TABLE II
PERFORMANCE COMPARISON OF MS-GAT AND BASELINE MODELS ON PEMSD3, PEMSD4, PEMSD7 AND PEMSD8.

Datasets	Metric	DCRNN	STGCN	ASTGCN(r)	Graph WaveNet	STSGCN	AGCRN	**MS-GAT**
PEMSD3	MAE	18.18 ± 0.15	17.49 ± 0.46	17.69 ± 1.43	19.85 ± 0.03	17.48 ± 0.15	16.13 ± 0.19	**15.68 ± 0.24**
	MAPE(%)	18.91 ± 0.82	17.15 ± 0.45	19.40 ± 2.24	19.31 ± 0.49	16.78 ± 0.20	-	**16.11 ± 0.13**
	RMSE	30.31 ± 0.25	30.12 ± 0.70	29.66 ± 1.68	32.94 ± 0.18	29.21 ± 0.56	28.42 ± 0.07	**26.54 ± 0.16**
PEMSD4	MAE	24.70 ± 0.22	22.70 ± 0.64	22.93 ± 1.29	25.45 ± 0.03	21.19 ± 0.10	19.75 ± 0.11	**19.54 ± 0.19**
	MAPE(%)	17.12 ± 0.37	14.59 ± 0.21	16.56 ± 1.36	17.29 ± 0.24	13.90 ± 0.05	13.00 ± 0.18	13.44 ± 0.28
	RMSE	38.12 ± 0.26	35.55 ± 0.75	35.22 ± 1.90	39.70 ± 0.04	33.65 ± 0.20	32.41 ± 0.20	**31.69 ± 0.15**
PEMSD7	MAE	25.30 ± 0.52	25.38 ± 0.49	28.05 ± 2.34	26.85 ± 0.05	24.26 ± 0.14	21.19 ± 0.09	**20.47 ± 0.13**
	MAPE(%)	11.66 ± 0.33	11.08 ± 0.18	13.92 ± 1.65	12.12 ± 0.41	10.21 ± 1.65	8.95 ± 0.08	**8.84 ± 0.14**
	RMSE	38.58 ± 0.70	38.78 ± 0.58	42.57 ± 3.31	42.78 ± 0.07	39.03 ± 0.27	35.12 ± 0.12	**34.17 ± 0.08**
PEMSD8	MAE	17.86 ± 0.03	18.02 ± 0.14	18.61 ± 0.40	19.13 ± 0.08	17.13 ± 0.09	16.10 ± 0.11	**14.78 ± 0.09**
	MAPE(%)	11.45 ± 0.03	11.40 ± 0.10	13.08 ± 1.00	12.68 ± 0.57	10.96 ± 0.07	10.27 ± 0.08	**10.07 ± 0.12**
	RMSE	27.83 ± 0.05	27.83 ± 0.20	28.16 ± 0.48	31.05 ± 0.07	26.80 ± 0.18	25.62 ± 0.17	**24.15 ± 0.07**

TABLE III
PERFORMANCE COMPARISON OF MS-GAT AND BASELINE MODELS ON PEMS-BAY.

Dataset	Metric	DCRNN	STGCN	Graph WaveNet	STSGCN	STGCNN	AGCRN	**MS-GAT**
15min	MAE	1.38	1.36	1.30	2.54	1.20	1.16	**1.13 ± 0.02**
	MAPE(%)	2.90	2.90	2.73	5.88	2.34	2.47	**2.44 ± 0.06**
	RMSE	2.95	2.96	2.74	4.79	2.43	2.40	**2.38 ± 0.04**
30min	MAE	1.74	1.81	1.63	2.60	1.46	1.41	**1.35 ± 0.01**
	MAPE(%)	3.90	4.17	3.67	6.03	3.09	3.12	**3.09 ± 0.04**
	RMSE	3.97	4.27	3.70	4.93	3.27	3.10	**2.94 ± 0.02**
60min	MAE	2.07	2.49	1.95	2.71	1.83	1.79	**1.74 ± 0.03**
	MAPE(%)	4.90	5.79	4.63	6.39	4.15	4.01	**3.97 ± 0.05**
	RMSE	4.74	5.69	4.52	5.28	3.20	3.78	3.88 ± 0.02

adaptively learning from the ground-truth. As a result, the leaning ability of the regression model is strengthened, and the comparative results shown in Table II and Table III also demonstrate that the innovation in MS-GAT is beneficial to improving the accuracy of traffic prediction.

On the other, the model architecture is critical to the performance of traffic prediction. The comparative experiments show that the predictive models with the multi-component architecture perform significantly better those without considering multi-components. The reason might be that the multi-component architecture takes advantage of ensemble learning, like what is observed in ASTGCN [43] and MS-GAT. However, they work differently. For ASTGCN, first, it adopts ChebNet [58] as a graph convolution operation to model spatial correlations, which makes it inferior to other predictive models using more excellent graph convolution networks, e.g., STSGCN [35] and STFGNN [34]. Second, it stacks a standard convolution layer in the temporal dimension to merge the information at the neighboring time slice, which makes it hard to gain a promising ability of modeling temporal correlation as Graph WaveNet [24] using dilated casual convolution. Third, it develops a very complicated spatial-temporal attention mechanism to capture the dynamic spatial and temporal correlations in the traffic network, which instead brings about a large amount of model parameters and makes it hard to train and easy to overfit. In contrast, MS-GAT pursues to fill these gaps. Except for designing a distinctive time-gating multi-component architecture, MS-GAT also focuses on the following aspects. First, by utilizing a first-order Chebyshev polynomial to simplify ChebNet [58] as our efficient GCN operation, MS-GAT significantly reduces the complexity of aggregating spatial information and achieves the approximate effect of ordinary higher-order ChebNet by stacking multiple *GCN* operations in MEAM. Second, MS-GAT employs TCN (i.e., temporal convolutional networks [54]) to capture complex relations between traffic conditions in the temporal dimension rather than using prior CNN- or RNN-based methods (e.g., STGCN [22]). Since TCN has the advantage of causal convolution and more flexible receptive field in sequence modeling, it thus performs better than CNNs and RNNs in aggregating temporal information. Third, MS-GAT emphasizes the significance the channel relations hidden among traffic conditions, which is verified benefitable for accurate traffic prediction. Lastly, MS-GAT develops a simple yet effective multi-dimensional self-attention scheme to generalize existing ordinary self-attention mechanism to all information dimensions of traffic conditions and to improve the flexibility of handling multi-dimensional data based on attention mechanism. The above analysis explains why MS-GAT performs best in the comparative experiments. Moreover, MS-GAT also maintains a competitive parameter scale that ensures the efficiency of training and performing the prediction.

C. The ablation study

To evaluate the effectiveness of three critical ingredients in the architecture of MS-GAT, we conduct ablation studies on datasets PEMSD3, PEMSD4, PEMSD7 and PEMSD8. Table IV shows their experimental results in terms of the MAE, RMSE and MAPE metrics. Fig. 6 presents the sensitivity of MAE to different settings of MS-GAT. We can draw the following conclusions from these results.

TABLE IV
THE ABLATION STUDY OF MS-GAT ON PEMSD3, PEMSD4, PEMSD7 AND PEMSD8 (BEST NUMBER PER ROW IS SHOWN IN BOLD).

Dataset	Metrics	TPC	TPC × 2	TPC × 3	TPC × 4	TPC × 5	no CAttention
			without/with TE	without/with TE	without/with TE	without/with TE	
PEMSD3	MAE	18.87	16.70/16.25	16.38/16.04	16.08/**15.68**	17.38/16.88	15.82
	MAPE(%)	32.04	17.79/17.55	17.22/16.98	16.88/**16.11**	17.06/17.21	16.37
	RMSE	29.06	27.11/26.74	26.87/26.62	26.84/**26.54**	29.97/29.07	27.05
PEMSD4	MAE	25.42	22.77/20.48	21.43/20.30	21.42/20.11	20.58/**19.56**	20.41
	MAPE(%)	21.94	15.95/14.12	15.45/13.70	15.24/13.70	14.20/**13.44**	14.39
	RMSE	37.95	35.63/32.34	33.67/32.17	33.61/31.98	32.95/**31.69**	32.73
PEMSD7	MAE	27.86	24.40/22.50	23.33/22.37	23.04/21.55	20.84/**20.47**	20.76
	MAPE(%)	16.18	10.66/9.67	10.51/9.71	10.02/9.35	9.13/**8.84**	9.01
	RMSE	41.53	38.07/35.67	36.29/35.64	35.95/34.68	34.17/**34.17**	34.08
PEMSD8	MAE	20.63	17.68/16.50	16.82/16.43	16.77/15.71	15.34/**14.78**	15.65
	MAPE(%)	18.89	11.44/10.73	11.02/10.81	11.32/10.43	10.21/**10.14**	11.21
	RMSE	30.39	27.82/25.96	26.39/25.75	26.27/24.74	24.73/**24.15**	25.32

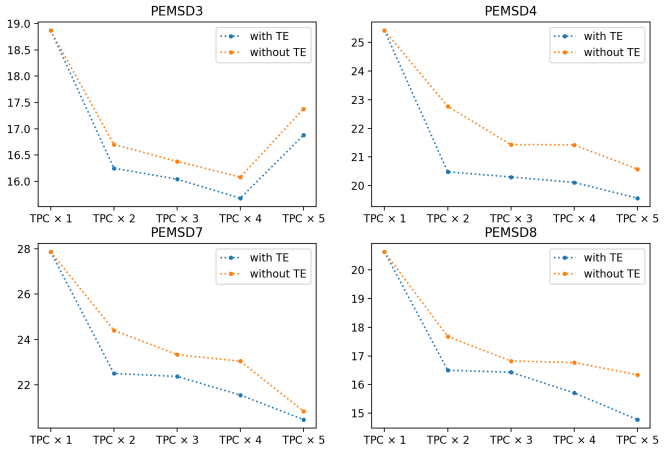

Fig. 6. The MAE sensitivity to different settings of MS-GAT on PEMSD3, PEMSD4, PEMSD7 and PEMSD8.

First, the multi-component structure is shown effective for spatio-temporal forecasting. Importantly, adding more components (*TPCs*) to MS-GAT can contribute to accurate prediction. However, it does not mean more components lead to the better results. As shown in Table IV, the model setting with four *TPCs* achieves the best performance on PEMSD3, while the prediction on PEMSD4, PEMSD7 and PEMSD8 requires five *TPCs* for the best. This is owing to that the traffic conditions of different physical road networks show specific spatio-temporal system complexities. When taking more historical horizons as the input of each *TPC* separately, the prediction of MS-GAT would worsen instead. In other words, more input sequences to model could be harmful or meaningless. Therefore, the number of *TPC* components is an important hyperparameter, which determines the performance of MS-GAT.

Second, the *TE* is proven very necessary for ensuring the model accuracy. The results in Table IV show the model settings with *TE* are consistently superior to those without *TE*. It demonstrates the traffic conditions at the past distinct time horizons have different correlation strengths with the one at the same future time horizon. Thus, the prediction effect can be significantly improved through deploying *TE* in MS-GAT.

Third, the channel relation is confirmed essential for modeling the dynamics of traffic systems. The results exhibited in the last column of Table IV draw from the case of skipping over the channel relation when MS-GAT is under its best model setting (i.e., the one that produces the bold number in each row of Table IV). Clearly, when those potential channel relations coupled among traffic signals are not taken care of, the performances of MS-GAT degrade significantly. The experimental results also substantiate the effectiveness of *CAttention* that applies attention mechanism to capture the channel relation of traffic signals. As a result, focusing on more interactive relations coupled within complex traffic systems contributes to improving the capability of predictive model.

D. Case study

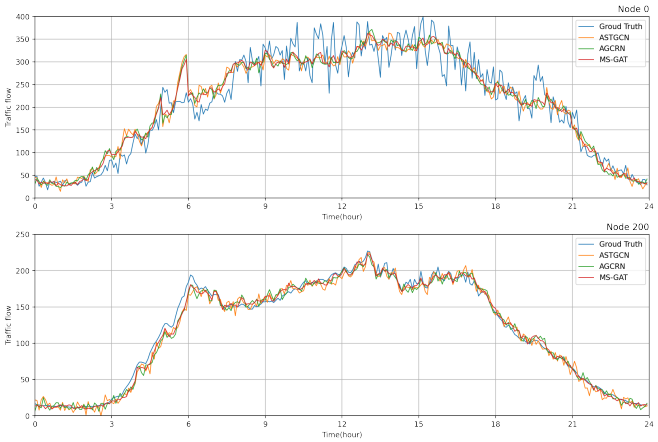

Fig. 7. A case study of the traffic flow prediction at Node 0 and Node 200.

To further investigate the performance of MS-GAT, we also conduct a case study to intuitively show its prediction effect. We select two traffic nodes Node 0 and Node 200 in PEMSD4 and plot their future 24-hour traffic flow prediction results separately in Fig. 7 to compare two state-of-the-art models ASTGCN and AGCRN against our proposed MS-GAT to carry out every 1-hour interval forecasting. It can be observed that,

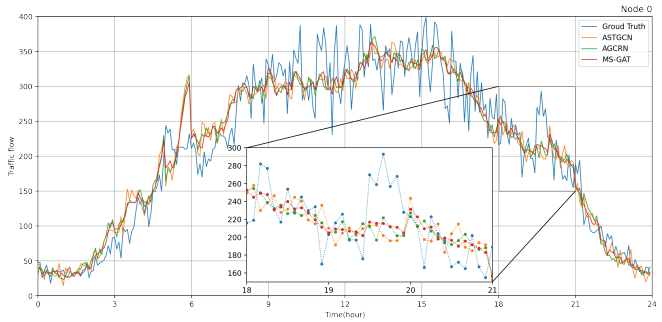

Fig. 8. Illustration of the prediction error at Node 0.

compared to those competitive models, the prediction curves of MS-GAT on both Node 0 and Node 200 are better aligned to the temporal trends of their ground-truth traffic conditions. Fig. 7 shows that the prediction error at Node 0 is significantly higher than that at Node 200. This is because there exist some frequent uncertain interference factors or random traffic events at Node 0 which induce the abrupt changes of traffic flow, thus exhibiting a kind of short-term recurrent blocked-unblocked traffic condition that is very challenging to be accurately estimated in practice. Fig. 8 further shows that MS-GAT is slightly superior to the other two models, indicating that MS-GAT could adapt to such abrupt changes of traffic conditions by its distinctive model design and parameter learning. Therefore, when performing the traffic flow prediction on the whole road network consisting of various geospatial nodes, the RMSE improvement of MS-GAT is not as significant as that in terms of MAE, which is consistent with our prior experimental results presented in Table II. The reason for that is the physical road network in PEMSD4 contains many traffic nodes with short-term recurrent blocked-unblocked traffic condition like Node 0. This also triggers a future direction, i.e., further improving the stability and robustness of traffic prediction models by exploring the influence of the sharp changes of traffic conditions over nodes.

VI. Discussion

Machine learning technique motivates the development of the ITS community in recent years. In particular, many deep-learning-based models including the aforementioned modern neural networks are successively proposed to meet the challenging traffic prediction task. The design and formulation of these impressive predictive models usually relies on some assumptions or specific observations on the recorded traffic signal data, and thus forming diverse network architectures of such learning models in terms of their respective inductive biases. Specifically, early deep-learning-based efforts started with treating traffic prediction as a sequence learning issue. They further naturally employed canonical RNNs (e.g., LSTM) to build the roughly similar backbone of their deep neural network architectures. The backbone models the dynamics of traffic system by focusing on the temporal dependencies hidden inside traffic conditions. After that, other studies also paid attention to the influence of potential spatial dependencies among traffic conditions on dynamic traffic system. They successively exploited well-established CNNs or emerging GNNs to enrich the above backbone for elaborating their distinctive model architectures. These models (e.g., the above baseline methods) all deliver excellent results, and thus a unified framework, called Spatial-Temporal Graph Neural Networks (STGNNs) [59], is formed for modeling dynamic traffic systems based on the pioneering works.

We argue that, like other complex systems, a traffic system is a coupled traffic network consisting of various channels and nodes, where traffic signals interact and are coupled with each other. Specifically, the dynamics of traffic conditions are associated with a variety of traffic signals and external environment (such as weather data), which interact and are coupled with each other as well. By viewing a traffic system as a coupled traffic network with multi-sensor traffic signals interacting and coupled, we can further model couplings in multi-modal traffic data and external data. We notice that the current widely-adopted framework STGNNs implies its inductive bias of modeling traffic systems likewise. Those models following the framework prefer to merely focus on two kinds of acquainted dependency relations (i.e., spatial and temporal relations) coupled inside traffic conditions. And they use an alternating neural network forward computation manner to pick up the two relations in turn, so as to excavate their coupling relations (i.e., so-called spatio-temporal relations) indirectly. Compared with them, MS-GAT attempts to improve the model bias by (1) shedding light on another noteworthy dependency relations (i.e., channel relations) that are also coupled inside traffic conditions, and (2) conceiving a synchronous neural network forward computation manner to separately identify all possible dependency relations of concern to us and explicitly assess their coupling strengths with each other to predictions. Ultimately, the above experimental results have also verified the superiority of our inductive bias to modeling dynamic traffic conditions.

Furthermore, we notice that a burgeoning neural network architecture, called Transformer [40], has flourished in the field of nature language processing (NLP) in the past few years. This architecture different from CNNs, RNNs and GNNs, can provide the relationships between features of different input data via the attention mechanism. Its power in sequence learning issue inspires other communities (e.g., computer vision, CV) to investigate the use of Transformer for their specific tasks. Not surprisingly, the novel architecture has also attracted a few ITS researchers to explore its potential in traffic predictions. The most recent ASTGNN [60] is such a impressive transformer-like predictive model. It leverages Transformer to conduct the forward computation of the STGNNs framework for long-term prediction, and elaborates its distinctive inductive bias to concretize the transformer-based encoder-decoder framework with valuable considerations (e.g., explicitly modeling the periodicity and spatial heterogeneity).

By experimental comparisons on the PEMSD4 dataset, ASTGNN performs slightly better than MS-GAT on the metrics of MAE, MAPE and RMSE by 5.6%, 7.9% and 2.1% respectively for the next 1-hour prediction task. Definitely, the transformer-based architecture of ASTGNN has played a great

role for the learning ability of the transformer-like model to make long-term predictions. It is because that massive multi-head attentions in Transformer with global receptive fields as well as unique embedding mechanisms of relevant information can effectively assist those transformer-based models to excavate more significant features for the sequence-to-sequence task. Meanwhile, ASTGNN also reveals its own inductive bias of modeling the dynamics of traffic conditions across both spatial and temporal dimensions. Concretely, except for taking the transformer-like framework as its backbone, ASTGNN also deliberately devises two important modules: (1) a trend-aware self-attention module that enables the self-attention being aware of the local temporal context, and (2) a dynamic graph convolution module that models the spatial dependencies. Notwithstanding, in contrast to MS-GAT, this kind of transformer-like predictive model also faces extra challenges, including: (1) *huge amount of calculation*. Due to complex learning mechanisms and overparametered networks in Transformer, those transformer-like predictive models are typically difficult to train. They depend on a huge number of training data. For instance, on our above experimental setting with 2080TI, ASTGNN need to cost around 220s to conduct an epoch of training, while MS-GAT just costs about 100s for every epoch; (2) *insufficiency of parallel capability*. The transformer-like predictive models like ASTGNN are all equipped with a heavy decoder that performs in an auto-regressive manner. It means that every output sequence of their decoders is generated one by one. Namely, the current prediction relies on the previous result. Instead, the decoding process of MS-GAT is developed to be a simple regressor based on neural network, which produces output sequences in a one-step manner. Its encoding process is designed to be a structure with multiple independent branches, where each branch is responsible for identifying a specific dependency relation coupled inside traffic conditions (e.g., spatial, temporal or channel relations). Importantly, each branch can be conveniently deployed on a separate computation resource (e.g., CPU or GPU) for achieving parallel acceleration.

To sum up, owing to its own architectural characteristics, MS-GAT is more applicable for short-term traffic prediction, and also suitable for the case of deploying parallel settings for pursuing real-time effect. Moreover, the newly proposed ASTGNN paves the way for transferring the success of pure transformer model in NLP to the traffic prediction task in ITS. Inspirited by it, we are also thinking about applying Transformer to MS-GAT, e.g., attaching a transformer-based decoder in MS-GAT for more accurate long-term prediction.

VII. Conclusions

Despite of the great success of prior works, traffic prediction is still a challenging task in ITS. By viewing a traffic system as a coupled traffic network, in this paper, we focus on the continuous signals from various channels and nodes interacting and coupling with each other in terms of temporal, spatial and spatio-temporal aspects of traffic conditions. The multi-aspect traffic signal couplings are modeled by a deep graph network MS-GAT, which characterizes multi-dimensional, time-evolving and multi-relational traffic interactions in the road network. Concretely, MS-GAT explicitly models the latent relations between diversified traffic signals in terms of *channel relations* and integrates them with *spatial* and *temporal relations* in a synchronous way into the neural traffic graph for forecasting. These are modeled by (1) a module MEAM stacking replicas of deep learning architectures to capture the complicated spatio-temporal dynamics of traffic conditions and their multi-relations and influence on future traffic conditions, and (2) a multi-component structure TPCs that adopts a time gated fusion mechanism to adaptively focus on the traffic conditions at different past stages. Further, MS-GAT maintains a small number of parameters with a convenient multi-dimensional self-attention scheme applicable to any data with multi-dimensional features. We substantially test MS-GAT on five real-world datasets against six state-of-the-art graph neural networks for traffic prediction, which show that MS-GAT outperforms the baselines.

MS-GAT is a general spatio-temporal forecasting framework, applicable to other spatio-temporal structured sequence forecasting scenarios, such as preference prediction in recommender systems and air quality forecasting, since it is characterized by providing a multi-view representation for each node in graphs and a multi-relational representation for graph-based complex systems. In ITS, in the case of the data from different traffic modes (e.g., bus, private car, bike) or external modes (e.g., passengers or weather) in an identical traffic region is available, MS-GAT can also attempt to handle these multi-modal data and their interactions through treating each modal data as a separate information channel to manipulate. We argue that there exist some potential causal associations between those available multi-modal data and expectations in traffic prediction tasks, such as the impact of current rainfall on future traffic flow in a road network, whereas MS-GAT just can deal with that.

Besides, MS-GAT can be also expanded in the following directions. First, we will further optimize the network structure and parameters to improve the prediction accuracy on sharp changes of traffic conditions that may be induced by external factors, e.g., weather or abnormal events. Second, like other spatio-temporal forecasting methods, we will also explore the mechanism of modeling the spatio-temporal dependencies for evolving graph structures.

References

[1] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, 2014.

[2] Cao, L., "Coupling learning of complex interactions," *Information Processing & Management*, 2015.

[3] C. Wang, F. Giannotti, and L. Cao, "Learning complex couplings and interactions," *IEEE Intell. Syst.*, vol. 36, no. 1, pp. 3–5, 2021.

[4] L. Cao, "Non-iid recommender systems: A review and framework of recommendation paradigm shifting," *Engineering*, vol. 2, no. 2, pp. 212–224, 2016.

[5] C. Zhu, L. Cao, and J. Yin, "Unsupervised heterogeneous coupling learning for categorical representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[6] J. Liu and W. Guan, "A summary of traffic flow forecasting methods [j]," *Journal of Highway and Transportation Research and Development*, vol. 3, pp. 82–85, 2004.

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

[8] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through kalman filtering theory," *Transportation Research Part B: Methodological*, vol. 18, no. 1, pp. 1–11, 1984.

[9] E. Zivot and J. Wang, "Vector autoregressive models for multivariate time series," *Modeling Financial Time Series with S-Plus®*, pp. 385–429, 2006.

[10] C.-H. Wu, J.-M. Ho, and D. Lee, "Travel-time prediction with support vector regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 2004.

[11] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.

[12] X.-l. Zhang, G.-g. HE, and H.-p. LU, "Short-term traffic flow forecasting based on k-nearest neighbors non-parametric regression," *Journal of Systems Engineering*, vol. 24, no. 2, pp. 178–183, 2009.

[13] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.

[14] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," in *Proceedings of the 2017 SIAM international Conference on Data Mining*. SIAM, 2017, pp. 777–785.

[15] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "Lstm network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[16] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[17] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," *arXiv preprint arXiv:1612.01022*, 2016.

[18] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[19] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[20] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, 2020.

[21] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *arXiv preprint arXiv:2101.11174*, 2021.

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

[23] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[24] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.

[25] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *arXiv preprint arXiv:2007.02842*, 2020.

[26] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, "Traffic flow prediction via spatial temporal graph neural network," in *Proceedings of The Web Conference 2020*, 2020, pp. 1082–1092.

[27] E. Cascetta, *Transportation systems engineering: theory and methods*. Springer Science & Business Media, 2013, vol. 49.

[28] Z. Pan, W. Zhang, Y. Liang, W. Zhang, Y. Yu, J. Zhang, and Y. Zheng, "Spatio-temporal meta learning for urban traffic prediction," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[31] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 240–254, 1994.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[33] S. Fang, Q. Zhang, G. Meng, S. Xiang, and C. Pan, "Gstnet: Global spatial-temporal network for traffic flow prediction." in *IJCAI*, 2019, pp. 2286–2293.

[34] L. Mengzhang and Z. Zhanxing, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," *arXiv preprint arXiv:2012.09641*, 2020.

[35] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 914–921.

[36] L. Hu, S. Jian, L. Cao, Z. Gu, Q. Chen, and A. Amirbekyan, "HERS: modeling influential contexts with heterogeneous relations for sparse and cold-start recommendation," in *AAAI'2019*, 2019, pp. 3830–3837.

[37] Q. Zhang, L. Cao, C. Zhu, Z. Li, and J. Sun, "Coupledcf: Learning explicit and implicit user-item couplings in recommendation for deep collaborative filtering," in *Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18*, 2018.

[38] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[39] W. Cheng, Y. Shen, Y. Zhu, and L. Huang, "A neural attention model for urban air quality inference: Learning the weights of monitoring stations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[41] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[42] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "Geoman: Multi-level attention networks for geo-sensory time series prediction." in *IJCAI*, 2018, pp. 3428–3434.

[43] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 922–929.

[44] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

[45] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, "Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting," *Transactions in GIS*, vol. 24, no. 3, pp. 736–755, 2020.

[46] X. Fang, J. Huang, F. Wang, L. Zeng, H. Liang, and H. Wang, "Constgat: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2697–2705.

[47] C. Park, C. Lee, H. Bahng, Y. Tae, S. Jin, K. Kim, S. Ko, and J. Choo, "St-grat: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 1215–1224.

[48] W. Chen, L. Chen, Y. Xie, W. Cao, Y. Gao, and X. Feng, "Multi-range attentive bicomponent graph convolutional network for traffic forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3529–3536.

[49] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers of Computer Science*, vol. 14, no. 2, pp. 241–258, 2020.

[50] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3656–3663.

[51] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-head r-cnn: In defense of two-stage object detector," *arXiv preprint arXiv:1711.07264*, 2017.

[52] K. W. Church, "Word2vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[54] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[55] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: mining loop detector data," *Transportation Research Record*, vol. 1748, no. 1, pp. 96–102, 2001.
[56] N. Ketkar, "Introduction to pytorch," in *Deep learning with python*. Springer, 2017, pp. 195–208.
[57] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*. Springer, 1992, pp. 492–518.
[58] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *arXiv preprint arXiv:1606.09375*, 2016.
[59] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.
[60] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, no. 99, pp. 1–1, 2021.

Jing Huang received Ph.D. in computer science from Huazhong University of Science and Technology, China in 2006. Before joining Wuhan University of Technology as an Associate Professor, he worked in Huazhong University of Science and Technology as a post-doctoral fellow. He visited University of Technology Sydney as a visiting scholar between August 2015 and August 2016. His research interests include but not limited to machine learning, data mining, intelligent transportation systems, pattern recognition and computer vision.

Kun Luo received his B.S. degree in logistics engineering from Wuhan University of Technology, Wuhan, China in 2020. He was a postgraduate in the school of computer science and technology, Wuhan University of Technology, China when this work was done. His research interests include machine learning and pattern recognition in intelligent transportation systems (ITS).

Longbing Cao (SM'06) received a PhD degree in pattern recognition and intelligent systems and another PhD in computing sciences. He is a Professor at the University of Technology Sydney and an Australian Research Council Future Fellow (professorial level). His current research interests include data science, data mining, machine learning, behavior informatics, artificial intelligence and intelligent systems, and their enterprise applications.

Yuanqiao Wen received Ph.D. in computer science from Huazhong University of Science and Technology, China in 2006. He is currently a Professor and Ph.D. supervisor in the National Engineering Research Center for Water Transport Safety, Wuhan University of Technology, China. He visited Technische Universiteit Delft as a senior visiting scholar between November 2016 and December 2016. Dr. Wen's research interests mainly include but not limited to intelligent transportation systems, data mining, artificial intelligence, and transport safety.

Shuyuan Zhong received his B.S. degree in computer science from Wuhan University of Technology, Wuhan, China, in 2019. He was a postgraduate in the school of computer science and technology, Wuhan University of Technology, China when this work was done. His research interests include deep learning and graph neural networks.