Graph Construction for Traffic Prediction: A Data-Driven Approach

James J. Q. Yu, *Senior Member, IEEE*

***Abstract*— Graph learning-based algorithms are becoming the prevalent traffic prediction solutions due to their capability of exploiting non-Euclidean spatial-temporal traffic data correlation. However, current predictors primarily employ heuristically constructed static traffic graphs in forecasting, which may not describe the latent traffic dynamics well. Existing attempts on dynamically generated traffic graphs also face challenges like prolonged model training time and undermined model expressibility. In this paper, a novel data-driven graph construction scheme based on graph adjacency learning is proposed for graph learning-based traffic predictors. The proposed scheme explores inter-time-series dependency with the graph attention mechanism to embed the sensor correlation in a latent attention space, which determines the correlation of any possible sensor pairs for traffic graph construction. Comprehensive case studies on three real-world traffic datasets reveal that the proposed scheme outperforms state-of-the-art static and dynamic graph construction baselines. Additionally, time-varying and sparse graph construction schemes are devised and assessed to boost the efficacy, and a hyper-parameter test develops guidelines for parameter and model architecture selection.**

***Index Terms*— Traffic graph construction, traffic prediction, graph attention, intelligent transportation systems, deep learning, data mining.**

I. Introduction

INTELLIGENT transportation systems (ITS) are among the vital infrastructure in modern smart cities with the rapid urbanization process [1]. Benefit from the boom of big data collected by a variety of sources, the community is embracing a rise in new data-driven solutions to transportation problems [2]. Within the diversity of ITS sub-domains, traffic prediction is among the essential support to the daily commuting and traveling of millions of people [3]. Accurate and timely traffic prediction data are highly valued in enhancing traffic management and implementing congestion prevention and remediation operations [4], [5]. Motivated by the indispensable role in smart city transportation, both academia and industry are devoting efforts to devising traffic prediction algorithms in favor of their significant social influence.

Among the plethora of recent literature on traffic prediction, deep learning methods have engrossed significant attention [6]. Due to the nature of traffic prediction being high-dimensional, irreducibly complicating, and massive in data volume, the utilization of deep learning methods is justified by its capability of modeling highly complex and non-linear functions from data. In the past a few years, ITS scholars are witnessing a gradual and steady transition of deep learning techniques from using recurrent neural networks for multivariate time-series forecasting to using convolutional neural networks for exploiting spatial data correlation, and more recently to adopting graph learning-based networks, e.g., graph convolution networks (GCN), to facilitate learning from traffic data aligned in non-Euclidean spaces [6], [7]. This transition advances the performance of traffic predictors by incorporating more latent information from the data but is also introducing a presumption — the inter-series traffic data correlation follows heuristically generated adjacency matrices, typically based on the distance between traffic sensors or the topology of underlying transportation networks [7]. Graph learning-based predictors rely heavily on the quality of traffic graphs derived from the adjacency matrices. Despite the improving predictions over canonical methods, there is no proof showing that these heuristically generated matrices lead to optimal performance [7], [8].

To rebut the presumption, a number of learning-based traffic graph construction approaches are proposed in the literature [7]. As will be detailed in the next section, these approaches can be generally classified into three categories. A most straightforward approach is to use a fully learnable matrix to substitute the original adjacency or Laplacian matrix during graph convolutions in the spectral domain [9]. For regularization, the fully learnable matrix is sometimes replaced by multiplying two learnable node embedding matrices [10], [11]. The second line of research exploits the spatial design of graph convolution operations and employs the attention mechanism to refine the heuristic adjacency matrix and develop a more precise nodal correlation [12], [13]. Finally, there is also a series of investigations concentrating on superimposing a learnable local graph Laplacian over the global one derived from the adjacency matrix [14], [15]. All three classes of approaches are reported to outperform graph learning-based predictors using static and heuristic traffic graphs.

However, there are still open challenges in constructing graphs for graph learning-based traffic predictors [6], [7]. While fully learnable adjacency matrices render predictors the largest model capacity, the corresponding parameter searching space is drastically extended, which leads to notably prolonged model training time. Furthermore, the learned matrices are not

Manuscript received June 19, 2021; revised October 15, 2021; accepted December 14, 2021. This work was supported in part by the Stable Support Plan Program of Shenzhen Natural Science Fund under Grant 20200925155105002 and in part by the Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation under Grant 2020B121201001. The Associate Editor for this article was M. Mesbah.

The author is with the Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: yujq3@sustech.edu.cn).

Digital Object Identifier 10.1109/TITS.2021.3136161

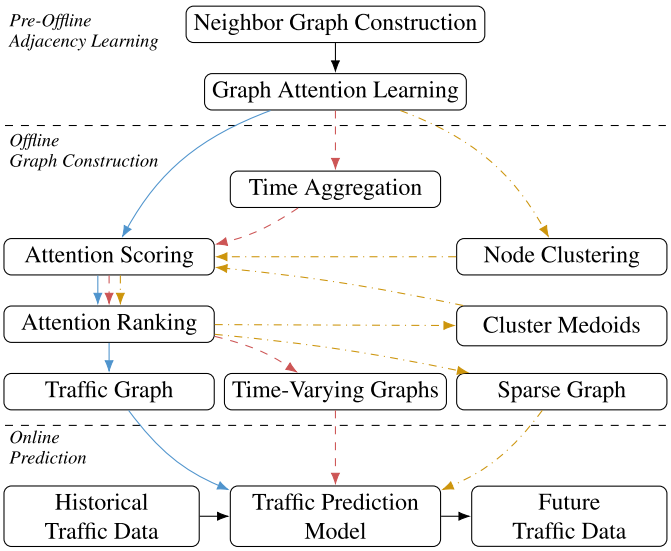


Fig. 1. Data flow of the proposed GALEN. The three alternative graph construction mechanisms are differently styled and colored.

guaranteed to be positive semi-definite, which cannot serve as the graph Laplacian and undermines the expressibility of graph learning models [16]. The attention-driven spatial graph convolution circumvents this positive semi-definite constraint but still suffers from its lengthy offline training and online inference time. Additionally, it can only work on existing nodal connections defined by the adjacency matrix despite that informative graph edges may be missing [17]. Last but not least, the local graph Laplacian and the aforementioned fully learnable Laplacian have minuscule differences from the perspective of model training, and the learned matrices lack interpretability.

To bridge the research gap and address these challenges, we devise a Graph Adjacency LEarning Network (GALEN) to construct traffic graphs for graph learning-based traffic predictors. Fig. 1 presents an outline of the proposed graph construction scheme. The proposed scheme can capture the inter-sensor correlation from raw historical traffic data using attention on graphs. As interpretable indicators of data dependency strength, The learned attention coefficients are employed to construct traffic graphs for graph learning-based traffic predictors. GALEN tackles previous challenges by offloading the additional model training effort in an individual pre-offline adjacency learning phase, effectively reducing the predictor model training and inference complexity. This model is also not restricted to available edges derived heuristically as canonical graph attention-based models do. Note that the primary objective of this work is not to devise a new traffic predictor, but instead to rethink the physical meaning of the graph attention values and to use them to dynamically create graphs for traffic prediction. This is among the pioneer work of traffic graph construction using attention. The proposed scheme can be applied to existing and future graph learning-based traffic predictors with static or dynamic graphs as an add-on, leading to potential performance improvement almost for free. The main contribution of this work is summarized as follows:

- We propose an adjacency learning mechanism to capture the inter-sensor correlation from the raw data, which is embedded by attention coefficients for generalization.
- We devise a graph construction scheme to build traffic graphs for graph learning-based predictors. The graphs are computationally efficient and provide outstanding abstraction on the underlying data correlation.
- We design a time-varying dynamic graph construction algorithm and a sparse graph construction mechanism to cope with traffic dynamics and achieve quadratic speed-up.
- We carry out comprehensive case studies on three large-scale real-world traffic datasets with three state-of-the-art graph learning-based traffic predictors to show the efficacy of the proposed graph construction scheme.

The remainder of this paper is organized as follows. Section II presents a literature study on traffic graph construction and graph link prediction state-of-the-arts. Section III elaborates on the proposed adjacency learning mechanism with a comprehensive analysis of the design principle and alternative designs. Section IV introduces the proposed graph construction scheme together with its time-varying and sparse graph construction algorithms. Section V demonstrates the numerical results of the case studies with detailed discussions. Finally, this paper is concluded in Section VI.

II. Related Work

Graph learning-based traffic prediction has received much research attention in the past few years. In this section, we present a brief summary on the development of such predictors, emphasizing their utilization of static or dynamic traffic graphs. Interested audiences are referred to recent surveys for a more thorough investigation in this context [6], [7].

In the last decade, the renaissance of deep learning gathered attention from the transportation industry and research community to re-visit transportation problems from the data-driven perspective [2]. Deep learning techniques overcome difficulties in handcraft feature engineering and resolve the linearity and stationary assumptions of statistical methods such as ARIMA and Kalman filter [7]. As introduced in Section I, learning-based traffic predictors are experiencing a shift from recurrent neural networks to convolutional neural networks and, more recently, to graph learning-based neural networks. Among them, the recurrent neural network family, including but not limited to the widely adopted long short-term memory (LSTM) and gated recurrent unit (GRU), concentrates more on exploiting the temporal correlation within the traffic data [18]. In the meantime, convolutional neural networks and variants are generally more focused on capturing the spatial latent correlation embedded in grid-based traffic systems [6]. Considering that transportation networks, e.g., road and subway networks, are typically graph-structured, graph neural networks are natural fits to handle such complicated data correlation in non-Euclidean space [7].

Contemporary graph learning-based traffic predictors come into two flavors regarding the underlying convolution

operation, namely, spectral and spatial graph convolutions. The former performs an eigendecomposition of the graph Laplacian to help deep learning models understand the graph structure. Common representatives are ChebNet [19] and GCN [16] who employ k-order Chebyshev polynomials of graph Laplacian $T_k(2\mathbf{L}/\lambda_{\max} - \mathbf{I})$ and their first-order approximation $\deg(\hat{\mathbf{A}})^{-\frac{1}{2}}\hat{\mathbf{A}}\deg(\hat{\mathbf{A}})^{-\frac{1}{2}}$ to dictate how data signals diffuse in the graph, respectively, where $\mathbf{L}$ is the graph Laplacian, $\lambda_{\max}$ is the maximum eigenvalue, and $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the graph adjacency matrix with self-loops. The latter works on nodal locality and understands the properties of a node based on its local neighbors according to the adjacency $\mathbf{A}$. In this context, GraphSage [20] and graph attention network (GAT) [17] are widely recognized as effective alternatives of spectral graph convolutions.

A major difference of graph learning-based traffic predictors is the explicit adoption of non-Euclidean topology information, typically embedded in the graph adjacency [8]. While sensors that produce traffic data may be fixed by physical constraints, traffic predictors have control over the design of the graph, either static or dynamic. Current static traffic graphs are primarily constructed data-agnostically, following 1) the road network topology, including connection [21]–[23], transportation connectivity [24], and direction matrices [25], 2) the spatial closeness between sensors, e.g., the famous thresholded Gaussian kernel [26] commonly used for METR-LA and PeMS datasets, or 3) the traffic pattern and functional similarity [24]. Nevertheless, static graphs all impose the assumption that these heuristically generated matrices can capture the non-Euclidean spatial-temporal data correlation, which is still disputable. In this context, dynamic graphs provide alternatives to these predictors.

Constructing dynamic graphs for graph learning-based traffic predictors can be generally classified into three categories. The first category adopts the hypothesis that graph structure directly learned from raw data outperforms heuristic ones and uses a fully learnable network parameter matrix $\mathbf{S}$ or two node embedding matrices $\mathbf{E}_1\mathbf{E}_2^\top$ to replace the adjacency $\mathbf{A}$, see [9]–[11], [27] for some examples. This approach, however, cannot ensure that the learned matrix is semi-definite, let alone a feasible graph Laplacian, rendering the theoretical supports to spectral graph convolution invalid. Additionally, the learnable matrices remarkably expand the parameter searching space, leading to prolonged model training time. The second category employs the attention mechanism to adaptively distinguish neighbors from each other, making the traffic graph dynamic. A most prominent technique falling into this category is to utilize GAT and siblings to formulate learning models, see [12], [28]–[31] for some examples. Nonetheless, such approaches are based on a pre-defined graph topology to enhance nodes with larger impact and fade out the rest. They cannot establish new edges within the traffic graph, thus limiting their graph construction efficacy. Finally, the third category incorporates learnable local Laplacian matrices in addition to the heuristically generated static ones to refine the static graphs, see [14], [15], [32] for some examples. While these approaches partially resolve the semi-definite issue of the first category, the parameter searching space is as ample, and spatial graph convolution cannot benefit from this graph Laplacian-oriented modification.

In this work, inspired by the interpretability of the attention mechanism, we propose a new graph construction scheme for graph learning-based traffic predictors. The proposed scheme overcomes the aforementioned challenges and can be applied in existing and future graph learning-based traffic predictors as an addon for possible performance improvement.

III. Graph Adjacency Learning Network

To overcome the challenges in graph construction for traffic prediction, we propose GALEN to construct traffic graphs based on the historical data dynamically. Different from the existing efforts, the proposed GALEN model constructs traffic graphs from a new perspective. GALEN is capable of extracting the interdependence among traffic sensors from the raw data instead of the traffic network structure. Additionally, GALEN performs the extraction in an offline manner, distinguished from subsequent traffic prediction learning to alleviate the model training complexity.

In this section, we first present an overview of GALEN with illustrations on the design principle. We then elaborate on the key concept that drives the model to capture traffic data correlation, i.e., attention on graph. Subsequently, we present the detailed learning process of GALEN.

A. GALEN Overview

The main objective of this model is to train a data-driven deep learning model to capture transferrable interdependence information among traffic sensors. Such information is capable of identifying whether a latent but strong correlation can be observed between pairs of traffic sensors based on their sampled time-series. GALEN follows the hypothesis that if two traffic sensors demonstrate significant correlation according to the historical data, they shall be connected when constructing traffic graphs for traffic prediction with graph neural networks. Note that this hypothesis serves as the foundation of the contemporary graph-based traffic prediction approaches, in which traffic information is propagated along the connections, see [33], [34] for some examples.

Fig. 2 presents the architecture of the proposed data-driven GALEN model for traffic graph construction. There are two major phases in GALEN to construct graphs for traffic prediction, i.e., *adjacency learning* (Section III-C) and *graph construction* (Section IV-A), roughly resembling the training and inference phases of typical learning systems. In the former phase, sensor correlations are extracted and resembled by the attention mechanism (Section III-B). Stronger dependencies are perceived if the attention mechanism concentrates only selected edges on the sub-graph of sensors. We utilize this characteristic to learn the latent adjacency information among traffic measurements, which can be further adopted to construct graphs for traffic prediction (Section IV-A).

B. Attention on Graph

As previously introduced in Section I, the recent state-of-the-art performance of data-driven traffic prediction is primarily developed based on graph learning techniques. The

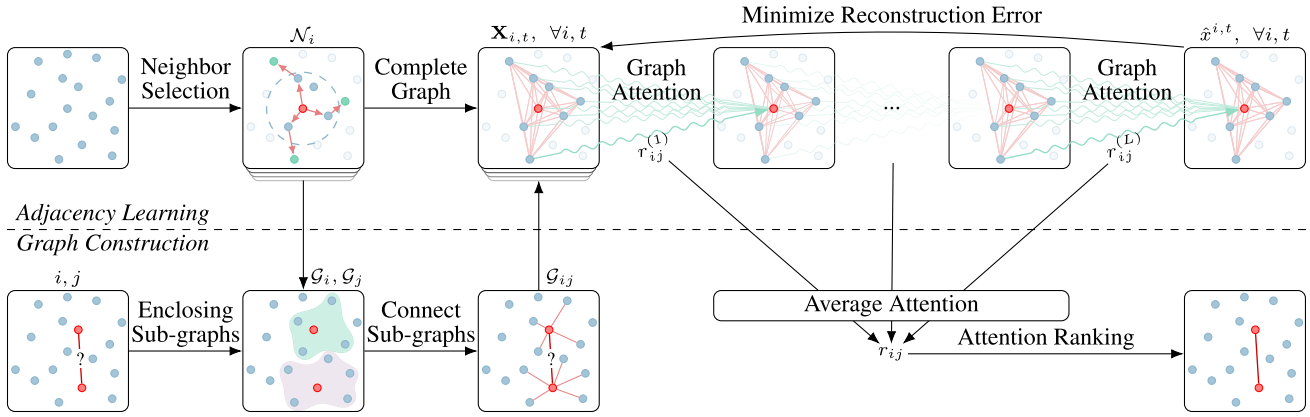


Fig. 2. Overview of GALEN on adjacency learning and graph construction.

literature focuses on investigating the non-Euclidean data correlation among nodes, which are the counterparts of traffic sensors or road segments in a transportation network. Nonetheless, GCN — arguably the most popular graph learning model for traffic prediction — suffers from two limitations that hinder its efficacy: 1) subpar performance on inductive learning tasks due to the embedded static adjacency matrix, and 2) all neighboring nodes are equal, which deviates from the nature of transportation where connecting roads can have a different influence, e.g., arterial versus collector roads.

In view of these two defects, GAT [17] is a notable alternative that has been recently adopted in the context. By incorporating the attention mechanism initially designed for neural machine translation, GAT achieves better neighbor aggregation through adaptively learning the weight (attention coefficient) for each neighboring node. Besides, the attention also grants partial model interpretability to typical graph learning models [8], which serves as the basis of GALEN. Considering a graph $\mathcal{G}(\mathcal{N},\mathcal{E})$, GAT defines the *attention coefficient* on an edge (i,j) as follows:

$$e_{ij} = \text{LeakyReLU}\left(\vec{\mathbf{a}}^{\top}\left[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j\right]\right), \quad \forall (i,j)\in\mathcal{E}, \quad (1)$$

where $\vec{h}_i$ and $\vec{h}_j$ are the node features of nodes i and j, respectively, $\vec{\mathbf{a}}$ and $\mathbf{W}$ are the learnable *attention vector* and weight matrix, respectively, $\|$ denotes concatenation, and LeakyReLU$(\cdot)$ is the Leaky Rectified Linear Unit [35]. This attention coefficient indicates the importance of node j's feature to node i, which resembles the influence of sensor j on its connecting sensor i. Furthermore, the influences on the same node i are normalized using the softmax function for easy computation and comparison as follows:

$$\alpha_{ij} = \text{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(e_{ik})}, \quad (2)$$

where $\mathcal{N}_i = \{k\in\mathcal{N} : (i,k)\in\mathcal{E}\}\cup\{i\}$ is the self-containing neighboring set of node i. With the relative influence of neighboring nodes, the new node feature of i, denoted by $\vec{h}'_i$, can be aggregated by

$$\vec{h}'_i = \sigma\left(\sum\nolimits_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{W}\vec{h}_j\right), \quad (3)$$

where σ is a pre-determined non-linear activation function, and ReLU is adopted herein. In this process, the weight matrix $\mathbf{W}$ is reused from (1) to reduce the model complexity. This aggregation resembles the intuition that the traffic speed or flow of a road can be determined by its connecting ones of different impact due to the spatio-temporal correlation of transportation networks [6].

C. Adjacency Learning

When GAT is employed in time-series regression tasks (e.g., traffic prediction), the model training process learns the inter-dependency between nodes and their neighbors explicitly embedded in attention coefficients. Connecting roads with greater influence render their corresponding α_{ij} values larger. Previous research takes advantage of this characteristic and follows an intuitive "Previous Data + Adjacency = Future Data" equation to perform traffic prediction, see [10], [11] for some examples. Meanwhile, a hypothesis is developed by observing this equation: adjacency[1] information can be obtained from both previous and future traffic data. GALEN proposes an adjacency learning phase to reconstruct this information.

In the adjacency learning phase, two schemes are launched sequentially to reconstruct the traffic data dependency, namely, *neighbor graph construction* and *graph attention learning*. Given a collection of N traffic sensors with their position and optional the underlying traffic network topology, GALEN first employs a neighbor selection scheme to construct N complete sub-graphs.[2] In particular, each sub-graph starts with an arbitrary node i. All other nodes within a pre-defined distance d from i are gathered to formulate a neighbor candidate set $\mathcal{N}_i^{\text{near}}$. If nodes correspond to road segments instead of traffic sensors, their midpoints are employed in this step. Furthermore, the k-hop neighbors of i, denoted by set $\mathcal{N}_i^{\text{hop}}$, is incorporated as candidates granted that the relevant information (i.e., road connectivity) is available. Otherwise,

[1] "Adjacency" here and in the sequel refers to the traffic data dependency in non-Euclidean space instead of the canonical road connectivity.

[2] A complete (sub-)graph is a graph in which each pair of graph nodes is connected by an edge.

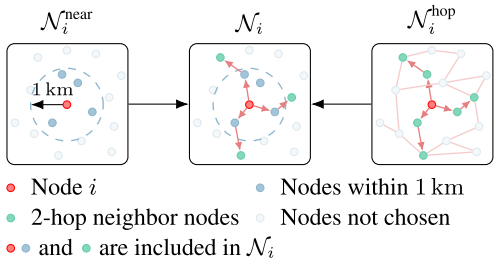


Fig. 3. A illustration of the self-containing neighbor set of node i.

$\mathcal{N}_i^{\text{hop}} = \emptyset$. The self-containing neighbor set of node i is accordingly developed by $\mathcal{N}_i = \mathcal{N}_i^{\text{near}} \cup \mathcal{N}_i^{\text{hop}} \cup \{i\}$, which is later used to override its original definition in (2) and (3). Fig. 3 depicts an example of a neighbor set with nodes within 1 km and 2-hop neighbors. Finally, an *enclosing sub-graph* of i is constructed as $\mathcal{G}_i(\mathcal{N}_i, \mathcal{E}_i)$, where $\mathcal{E}_i = \{(i, j) : \forall i, j \in \mathcal{N}_i\}$. The generation of these sub-graphs is critical to the subsequent graph attention learning scheme, which presumes that near-distance data dependency of i is within the coverage of $\mathcal{N}_i$.

In the second scheme of adjacency learning, a deep neural network based on stacked GAT is adopted to predict the traffic data and exploit the latent node adjacency information. Let $\mathbf{X} = (x_{i,t}) \in \mathbb{R}^{N \times |\mathcal{T}|}$ be the available traffic data of the transportation network during the discrete time horizon $\mathcal{T} = \{0, -1, -2, \cdots\}$. Given a sub-graph $\mathcal{G}_i$ and an arbitrary time t, the past traffic data within T time instances from t can be aggregated as $\mathbf{X}_{i,t} = (\vec{x}_{i,t-T}^{\top}, \vec{x}_{i,t-T+1}^{\top}, \cdots, \vec{x}_{i,t-1}^{\top}) \in \mathbb{R}^{|\mathcal{N}_i| \times T}$, where $\vec{x}_{i,t} = (x_{j,t})_{j \in \mathcal{N}_i}$. Subsequently, $\mathbf{X}_{i,t}$ is input into an L-layer GAT model to predict the following traffic data at t, i.e., $\vec{x}_{i,t}^{\top}$. In this model, (3) is calculated L times for every node $j \in \mathcal{N}_i$, in which the l-th layer takes the output of the previous layer $\vec{h}'_j$ as its input $\vec{h}_j$, i.e., $\vec{h}_j^{(l)} \equiv \vec{h}_j^{(l-1)'}$ where the super-indices indicate the layer index. Adopting $\mathbf{X}_{i,t}$ as the model input by $\vec{h}_j^{(1)} \equiv (x_{j,t-T}, x_{j,t-T+1}, \cdots, x_{j,t-1})^{\top}, \forall j \in \mathcal{N}_i$ and $\vec{x}_{i,t}^{\top}$ as the output by $\vec{h}_j^{(L)'} \equiv (x_{j,t})$, all learnable parameters $\vec{\mathbf{a}}$ and $\mathbf{W}$ can be trained by minimizing the L^2 predictive error

$$\mathcal{L}(\vartheta, i, t) = \sum\nolimits_{j \in \mathcal{N}_i} \|x_{j,t} - \hat{x}_{j,t}\|_2, \tag{4}$$

where ϑ is the collection of all learnable parameters in the L GAT layers and $\hat{x}_{j,t}$ is the reconstructed prediction of ground truth $x_{j,t}$ using $\mathbf{X}_{i,t}$. Consequently, the optimal layer-wise attention vector $\vec{\mathbf{a}}^{(l)}$ can be obtained.

There is one key observation on the model training process. Equations (1) and (3) suggest that only the attention coefficients between connecting roads or sensors are computed and employed in predicting future traffic data. Nonetheless, (1) does not mathematically prohibit one from calculating the attention coefficient of any arbitrary pair of nodes in the graph. In this equation, $\mathbf{W}$ and $\vec{\mathbf{a}}$ are not syntactically mergeable: $\mathbf{W}$ serves the purpose of casting the nodal features $\vec{h}_i$ into an embedding space, while $\vec{\mathbf{a}}$ focuses on deriving the correlation between two embeddings [17]. This nature offers the possibility of inductive attention learning, where the edge attention is calculated on nodes and graphs that are unseen during training. This is one of the major breakthroughs of GAT over other graph learning models and is utilized to construct traffic graphs in GALEN.

D. Discussion

There are also alternative designs of the aforementioned adjacency learning, e.g., including all the N nodes in $\mathcal{N}_i$ or removing edges in $\mathcal{E}_i$ that do not involve i. Notwithstanding, GALEN explores the node adjacency information following the design in Section III-B considering the following reasons. When all nodes are included in the enclosing "sub-graph", GAT degenerates into a global graph attention model, arguably equivalent to a multilayer perceptron. This design leads to significantly higher computational complexity, and the convergence capability is notably undermined due to the increased data volume and complexity. Furthermore, the invariant graph structure poses a great challenge in the model generality, which is the key to inductive learning. It is true that the "boundary" nodes have relatively less number of neighbors than others if only the distance metric is employed. However, this can be countered by adaptively increasing the d value for a larger coverage. Adopting a global graph attention may also mitigate this issue, but the previously discussed training difficulty becomes a more significant challenge. On the other hand, using smaller enclosing sub-graphs reduces the model capacity requirement, improves the graph structure variety, and helps reducing overfitting by data augmentation. Removing inter-connections in $\mathcal{E}_i$ also removes the data inter-dependency among nodes in $\mathcal{N}_i$, rendering difficulties in convergence. While the proposed adjacency learning is based on the distance and/or topological connectivity, the subsequent graph construction phase to be introduced in the next section relax the limitation of using only "local" dependency, enforcing exploitation of long-range data correlation. Offline experiments on these alternative designs also indicate that the models can barely converge to much inferior performance and the learned attention vectors are not applicable to the subsequent graph construction phase.

It is also worth noting that this adjacency learning is not without limitations. Among others, a significant issue is that the size of nodal neighbor set is heuristically determined by two hyperparameters d and k as elaborated before. While extensive hyperparameter search as will be demonstrated in Section V-E can develop guidelines for setting their values, it imposes notable computation burden to achieve the best performance. Therefore, developing a self-adaptive neighbor set generation scheme is a good addition to GALEN, and we plan to further investigate this topic in future research.

IV. Graph Construction for Traffic Prediction

With the previous adjacency learning phase, GALEN is capable of extracting the relative influence between pairs of traffic sensors or road segments from a data-driven perspective. In this section, we propose a novel graph construction algorithm based on the graph attention learned from historical data. We then discuss the possibility of time-varying graph

construction and its integration to existing traffic prediction models. Finally, we devise a sparse graph construction mechanism to alleviate the computation burden of GALEN.

A. Graph Construction

Credited to GALEN's graph attention learning capability, the model can determine the influence between any pairs of nodes in a new sub-graph within the transportation network. Accordingly, GALEN constructs traffic prediction graphs following an enumerative approach.

In particular, we consider a set of sensors or roads $\mathcal{N}$, the corresponding historical traffic data $\mathbf{X}$, and the optional underlying road connectivity. GALEN iterates over all possible node pairs in $\{\forall i, j \in \mathcal{N} : i \neq j\}$ and determines whether a uni-directional connection shall be created from j to i. The model adopts the neighbor graph construction scheme elaborated in Section III-C to respectively create two enclosing sub-graphs $\mathcal{G}_i$ and $\mathcal{G}_j$. These two sub-graphs are subsequently connected to further create a new sub-graph $\mathcal{G}_{ij}\left(\mathcal{N}_{ij}, \mathcal{E}_i \cup \mathcal{E}_j \cup \{(i,j),(j,i)\}\right)$, where $\mathcal{N}_{ij} = \mathcal{N}_i \cup \mathcal{N}_j$. In principle, $\mathcal{G}_{ij}$ is constructed by merging $\mathcal{G}_i$ and $\mathcal{G}_j$ with a bi-directional bridge between nodes i and j.

Subsequently, we select R different time instances $t \in \mathcal{T}$. The respective past traffic data of $\mathcal{N}_{ij}$ are input into the trained L-layer GAT using the connectivity of $\mathcal{G}_{ij}$ and develops the corresponding predictions $\{\hat{x}_{k,t}\}_{k\in\mathcal{N}_{ij}}$ using (3). At each time instance t, we have two sets of attention-related information:

1) Softmaxed layer-wise attention coefficients $\alpha_{ij}^{(l),t}$ indicating the influence of nodes j on i at time t, which is calculated by (2).
2) Node i reconstruction percentage error $\epsilon_{i,t} = (\hat{x}_{i,t} - x_{i,t})/x_{i,t}$ between prediction $\hat{x}_{i,t}$ and ground truth $x_{i,t}$. This error evaluates the reliability of the influence on i.

The influence and reliability are jointly considered by defining an *attention score* of a possible edge from j to i as follows:

$$r_{ij} = \frac{1}{LR}\sum_{l=1}^{L}\sum^{t}\alpha_{ij}^{(l),t}\epsilon_{i,t}, \tag{5}$$

which is the average value of all layer-wise attention coefficients multiplied by the corresponding reconstruction percentage error.

The construction of graph edges follows the intuition of graph-based traffic prediction that strong influence shall be represented as an edge. Therefore, we sort all r_{ij} for every i, j pair within $\mathcal{N}$ in descending order, and the ones with top $S \times N$ scores are adopted to construct the edge set $\mathcal{E}$ of $\mathcal{N}$, where S is a control parameter regulating the degree of each node. Consequently, a traffic graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ is created for traffic prediction, whose adjacency matrix can be naturally derived. By this graph construction process, we note that both short- and long-range data correlations can be validated thanks to the iteration over $\{\forall i, j \in \mathcal{N} : i \neq j\}$, regardless of the corresponding node distance or physical connectivity. Two unrelated nodes on the surface can also be connected if they exhibit similar correlation as neighboring ones do. We demonstrate the efficacy of such data correlation in the subsequent case studies.

One may note that as all edges in $\{\forall(i,j) \in \mathcal{N}\times\mathcal{N} : i \neq j\}$ have attention scores r_{ij}, it is possible that the constructed traffic graph comprises virtual connections between roads that are not physically connected. Such connections are indeed important in traffic prediction as they indicate strong traffic dynamics correlations learnt from data. Contemporary geometric deep learning techniques, e.g., graph convolution and graph attention networks, rely on these connections to perform information propagation, see [36], [37] for some examples.

B. Time-Varying Graph

In the previous sub-section, we proposed the graph construction algorithm based on the trained L-layer GAT, where the attention coefficients on randomly selected time instances are averaged for computation. Albeit this scheme can effectively capture the time-invariant adjacency information that describes the general spatio-temporal data correlation in the traffic data, we argue that such correlation is indeed time-varying. For instance, peak hours demonstrate a strong correlation between residential and commercial/industrial regions, while distant shopping areas may exhibit high similarity during weekends.

To capture the time-varying traffic data correlation, a simple yet effective time-varying graph construction algorithm is devised whose potency will be evaluated in Section V-B. Instead of randomly selecting R time instances from the past time horizon $\mathcal{T}$, we aggregate all time instances in each hour of a day and compute hourly reliability scores by averaging the corresponding layer-wise attention score within the hour, denoted by $r_{ij}^{\mathrm{HoD,DoW}}$ where HoD and DoW are placeholders for the hour of the day and day of the week, respectively. For example, if $\mathbf{X}$ contains the traffic data of the past four weeks, $r_{ij}^{\mathrm{18,Tue}}$ is calculated by taking the mean of $\frac{1}{L}\sum_{l=1}^{L}\alpha_{ij}^{(l),t}\epsilon_{i,t}$ values at all time instances from 6 PM to 7 PM (exclusive) in the past four Tuesdays. The subsequent edge construction scheme still follows the original design, and multiple time-varying graphs $\mathcal{G}^{\mathrm{HoD,DoW}}$ are created and used to predict traffic data during specific periods.

C. Sparse Graph Construction

As introduced in Section IV-A, GALEN computes $N \times (N-1)$ attention scores for all possible edges in $\mathcal{G}$. Since each calculation of r_{ij} requires R forward-passes on the L-layer GAT, obtaining the scores can be highly computationally expensive when N is large (e.g., greater than a thousand). Although the graph construction process is conducted offline, a lengthy computation time may hinder GALEN from frequent updates with the latest traffic data.

To alleviate the computation burden of GALEN, we propose a sparse graph construction mechanism for fast edge creation. We consider an intuitive but effective principle that similar traffic sensors can be grouped as a meta-sensor, and data correlation can be abstracted among meta-sensors with minuscule loss. Following this principle, we first cluster all traffic sensors into $K = \lceil\sqrt{N}\rceil$ equal groups using k-medoids algorithm with respect to the historical traffic data at the beginning of the graph construction phase. Subsequently, the edges between nodes within each group are created using the aforementioned

TABLE I
SUMMARY OF PEMS-BAY, NAVINFO BEIJING, AND NAVINFO SHANGHAI DATASETS

	NavInfo Beijing	NavInfo Shanghai	PeMS-BAY
No. sensors	1569	1830	325
Avg. length	332.97 m	342.83 m	6.93 km†
Avg. speed	35.03 km/h	34.37 km/h	100.78 km/h
Std. speed	12.80 km/h	13.57 km/h	15.44 km/h

†The average distance between sensors.

graph construction algorithm, and K disjoint union of graphs is produced. Finally, the traffic graph $\mathcal{G}$ is constructed by creating edges on all possible pairs of medoids in the K groups with the same algorithm. This simplification can significantly reduce the total number of GAT forward-passes from $N^2 - N$ to roughly $N^2/K - K$, i.e., approximately $\sqrt{N}$ times speed-up. The time efficiency and performance drop of this sparse graph construction mechanism will be investigated in Section V-D.

V. CASE STUDIES

In this work, we propose a novel GALEN model to construct traffic graphs for graph learning-based traffic prediction. We carry out a series of comprehensive case studies on three real-world datasets to evaluate the efficacy of GALEN using three state-of-the-art graph learning-based traffic prediction approaches. Specifically, we first investigate the performance improvement of GALEN over existing static and dynamic graph construction schemes. Then, we conduct an ablation study to verify the necessity of multiple constituting components of GALEN. Subsequently, the sensitivity of GALEN hyper-parameters is evaluated. Finally, we demonstrate the efficiency of the proposed sparse graph construction with meta-sensor clustering.

A. Experimental Configurations

We employ three practical traffic speed datasets for testing in this work: NavInfo Beijing,[3] NavInfo Shanghai, and PeMS-BAY[4] datasets. In particular, the NavInfo Beijing and Shanghai datasets comprise proprietary floating car data of Beijing, China and Shanghai, China from Jan. 2019 to Jun. 2019 with a 5 min sample interval. The respective default adjacency matrices are derived from the road connectivity information, and sensors are assumed to be midway-located. The PeMS-BAY dataset is generated based on 325 traffic speed sensors in the Bay area of California, US from Jan. 2017 to May 2017. Following previous work [10], [34], all traffic speed readings are aggregated into 5 min windows. The default adjacency matrix is constructed by the pairwise road network distances with a thresholded Gaussian kernel, and there is no connectivity information available. Z-score normalization is applied to all datasets, which are split in chronological order with 70% for training, 10% for validation, and 20% for testing. A summary of the datasets is presented in Table I.

[3]https://nitrafficindex.com/
[4]https://pems.dot.ca.gov/

We adopt the widely used Root Mean Square Error (RMSE) and Mean Average Percentage Error (MAPE) as the performance metrics in all case studies. Data from the past 12 time instances are used to predict 5 min, 15 min, 30 min, and 60 min ahead traffic data for all dataset and predictor combinations. Unless otherwise stated, the distance of neighboring sensors d is set to double the average road length, the number of neighboring hops $k = 2$, the number of GAT layers in GALEN $L = 3$ with 128 neurons in each layer, the number of attention-averaging time instances $R = 8$, and the average number of nodal edges $S = 8$. GALEN is optimized by Adam optimizer with a learning rate 10^{-4}. All case studies are conducted on computing servers with two Intel Xeon E5 CPUs and 128 GB RAM. The simulation is developed in Python and PyTorch. Eight nVidia GTX 2080 Ti GPUs are employed on each server for neural network computing acceleration.

B. Quantitative Results

In this case study, we employ the proposed GALEN on all three testing datasets to forecast traffic speed data. As GALEN produces traffic graphs for graph learning-based traffic prediction approaches, we adopt the following three state-of-the-art traffic predictors and assess the performance improvement developed by incorporating GALEN:

- T-GCN [33]: Temporal Graph Convolutional Network (T-GCN) model is a traffic prediction approach combining GCN and GRU to build the spatial-temporal correlation among traffic data.
- STSGCN [36]: Spatial-Temporal Synchronous Graph Convolutional Network (STSGCN) captures the complex localized spatial-temporal traffic correlation by a spatial-temporal synchronous modeling mechanism, which effectively learns from the heterogeneities in localized spatial-temporal graphs.
- GA2 [22]: Generative Adversarial Graph Attention (GA2) network captures the geometric traffic data dependency with graph convolution and attention mechanisms, and the temporal data correlation is extracted and expanded using the encoder-decoder architecture within a generative adversarial learning framework.

For all three predictors, we adopt the source code provided in the respective literature with minuscule changes. As all approaches by default employ the static traffic graph developed by either geographical adjacency or connectivity information, multiple variants with different graph construction schemes are incorporated for a more comprehensive comparison. Besides GALEN and the default static adjacency matrices as introduced above, the following dynamic graph construction scheme baselines are employed:

- Attn: Instead of the typical spectral graph convolution adopted in the original formulation, GAT is incorporated to capture the inter-dependency among sensors during the model training process using the attention mechanism on the default adjacency matrix.
- LocLap: Besides the global Laplacian matrix developed from the default adjacency matrix, an additive local

TABLE II
RMSE PERFORMANCE COMPARISON OF GALEN WITH GRAPH CONSTRUCTION SCHEMES

RMSE (km/h)	NavInfo Beijing				NavInfo Shanghai				PeMS-BAY			
	5 min	15 min	30 min	60 min	5 min	15 min	30 min	60 min	5 min	15 min	30 min	60 min
T-GCN+Default	4.83	5.18	5.60	6.10	5.28	5.55	5.98	6.59	2.88	4.02	4.82	6.30
T-GCN+Attn	4.77	5.08	5.47	6.00	5.19	5.51	5.88	6.46	2.85	4.01	4.76	6.28
T-GCN+LocLap	4.66	5.03	5.43	5.84	5.13	5.45	5.78	6.38	2.84	3.98	4.73	6.24
T-GCN+SlfAdp	4.67	4.95	5.49	5.88	5.13	5.42	5.82	6.34	2.84	3.98	4.72	6.18
T-GCN+GALEN	4.36	4.71	5.41	5.81	5.00	5.23	5.47	6.06	2.83	3.89	4.58	6.10
STSGCN+Default	4.77	5.10	5.56	6.03	5.25	5.49	5.87	6.55	2.87	4.00	4.79	6.25
STSGCN+Attn	4.67	5.08	5.53	6.00	5.18	5.40	5.79	6.46	2.83	3.95	4.75	6.23
STSGCN+LocLap	4.66	5.00	5.47	5.80	5.09	5.30	5.79	6.36	2.84	3.96	4.75	6.21
STSGCN+SlfAdp	4.62	4.91	5.34	5.90	5.23	5.30	5.74	6.43	2.79	3.95	4.73	6.22
STSGCN+GALEN	4.47	4.71	5.20	5.62	5.00	5.04	5.56	6.30	2.79	3.90	4.72	6.17
GA2+Default	4.77	5.11	5.58	6.07	5.25	5.48	5.89	6.48	2.86	4.02	4.79	6.26
GA2+Attn	4.61	5.05	5.48	5.94	5.14	5.42	5.86	6.33	2.84	3.96	4.82	6.22
GA2+LocLap	4.60	4.96	5.39	5.82	4.99	5.37	5.86	6.21	2.84	3.95	4.75	6.23
GA2+SlfAdp	4.66	5.01	5.45	5.82	5.17	5.14	5.67	6.24	2.82	3.95	4.69	6.16
GA2+GALEN	4.44	4.79	5.19	5.45	4.75	5.08	5.65	6.07	2.76	3.77	4.71	6.02

TABLE III
MAPE PERFORMANCE COMPARISON OF GALEN WITH GRAPH CONSTRUCTION SCHEMES

MAPE	NavInfo Beijing				NavInfo Shanghai				PeMS-BAY			
	5 min	15 min	30 min	60 min	5 min	15 min	30 min	60 min	5 min	15 min	30 min	60 min
T-GCN+Default	10.30%	11.08%	11.97%	13.02%	10.77%	11.32%	12.17%	13.41%	2.27%	3.17%	3.80%	4.97%
T-GCN+Attn	10.19%	10.86%	11.69%	12.79%	10.58%	11.24%	12.01%	13.20%	2.25%	3.15%	3.76%	4.96%
T-GCN+LocLap	9.95%	10.73%	11.59%	12.45%	10.48%	11.09%	11.85%	13.05%	2.24%	3.14%	3.74%	4.92%
T-GCN+SlfAdp	9.99%	10.56%	11.73%	12.54%	10.44%	11.05%	11.91%	12.94%	2.25%	3.14%	3.72%	4.87%
T-GCN+GALEN	9.29%	10.05%	11.54%	12.40%	10.19%	10.71%	11.16%	12.33%	2.23%	3.07%	3.60%	4.82%
STSGCN+Default	10.20%	10.91%	11.84%	12.88%	10.71%	11.21%	12.04%	13.39%	2.26%	3.15%	3.78%	4.93%
STSGCN+Attn	9.98%	10.83%	11.82%	12.78%	10.60%	11.03%	11.82%	13.17%	2.24%	3.12%	3.74%	4.92%
STSGCN+LocLap	9.92%	10.67%	11.66%	12.38%	10.41%	10.83%	11.81%	13.00%	2.24%	3.12%	3.73%	4.89%
STSGCN+SlfAdp	9.85%	10.48%	11.40%	12.56%	10.64%	10.83%	11.71%	13.19%	2.19%	3.12%	3.72%	4.91%
STSGCN+GALEN	9.51%	10.05%	11.08%	11.99%	10.22%	10.30%	11.37%	12.90%	2.20%	3.07%	3.72%	4.87%
GA2+Default	10.17%	10.91%	11.89%	12.96%	10.74%	11.20%	12.02%	13.28%	2.25%	3.17%	3.78%	4.93%
GA2+Attn	9.84%	10.77%	11.68%	12.67%	10.55%	11.04%	11.96%	12.92%	2.24%	3.13%	3.80%	4.91%
GA2+LocLap	9.82%	10.57%	11.48%	12.40%	10.17%	10.97%	11.92%	12.67%	2.24%	3.12%	3.75%	4.91%
GA2+SlfAdp	9.95%	10.69%	11.64%	12.42%	10.56%	10.53%	11.60%	12.73%	2.23%	3.12%	3.69%	4.85%
GA2+GALEN	9.48%	10.22%	11.07%	11.64%	9.70%	10.33%	11.55%	12.39%	2.17%	2.97%	3.72%	4.75%

Laplacian matrix is trained in real-time and serves as a short-term perturbation.

- SlfAdp: A self-adaptive adjacency matrix defined by $\tilde{\mathbf{A}} = \text{softmax}(\text{ReLU}(\mathbf{E}_1\mathbf{E}_2^\top))$ is utilized in graph convolution, where the source node embedding $\mathbf{E}_1$ and target node embedding $\mathbf{E}_2$ are multiplied to derive the spatial dependency among sensors.

We follow a "Predictor + GraphScheme" naming pattern to tag all tested approaches. For example, "T-GCN + GALEN" stands for using GALEN to generate graphs for T-GCN, "STSGCN + LocLap" refers to including local Laplacian in STSGCN, and "GA2 + Default" means that the original GA2 algorithm with the default adjacency matrix is tested.

Tables II and III presents the traffic prediction MAPE and RMSE of T-GCN, STSGCN, and GA2 using GALEN and other graph construction schemes. The best performing results are highlighted by underlines. When constructing traffic graphs for data prediction, GALEN outperforms compared graph construction scheme baselines on all three predictors and datasets. In particular, an average 5.60% prediction error reduction in RMSE can be observed by substituting the default traffic graphs with ones generated by GALEN across all tested datasets and predictors. The advantage is maintained compared with the state-of-the-art dynamic graph construction schemes, where GALEN can facilitate generating predictions with 4.37% (Attn), 3.25% (LocLap), and 3.06% (SlfAdp) less error. This statistical summary of raw results indicates that a tailor-made traffic graph construction scheme considering the transportation domain knowledge like GALEN can notably improve the performance of traffic predictors with trivial additional effort. The comparison also shows that both global adjacency information given by the transportation topology and adaptive adjacency information learned from the data is critical to traffic prediction as LocLap and SlfAdp imply, where the former makes use of both, and the latter only utilizes the local one. GALEN takes a step forward beyond LocLap and learns the adaptive adjacency information from both the topology and the data. Therefore, the trade-off of balancing the two adjacencies can be avoided, and better

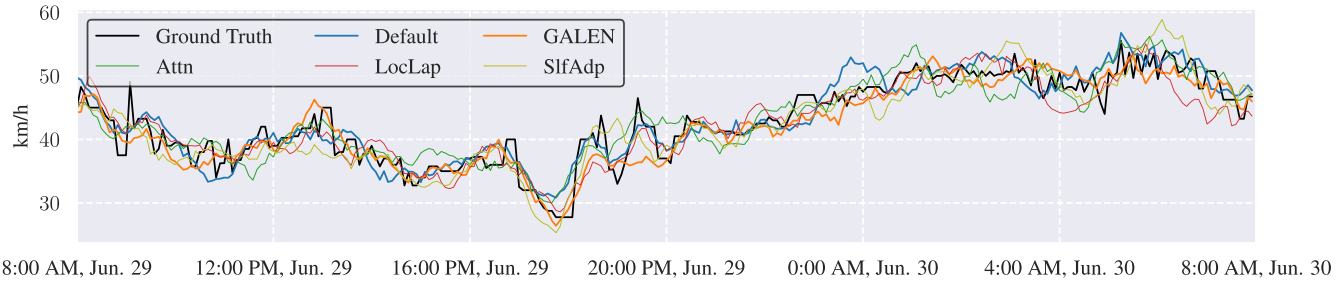


Fig. 4. Traffic speed prediction of West Chang'an Avenue with T-GCN in NavInfo Beijing dataset.

performance is achieved. At the same time, attention is another viable solution to exploit the latent inter-sensor data correlation further. Both GALEN and Attn helps predictors to capture the heterogeneous edge dependencies in traffic graphs. Fig. 4 provides the predicted speed values of West Chang'An Avenue in NavInfo Beijing dataset using GALEN and baseline graph construction approaches for a direct illustration.

Furthermore, due to the unique design of GALEN pre-training and training phases as illustrated in Fig. 1, the training time of all predictors is not significantly increased (<5% difference). Attn, LocLap, and SlfAdp all notably increase the neural network parameter searching space, rendering approximately 25% to 90% training time increase on all three predictors, respectively. Take T-GCN and NavInfo Beijing as an example, GALEN requires 1200.53 s as the additional computation time to construct the traffic graph, which can be read as a 3.75% increase on the model training time (from 31 960.95 s, default FP32 implementation). Its time footprint is minuscule compared to Attn (+34.42%), LocLap (+49.15%), and SlfAdp (+77.90%). Therefore, GALEN is an effective and efficient graph construction scheme. It can be easily adopted in any traffic predictors that use traffic graphs in the forecast and incurs negligible additional offline training or online inference time.

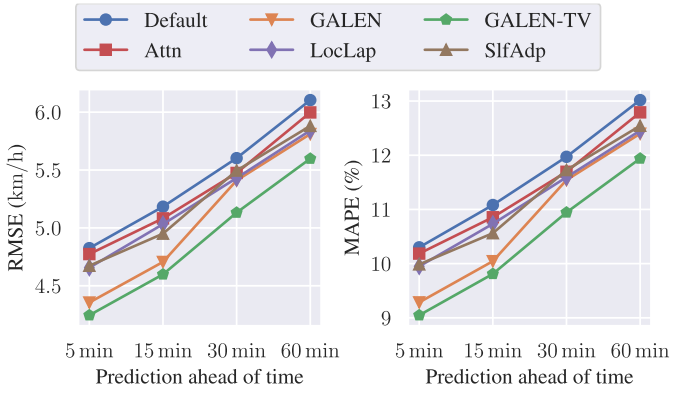
From the comparison, another interesting observation can be developed. While GALEN consistently improves the traffic prediction performance, the magnitude differs across the three datasets. In particular, NavInfo Beijing and Shanghai datasets witness an average 6.89% and 6.38% improvement, respectively, while PeMS-BAY is ameliorated to 3.03%. This difference can be credited to the complexity of the underlying transportation networks. For PeMS-BAY, the traffic sensors are located along the same multiple highways, and the correlation among geographically adjacent sensors is notably significant. In addition, the respective traffic dynamics, or rather speed changes, are much "smoother" than in urban environments. In this context, it is sufficient to construct sensors according to their relative distance for traffic prediction. Notwithstanding, the incorporation of GALEN can still further exploit the latent sensor correlation that is not covered by the aforementioned default graph construction scheme, leading to a slight but not negligible performance improvement. On the other hand, NavInfo Beijing and Shanghai significantly benefit from the proposed data-driven graph construction scheme. These two datasets describe the traffic dynamics of two metropolitan areas and thus contain more complex spatial correlations. The traffic dynamics are more diverse, and random traffic factors exhibit stronger influence than on highways. By GALEN, distant but highly correlated areas can be connected to allow information propagation during the prediction. To conclude, GALEN can be applied to any traffic prediction dataset, and a more significant performance augmentation is expected when the spatial data correlation is complex and cannot be fully captured by geographical adjacency.

C. Time-Varying Graph Performance

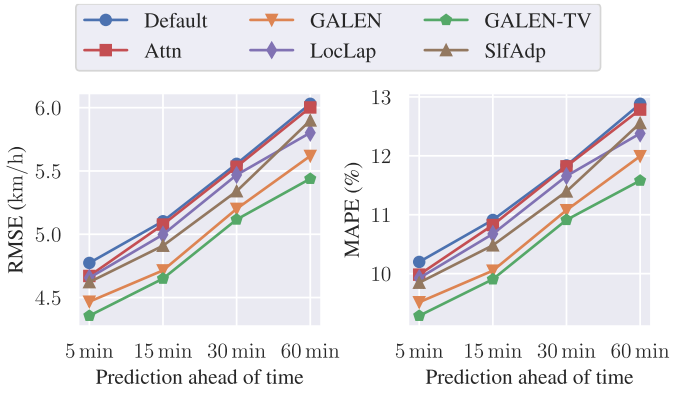
In Section IV-B, we propose a novel time-varying graph construction scheme for traffic predictors based on GALEN, following the hypothesis that the spatial-temporal traffic data correlation may vary at a different time of a day. In this section, we employ the time-varying graph construction scheme on NavInfo Beijing dataset to illustrate its efficacy. Fig. 5 depicts the MAPE and RMSE performance comparison.

From the comparative results, it can be concluded that time-varying graph construction based on GALEN can further reduce the prediction error when employed in all three state-of-the-art traffic predictors and potentially more general graph learning-based traffic predictors. Additionally, the performance improvement is more apparent for mid-term predictions compared to short-term ones. Take T-GCN (Fig. 5a) as an example, while 2.52% and 2.28% error reductions are achieved for 5 min and 15 min ahead predictions, 5.13% and 3.66% of the residuals are eliminated for 30 min and 60 min ones, respectively. This result follows the intuition on urban transportation, where the spatial-temporal dependency is dynamic throughout the day. Therefore, dynamic graphs tailored for specific periods can better capture the correlation than a static one. Due to the error accumulation nature of multi-stage traffic predictions [23], 60 min performs worse than 30 min in both RMSE and MAPE. Nevertheless, the improvement of time-varying graphs over the original GALEN and default graph construction schemes still indicates the superiority of using dynamic adjacency information in graph learning-based traffic predictors.

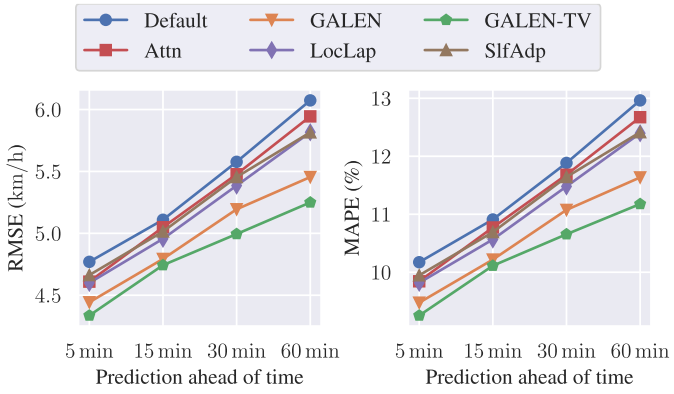
In the proposed time-varying graph construction scheme with GALEN, an important hyper-parameter influences the model performance, i.e., the length of time period for traffic predictors to use each traffic graph. Fig. 6 presents the result of a comparative study utilizing time-varying graphs with 1 h, 2 h, 4 h, 5 min, 15 min, and 30 min graph changing intervals, respectively. Due to space limit, only results



(a) T-GCN.



(b) STSGCN.



(c) GA2.

Fig. 5. Performance comparison of time-varying graph construction on NavInfo Beijing dataset.

developed from T-GCN are presented; STSGCN and GA2 demonstrate similar trends as in Fig. 6. Note that the curve labeled by "1 h" is identical to that of "GALEN-TV" in Fig. 5 and is the default configuration in Section IV-B. Two observations can be developed from the result. On the one hand, large intervals, e.g., 2 h and 4 h, lead the performance curve to converge towards that of the static graphs by GALEN. Along with the increased time period length, more complex spatial-temporal traffic correlation spanning a more extended period of time needs to be embedded in one adjacency matrix, which overwhelms its information capacity and renders inferior performance. On the other hand, small intervals, e.g., 5 min and 15 min, also undermine the prediction accuracy. As the interval shortens, the traffic dynamics reflect more

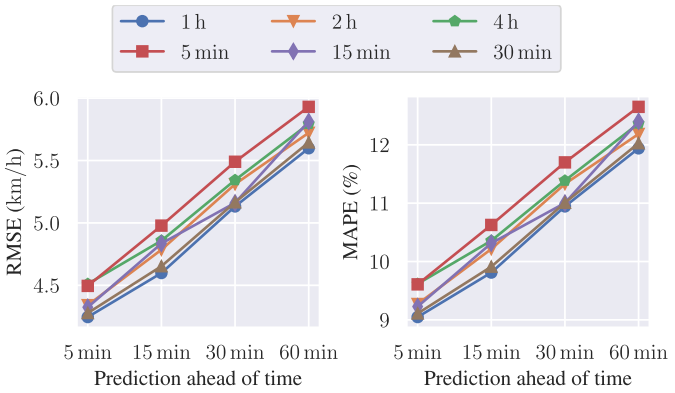


Fig. 6. Performance comparison of time-varying graph construction with T-GCN and different graph changing intervals on NavInfo Beijing dataset.

on the transient instead of the trend of the traffic flow, the former of which is highly stochastic. Therefore, an adequately configured graph changing interval is required for the proposed time-varying graph construction scheme to work well for graph learning-based traffic predictors. Following the results and discussion, we recommend an interval between 30 min and 1 h as a desirable option.

D. Sparse Graph Performance

To further alleviate the computation complexity of graph construction in GALEN, we propose a sparse graph construction scheme in Section IV-C. In this section, we compare the performance degradation and computation speed improvement of the proposed sparse graph construction. Particularly, both the complete and the sparse graphs constructed by GALEN are used to train T-GCN on NavInfo Beijing dataset, which is later employed to examine the accuracy of prediction. The simulation results are presented in Table IV, where the proposed GALEN sparse graph construction scheme is labeled by "+Sparse" under the "Scheme" column. A straightforward conclusion can be developed from the comparison that the sparse graph scheme can dramatically accelerate the graph construction progress without notable performance degradation. On the three investigated datasets, 38.84, 41.69, and 18.95 times speed-up can be obtained, respectively. Considering that 40, 43, and 19 meta-sensors ($\lceil\sqrt{N}\rceil$) are created for each dataset, the sparse graph construction scheme is highly effective.

E. Hyper-Parameter Sensitivity

In the design of GALEN, several hyper-parameters are incorporated to orchestrate the graph construction process. The selection of these hyper-parameters is crucial in determining the quality of the generated traffic graphs and, subsequently, predictor accuracy. In this sub-section, we carry out a series of hyper-parameter sweep tests to illustrate the sensitivity of GALEN on them. Specifically, we consider four GALEN hyper-parameters: neighboring sensor distance d, number of neighboring hops k, number of attention-averaging time instances R, and number of average nodal edges S. Multiple

TABLE IV
SPARSE GRAPH PERFORMANCE COMPARISON

Dataset	Scheme	RMSE (km/h)				MAPE (%)				Graph Time (s)	Speed-up
		5 min	15 min	30 min	60 min	5 min	15 min	30 min	60 min		
NavInfo Beijing	T-GCN+GALEN	4.36	4.71	5.41	5.81	9.29%	10.05%	11.54%	12.40%	1200.53	38.84×
	T-GCN+Sparse	4.42	4.75	5.37	5.72	9.27%	10.17%	11.53%	12.49%	30.91	
NavInfo Shanghai	T-GCN+GALEN	5.00	5.23	5.47	6.06	10.19%	10.71%	11.16%	12.33%	1530.24	41.69×
	T-GCN+Sparse	5.06	5.16	5.41	5.96	10.25%	10.75%	11.29%	12.61%	36.70	
PeMS-BAY	T-GCN+GALEN	2.83	3.89	4.58	6.10	2.23%	3.07%	3.60%	4.82%	52.70	18.95×
	T-GCN+Sparse	2.82	3.95	4.50	6.23	2.21%	3.05%	3.56%	4.82%	2.78	

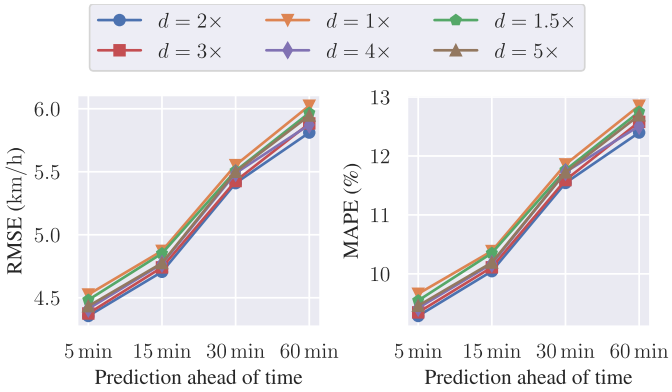


Fig. 7. Model sensitivity of neighboring sensor distance d.

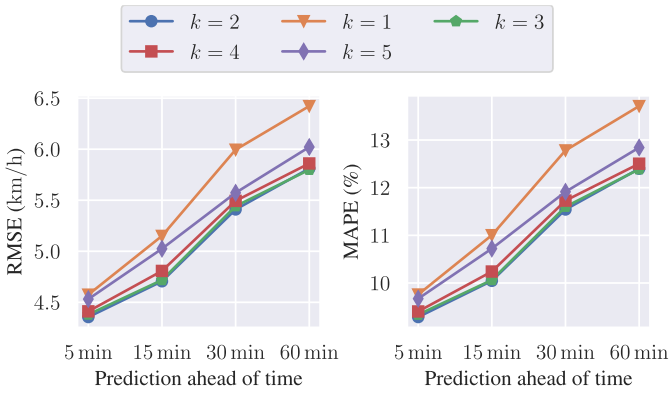


Fig. 8. Model sensitivity of neighboring hop number k.

values of each hyper-parameters are tested on T-GCN and NavInfo Beijing dataset, and all other simulation configurations are identical to the previous case studies.

We first discuss the sensitivity of neighbor graph construction hyper-parameters, i.e., d and k as introduced in Section III-C. Besides the default doubled average road length for d, namely, $d = 2\times$, we further test the performance of GALEN with d set to $1\times$, $1.5\times$, $3\times$, $4\times$, and $5\times$ of the average road length in NavInfo Beijing. Additionally, $k \in \{1, 3, 4, 5\}$ are tested besides the default $k = 2$ setting. The simulation results are presented in Figs. 7 and 8. A general conclusion can be developed from these two figures that GALEN is relatively less sensitive to d than to k. In particular, 3.31%, 2.54%, 0.71%, 1.27%, and 1.80% more prediction

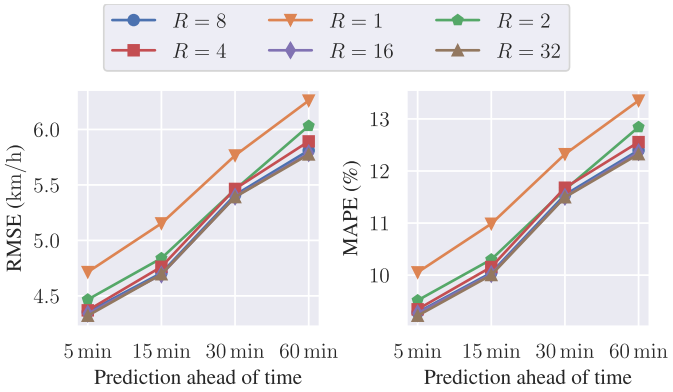


Fig. 9. Model sensitivity of attention-averaging time instance number R.

errors are introduced by changing the d value to $1\times$, $1.5\times$, $3\times$, $4\times$, and $5\times$, respectively, while setting k to 1, 3, 4, or 5 renders 8.40%, 0.20%, 1.42%, or 4.11% more error. The worst performing parameter values, namely, $d = 1\times$ and $k = 1$, both notably shrink the size of the neighboring sensor set. This may further hinder GALEN from learning the complex spatial-temporal traffic correlation via more local sensor edges. In the meantime, we also notice performance degradation if either of the two parameters is set to a large value, e.g., $d = 5\times$ or $k = 5$, albeit not as significant. This is due to the adjacency learning design of GALEN, where the neighboring sensor set is fully connected to learn the graph attention. While this nature advocates the model to exploit data inter-dependency, the complete graph also degenerates the multilayer GAT into a multilayer perceptron, which requires a drastically increased volume of training data for training and avoiding over-fitting. Apart from configuring extreme values, GALEN is insensitive to d and k, and we recommend $2\times$ and 2 respectively as a general setting for typical traffic datasets.

We further present the sensitivity of the other two hyper-parameters, i.e., R and S as introduced in Section IV-A in Figs. 9 and 10, where $R \in \{1, 2, 4, 16, 32\}$ and $S \in \{1, 2, 4, 16, 32\}$ are tested besides the default $R = 8$ and $S = 8$. The simulation results indicate that GALEN does require the attention coefficients derived from multiple time instances to construct a well-performing traffic graph. The 7.36% deterioration of $R = 1$ notably deviates from the others, where 2.48% and 1.01% degradation are observed for $R = 2$ and $R = 4$. This result can be credited to the volatile and

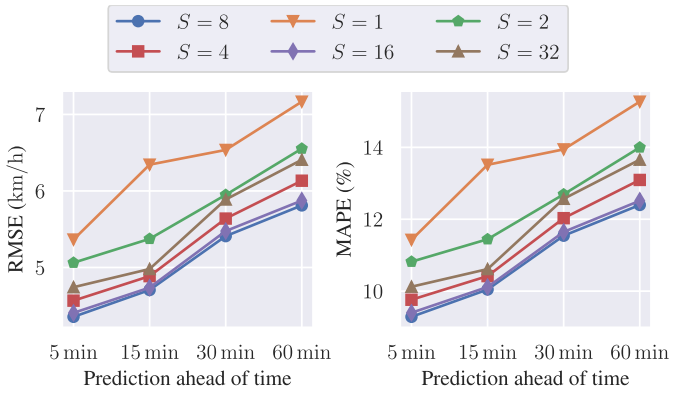


Fig. 10. Model sensitivity of average nodal edge number S.

TABLE V
GRAPH ATTENTION LEARNING VARIANTS COMPARISON

Variant	Neurons	RMSE (km/h)	MAPE (%)	Graph Time (s)
GALEN	3×128	5.07	10.82%	1200.53
Model A	2×128	5.35	11.42%	775.85
Model B	4×128	5.04	10.77%	1504.27
Model C	2×64	6.21	13.23%	773.63
Model D	3×64	5.33	11.37%	1164.31
Model E	4×64	5.14	10.95%	1505.82
Model F	5×64	5.10	10.90%	1865.35
Model G	2×256	5.56	11.87%	773.96
Model H	3×256	5.10	10.89%	1242.93

stochastic nature of traffic. In principle, $R = 1$ and adopting time-varying traffic graphs every 5 min (Section V-C) catalyze the same issue: the transient traffic flow characteristics are captured instead of the trend. On the other hand, increasing R value beyond the default 8 can further improve the prediction accuracy, namely, by 0.34% for $R = 16$ and 0.50% for $R = 32$. Nevertheless, neither option is preferred due to the computation time concern. According to the design of graph construction phase in GALEN, the construction time is linearly proportional to R. Considering the time performance in Table IV, doubling or quadrupling the graph construction time may potentially undermine the real-time nature of GALEN in huge transportation systems. Therefore, the selection of R depends on the computation burden of graph construction, and we choose $R = 8$ as the default value with the aforementioned experimental configuration.

For the number of average nodal edges S, a similar performance deviation can be observed when the parameter value is reduced. As graph learning models heavily rely on graph edges for nodal information propagation and exchange, a small S typically renders a sparse traffic graph that cannot captures the spatial-temporal data correlation. Meanwhile, increasing S beyond the default 8 cannot further improve the prediction accuracy as opposed to R. This can also be accredited to the overfitting issue of the degenerated multilayer perceptron, similar to the scenarios with large k values albeit less significant.

F. Graph Attention Learning Architecture

GALEN employs a stacked GAT neural network for graph attention learning when constructing traffic graphs. In this section, we propose a series of stacked GAT variants with different number of layers and neurons as the graph attention learning neural network for an architecture test. Particularly, Table V presents a summary of the GALEN variants and their respective traffic prediction performance on T-GCN and NavInfo Beijing dataset. The labels under the "Neurons" column refer to the number of GAT layers and neurons. For example, "3×128" refers to the default configuration where three layers of GAT are utilized to learn the graph attention and each layer has 128 neurons. From the comparison, it can be concluded that the performance of GALEN is not sensitive to the neural network architecture as long as the model capacity is saturated, which is derived by observing and comparing the performance of variant models B, E, F, and H. Considering that the graph construction time is linearly proportional to the number of layers L but not the number of neurons due to the parallel computing nature of neural networks, the default architecture is preferred over these variants. On the other hand, reducing L renders remarkably attenuation on the model capacity (models A and C), even if the number of neurons is doubled (model G). As model capacity is crucial in capturing the spatial-temporal correlation among traffic data, which is adopted to construct traffic graphs, these variants lead to inferior prediction accuracy is thus not favored despite the reduced graph construction time.

VI. CONCLUSION

In this paper, we propose a novel Graph Adjacency Learning Network to dynamically construct traffic graphs based on the historical traffic data for graph learning-based traffic predictors. This research is grounded on the hypothesis that traffic network topology-based or relative distance-based adjacency matrices cannot fully exploit the complex spatial-temporal correlation of traffic data, which is empirically verified by the comprehensive case studies. The proposed graph construction scheme is capable of capturing such correlation from a data-driven perspective. By adopting a stacked graph attention neural network, the dependency among traffic sensors is embedded in the layer-wise attention coefficients, which are later employed to determine the necessity of incorporating an edge between an arbitrary pair of sensors in the traffic graph. Subsequently, a time-varying graph construction scheme is proposed to cater to the dynamic traffic data correlation for performance improvement. We further devise a sparse graph construction scheme to alleviate the computation burden of the proposed graph construction scheme without compromising the quality of traffic graphs. To evaluate the efficacy of the proposed schemes, a series of comprehensive case studies are conducted on three real-world traffic datasets of both urban and highway transportation networks. The results indicate that the proposed scheme can outperform typical traffic graphs by 5.60% prediction error reduction with negligible time performance degradation, and time-varying graphs can further improve the accuracy by 2.28% to 5.13% under different scenarios. Furthermore, the proposed sparse graph scheme drastically accelerates the graph construction process. Finally, hyper-parameter sensitivity tests are carried out to illustrate the impact of and develop guidelines for hyper-parameter and neural network architecture selection.

In the future, we plan to explore other attention learning mechanisms from both the spatial and the spectral graph theory perspectives. We look forward to follow-up research on developing further general schemes for boosting graph learning-based traffic predictors, including but not limited to the construction of traffic graphs.

REFERENCES

[1] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[2] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[3] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.

[4] R. Jia, P. Jiang, L. Liu, L. Cui, and Y. Shi, "Data driven congestion trends prediction of urban transportation," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 581–591, Apr. 2018.

[5] Q.-J. Kong, Y. Xu, S. Lin, D. Wen, F. Zhu, and Y. Liu, "UTN-model-based traffic flow prediction for parallel-transportation management systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1541–1547, Sep. 2013.

[6] E. L. Manibardo, I. Lana, and J. D. Ser, "Deep learning for road traffic forecasting: Does it make a difference?" *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 7, 2021, doi: 10.1109/TITS.2021.3083957.

[7] J. Ye, J. Zhao, K. Ye, and C. Xu, "How to build a graph-based deep learning architecture in traffic domain: A survey," *IEEE Trans. Intell. Transp. Syst.*, early access, Dec. 29, 20210, doi: 10.1109/TITS.2020.3043250.

[8] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[9] Z. Li *et al.*, "A hybrid deep learning approach with GCN and LSTM for traffic flow prediction," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Auckland, New Zealand, Oct. 2019, pp. 1929–1933.

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

[11] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Neural Inf. Process. Syst.*, Dec. 2020, pp. 1–16.

[12] J. Zhang, F. Chen, Z. Cui, Y. Guo, and Y. Zhu, "Deep learning architecture for short-term passenger flow forecasting in urban rail transit," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7004–7014, Nov. 2021.

[13] G. Li, V. L. Knoop, and H. V. Lint, "Dynamic graph filters networks: A gray-box model for multistep traffic forecasting," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.

[14] Z. Li *et al.*, "A multi-stream feature fusion approach for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, early access, Oct. 7, 2020, doi: 10.1109/TITS.2020.3026836.

[15] K. Guo, Y. Hu, Z. Qian, Y. Sun, J. Gao, and B. Yin, "Dynamic graph convolution network for traffic forecasting based on latent network of Laplace matrix estimation," *IEEE Trans. Intell. Transp. Syst.*, early access, Sep. 9, 2020, doi: 10.1109/TITS.2020.3019497.

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

[17] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, Apr. 2018, pp. 1–12.

[18] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, *arXiv:1506.00019*.

[19] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 3844–3852.

[20] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 1025–1035.

[21] J. J. Q. Yu and J. Gu, "Real-time traffic speed estimation with graph convolutional generative autoencoder," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3940–3951, Oct. 2019.

[22] J. J. Q. Yu, "Citywide traffic speed prediction: A geometric deep learning approach," *Knowl.-Based Syst.*, vol. 212, Jan. 2021, Art. no. 106592.

[23] J. J. Q. Yu, C. Markos, and S. Zhang, "Long-term urban traffic speed prediction with deep learning on graphs," *IEEE Trans. Intell. Transp. Syst.*, early access, Apr. 9, 2021, doi: 10.1109/TITS.2021.3069234.

[24] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, and S. Ji, "Temporal multi-graph convolutional network for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3337–3348, Jun. 2021.

[25] Y. Shin and Y. Yoon, "Incorporating dynamicity of transportation network with multi-weight traffic graph convolutional network for traffic forecasting," *IEEE Trans. Intell. Transp. Syst.*, early access, Oct. 26, 2020, doi: 10.1109/TITS.2020.3031331.

[26] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, Apr. 2013.

[27] X. Kong, W. Xing, X. Wei, P. Bao, J. Zhang, and W. Lu, "STGAT: Spatial-temporal graph attention networks for traffic flow forecasting," *IEEE Access*, vol. 8, pp. 134363–134372, 2020.

[28] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, Feb. 2019, pp. 922–929.

[29] X. Zhang, C. Huang, Y. Xu, and L. Xia, "Spatial-temporal convolutional graph attention networks for citywide traffic flow forecasting," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1853–1862.

[30] B. Yu, Y. Lee, and K. Sohn, "Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN)," *Transp. Res. C, Emerg. Technol.*, vol. 114, pp. 189–204, May 2020.

[31] C. Park *et al.*, "ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1215–1224.

[32] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, Feb. 2019, pp. 890–897.

[33] L. Zhao, Y. Song, C. Zhang, and Y. Liu, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[34] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, Apr. 2018, pp. 1–16.

[35] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, Ft. Lauderdale, FL, USA, Apr. 2011, pp. 315–323.

[36] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proc. AAAI Conf. Artif. Intell.*, New York, NY, USA, Jan. 2020, pp. 1177–1185.

[37] C. Zhang, J. J. Q. Yu, and Y. Liu, "Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting," *IEEE Access*, vol. 7, pp. 166246–166256, 2019.

James J. Q. Yu (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees in electrical and electronic engineering from The University of Hong Kong, Hong Kong, in 2011 and 2015, respectively. He was a Post-Doctoral Fellow at The University of Hong Kong from 2015 to 2018. He is currently an Assistant Professor at the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China, and an Honorary Assistant Professor at the Department of Electrical and Electronic Engineering, The University of Hong Kong. He also works as the Chief Research Consultant at GWGrid Inc., Zhuhai, and Fano Labs, Hong Kong. His general research interests are in smart city and urban computing, deep learning, intelligent transportation systems, and smart energy systems. His work is now mainly on forecasting and decision making of future transportation systems and basic artificial intelligence techniques for industrial applications. He was ranked World's Top 2% Scientists by Stanford University in 2020. He is an Editor of the *IET Smart Cities* journal.