Long-Term Urban Traffic Speed Prediction With Deep Learning on Graphs

James J. Q. Yu, *Senior Member, IEEE*, Christos Markos, and Shiyao Zhang, *Member, IEEE*

***Abstract*— Traffic speed prediction is among the foundations of advanced traffic management and the gradual deployment of internet of things sensors is empowering data-driven approaches for the prediction. Nonetheless, existing research studies mainly focus on short-term traffic prediction that covers up to one hour forecast into the future. Previous long-term prediction approaches experience error accumulation, exposure bias, or generate future data of low granularity. In this paper, a novel data-driven, long-term, high-granularity traffic speed prediction approach is proposed based on recent development of graph deep learning techniques. The proposed model utilizes a predictor-regularizer architecture to embed the spatial-temporal data correlation of traffic dynamics in the prediction process. Graph convolutions are widely adopted in both sub-networks for geometrical latent information extraction and reconstruction. To assess the performance of the proposed approach, comprehensive case studies are conducted on real-world datasets and consistent improvements can be observed over baselines. This work is among the pioneering efforts on network-wide long-term traffic speed prediction. The design principles of the proposed approach can serve as a reference point for future transportation research leveraging deep learning.**

***Index Terms*— Traffic speed prediction, long-term forecast, intelligent transportation systems, deep learning, data mining.**

I. Introduction

ACCURATE and timely traffic speed prediction based on internet of things data is an essential functionality of the modern smart transportation system [1], [2]. With the gradual adoption of inter-connected stationary and dynamic traffic sensors, a massive volume of traffic data is streamlined to advanced traffic management systems for transportation planning and control [3]. The proper utilization of such data for traffic decision-making serves as the foundation of a number of smart city applications, including but not limited to traffic congestion control, transport resource allocation, and personalized trip recommendations. Specifically, future traffic speed predictions are widely recognized as indispensable towards smarter city transportation [4], [5].

Over the last decades, traffic prediction has attracted attention from intelligent transportation communities due to its welcoming influence on the social efficiency [6]. In typical settings, traffic prediction expects the forecast of traffic measures—e.g., speed and flow—in the short-term future ranging from the next few minutes to hours [7]. Boosted by the deployment of internet of things and deep learning approaches, traffic prediction is embracing new data-driven techniques along with the traditional traffic model-based ones [4]. Furthermore, the plentiful historical data enable traffic predictions to be performed over one or more days in the future, i.e., long-term traffic predictions so as to maximize operational efficiency of transportation management [7], [8]. Such long-term predictions are also important for transportation management. While the system operators benefit from the canonical four-stage traffic prediction approach, i.e., traffic generation, distribution, mode choice, and assignment, it is primarily a qualitative strategic approach rather than a quantitative one [9]. Long-term quantitative traffic speed prediction methods are still fundamental for long-term traffic management, control, and route recommendations [10]. Traffic management agencies can utilize the information for better traffic management and congestion avoidance [7], [11], [12]. Additionally, road users also benefit from the early and possibly better route planning service provided by location-based applications such as crowdsourced navigation, in which the long-term prediction data play a critical role [8], [13].

There has been a noteworthy amount of research on short-term traffic speed prediction with data-driven techniques [14]. In general, the proposed approaches can be classified into three categories, namely, naïve (e.g., historical average), parametric (e.g., auto-regressive integrated moving average, ARIMA), and non-parametric (e.g., Bayesian networks). Compared with traffic simulation models, such approaches are generally considered more robust to noise and better-performing given sufficient prior knowledge in the form of historical data [4], [7]. In recent years, traffic speed prediction is witnessing the emergence of deep learning as a new data-driven solution to many engineering problems. Powered by specialized multi-layer architectures and training techniques, deep learning models can learn highly complex traffic flow processes from historical traffic data without prior domain-specific knowledge. Existing results have demonstrated their outstanding prediction performance, see [4], [15]–[18] for some examples.

Manuscript received August 12, 2020; revised November 12, 2020 and February 17, 2021; accepted March 11, 2021. This work was supported in part by the General Program of Guangdong Basic and Applied Basic Research Foundation under Grant 2019A1515011032 and in part by the Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation under Grant 2020B121201001. The Associate Editor for this article was Y. Kamarianakis. *(Corresponding author: Shiyao Zhang.)*

James J. Q. Yu is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: yujq3@sustech.edu.cn).

Christos Markos is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China, and also with Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia.

Shiyao Zhang is with the Academy for Advanced Interdisciplinary Studies, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: syzhang@ieee.org).

Digital Object Identifier 10.1109/TITS.2021.3069234

However, there is a research gap with respect to long-term traffic speed prediction. When forecasting day-ahead traffic conditions, the majority of previous work employs the recursive multi-stage/step prediction paradigm that uses the predicted value of the current time instance to determine its value in the next one, see [7], [8] for some examples. Nonetheless, this prediction scheme is subject to two issues. First, the recursive strategy allows forecasting errors to accumulate along the time horizon because predictions are used in place of ground truth observations. Hence, the performance can quickly be degraded as the time horizon lengthens. Second, since these approaches are particularly designed for one-step prediction, exposure bias arises where only ground-truth data are used for prediction in offline training. However, predicted data are considered as ground-truth in the forecast during the online inference, resulting in train-test inconsistency. Models that take historical data to directly and non-recursively forecast into the future are favored in the context. While a few recent studies on long-term traffic forecasting do without multi-stage predictions, they either focus on a particular type of road, e.g., highways [9], [19], or produce predicted speed and flow data at a low granularity, e.g., 30 min to 1 h in [8]. Both improving the long-term prediction accuracy and providing high-resolution forecast data can enhance traffic speed prediction for advanced traffic management [9], [20], [21].

To address the aforementioned issues and fill the research gap, we propose a new data-driven, day-ahead, high-granularity traffic speed prediction approach for general urban areas based on recent developments in graph deep learning techniques. In particular, the proposed spatial-temporal graph neural network (ST-GNet) employs a predictor-regularizer design to incorporate the spatial-temporal data correlation of traffic network dynamics in day-ahead speed predictions. Different from existing research, ST-GNet extracts the geometric traffic characteristics by utilizing historical data at various temporal resolutions and transportation network topological information. ST-GNet constructs a speed predictor following the graph convolution principle of graph deep learning for geometric feature extraction. A graph neural network structure is further formulated for prediction regularization.

To summarize, the highlights of this work are as follows:

- A novel data-driven, long-term, high-granularity traffic speed prediction approach is proposed based on a predictor-regularizer structure.
- A graph convolution-based multi-scale latent information extraction mechanism is devised and adopted to construct the predictor.
- An effective regularizer for prediction regularization is introduced, which is new to traffic speed prediction and can be extended to related research.

The rest of this paper is organized as follows. In Section II, we briefly summarize the research related to this work. In Section III, we introduce the formulation of the proposed ST-GNet, and elaborate on its constituting components and training details. Section IV presents the case studies and discussions, and this paper is concluded in Section V.

II. Related Work

Data-driven traffic prediction has attracted much research effort in the past decade. In this section, we summarize the related work of traffic prediction with an emphasis on recent long-term traffic speed and flow prediction developments.

As discussed in Section I, data-driven traffic prediction approaches can be generally grouped into three classes [22]. While "naïve" ones are still widely adopted by the industry, parametric approaches have been important solutions to the prediction problem for decades [14]. Among them, ARIMA [23], Kalman filtering [24], and their variants [25], [26] are the prominent ones in the literature. Based on the traditional historical-data-only prediction paradigm, recent studies incorporate external traffic context data in the time-series prediction for performance enhancement, e.g., weather conditions, event information, and spatial information [27], [28]. Nonetheless, these approaches are not capable of modeling the complex non-linear spatial-temporal correlation of traffic dynamics. Their prediction performance is generally inferior to non-linear models [4], [22], [29].

In the past a few years, the rise of deep learning techniques provides the smart transport community with a new solution for capturing highly-complex data correlations, which has achieved great success. Among the early results, [4] marks a milestone of using deep learning to forecast traffic data, where the authors proposed a deep belief network to extract spatial-temporal traffic data correlations. Since then, various deep learning techniques have been adopted to handle this problem, including but not limited to Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) modules [15], [16] for capturing temporal data correlation and Convolutional Neural Networks (CNN) [17], [18] for extracting spatial latent information.

A recent body of work explores traffic data forecasting at the granularity of lanes. [30] predicts the traffic speed for each lane segment by first selecting those that exhibit the strongest correlation with the target segment, and then feeding all of them to an RNN leveraging both LSTM and Gated Recurrent Unit (GRU) modules. [31] trains a two-stream CNN on traffic volume and speed features for multi-lane traffic speed prediction, modeling correlations not only between lanes, but also between the spatial-temporal dimensions of the input features. [32] utilizes a convolutional LSTM architecture for joint multi-lane prediction of traffic flow and speed. Despite their promising results, the above road- and lane-level approaches focus on short-term predictions of 5 min up to a few hours into the future, and need to adopt the multi-stage forecasting paradigm for long-term predictions.

Compared with the previous short-term forecasting research, long-term traffic prediction results are relatively scarce. While there is no established definition on the "length" of a long-term forecasting horizon, predictions of more than one hour into the future are generally considered as long-term [33]. In this context, references [9], [19] present two data-driven approaches for day-ahead highway traffic predictions during holidays and workdays, respectively. Reference [34] introduces a CNN-based traffic flow forecasting network to perform day-ahead prediction throughout a city. Reference [8] provides up to one week-ahead traffic prediction at 30 min to 1 h granularity based on convolutional LSTM networks. Yet none of the above approaches can provide long-term, high-granularity traffic predictions with one-stage prediction to prevent the multi-stage issues.

Even though standard CNNs have been shown to be highly effective when handling Euclidean-space inputs, emerging research suggests that graph neural networks may be more suitable for learning geometric characteristics such as the complex topology of urban road networks [6], [7], [35]. References [35] and [6] combine graph and recurrent neural networks to model spatial-temporal dependencies in traffic data. Concretely, [35] proposes a graph convolutional LSTM with regularization applied to both the model's weights and features to encourage interpretability, while [6] instead endows a graph convolutional network with GRUs. The work in [7] designs a graph CNN-LSTM coupled with a correlation-based method for optimizing the adjacency matrix by including the most relevant road links. Other works [36], [37] take into account weekly and periodical flows: [36] utilizes a residual recurrent graph neural network, and [37] integrates a graph convolutional network with a spatial-temporal attention mechanism. In this work, inspired by the recent success of graph convolutional networks in traffic prediction, we present a new graph-deep-learning-based, long-term, high-granularity traffic speed prediction approach for day-ahead forecast.

III. Deep Spatial-Temporal Graph Network

In this section, we propose a new ST-GNet for beyond day-ahead long-term urban traffic speed predictions on modern urban transportation networks based on deep neural networks. The latter are based on artificial neural networks, leveraging multiple hidden layers for data correlation extraction, and have been widely adopted in modeling complex non-linear and stochastic systems in the engineering research. The proposed ST-GNet is developed on the hypothesis that the traffic speed data in urban areas demonstrate strong temporal correlation along the time axis and strong spatial correlation according to the transportation network topology. This hypothesis is widely recognized in transportation engineering, see [4], [15] for some examples. We first introduce the architecture of the proposed ST-GNet. Then, the design principle and structures of the sub-networks employed in ST-GNet are elaborated. Lastly, we discuss the training method of the proposed deep neural network.

A. ST-GNet Overview

Figure 1 depicts the overview of ST-GNet from the data flow point of view. The network is composed of two major sub-networks, namely, a *predictor* and a *regularizer*. While the former concentrates more on extracting the spatial correlation by aggregating the recent, near, and distant historical data, the latter fully utilizes the spatial-temporal data correlation to regularize the predicted speed data. These two sub-networks cooperate to provide high-granularity long-term urban traffic speed predictions.

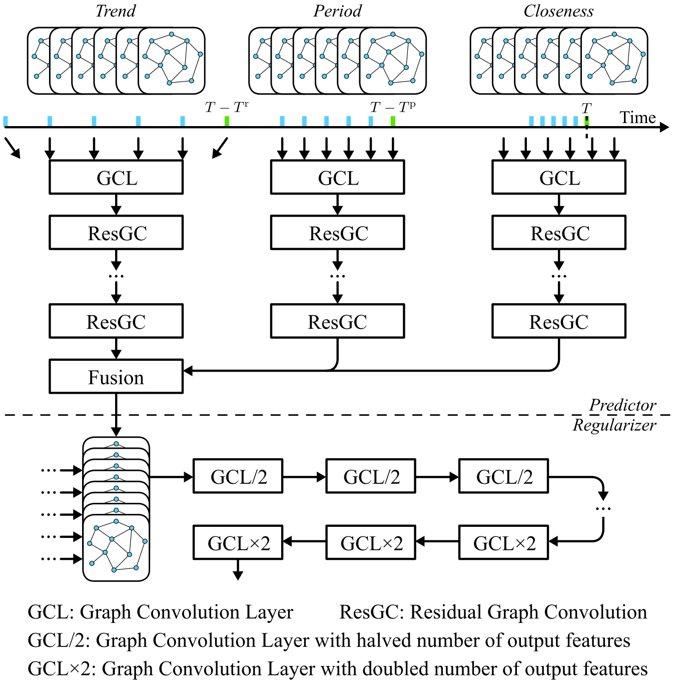
Trend
Period
Closeness
T − T^τ
T − T^p
T
Time
GCL
ResGC
...
ResGC
Fusion
Predictor
Regularizer
GCL/2
GCL×2
GCL: Graph Convolution Layer
ResGC: Residual Graph Convolution
GCL/2: Graph Convolution Layer with halved number of output features
GCL×2: Graph Convolution Layer with doubled number of output features

Fig. 1. Overview of the proposed deep spatial-temporal graph network.

We consider a discrete time horizon $\mathcal{T}$ in which each time instance corresponds to a 5 min interval. At an arbitrary time, the city-wide historical traffic semantic speed data are input into the proposed ST-GNet, which first segregates them into three groups following the common practice for long-term predictions [29], [38], [39]. Each group of data is manipulated individually by the predictor to produce a corresponding latent information embedding, all of which are later fused to generate rough day-ahead fixed-point speed forecasts. The above process can be repeated to provide high-granularity predictions within the following day with a prediction interval of several minutes. The predictor aims to predict the future city-wide traffic speed considering the historical data. Nonetheless, the potential spatial-temporal correlations among the developed high-granularity predictions is not accounted for during this data generation process. This is to be achieved by the regularizer sub-network. After obtaining the high-granularity predictions, they are input into a regularizer, which utilizes a graph-convolution-driven neural network for data regularization considering spatial-temporal data correlations. The outputs are the final day-ahead high-granularity traffic speed predictions.

In the proposed ST-GNet, the predictor learns from the spatial-temporal data correlation within each time group. Contributed by the underlying graph convolution operation (Section III-B), the transportation network topology is included in the computation as the spatial information, and the temporal data correlation is captured by convolution. The subsequent regularizer further enhances the spatial-temporal correlation exploitation by convolving consecutive predictions along the network topology. Therefore, both the spatial and temporal information on the traffic and transportation network can be handled, and the prediction is not limited to grid partitions of the investigated region.

B. ST-GNet Predictor

In ST-GNet, the predictor is composed of three parallel deep neural networks with identical structure but different sets of network parameters. In particular, each neural network is the stack of one Graph Convolution Layer (GCL) [40] and L layers of Residual Graph Convolution (ResGC) operations, appended by a final fusion layer which aggregates the outputs of the three networks to produce the day-ahead prediction.

GCL, proposed by [40], refers to a neural network layer which aims at feature extraction out of non-Euclidean structured data. It inherits the design principle of convolution operations in typical CNN, which instead handles Euclidean space data. Given a graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ whose adjacency matrix is denoted by $\mathbf{A}$, GCL adopts the nodal connectivity as the convolution filter for neighborhood mixing [41]. In particular, road segments in the transportation network represent nodes in $\mathcal{N}$, and two geographically connected segments in $\mathcal{N}$ have an edge in $\mathcal{E}$. Consequently, the adjacency matrix can be constructed by $\mathcal{E}$. Let the input data be $\mathbf{h}^{(l)} \in \mathbb{R}^{|\mathcal{N}| \times F^{(l)}}$, where $F^{(l)}$ is the number of input features. The l-th GCL performs the mixing by the following layer propagation rule to produce an output $\mathbf{h}^{(l+1)} \in \mathbb{R}^{|\mathcal{N}| \times F^{(l+1)}}$:

$$\mathbf{h}^{(l+1)} = \mathrm{GC}\left(\mathbf{h}^{(l)}\right) = \sigma\left(\mathbf{L}\mathbf{h}^{(l)}\mathbf{W}^{(l)} + \mathbf{b}^{(l)}\right), \tag{1}$$

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the learnable weight and bias parameters of the l-th layer, respectively. $\mathbf{L} = \hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}$, $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\hat{\mathbf{D}}$ is the diagonal node degree matrix of $\hat{\mathbf{A}}$, and σ is the nonlinear activation function. This propagation rule is motivated by the first-order approximation of Chebyshev polynomials of the eigenvalues in the input graph spectral domain [40].

The adoption of GCL in constructing the predictor follows the intuition: the traffic speed/conditions of adjacent roads within the transportation network may typically affect each other, though with different intensity. This physical correlation can be captured by the convolution operations on graphs with efficacy [42]. Without $\mathbf{L}$ in (1), $\mathbf{h}^{(l)}\mathbf{W}^{(l)}$ mixes all input data to produce the output. The prepended $\mathbf{L}$ limits the information fusion to be conducted regarding the nodal connectivity. Therefore, only upstream and downstream link data are involved in the convolution besides the information of the target road segment. Meanwhile, the mutual influence does not terminate after the one-hop propagation, rendering the one-hop neighborhood mixing performed by GCL insufficient in exploiting the data correlation [43]. This motivates the stack of multiple GCLs to extend the coverage of convolution operations. Nonetheless, such stacking may lead to over-smoothing issues [44], where repeated graph convolutions eventually make node embeddings indistinguishable, rendering inferior network training performance [45]. To address this issue, we incorporate the identity mapping principle to guide the design of deep neural networks, i.e., including residual links from deep residual networks [46] to skip selected layers automatically. By utilizing skip connections (shortcuts, or identity mapping), the number of graph convolutions can be adaptively selected so that the neural network is deep enough for latent feature extraction but is not too deep to over-smooth nodal embeddings. Such skipping effectively simplifies the network and reduces graph convolutions to avoid both gradient vanishing and over-smoothing [45], [46]. In the proposed ST-GNet predictor, L layers of ResGC are stacked after the first GCL, all of which share the following propagation rule:

$$\mathbf{h}^{(l+1)} = \mathbf{h}^{(l)} + \mathrm{GC}(\mathbf{h}^{(l)}). \tag{2}$$

Batch normalization is also included between every pair of layers in the predictor to alleviate the over-fitting issue commonly observed in deep neural networks.

From the overview of ST-GNet, it is clear that the structured historical traffic speed data is the essential input information to the neural network. Besides the raw speed data, additional road network knowledge and meteorology data can notably improve the prediction performance by significantly increasing the volume of effective input information [38]. Such knowledge is integrated in the computation graph by modifying the forward-pass of the first GCL to the following:

$$\mathbf{h}^{(2)} = \sigma\left(\mathbf{L}\mathbf{x}\mathbf{W}^{(1)} + \mathbf{x}'\mathbf{U} + \mathbf{m}\mathbf{V} + \mathbf{b}^{(1)}\right), \tag{3}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are the input traffic speed values and the additional road network knowledge of all roads in the transportation network at multiple past time instances, respectively, $\mathbf{m}$ is the global meteorology data of the investigated transportation network, and $\mathbf{U}$ and $\mathbf{V}$ are two additional learnable network parameters. In ST-GNet, we employ the speed limit, road length, number of lanes, and number of nearby points of interest as the additional road network data.

Furthermore, the temporal data dependencies of the traffic speed data need to be utilized for the long-term forecasting. In ST-GNet, we follow the common practice of long-term predictions (see [29], [38], [39] for some examples) and aggregates three groups of past time instances, namely:

$$\textit{Closeness}: \mathcal{T}_{\mathrm{c}} = \{t, t - T_{\mathrm{c}}, \cdots, t - (N_{\mathrm{c}} - 1)T_{\mathrm{c}}\}, \tag{4a}$$

$$\textit{Period}: \mathcal{T}_{\mathrm{p}} = \{t - T_{\mathrm{p}}, \cdots, t - N_{\mathrm{p}}T_{\mathrm{p}}\}, \tag{4b}$$

$$\textit{Trend}: \mathcal{T}_{\mathrm{r}} = \{t - T_{\mathrm{r}}, \cdots, t - N_{\mathrm{r}}T_{\mathrm{r}}\}, \tag{4c}$$

where t is the current time instance, T_{c}, T_{p}, and T_{r} are the intervals between time instances of each respective group, and N_{c}, N_{p}, and N_{r} are the number of time instances of each group, respectively. Subsequently, the input to each parallel deep neural network of the predictor is constructed by stacking the city-wide traffic speed values X_τ, road network knowledge X'_τ, and global meteorology data m_τ at time τ along the time axis, where τ enumerates all elements in $\mathcal{T}_{\mathrm{c}}$, $\mathcal{T}_{\mathrm{p}}$, or $\mathcal{T}_{\mathrm{r}}$.

Lastly, the output embeddings of all three parallel deep neural networks are fused to develop the final day-ahead traffic speed prediction. In accordance with the empirical analysis in [29] which indicates that closeness, period, and trend all affect future traffic dynamics with different degrees of influence, we employ a parametric-matrix-based learnable fusion method as follows:

$$\tilde{X}_{t+D} = \tanh\Big(\mathbf{W}_{\mathrm{c}} \odot \mathbf{h}_{\mathrm{c}}^{(L+1)} + \mathbf{W}_{\mathrm{p}} \odot \mathbf{h}_{\mathrm{p}}^{(L+1)} + \mathbf{W}_{\mathrm{r}} \odot \mathbf{h}_{\mathrm{r}}^{(L+1)}\Big), \tag{5}$$

where $\tilde{X}_{t+D}$ is the day-ahead speed prediction ($D = 24\,\text{h}/5\,\text{min} = 288$), $\mathbf{h}_{\text{c}}^{(L+1)}$, $\mathbf{h}_{\text{p}}^{(L+1)}$, and $\mathbf{h}_{\text{r}}^{(L+1)}$ are the output of neural networks handling closeness, period, and trend group of data, respectively, and symbols $\mathbf{W}$ with sub-indices are learnable parameters. The predictor is trained to predict day-ahead traffic speed X_{t+D} by minimizing the L2 loss $\mathcal{L}(\theta) = ||X_{t+D} - \tilde{X}_{t+D}||_2^2$, where θ is the collection of all learnable neural network parameters in the predictor.

C. ST-GNet Regularizer

With the aforementioned predictor, ST-GNet is capable of producing day-ahead traffic speed predictions given historical data. While traffic speed of consecutive time instances follows the transportation dynamics, the predictions alone do not respect such correlation if the results of multiple prediction runs are accumulated without post-processing. In this work, we propose a graph-convolution-driven neural network to incorporate the transportation dynamics and regularize the predictions. The regularizer adopts an architecture where the layer-wise output dimensionality is first reduced and then increased. Note that though it shares a similar structure with auto-encoders, the training is different, as will be introduced in the sequel.

The objective of this regularizer is to capture the spatial-temporal correlation among time-series traffic speed data, so that accurate predictions can be recovered from the first step's rough predictions. The working principle is simple yet effective as will be demonstrated in the case studies. In the proposed predictor, traffic speed data of one standalone future time instance are predicted. With the predictor itself, multiple individual and independent inferences are required to develop a time-series forecast. However, the spatial-temporal traffic dynamics among all the time instances in the time-series are not embedded during the process, rendering potential performance degradation. The regularizer aims to "regularize" the predicted time-series so that traffic dynamics are respected. Note that the principle of this regularizer is not limited to the proposed ST-GNet: it can be employed in other time-series prediction tasks in which each data points are individually (or non-recursively) generated. In the following, we present one possible implementation of the regularizer that fits in well with ST-GNet and long-term traffic speed prediction.

Let the prediction granularity be 5 min, and the prediction horizon start from $t + 1$ until $t + D$ where D is the number of future predictions. The predictor is executed for D times—each with a different set of input historical data—to develop D speed predictions $\tilde{X}_\tau, t < \tau \leq t + N$ in the form of time-series graphs. These graphs are then stacked and input into a sequence of R GCLs defined by (1), where each time instance is considered as a channel of the graph convolution [40]. The employed GCL is evenly divided into two groups and are labeled by "GCL/2" or "GCL×2" respectively in Figure 1. While they share the same forward pass rule as in (1), these layers serve the functionality of dimensionality reduction and recovery (up-sampling). The main difference is that the number of output features GCL/2 is reduced by half from the input, i.e., $F^{(l+1)} = \lceil F^{(l)}/2 \rceil$, while GCL×2 doubles the number of output features,[1] i.e., $F^{(l+1)} = 2F^{(l)}$. By minimizing the reconstruction loss

$$\mathcal{L}(\vartheta) = \frac{1}{2}\sum_{i=1}^{N} ||X_{t+i} - \hat{X}_{t+i}||^2 \tag{6}$$

where ϑ is the collection of all learnable parameters in the R GCLs and $\hat{X}_t$ is the reconstructed prediction corresponding to X_t, the optimal model parameters can be obtained.

D. Network Training

In the previous subsections, the architecture and computation graph of ST-GNet is given, which contains a large number of learnable parameters. Before employing ST-GNet for long-term high-granularity prediction, these parameters need to be properly tuned first. Although it is viable to directly apply common back-propagation optimization algorithms to train ST-GNet end-to-end, such a training scheme may lead to an over-trained predictor and under-trained regularizer. This is because when the prediction loss is back-propagated along the computation graph, one gradient update on the regularizer corresponds to N predictor updates, as the input of regularizer is aggregated from N predictions. Therefore, the network training is split into two parts that focuses on minimizing $\mathcal{L}(\theta)$ and $\mathcal{L}(\vartheta)$, respectively.

Given a set of historical city-wide urban traffic speed and related data defined by $\mathbf{x}_\tau = \langle X_\tau, X'_\tau, m_\tau \rangle, \tau \in \mathcal{T}$ over time horizon $\mathcal{T}$, the whole training process is summarized in the following steps:

1) Initialize neural network parameters θ and ϑ randomly.
2) Construct the predictor training set, in which each training case adopts $\{\mathbf{x}_\tau\}_{\tau \in \mathcal{T}_{\text{c}} \cup \mathcal{T}_{\text{p}} \cup \mathcal{T}_{\text{r}}}$ as inputs and $X_{\tau+D}$ as the target output.
3) Adjust the values of θ using Adam optimizer as the gradient descent optimization algorithm until convergence with respect to $\mathcal{L}(\theta)$.
4) Freeze θ and use the predictor to forecast all possible $\tilde{X}_{\tau+D}$.
5) Construct the regularizer training set, which is composed of both ground truth and predictor data:
   a) Adopt $\{X_\tau\}_{\tau \in \mathcal{D}}$ as both the input and the output (reconstruction target), where $\mathcal{D}$ is any arbitrary day within $\mathcal{T}$ starting from an arbitrary time.
   b) Adopt $\{\tilde{X}_\tau, X_\tau\}_{\tau \in \mathcal{D}}$ as the input and output, respectively.
6) Adjust the values of ϑ using Adam until convergence with respect to $\mathcal{L}(\vartheta)$.

In the above steps, the predictor is fine-tuned with the standard neural network training paradigm with a typical data pre-processing scheme. Meanwhile, the inclusion of $\tilde{X}_\tau$ in training the regularizer is beyond the common practice of adjusting similar neural networks such as auto-encoders. This design can effectively eliminate the adversarial impact of exposure bias where only ground-truth contexts are referred to during training, causing a train-test discrepancy. Additionally,

[1] In the case that the regularizer produced an output of different dimensionality from the input, the last GCL×2 is modified to output D features.

TABLE I
SUMMARY OF BJ, SH, AND GZ DATASETS

	BJ	SH	GZ
No. roads	1569	1830	1180
Avg. length	332.97 m	342.83 m	276.38 m
Avg. speed	35.03 km/h	34.37 km/h	34.93 km/h
Std. speed	12.80 km/h	13.57 km/h	13.24 km/h

it also enforces the training algorithm to adjust θ first before fine-tuning the regularizer.

IV. Case Studies

In this work, we propose ST-GNet for long-term high-granularity urban traffic speed prediction. To thoroughly assess the performance of the proposed model, we conduct a series of comprehensive case studies on two real world data. We first investigate the prediction accuracy of the proposed model and compare it with existing traffic speed prediction state-of-the-art. An ablation test is carried out to validate the effectiveness of both sub-networks in ST-GNet. Next, we investigate the model sensitivity on architectural designs of the neural networks, in particular the hyper-parameter selections. Lastly, we extend the proposed day-ahead predictor to handle week-ahead traffic speed predictions.

A. Dataset and Configurations

In this section, we adopt the real-world traffic data of three major cities in China for investigation, namely, Beijing (BJ), Shanghai (SH), and Guangzhou (GZ). These datasets include city-wide traffic speed and meteorology data obtained from NavInfo Traffic Data Platform,[2] and transportation network topology developed from OpenStreetMap.[3] The topology is used in the calculation of graph convolutions to embed spatial information in the model. All traffic speeds are calculated by NavInfo according to its proprietary floating car data, and a summary of these datasets are presented in Table I. The data from 8:00AM January 1st, 2019 to 7:55AM June 30th, 2019 are employed in the case studies, leading to 52 128 time instances with a 5 min sampling interval. After Z-score normalization, missing speed values are encoded by zeros.

In addition to the three datasets, we also employ the TaxiBJ dataset [29] in the test for a fair comparison with existing state-of-the-art that does not handle topological information. We strictly follow [8] in preprocessing the taxicab trajectory data in TaxiBJ and construct 32×32 grids of the traffic speed as the historical data, except that the time interval is set to 5 min instead of the original 30 min for high-granularity predictions. Since ST-GNet requires a topology among the input data, we manually create a connectivity matrix by connecting each grid in TaxiBJ with its eight surrounding neighborhoods. Lastly, we embed the time, day-of-week, and week-of-year data using sine and cosine functions as $\mathbf{m}$ in ST-GNet computing.

For cross-validation, each of the four datasets is split into two non-overlapping subsets, i.e., a training and a testing set: the first 80% samples of each dataset are the training data, and the remaining 20% are the testing data. In all case studies, we adopted the widely used Mean Average Percentage Error (MAPE) and Root Mean Square Error (RMSE) as the performance metrics. Unless otherwise stated, the number of ResGC layers in the predictor $L = 4$, the number of GCLs in the regularizer $R = 6$, the number of output features for all convolution layers in the predictor is 256, the number of time instances in each time group $N_\mathrm{c} = N_\mathrm{p} = N_\mathrm{r} = 6$, the time intervals of each time group $T_\mathrm{c} = 5\,\mathrm{min}$, $T_\mathrm{p} = 1\,\mathrm{h}$, and $T_\mathrm{r} = 1\,\mathrm{d}$. ST-GNet is optimized by Adam according to the training algorithm presented in Section III-D with mini-batch size 32 and learning rate 0.001. The proposed ST-GNet is modeled with PyTorch, and all simulations are conducted on a computing server with two Intel Xeon E5 CPUs, and nVidia GTX 2080 Ti GPUs are employed for neural network computing acceleration.

B. Prediction Accuracy

In this case study, we employ the proposed ST-GNet on all four testing datasets to develop day-ahead speed predictions. Additionally, the following baseline approaches are implemented for comparison, some of which are the existing state-of-the-art for long-term traffic speed prediction:

- *History Average (HA):* HA models traffic speed dynamics of roads as a seasonal process. We average the value of historical traffic speed in the corresponding time period of the past four weeks as the prediction, in accordance with the configuration in [29].
- *ARIMA with Kalman Filter (ARIMA-K):* ARIMA-K makes use of the inter-data point dependency for predictions considering a stationary regression-type time-series. We consider three lagged data points, zero degree of differencing and window size one in the test.
- *Spatial-temporal CNN (STCNN) [8]:* STCNN extracts the spatio-temporal traffic dependencies with an encoder-decoder structure based on convolutional LSTM.
- *Graph CNN-LSTM (GCNN-LSTM) [7]:* GCNN-LSTM employs a sequential graph convolution model to capture the spatial data dependency and a subsequent LSTM network for time-series generation.
- *Spatial-temporal Dynamic Network (STDN) [14]:* STDN utilizes a shifted attention mechanism to capture the long-term periodic data dependency which experiences temporal shifting.
- *Diffusion Convolutional Recurrent Neural Network (DCRNN) [22]:* DCRNN captures the spatial dependency using bidirectional random walks on the graph and the temporal dependency with an encoder-decoder architecture.

We follow the descriptions in the previous work to implement HA [29], ARIMA-K [47], STCNN [8], and GCNN-LSTM [7] in this test, and all hyper-parameters remain unaltered. For STDN and DCRNN, we adopt the source code provided in the respective literature with minuscule non-algorithmic changes.

[2] https://nitrafficindex.com

[3] https://www.openstreetmap.org

TABLE II

PREDICTION ACCURACY COMPARISON WITH BASELINES

MAPE	BJ			SH			GZ			TaxiBJ [29]		
	$T+1$	Avg.	$T+D$	$T+1$	Avg.	$T+D$	$T+1$	Avg.	$T+D$	$T+1$	Avg.	$T+D$
ST-GNet	9.0%	9.1%	9.5%	9.5%	9.5%	9.7%	9.4%	9.4%	10.0%	7.2%	7.3%	7.6%
HA (c.f. [29])	13.1%	13.1%	13.2%	12.9%	13.3%	13.8%	13.4%	13.4%	13.6%	11.6%	11.8%	12.1%
ARIMA-K (c.f. [47])	12.5%	15.7%	20.5%	11.9%	15.9%	18.8%	12.1%	15.6%	20.5%	9.6%	12.4%	15.5%
STCNN [8]	10.7%	12.4%	13.9%	11.6%	13.0%	14.6%	10.4%	13.1%	14.2%	10.6%	12.7%	13.8%
GCNN-LSTM [7]	15.5%	15.7%	16.6%	15.5%	15.5%	16.3%	15.8%	15.8%	17.2%	14.0%	14.2%	14.7%
STDN [14]	12.0%	13.3%	13.9%	12.5%	13.7%	14.3%	12.0%	12.6%	13.7%	9.8%	10.1%	11.0%
DCRNN [22]	10.5%	13.6%	17.5%	10.5%	13.9%	16.4%	10.9%	13.9%	17.3%	10.2%	12.7%	15.4%
STCNN-CPT	12.4%	14.6%	15.6%	13.4%	15.2%	15.3%	13.9%	14.2%	15.4%	11.7%	12.2%	12.5%
DCRNN-CPT	12.6%	15.3%	17.4%	12.2%	15.7%	18.2%	12.1%	14.8%	16.7%	11.2%	13.4%	14.7%

RMSE (km/h)	BJ			SH			GZ			TaxiBJ [29]		
	$T+1$	Avg.	$T+D$	$T+1$	Avg.	$T+D$	$T+1$	Avg.	$T+D$	$T+1$	Avg.	$T+D$
ST-GNet	3.64	3.64	3.80	3.75	3.78	3.82	4.20	4.25	4.47	3.49	3.57	3.61
HA (c.f. [29])	5.29	5.32	5.40	5.16	5.27	5.32	5.94	5.95	6.06	5.67	5.77	5.86
ARIMA-K (c.f. [47])	5.05	6.26	8.20	4.69	6.30	7.55	5.32	6.89	9.12	4.62	6.01	7.34
STCNN [8]	4.36	4.99	5.48	4.58	5.18	5.63	4.65	5.90	6.39	5.20	6.16	6.56
GCNN-LSTM [7]	6.16	6.33	6.70	6.11	6.12	6.44	7.02	7.07	7.65	6.72	6.94	7.01
STDN [14]	4.82	5.38	5.61	4.95	5.27	5.60	5.27	5.54	6.05	4.72	4.92	5.34
DCRNN [22]	4.22	5.44	7.02	4.13	5.48	6.48	4.92	6.10	7.61	4.91	6.18	7.46
STCNN-CPT	4.93	5.82	6.21	5.37	5.94	6.13	6.20	6.32	6.82	5.65	5.84	6.09
DCRNN-CPT	5.05	6.13	6.88	4.75	6.18	7.07	5.36	6.60	7.53	5.36	6.47	7.16

When adopting all baselines for long-term traffic speed prediction, the multi-stage prediction paradigm is adopted for ones that cannot develop forecasts of all D future time instances in a single step. Note that since STCNN and STDN can only handle grid-like urban traffic speed data, we divide the investigated area of the four datasets into $1\,\text{km} \times 1\,\text{km}$ grids as instructed in [14]. All other simulation configurations are identical to those stated in Section IV-A for a fair comparison. Besides, we also conduct a series of Wilcoxon rank-sum tests on the null hypothesis that the proposed ST-GNet performs similarly with any of the compared baselines when performing long-term traffic speed predictions. The statistical test result at 95% significance level is presented next to the corresponding result where a “†” symbol without quotes indicates that the Wilcoxon rank-sum test cannot distinguish between the predictions made by ST-GNet and the respective approach.

Table II summarizes the simulation results of the proposed and baseline approaches. In this table, the MAPE and RMSE values of all D future predictions in the next day of the current time are averaged and listed under the “Avg.” column. In addition, the prediction accuracies of 5 min-ahead and 1 d-ahead forecast are labeled by “$T + 1$” and “$T + D$”, respectively. From the simulation results it is obvious that the proposed ST-GNet outperforms all competing baselines by achieving the lowest MAPE and RMSE on all datasets, and Wilcoxon rank-sum tests approve the statistical significance of all results by ST-GNet. In particular, traditional time-series prediction methods, i.e., HA and ARIMA-K in the comparison, generally produce worse forecasted future speed values. This is due to the fact that such time-series predictions can only capture linear temporal correlation within datasets. Neither non-linear temporal correlation nor spatial data dependency can be extracted, rendering worse performance. Compared to ARIMA-K, HA develops significantly more accurate and stable predictions. This is because HA manipulates the raw data points based on historical data samples instead of time-series, therefore preventing forecast errors being accumulated in the multi-stage data prediction process that ARIMA-K performs.

In terms of deep learning-based approaches, the proposed ST-GNet outperforms the state-of-the-art STCNN, GCNN-LSTM, STDN, and DCRNN. Compared with STCNN and GCNN-LSTM, ST-GNet is capable of regularizing the high-granularity predictions using the regularizer, which does not have a counterpart in the two compared baselines. As a critical temporal dependency extraction module, the regularizer prevents ST-GNet from the adversarial prediction error accumulation issue that STCNN and GCNN-LSTM may experience, whose $T + D$ accuracy is notably inferior than that at $T + 1$. While this issue is alleviated in STDN by its explicit temporal shifting attention mechanism, the model concentrates on peak value shifts, making it potentially sensitive to data noise. Additionally, STDN does not embed the transportation network topology information in the computation, which may undermine its capability to extract spatial data correlation. Finally, as DCRNN is tailor-made for short-term traffic speed prediction, its performance significantly degrades with the prediction window. This observation also accords with the original results published in the respective literature [22, Table II]. The better performance of the proposed ST-GNet demonstrates the efficacy of the proposed graph deep learning-based traffic speed predictor.

One may note that the input time-series of deep learning baselines, e.g., STCNN and DCRNN, are not identical to those of ST-GNet. In particular, both STCNN and DCRNN take the immediately preceding twelve time instances, i.e., $N_\text{c} = 12$

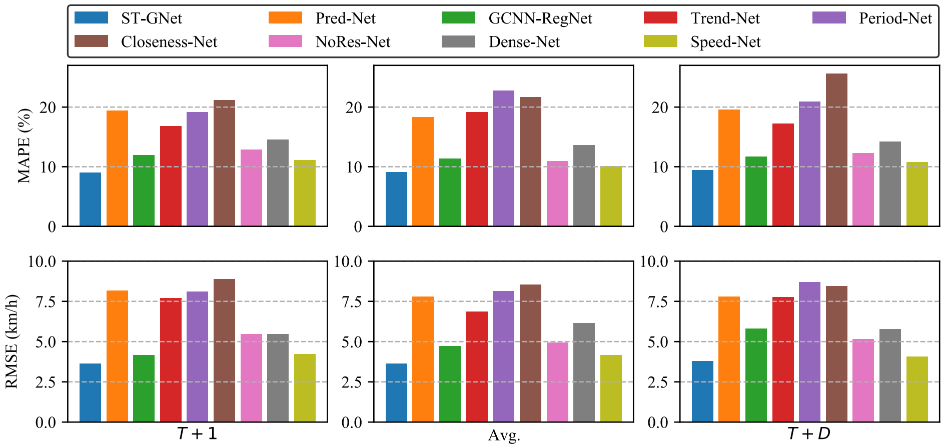

Fig. 2. Performance comparison of ST-GNet ablation test models on BJ dataset.

and $N_\mathrm{p} = N_\mathrm{r} = 0$. In order to rule out the impact of input data being different, we additionally test the variants of these two baselines by adopting $N_\mathrm{c} = N_\mathrm{p} = N_\mathrm{r} = 6$ historical data as the input. The results are presented in Table II where the variants are labelled by "STCNN-CPT" and "DCRNN-CPT". From the simulation results, it can be observed that both baselines achieved worse short-term prediction accuracy while the long-term performance remains on the same level as the vanilla models. We hypothesize that the result is contributed by the recurrent structure of STCNN and DCRNN. While such structure is capable of accepting variable-length inputs, the different sampling intervals within closeness, period, and trend time instances impose additional difficulty for the models to extract the correct multi-scale temporal correlation.

Last but not least, we summarize the computation time of training ST-GNet and compared deep learning-based approaches, namely, STCNN, GCNN-LSTM, STDN, and DCRNN. On the BJ dataset,[4] the training takes approximately 37.4 h, 21.0 h, 25.5 h, and 14.5 h on one testing GPU, respectively. In the meantime, the proposed ST-GNet takes around 11.3 h to be fully trained with the same dataset and computing facility. The highly notable improvement in training efficiency is due to the removal of recurrent neurons from the speed predictor. All of the four compared approaches employ recurrent neural network principles to guide the design, which on the one hand handles time-series data processing by nature but on the other hand can hardly be parallelized to benefit from modern computing acceleration technologies by GPU. The trained model can be directly used for online inference, which takes <1 s to predict the traffic speed of the next full day with a 5 min resolution. We argue that the proposed ST-GNet is capable for real-time implementation in realistic scenarios. As hundreds to thousands of measurement locations may be involved in practice, the practical training time of ST-GNet is on par with the BJ, SH, and GZ case studies. Nonetheless, the resulting deep learning model remains valid once well-trained until remarkable traffic dynamics changes occurred, which are in most cases led by new roads (measurement locations). Considering that such situations are typically well-planned weeks if not months earlier and are not frequent, the model training time is indeed insignificant. On the other hand, the sub-second online inference time grants service providers the possibility of making day-ahead predictions at the start of every hour or minute. Therefore, we consider the forecast of ST-GNet to be online.

C. Ablation Test

In the proposed ST-GNet, two deep neural networks, namely, the predictor and regularizer, are orchestrated for long-term traffic speed prediction. In this case study, we carry out a comprehensive ablation test to validate the contribution of these sub-networks to the overall model performance. Specifically, new ST-GNet variants are constructed based on the original ST-GNet design as follows:

- Pred-Net: Only the predictor as shown in Figure 1 is used for long-term prediction.
- GCNN-RegNet: The original predictor is substituted by GCNN-LSTM, whose output is regularized by the original regularizer.
- Trend-Net, Period-Net, and Closeness-Net: Both the predictor and the regularizer are used, but only trend, period, or closeness historical data serve as the model input.
- NoRes-Net: All residual links in ResGC layers are removed from the original predictor.
- Dense-Net: $\mathbf{L} = \mathbf{I}$ in (1) and (3) so that the additional transportation network topology information is discarded.
- Speed-Net: (3) is replaced by (1) and no meteorology data is used in the training process.

For conciseness without loss of generality, all compared models are trained and tested on the BJ dataset. The simulation results are depicted in Figure 2.

A general conclusion can be derived from the comparison: the original ST-GNet unsurprisingly outperforms all other variants. This conclusion supports the use of constituting components of ST-GNet, namely, the predictor-regularizer design, the trend-period-closeness data pre-processing, the residual

[4] The order of training time on SH, GZ, and TaxiBJ is identical to that of the BJ dataset.

links and the particular architecture of ResGC-based predictor. Comparing ST-GNet with Pred-Net, the results clearly indicate that the introduction of the proposed regularizer dramatically improves the overall prediction accuracy. This supports the previous hypothesis that spatial-temporal traffic dynamics are among the critical factors influencing speed forecast performance, and the proposed regularizer successfully extracts and incorporates such information.

Furthermore, replacing the proposed predictor with GCNN-LSTM leads to undermined overall performance: the inclusion of residual links notably alleviates the possibility of over-fitting issue during the training process. This property also explains the performance gap between ST-GNet and NoRes-Net. None among Trend-Net, Period-Net, or Closeness-Net performs comparably to ST-GNet, which emphasizes the importance of all past time inputs.

Finally, the additional knowledge provided to the ST-GNet besides traffic speed, namely, transportation network topology and meteorology information, also contributes to its performance improvement over baseline approaches. This is in accordance with the intuition. While the road topology increases the volume of input data, it does not necessarily lead to over-fitting: in Dense-Net, extra computation effort shall be made to learn the spatial data dependency among road segments, which is greatly captured by the road topology. Besides, the additional trainable parameters introduced with meteorological data in (3) is minuscule in volume compared with others. Therefore, the parameter searching space is not notably expanded, and the respective information positively contributes to the data learning.

A direct comparison of Table II and Fig. 2 reveals that Pred-Net is notably inferior than state-of-the-art baselines especially in short-term predictions. This is mainly contributed by two factors: the sub-optimal hyper-parameters and the simple design of the fusion layer (5). If the fusion operation is substituted by two consecutive Res-GC operations and a better hyper-parameter configuration is adopted, the new Pred-Net variant can deliver 11.2% and 14.5% MAPE at $T+1$ and $T+D$ on BJ dataset, respectively. Nonetheless, the non-negligible increased model capacity (and in turn the parameter space) does not contribute to an expected performance improvement over ST-GNet if the regularizer is appended. As a result, the significantly increased training time hinders the new Pred-Net variant from being included in ST-GNet, and we maintain (5) as the fusion operation.

D. ST-GNet Hyper-Parameters

In addition to the constituting components of ST-GNet, the selection of hyper-parameters that determine the model architecture is another critical performance-influencing factor. In particular, we first evaluate the impact of layer count (L for predictor and R for regularizer) and number of neurons (equivalent to $F^{(l+1)}$) on prediction accuracy by considering hyper-parameter configurations presented in Table III. All variants are trained on the same BJ dataset.

From the simulation results, a few conclusions can be derived. First, reducing the model capacity by removing either layers (model Param-A) or neurons (model Param-B) leads to inferior prediction accuracy, although the training time is accordingly reduced. This is partly due to the nature of graph convolution operations. While each graph convolution carries out one-hop neighborhood mixing along the transportation network adjacency matrix, it is highly plausible that traffic dynamics are highly correlated within a larger range. Therefore, increasing the number of consecutive graph convolutions potentially helps local information propagate to a larger regions, leading to better accuracy. In the meantime, making neural networks deeper does not come without drawbacks: training time increases with the total number of learnable parameters in the model. Furthermore, increasing number of parameters may lead to an exponentially larger parameter searching space, which introduces the over-fitting issue to the network training. This is the main reason of insignificant performance improvement by models Param-C, Param-D, Param-E, and Param-F, which all increase the model capacity. To conclude, by increasing the number of neurons and layers in ST-GNet predictor and regularizer, the prediction accuracy can be improved until the increasing training difficulty offsets the model capacity growth.

TABLE III

ST-GNet Hyper-Parameter Models and Prediction Accuracy

Model	Predictor		Regularizer		Average		Training
	L	$F^{(l+1)}$	R	$F^{(l+1)}$	MAPE	RMSE	
ST-GNet	4	256	6	256	9.1%	3.64	11.3 h
Param-A	3	256	4	256	10.1%	4.06	8.6 h
Param-B	4	128	6	128	10.3%	4.21	7.2 h
Param-C	5	256	8	256	9.1%	3.63	16.7 h
Param-D	4	512	6	256	9.3%	3.77	14.0 h
Param-E	4	256	6	512	9.4%	3.79	15.2 h
Param-F	3	512	4	512	9.3%	3.80	19.2 h

TABLE IV

ST-GNet Input Data Models and Prediction Accuracy

Model	N_c	N_p	N_r	Average		Training
				MAPE	RMSE	
ST-GNet	6	6	6	9.1%	3.64	11.3 h
Param-G	3	6	6	9.5%	3.77	10.9 h
Param-H	6	3	6	9.4%	3.76	10.7 h
Param-I	6	6	3	9.2%	3.76	10.9 h
Param-J	3	3	3	9.9%	4.06	10.2 h
Param-K	9	9	9	9.1%	3.63	12.5 h

Besides the traditional neural network hyper-parameters investigated above, ST-GNet also employs a set of hyper-parameters to control the input data volume, i.e., the number of time instances in each time group N_c, N_p, and N_r. To assess the model's sensitivity to these parameters, an additional set of ST-GNet variants is constructed as shown in Table IV. The simulation results imply that the proposed ST-GNet is not sensitive to the volume of input data as long as sufficient historical information is provided. While reducing N_c, N_p, or N_r leads to slightly worse MAPE and RMSE performance, even the worse-performing hyper-parameter setting, namely, $N_\mathrm{c} = N_\mathrm{p} = N_\mathrm{r} = 3$, can still achieve a better prediction accuracy than the state-of-the-art on the BJ dataset, i.e., STCNN and STDN according to Table II. This indicates that proper

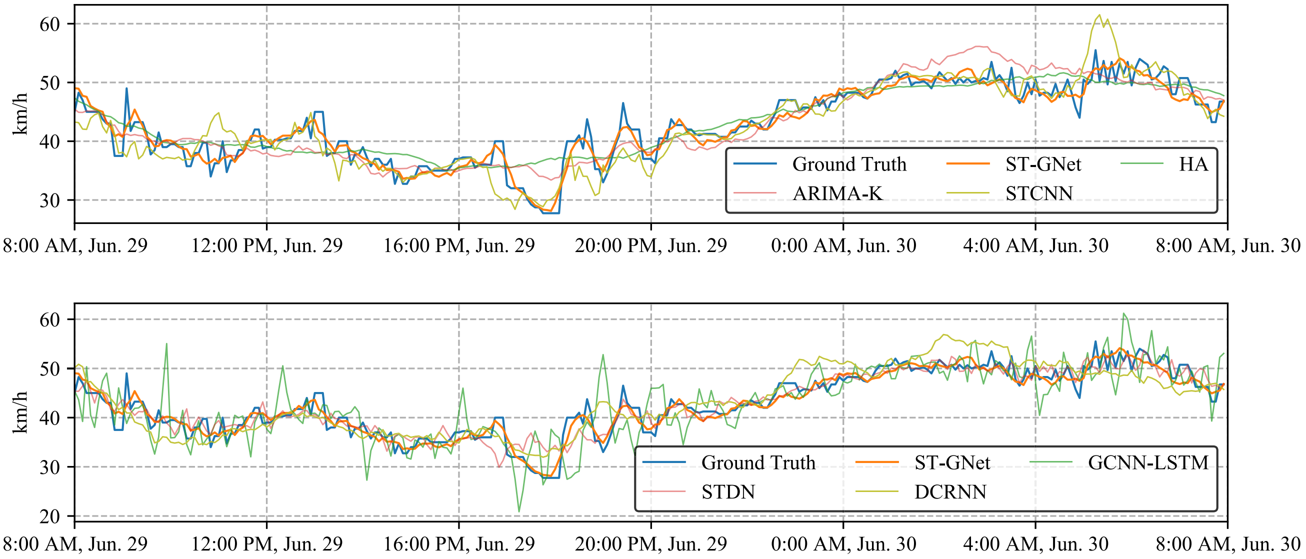

Fig. 3. Week-ahead traffic speed prediction of West Chang'an Avenue in Beijing.

hyper-parameter selection can help optimize the prediction performance of ST-GNet, yet is not required to supersede existing long-term traffic speed prediction approaches. This conclusion is also supported by a test that employs all models from Param-A to Param-K in Tables III and IV on the other three datasets besides the BJ one, where similar results are developed. For conciseness, the respective case studies are not presented.

E. Week-Ahead Prediction

In advanced traffic management system operations, week-ahead knowledge of transportation conditions can potentially benefit the decision making of system operators [9], [19]. In this case study, we conduct a preliminary test on the week-ahead forecast performance of the proposed ST-GNet and compare it with existing state-of-the-art. Please note that as ST-GNet is not particularly designed for this task, it is highly possible that adjustments can further improve the performance, which is beyond the scope of this paper. We employ ST-GNet to progressively generate future traffic speed predictions until week-ahead ones are developed. Such multi-stage prediction paradigms are widely adopted in related research, see the baselines [7], [14] for examples.

TABLE V
D+7 WEEK-AHEAD PREDICTION ACCURACY COMPARISON

MAPE	BJ	SH	GZ	TaxiBJ
ST-GNet	12.4%	12.4%	12.5%	11.5%
HA	12.7%[†]	13.2%	13.8%	11.9%
ARIMA-K	22.4%	20.9%	22.3%	19.1%
STCNN	15.4%	15.9%	15.4%	14.5%
GCNN-LSTM	16.9%	17.1%	17.6%	15.9%
STDN	18.6%	18.5%	19.3%	16.2%
DCRNN	18.9%	17.9%	17.9%	16.2%
RMSE (km/h)	BJ	SH	GZ	TaxiBJ
ST-GNet	5.04	4.86	5.65	5.51
HA	5.15[†]	5.17	6.07	5.72
ARIMA-K	8.79	8.14	10.08	9.17
STCNN	6.11	6.18	6.77	7.10
GCNN-LSTM	6.66	6.70	7.90	7.82
STDN	7.51	7.21	8.48	7.77
DCRNN	7.69	7.03	7.94	7.81

[†] Wilcoxon rank-sum test approves the null hypothesis at 95% significance.

The simulation results are summarized in Table V, where the prediction accuracies of the compared approaches on the seventh day from T are presented. Please note that due to the error amplification nature of multi-stage predictions, ST-GNet and other baselines except for HA may perform better during the seven day from Day 1 to Day 7 than the presented figures. From the simulation results, it can be observed that the proposed ST-GNet and all baseline approaches except for HA experience performance degradation compared to the previous day-ahead prediction accuracy. This is mainly due to two factors. On the one hand, the progressive prediction scheme adopted for multi-stage forecast enables single-step prediction errors to accumulate, resulting in inferior week-ahead accuracy. On the other hand, the model training process of these methods is conducted only on the ground truth data. During the online forecasting process, however, model generated contexts are also considered as ground truth. Such train-test discrepancy also undermines the week-ahead prediction performance. In the meantime, neither of the above issues is adopted by HA. Therefore, its week-ahead prediction accuracy is almost identical to that of day-ahead predictions, with the differences being less than 1.5% MAPE. Wilcoxon rank-sum test also approves the null hypothesis that ST-GNet performs similarly to HA on the BJ dataset. Nonetheless, we would like to note that the main objective of this work is to propose a new long-term traffic speed prediction model for day-ahead prediction. ST-GNet does not rely on the multi-step forecasting paradigm to resolve the error amplification issue in the previous literature. However, when employing the model to week-ahead prediction, we have to resort to recursive forecasting, which brings along the issue. Regarding that

ST-GNet can still provide improvements over state-of-the-art baselines at D + 7, ST-GNet still benefits from its predictor-regularizer design despite the adversarial issue, and can produce the best week-ahead traffic speed predictions among all baselines. On the other hand, the insignificant improvements of ST-GNet over HA on D + 7 renders the proposed model not an obvious substitution of the naïve HA method for beyond-seven-day prediction due to the model complexity. An important furture research is to find solutions for extending the advantage of ST-GNet on day-ahead predictions to week-ahead ones.

To better illustrate the characteristics of ST-GNet, we visualize the week-ahead traffic speed prediction results of West Chang'an Avenue in Beijing with the best-performing baselines in Figure 3. Compared with the other two deep learning-based prediction approaches, i.e., GCNN-LSTM and STCNN, ST-GNet develops much smoother week-ahead predictions thanks to the spatial-temporal data correlation extraction achieved by the regularizer, which can partially mitigate the negative effects of exposure bias and error accumulation. Additionally, ST-GNet benefits from the non-linear data modeling capability of deep learning models, which is absent in HA. Therefore, the smooth predictions by ST-GNet can follow the general trend of traffic dynamics more closely than HA. To conclude, while not specifically designed for the task, the proposed ST-GNet can generally better forecast week-ahead traffic speed dynamics than existing approaches.

V. Conclusion

In this paper, we propose a novel long-term high-granularity urban traffic speed prediction with graph deep learning techniques. Compared with the state-of-the-art deep learning-based traffic speed prediction approaches, the proposed ST-GNet employs a new predictor-regularizer neural network architecture to improve the capability of spatial-temporal correlation extraction. The proposed model utilizes transportation network topology information and divides past historical traffic data into trend, period, and closeness groups in order to capture the traffic dynamics from heterogeneous time scales. Furthermore, predicted time-series speed data are regularized by a graph convolution-driven neural network, which aims to reconstruct the time-series by incorporating spatial-temporal latent information from the ground truth.

To assess the performance of the proposed model, we conduct a series of comprehensive case studies on four real-world traffic datasets. Compared with baseline approaches, the proposed ST-GNet develops more accurate day-ahead and week-ahead traffic speed predictions with the same volume of training data. An ablation test is carried out to validate the efficacy of the constituting components of ST-GNet. Finally, we investigate the sensitivity of ST-GNet hyper-parameters on the prediction performance. In the future, we plan to integrate Bayesian deep learning techniques [48] in ST-GNet to produce future traffic speed probabilities for more robust predictions as well as exploring advanced models for far-future predictions.

References

[1] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 4, pp. 606–616, Aug. 2011.

[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] J. L. Adler and V. J. Blue, "Toward the design of intelligent traveler information systems," *Transp. Res. C, Emerg. Technol.*, vol. 6, no. 3, pp. 157–172, Jun. 1998.

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

[5] 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.

[6] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883–4894, Nov. 2019.

[7] T. Bogaerts, A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva, and P. Hellinckx, "A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data," *Transp. Res. C, Emerg. Technol.*, vol. 112, pp. 62–77, Mar. 2020.

[8] Z. He, C. Chow, and J. Zhang, "STCNN: A spatio-temporal convolutional neural network for long-term traffic prediction," in *Proc. IEEE Int. Conf. Mobile Data Manage.*, Hong Kong, Jun. 2019, pp. 226–233.

[9] Z. Hou and X. Li, "Repeatability and similarity of freeway traffic flow and long-term prediction under big data," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1786–1796, Jun. 2016.

[10] L. Qu, W. Li, W. Li, D. Ma, and Y. Wang, "Daily long-term traffic flow forecasting based on a deep neural network," *Expert Syst. Appl.*, vol. 121, pp. 304–312, May 2019.

[11] D. Zang, J. Ling, Z. Wei, K. Tang, and J. Cheng, "Long-term traffic speed prediction based on multiscale spatio-temporal feature learning network," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3700–3709, Oct. 2019.

[12] P. Duan, G. Mao, C. Zhang, and S. Wang, "STARIMA-based traffic prediction with time-varying lags," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1610–1615.

[13] Z. He, C.-Y. Chow, and J.-D. Zhang, "STANN: A spatio-temporal attentive neural network for traffic prediction," *IEEE Access*, vol. 7, pp. 4795–4806, 2019.

[14] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, Jan. 2019, pp. 5668–5675.

[15] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," 2017, *arXiv:1705.02699*. [Online]. Available: http://arxiv.org/abs/1705.02699

[16] 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.

[17] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, New York, NY, USA, Oct. 2016, pp. 1–4.

[18] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1–7.

[19] G. Lu, J. Li, J. Chen, A. Chen, J. Gu, and R. Pang, "A long-term highway traffic flow prediction method for holiday," in *Proc. Int. Conf. on. Future Inf. Technol.*, Salerno, Italy, Apr. 2018, pp. 153–159.

[20] I. Laña, J. L. Lobo, E. Capecci, J. Del Ser, and N. Kasabov, "Adaptive long-term traffic state estimation with evolving spiking neural networks," *Transp. Res. C, Emerg. Technol.*, vol. 101, pp. 126–144, Apr. 2019.

[21] Z. Cui, L. Lin, Z. Pu, and Y. Wang, "Graph Markov network for traffic forecasting with missing data," *Transp. Res. C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102671.

[22] 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.

[23] S. Lee and D. B. Fambro, “Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting,” *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1678, no. 1, pp. 179–188, Jan. 1999.

[24] J. Guo, W. Huang, and B. M. Williams, “Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification,” *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 50–64, Jun. 2014.

[25] M. Van Der Voort, M. Dougherty, and S. Watson, “Combining Kohonen maps with ARIMA time series models to forecast traffic flow,” *Transp. Res. C, Emerg. Technol.*, vol. 4, no. 5, pp. 307–318, Oct. 1996.

[26] C. P. I. J. van Hinsbergen, T. Schreiter, F. S. Zuurbier, J. W. C. van Lint, and H. J. van Zuylen, “Localized extended Kalman filter for scalable real-time traffic state estimation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 385–394, Mar. 2012.

[27] F. Wu, H. Wang, and Z. Li, “Interpreting traffic dynamics using ubiquitous urban data,” in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, New York, NY, USA, Oct. 2016, pp. 1–4.

[28] Y. Tong *et al.*, “The simpler the better: A unified approach to predicting original taxi demands based on large-scale online platforms,” in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2017, pp. 1653–1662.

[29] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, “Predicting citywide crowd flows using deep spatio-temporal residual networks,” *Artif. Intell.*, vol. 259, pp. 147–166, Jun. 2018.

[30] Y. Gu, W. Lu, L. Qin, M. Li, and Z. Shao, “Short-term prediction of lane-level traffic speeds: A fusion deep learning model,” *Transp. Res. C, Emerg. Technol.*, vol. 106, pp. 1–16, Sep. 2019.

[31] R. Ke, W. Li, Z. Cui, and Y. Wang, “Two-stream multi-channel convolutional neural network for multi-lane traffic speed prediction considering traffic volume impact,” *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2674, no. 4, pp. 459–470, Apr. 2020.

[32] Y. Ma, Z. Zhang, and A. Ihler, “Multi-lane short-term traffic forecasting with convolutional LSTM network,” *IEEE Access*, vol. 8, pp. 34629–34643, 2020.

[33] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, “Road traffic forecasting: Recent advances and new challenges,” *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 2, pp. 93–109, Apr. 2018.

[34] S. Sun, H. Wu, and L. Xiang, “City-wide traffic flow forecasting using a deep convolutional neural network,” *Sensors*, vol. 20, no. 2, p. 421, Jan. 2020.

[35] L. Zhao *et al.*, “T-GCN: A temporal graph convolutional network for traffic prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2019.

[36] C. Chen *et al.*, “Gated residual recurrent graph neural networks for traffic prediction,” in *Proc. AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, Jan. 2019, pp. 485–492.

[37] 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, Jan. 2019, pp. 922–929.

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

[39] X. Geng *et al.*, “Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, pp. 3656–3663.

[40] 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.

[41] J. Chen, T. Ma, and C. Xiao, “FastGCN: Fast learning with graph convolutional networks via importance sampling,” in *Proc. Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, May 2018, pp. 1–15.

[42] 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.

[43] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, “Simplifying graph convolutional networks,” in *Proc. Int. Conf. Mach. Learn.*, Long Beach, CA, USA, Jun. 2019, pp. 6861–6871.

[44] 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. 2020.

[45] L. Zhao and L. Akoglu, “PairNorm: Tackling oversmoothing in GNNs,” in *Proc. Int. Conf. Learn. Represent.*, Addis Ababa, Ethiopia, Apr. 2020, pp. 1–17.

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

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

[48] H. Wang and D.-Y. Yeung, “Towards Bayesian deep learning: A framework and some existing methods,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3395–3408, Dec. 2016.

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. From 2015 to 2018, he was a Postdoctoral Fellow with The University of Hong Kong. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China, and an Honorary Assistant Professor with the Department of Electrical and Electronic Engineering, The University of Hong Kong. He is also the Chief Research Consultant of GWGrid Inc., Zhuhai, and Fano Labs, Hong Kong. His research interests include smart city and urban computing, deep learning, intelligent transportation systems, and smart energy systems. He is an Editor of the *IET Smart Cities* journal.

Christos Markos received the Diploma degree in electrical and computer engineering from the University of Thessaly, Greece, in 2017. He is currently pursuing the joint Ph.D. degree with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. He is also with the Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. His research interests include deep learning, data mining, and their application to smart cities and transportation.

Shiyao Zhang (Member, IEEE) received the B.S. degree (Hons.) in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2014, the M.S. degree in electrical engineering (electric power) from the University of Southern California, Los Angeles, CA, USA, in 2016, and the Ph.D. degree from The University of Hong Kong, Hong Kong. He is currently a Postdoctoral Research Fellow with the Academy for Advanced Interdisciplinary Studies, Southern University of Science and Technology. His research interests include smart energy systems, intelligent transportation systems, optimization theory and algorithms, and deep learning applications.