A Universal Framework of Spatiotemporal Bias Block for Long-Term Traffic Forecasting

Fuqiang Liu, *Student Member, IEEE*, Jiawei Wang, *Student Member, IEEE*, Jingbo Tian, Dingyi Zhuang, Luis Miranda-Moreno, and Lijun Sun, *Member, IEEE*

***Abstract*— Recent studies have demonstrated the great success of graph convolutional networks in short-term traffic forecasting (e.g., 15-30 min ahead) tasks by capturing dependencies in road network structure. Based on these models, long-term forecasting can be achieved by two approaches: (1) recursively generating a one-step-ahead prediction and (2) adapting the models to sequence-to-sequence (seq2seq) learning. However, in practice, these two approaches often show poor performance in long-term forecasting tasks. The recursive approach suffers from the error accumulation problem, as the model is trained based on one-step-ahead loss. On the other hand, seq2seq shows convergence issues that limit its application. To address the issues for long-term forecasting, in this paper, we propose a universal framework that directly transforms any existing state-of-the-art models for one-step-ahead prediction to achieve more accurate long-term forecasting. The proposed framework consists of two components—a base model and a bias block. The base model is assumed to be a well-trained state-of-the-art one-step-ahead forecasting model, and the bias block is constructed by a spatiotemporal graph neural network composed of gated temporal convolution layers and graph convolution layers. The base model and the bias block are residually-connected so that we can substantially reduce the training complexity. Extensive experiments are conducted on existing benchmark datasets. We experiment with several state-of-the-art models in the literature as base models, and our results demonstrate the ability of the proposed universal framework to greatly improve the long-term prediction accuracy for all models.**

***Index Terms*— Traffic prediction, spatiotemporal graph neural network, gated convolutional network, residual connection.**

I. Introduction

TRAFFIC forecasting is a critical component of intelligent transportation systems (ITS) [1]–[4]. Accurate and reliable traffic forecasting benefits a wide range of agents from individual drivers, to commercial organizations, and to transport authorities. With recent advances in data acquisition technology, it becomes increasingly important to exploit the spatiotemporal patterns embedded in large-scale data to improve traffic forecasting. Deep learning has shown clear advantages in its ability of capturing complex nonlinear relations in data and utilizing latent features without tricky feature engineering. Generally, Graph Neural Networks (GNNs) and temporal deep learning models such as Long Short-Term Memory (LSTM) and Gated Convolutional Neural Networks (Gated-CNNs) are employed to capture spatiotemporal patterns in traffic data [5]–[11].

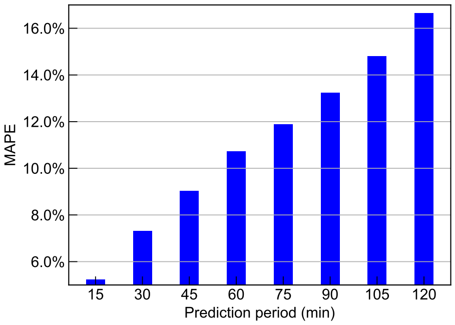


Fig. 1. An example of error accumulation. In this test, the spatiotemporal graph neural network [17] is tested on PeMS.

Based on these deep learning models, there are two general two approaches for multi-step traffic forecasting, namely: iterative multi-step forecasting and sequence-to-sequence (seq2seq) based forecasting [12]–[16]. However, these two approaches have some major limitations, especially when it comes to long-term traffic forecasting (over 30 min in lead time). The central idea of iterative multi-step scheme is to iteratively feed the output of current prediction as an input into the subsequent prediction. This idea is widely adopted by many state-of-the-art models [17]–[19]. However, the iterative use of previous results leads to diffusion and accumulation of prediction errors throughout the whole process. Consequently, prediction errors of the said methods grow substantially with the length of prediction horizons. Empirical experiments suggest a sharp increase in errors as the prediction horizon passes 30 minutes, as illustrated in Figure 1. The seq2seq-based models directly produce sequences of multi-period predictions [20]–[24] based on the sequences of historical ground truth. Seq2seq models are typically computationally expensive to train, largely due to the convergence problems

Manuscript received August 7, 2021; revised December 3, 2021 and February 19, 2022; accepted March 3, 2022. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, in part by the National Natural Science Foundation of China (NSFC)-Fonds de Recherche du Québec-Société et Culture (FRQSC) Research Program on Smart Cities and Big Data, and in part by the Canada Foundation for Innovation (CFI). The work of Fuqiang Liu was supported by the Fonds de Recherche du Québec-Nature et technologies (FRQNT) through the B2X Doctoral Scholarship. The Associate Editor for this article was H. Dong. *(Corresponding author: Lijun Sun.)*

The authors are with the Department of Civil Engineering, McGill University, Montreal, QC H3A 0C3, Canada (e-mail: fuqiang.liu@mail.mcgill.ca; jiawei.wang4@mail.mcgill.ca; jingbo.tian@mail.mcgill.ca; dingyi.zhuang@mail.mcgill.ca; luis.miranda-moreno@mcgill.ca; lijun.sun@mcgill.ca).

Digital Object Identifier 10.1109/TITS.2022.3157129

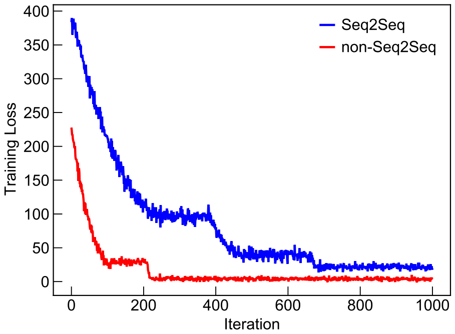


Fig. 2. Loss lines of the seq2seq-based model and iterative multi-step forecasting model. Both structure and parameter sizes of their input layers and intermediate layers are as same as the spatiotemporal graph convolutional network [17]. The loss of seq2seq have been divided by 24 because the final output channel is 24, while the output for non-seq2seq model is one.

rooted in the high complexity of gradients computations when updating model parameters. The complex gradients, even worse with gradient vanishing, make the model prohibitively expensive to train [25]. And enlarging a model is likely to worsen the performance because of the increased likelihood of converging to the local optima. As an example, Figure 2 shows the learning curves of deep learning-based traffic forecasting models under competing prediction schemes. It can be seen from Figure 2 that the seq2seq-based model takes more iterations to converge, with a training loss consistently higher than that of the non-seq2seq model.

To address the above issues, we propose a universal framework free of convergence problem and the associated error accumulation. The framework comprises a base model and a bias block. The base model, used to generate the base prediction, can be any well-designed/trained one-step-ahead forecasting model keeping its all original structure and parameters. The bias block, used to generate a bias sequence of following time steps, is constructed based on spatiotemporal modules composed of both Gated-CNNs and GNNs. The proposed framework can be regarded as dividing a complex seq2seq-based model into two parts. One part is replaced by a well-trained iterative multi-step forecasting model, while the other is a bias block with lower complexity. Instead of training an entire complex seq2seq-based model, we only need to train the bias block with a considerably less complex optimization space in the proposed framework, thus the convergence problem is resolved. Additionally, the accumulated errors of the base model are corrected by the bias block. More qualitative explanations on the proposed framework's effectiveness are detailed in section III-D. Empirical evidence suggests many state-of-the-art traffic prediction models work well under the proposed framework. According to our comparative tests of STGCN [17], DCRNN [18], Graph Wavenet [26], and ASTGNN [27] on PeMS and META-LA dataset, models enhanced by the proposed framework clearly outperform their counterparts (i.e., base) for prediction horizons from 30 min to 120 min.

To summarize, our contributions in this work are as follows:

- We propose an effective universal framework to improve the long-term prediction performance of any well-trained existing prediction model with no substantial changes.
- We design a bias block composed of temporal Gated-CNNs and GNNs. The bias block is able to increase state-of-the-art models' pattern capturing ability with few training penalty.
- We empirically demonstrate the effectiveness and generalizability of the proposed framework by conducting numerical experiments of traffic speed prediction on large-scale real-world data.

The rest of this paper is organized as follows. We first introduce the preliminary knowledge in Section II. The proposed framework is detailed in Section III. Section IV provides the design and setup of the experiments of the proposed framework. Section V concludes the paper with closing remarks and discussion on future work.

II. Preliminary

In this section, we first introduce the two forecasting schemes. Then we detail the graph convolution which lies as the foundation of most state-of-art traffic forecasting models. Then, we introduce the dilated convolution that is able to model the temporal sequence precisely and efficiently.

A. Sequence-to-Sequence and Iterative Multi-Step Prediction

One step traffic prediction can be summarized as

$$\mathcal{G}^*_{t+1} = f\left(\left\{\mathcal{G}_t, \mathcal{G}_{t-1}, \dots \mathcal{G}_{t-(H-1)}\right\}, \Phi\right), \tag{1}$$

where $\mathcal{G}_t = \{\mathcal{V}_t, \mathcal{E}, W\}$ denotes the graph-structured traffic condition at time step t, consisting of dynamic traffic information in vertices $\mathcal{V}_t$, edge set $\mathcal{E}$, and a weighted adjacency matrix $W \in R^{N\times N}$, and $\mathcal{G}^*_{t+1}$ denotes the prediction result at time step $t+1$. We use f to denote the prediction model parameterized by Φ, with H being the number of historical records used in the prediction model.

There are two main schemes for multi-step forecasting. The sequence-to-sequence (seq2seq) forecasting, which directly generates multi-step predictions, can be formulated as

$$\left\{\mathcal{G}^*_{t+T}, \mathcal{G}^*_{t+T-1}, \dots, \mathcal{G}^*_{t+1}\right\} = f\left(\left\{\mathcal{G}_t, \mathcal{G}_{t-1}, \dots \mathcal{G}_{t-(H-1)}\right\}, \Phi\right), \tag{2}$$

where T is the prediction window and assumed $T < H$ here.

Different from directly generating multiple predictions as seq2seq, the iterative multi-step forecasting, which generates one prediction first and then feeds the prediction into the model to generate the subsequent prediction, can be formulated as

$$\begin{cases} \mathcal{G}^*_{t+1} = f\left(\left\{\mathcal{G}_t, \mathcal{G}_{t-1}, \dots \mathcal{G}_{t-(H-1)}\right\}, \Phi\right), \\ \mathcal{G}^*_{t+2} = f\left(\left\{\mathcal{G}^*_{t+1}, \mathcal{G}_t, \dots \mathcal{G}_{t-(H-2)}\right\}, \Phi\right), \\ \vdots \\ \mathcal{G}^*_{t+T} = f\left(\left\{\mathcal{G}^*_{t+T-1}, \mathcal{G}^*_{t+T-2}, \dots \mathcal{G}^*_{t-(H-T)}\right\}, \Phi\right). \end{cases} \tag{3}$$

Most state-of-the-art traffic forecasting models (e.g., [17], [18]) utilize the iterative scheme, which is easy to train and converge. However, as mentioned previously, the scheme is vulnerable to the error accumulation problem in long-term forecasting.

B. Convolutions on Graph

Considering that traffic data are collected from irregular sensor networks (i.e. loop detectors deployed across the road network) and the spatial correlations among different sensors, the graph can be a good representation of traffic data. In this section, we introduce the graph convolution operation to capture spatial patterns embedded in the traffic data.

A common approach for graph convolution is to work in the spectral domain with graph Fourier transforms [28], which is referred to as "spectral graph convolution". The spectral graph convolution is defined as

$$\Theta *_g x = \Theta\left(L\right)x = \Theta\left(U\Lambda U^T\right)x = U\Theta\left(\Lambda\right)U^T x, \quad (4)$$

where $*_g$ denotes the spectral graph convolution, $x \in \mathbb{R}^n$ denotes a input signal, Θ denotes parameters of graph convolution kernel, $L = I_n - D^{-\frac{1}{2}}WD^{-\frac{1}{2}} = U\Lambda U^T$ denotes the normalized graph Laplacian (I_n represents an identity matrix, $W \in \mathbb{R}^{n\times n}$ represents the adjacency matrix, $D \in \mathbb{R}^{n\times n}$ represents the diagonal degree matrix with $D_{ii} = \sum_j W_{ij}$), $U \in \mathbb{R}^{n\times n}$ denotes the matrix composed of eigenvectors of L, and $\Lambda \in \mathbb{R}^{n\times n}$ denotes the diagonal matrix composed of eigenvalues of L. In addition, the filter $\Theta\left(\Lambda\right)$ is a diagonal matrix too.

By the definition in Eq. (4), the graph convolution between the graph signal x and the kernel Θ is transformed to the multiplication between the kernel Θ and graph Fourier transformed signal $U^T x$ in the graph spectral domain [28]. However, the computational complexity of the above spectral graph convolution operation is very expensive—$\mathcal{O}\left(n^2\right)$. To simplify the spectral graph convolution and reduce parameters, Chebyshev Polynomials approximation [17] is often used, which approximates kernel Θ by a polynomial of Λ:

$$\Theta\left(\Lambda\right) \approx \sum_{k=0}^{K-1} \theta_k \Lambda^k, \quad (5)$$

where $\theta \in \mathbb{R}^K$ is a vector composed of polynomial coefficients and K is the graph convolution kernel size.

Generally, graph convolution kernels are approximated by Chebyshev polynomial in a truncated form, which is defined as

$$\Theta\left(\Lambda\right) \approx \sum_{k=0}^{K-1} \theta_k T_k\left(\bar{\Lambda}\right), \quad (6)$$

where $T_k\left(x\right)$ denotes Chebyshev polynomial, scaled $\bar{\Lambda} = 2\Lambda/\lambda_{max} - I_n$, and λ_{max} is the largest eigenvalue. Then the spectral graph convolution in Eq. (4) becomes

$$\Theta *_g x = \Theta\left(L\right)x \approx \sum_{k=0}^{K-1} \theta_k T_k\left(\bar{L}\right)x. \quad (7)$$

To further reduce the complexity, the order k is generally set to 1 and assume $\lambda_{max} \approx 2$ empirically. Thus, we have

$$\begin{aligned}\Theta *_g x &\approx \theta_0 x + \theta_1\left(\frac{2}{\lambda_{max}L - I_n}\right)x\\ &\approx \theta_0 x + \theta_1\left(D^{-\frac{1}{2}}WD^{-\frac{1}{2}}\right)x, \quad (8)\end{aligned}$$

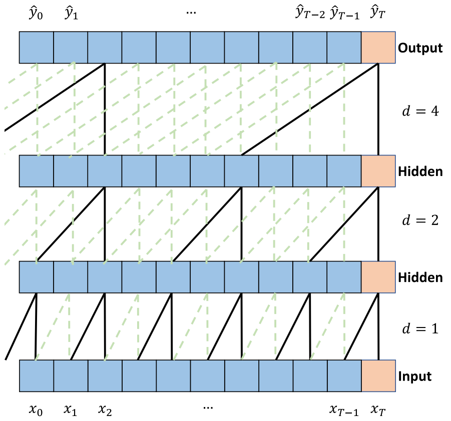


Fig. 3. An example of dilated convolutions [30]. Dilation factors d are equal to 1,2,4 and the convolutional filter size $k = 2$.

where θ_0 and θ_1 represent kernels sharing the same parameter, i.e., $\theta_0 = \theta_1 = \theta$, to improve the training efficiency. With the above Chebyshev polynomial approximation, the computational complexity of spectral graph convolution is reduced to linear $\mathcal{O}\left(n\right)$.

C. Dilated Convolution

Generally, the temporal sequence modeling and analysis is synonymous with recurrent neural networks (RNN), e.g., LSTM and GRU. Most existing deep learning-based traffic prediction works [29] are based on RNN. Yet recent studies indicate that CNNs can outperform recurrent architectures on sequence and time-series modeling [30]. Compared with RNN-like models, temporal convolutional network (TCN) is not only more accurate but also computationally cheaper. In this section, we introduce the basic knowledge and architecture of TCN.

Different from traditional CNN, TCN requires two strict assumptions: 1) the length of TCN outputs is as same as inputs and 2) no future information is leaked into the past. To satisfy the aforementioned two conditions, a 1D fully-convolutional network (FCN), of which the hidden layer has the same length as the input layer, is applied in TCN. Besides, TCN utilizes zero padding to keep the length of subsequent layers equal to the length of previous layers. In addition, the TCN applies causal convolutions, where output at time t is computed only from elements from time t and earlier. Overall, TCN [30] can be formulated as

$$TCN = 1D\ FCN + causal\ convolutions. \quad (9)$$

However, the basic architecture of TCN cannot be expanded to a long history size effectively, because the depth of the network on causal convolution operations increases significantly with the length of the historical sequence to process. Extremely large convolution filters, which are computationally expensive, are needed to analyze long time series. To process the long time series both effectively and efficiently, the dilated convolutions are introduced in [30] (see Figure 3). For a 1-D

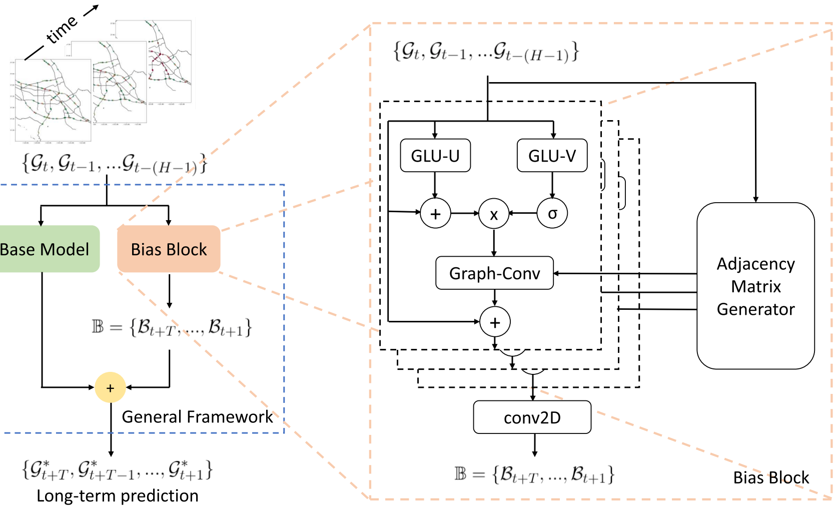

time
$\{\mathcal{G}_t, \mathcal{G}_{t-1}, ...\mathcal{G}_{t-(H-1)}\}$
Base Model
Bias Block
$\mathbb{B} = \{\mathcal{B}_{t+T}, ..., \mathcal{B}_{t+1}\}$
+
General Framework
$\{\mathcal{G}^*_{t+T}, \mathcal{G}^*_{t+T-1}, ..., \mathcal{G}^*_{t+1}\}$
Long-term prediction
$\{\mathcal{G}_t, \mathcal{G}_{t-1}, ...\mathcal{G}_{t-(H-1)}\}$
GLU-U
GLU-V
+
x
σ
Graph-Conv
+
Adjacency Matrix Generator
conv2D
$\mathbb{B} = \{\mathcal{B}_{t+T}, ..., \mathcal{B}_{t+1}\}$
Bias Block

Fig. 4. The overall structure of the universal framework.

sequence input $x \in \mathbb{R}^n$ and a filter $f : \{0, \ldots, k-1\} \rightarrow \mathbb{R}$, the dilated convolution operation F is defined as

$$F(s) = (\mathbf{x} *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-d \cdot i}, \tag{10}$$

where s denotes an element in the sequence, d denotes the dilation factor, k denotes the convolutional filter size, $*_d$ denotes the dilated convolution, $\mathbf{x}$ denotes the input, and $s-d \cdot i$ indicates the direction of the past.

As shown in Figure 3, a fixed skip operation between every two adjacency elements is introduced in the dilation. When $d = 1$, dilated convolutions becomes the same as simple causal convolutions. When we use larger dilation factors, the dilated convolution's top level output is able to represent a wider range of inputs and convolution filters' sizes are reduced compared to directly applying the causal convolution operations. Therefore, dilated convolutions greatly expand TCNs' receptive field. Compared with enlarging the filter size k, increasing the dilation factor d is a more effective and efficient way to mine long historical sequences.

III. Methodology

In this section, we first introduce the overall framework followed by the residual connection between the base model and bias block. Then, the design details of the bias block are illustrated. Following the explanation of the framework design, the reason why the proposed framework can fill in the aforementioned knowledge gap and enhance the long-term prediction ability of current forecasting models is discussed. It should be noted that the proposed framework does not modify any internal structure or parameter of the existing well-trained base model. Furthermore, unlike fusion models [31], [32], we restrict model inputs to those used by the base model without introducing external information (e.g., like daily/weekly/monthly periodic patterns). The purpose is to examine whether the improvement in long-term prediction does come from the novel framework design.

A. Overview of the Universal Framework

Figure 4 shows the structure of the proposed universal framework for long-term forecasting. The base model is a well-trained forecasting model (e.g., a state-of-the-art model) to generate preliminary prediction. It should be noted that only one-step-ahead prediction is applied when using the existing traffic forecasting model as the base model. The bias block is used to correct the long-term prediction error of the base model and consequently improve the base model's long-term prediction performance.

Both the base model and the bias block share the same input. Different from the iterative multi-step process as in Eq. (3) and the seq2seq-based forecasting as in Eq. (2), the universal framework integrates outputs of both the base model and bias block:

$$\left\{\mathcal{G}^*_{t+T}, \ldots \mathcal{G}^*_{t+1}\right\} = f\left(\left\{\mathcal{G}_t, \ldots \mathcal{G}_{t-(H-1)}\right\}, \Phi\right) + \mathbb{B}, \tag{11}$$

where f denotes the well-trained base model parameterized by Φ, $\mathbb{B}$ denotes the bias sequence and $\mathbb{B} = \{\mathcal{B}_{t+T}, \ldots, \mathcal{B}_{t+1}\} = f'\left(\left\{\mathcal{G}_t, \ldots \mathcal{G}_{t-(H-1)}\right\}, \Phi'\right)$, and f' denotes the bias block neural networks parameterized by Φ'. Parameters Φ of the base model are available since the base model is already well-trained. Thus, the framework only optimize $\Phi^{'*}$—neural network parameters for the bias block:

$$\Phi^{'*} = \underset{\Phi'}{\text{argmin}}\; O_{base} + f'\left(\left\{\mathcal{G}_t, \ldots \mathcal{G}_{t-(H-1)}\right\}, \Phi'\right) - L_{true}, \tag{12}$$

where $O_{base} = f\left(\left\{\mathcal{G}_t, \ldots \mathcal{G}_{t-(H-1)}\right\}\right)$ denotes the output of the base model and $L_{true} = \{\mathcal{G}_{t+1}, \ldots, \mathcal{G}_{t+T}\}$ denotes the ground truth in the training dataset.

We would like to reiterate that both structures and parameters of the base model are fixed all the time—the training process of the bias block does not affect the parameters of the base model. The connection of the base model and the bias block is a type of "residual connection" [33]. The advantage of training a deep learning model with this kind of connection is that the separated part are likely to be merged to be better than the original one. In other words, with the residual connection, the proposed framework can generate predictions no worse than the existing traffic forecasting model as the base model. An extreme example is that the bias block's outputs are forced to 0 consistently and then the final output of the framework is just that of the base model. The residual connection guarantees the lower bound of the proposed framework's performance and the bias block if carefully designed could act to correct the long-term prediction bias to a large degree.

The detailed structure of the bias block is shown in Figure 4, consisting of three spatiotemporal modules, and an adjacency matrix generator used to provide the adjacency matrix to graph convolution operations.

B. Bias Block—Spatiotemporal Module

All three spatiotemporal modules share the same structure of a temporal layer and spatial layer. The **temporal layer** is composed of a dilated TCN with a gated mechanism. Gated mechanisms have proved to be powerful to control information flow through TCNs' layers [26]. Specifically, we utilize the gated linear unit (GLU) to construct the gated mechanism. The output of GLU, x', is computed as

$$x' = \Gamma \otimes x = (U + \mathcal{C}(x)) \odot \sigma(V), \tag{13}$$

where Γ denotes the 1-D convolution kernel, $\otimes$ denotes the 1-D convolution, x denotes the input to the temporal layer, $[U, V]$ denotes the set of two input, which are split in half of same size of channels and fed into the gate mechanism, of TCN [30], $\odot$ denotes the element-wise Hadamard prodoct, $\mathcal{C}(x)$ denotes sampling the input channels to be the same as the half of output channels of TCN, which serves to balance dimensions of two sides in the residual connection, and $\sigma(x) = \frac{1}{1+e^{-x}}$ denotes the sigmoid function. The two intermediate outputs, $[U, V]$, are computed as

$$\begin{cases} U = \Theta_1 \star x_U + \mathbf{b} \\ V = \Theta_2 \star x_V + \mathbf{c}, \end{cases} \tag{14}$$

where Θ_1 and Θ_2 denote parameters of convolutional filters, $\mathbf{b}$ and $\mathbf{c}$ denote parameters of the bias, $\star$ denotes the dilated convolution operation introduced in section II-C, x_U denotes the first half of inputs, and x_V denotes the rest half of inputs.

The output of the temporal layer is then fed into the **spatial layer**, which uses the graph convolution operation delineated in section II-B to capture the dynamic similarity between different vertices, and hidden/non-linear spatial patterns inside the traffic data. The output of the spatial layer, s, is computed as

$$s = \Theta *_g x' = \Theta\left(\widetilde{L}_t\right) x', \tag{15}$$

where $\widetilde{L}_t$ denotes the truncated expansion of the generated adjacency matrix by the generator f'' and $\widetilde{L}_t = f''\left(\left\{\mathcal{G}_t, \ldots \mathcal{G}_{t-(H-1)}\right\}, \Phi''\right)$. Also, the skip connection based on residual network [33] is applied in the spatial layer to improve the effectiveness and efficiency of the training process.

The end output of the whole spatiotemporal module is $o = x + s$ (i.e., input to the temporal layer + output of the spatial layer), which is then fed into the next spatiotemporal module. The output of the third spatiotemporal module is fed into a 2D convolutional layer to get the final output of the bias block $\mathbb{B}$.

C. Bias Block—Adjacency Matrix Generator

Adjacency matrix plays a critical role in extracting spatial information for graph convolution layers. Here we introduce three different approaches to generate the adjacency matrix. First, the adjacency matrix can be computed by the distance between different vertices as in [17]:

$$\begin{cases} w_{ij} = \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right), & i \neq j \text{ and } \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \geq \epsilon \\ 0, & \text{otherwise,} \end{cases} \tag{16}$$

where w_{ij} denotes the weight between sensors i and j, d_{ij} denotes the distance between two sensors, σ and ϵ are hyperparameters controlling the distribution and sparsity of the adjacency matrix W, which are set to 10 and 0.5, respectively, following STGCN [17]. It is important to note that the method as in Eq. (16) requires the prior knowledge like network distance, and the computed matrix is fixed.

Second, the adjacency matrix can be derived directly from training data, just as the idea proposed in the Graph Wavenet [26]. This method requires no prior information, and the adjacency matrix is fixed after the training process.

The aforementioned settings overlook a plausible possibility that spatial correlation might be time-variant (i.e., varies over time) — for example, two sensors may look very similar in midnight but show substantial difference during morning peaks hours. To characterize the time-varying effect, the third approach is to utilize one more spatiotemporal module to generate the adjacency matrix corresponding to the dynamic input, which is named as temporally dynamic adjacency matrix [34]. This method requires no prior information, and it can generate the dynamic matrix under different traffic conditions. The effects of the aforementioned three adjacency matrix generators are analyzed in our numerical experiments. We refer to these three approaches as "Fix", "Com" and "TemD", respectively, in the remainder of this paper.

D. Qualitative Effectiveness Analysis

This section qualitatively explains why the proposed framework would outperform classic seq2seq and iterative multi-step forecasting. In an ideal situation, the performance of one deep neural network can be improved by increasing its scale and complexity, because a larger network has a larger capacity to capture more hidden patterns [35]. However, in practice, training DNNs with complex searching spaces and large numbers of parameters is extremely difficult, mainly because high-dimension parameters lead to serious gradient vanishing [25], [33] and the model tends to converge into local optima. In fact, simply expanding neural networks cannot always improve networks' performance and even a severe accuracy drop.

Adding the proposed framework to a well-trained base forecasting model can overcome this issue. The framework combining the base model and the bias block actually is a wider model. During the new training process, parameters of the base model are fixed and they provide a good initialization to update the left bias block. Training the bias block directly with simple and clean structures also becomes a much easier task. Moreover, the residual connection guarantees the bias block will not cause accuracy drop of the base model [33], [36]–[38]. Taken together, the proposed framework adds to the base model's complexity without few training penalties, which will lead to better performance in comparison with the original base model. Compared with training a larger seq2seq DNN-based forecasting model, it is less likely to encounter the gradient vanishing problem when training the proposed universal framework. In fact, considering that most state-of-the-art DNN-based forecasting models already demonstrate high degree of complexity [26], [34], directly training a larger model will be extremely challenging, time consuming, and even impossible. The proposed framework addresses these challenges by expanding state-of-the-art models (i.e., base), and then transform these models to an easy-to-train seq2seq model with improved long-term forecasting capability. In addition, compared with iterative multi-step forecasting models, there is no recursive process in the new framework, which ensures no error accumulation in the long-term forecasting.

IV. Experiment

In this section, we present the empirical study of the effectiveness and interpretability of the proposed framework. In particular, we compare the performance of different adjacency matrix generators discussed in section IV-B. We also interpret and visualize the results of the proposed universal framework.

A. Setup

1) Datasets: Traffic Speed datasets used to verify the proposed model are PeMS (Freeway Performance Measurement System) and METR-LA. PeMS is collated by the California Department of Transportation (Caltrans) with 12TB of data collected from 35,000 sensors across the state of California since 1999. We use data of 228 road segments and 44 days in our experiments, as in the PeMS data used in STGCN [17]. METR-LA contains traffic information from loop detectors in the Los Angeles County highway [39]. We select data of 207 sensors and 4 months for our experiments, as in DCRNN [18]. In addition, a traffic occupancy dataset, PeMS08 [40] collected from 170 stations in the California Freeway Network, is applied.

2) Baselines: We use vector autoregressive model (VAR) as a traditional base model and HA stands for historical average. Four state-of-the-art deep learning-based models are tested as advanced base models, including:

- **HA**—historical average [18],
- **STGCN**—spatiotemporal graph neural network [17],
- **DCRNN**—diffussion convolution RNN [18],
- **GWNet**—graph wavenet [26],
- **ASTGNN**—attention based spatialtemporal graph neural network [27].

TABLE I
Comparison on Different Adjacency Matrix Generators

Model	PeMS (60 / 90 min)			
	MAE	MAPE (%)	RMSE	TT(s)
STGCN	4.24 / 5.21	10.59 / 13.45	7.94 / 9.54	-
Fix	**3.83** / 4.14	9.41 / 10.28	6.97 / 7.41	**672**
Com	3.90 / 4.20	9.34 / 10.19	6.88 / 7.40	864
TemD	3.87 / **4.06**	**9.16** / **10.06**	**6.81** / **7.28**	3946

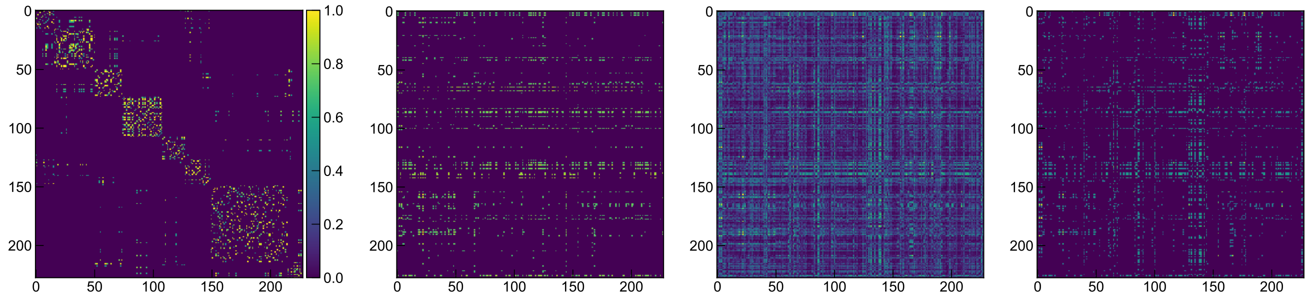
All base models are for single-step-ahead prediction. Our experiments are carried out on Ubuntu 18.04 LTS with tensorflow 1.18 and a Telsa V100 GPU.

3) Evaluation Metric: We choose MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), and RMSE (Root Mean Square Error) as key metrics to evaluate prediction accuracy.

B. Comparison on Different Adjacency Matrix Generators

This subsection details the experiment of three adjacency matrix generators detailed in section III-C, and then summarizes their advantages and disadvantages. It should be noted that all bias blocks except the adjacency matrix share the same parameters in this test. The well-trained STGCN [17] works as the base model and PeMS is utilized as the test dataset.

We compare the performance of 60 min and 90 min traffic predictions with different adjacency matrices, and results are shown in Table I. STGCN denotes the default STGCN model with iterative approach for long-term forecasting; "Fix" denotes using STGCN as the base model in the framework and the adjacency matrix in the bias block is computed using Eq. (16); "Com" denotes the adjacency matrix in the bias block is trained from training data; and "TemD" denotes that the spatiotemporal module is applied to generate temporally dynamic adjacency matrix. Besides MAE, MAPE, and RMSE, the training time (TT) for the bias block is also presented to evaluate the efficiency of different adjacency matrix generators. As we can see, STGCN with a bias block consistently outperforms the well-trained (base) STGCN for long-term prediction (over 60 min), regardless the adjacency matrix generator used. The three solutions show similar performances; however, we can still see that the temporally dynamic adjacency matrix outperforms the other two methods in most cases. Computing the adjacency matrix from vertices' distances is quick and easy, but this method only works when traffic network typology information is available. Temporally dynamic adjacency matrix provides an alternative solution when network information is not accessible, and this approach offers the best performance at the expense of computational costs. As can be seen from TT, the computational cost for "Com" (i.e., computing the adjacency matrix from data) is close to that of "Fix" (i.e., computing a fixed matrix from distance), while the cost of training the temporally dynamic matrix generator is almost 5 times the simple solution. Based on the above analysis, we choose to use the "Com" approach in the following experiments to balance the effectiveness and efficiency.



(a) Fix (b) Com (c) TemD-1 (d) TemD-2

Fig. 5. Visualization of different adjacency matrices.

Figure 5 visualizes different adjacency matrices. Compared with the fixed matrix computed based on the distance (see Figure 5(a)), the other two solutions can capture correlations between stations away from each other. The temporally dynamic matrix generator can generate different matrices based on different inputs (see Figure 5(c,d)).

C. Performance Comparison

Next, we empirically examine the effectiveness and generalizability of the proposed framework. Iterative multi-step forecasting and seq2seq versions are trained by the following respective loss functions

$$loss_{\text{iterative}} = \| f\left(\left\{\mathcal{G}_t, \dots \mathcal{G}_{t-(H-1)}\right\}\right) - \mathcal{G}_{t+1}\|_2, \quad (17)$$

$$loss_{\text{seq2seq}} = \| f\left(\left\{\mathcal{G}_t, \dots \mathcal{G}_{t-(H-1)}\right\}\right) - \left\{\mathcal{G}_{t+T}, \dots \mathcal{G}_{t+1}\right\} \|_2. \quad (18)$$

Note that all the baseline models share the same structures between the two versions. For instance, the only difference between the iterative STGCN and the seq2seq STGCN is the channel of their output layers. Additionally, the well-trained iterative multi-step forecasting models are used as the base model in the proposed framework.

Table II presents the detailed performance of tested models. We test three variants for each baseline model. Taking STGCN [17] as an example, “STGCN” means the default model that achieve multi-step forecasting in an iterative approach without introducing the bias block; “STGCN(S)” denotes the seq2seq version, which achieves long-term forecasting in a direct way without the bias block; finally, “STGCN+” follows the proposed universal framework, in which the base model is a well-trained single-step-ahead STGCN. Again, it should be noted that neural network parameters of the base model in “STGCN+” are the same as in “STGCN”, and there is no change in the base model used in the proposed framework. The same notation applies to VAR, DCRNN, GWNet, and ASTGNN. When comparing different models with simple HA baseline, we can clearly see that all models fall prey to error accumulation in long-term (multi-step-ahead) forecasting tasks. For example, most baseline models perform worse than HA for 90 min and 120 min ahead predictions. This is not surprising due to: (1) Traffic time series are often strongly non-stationary with clear periodic patterns at different scales (e.g., daily/weekly). A large body of literature shows that traffic time series demonstrate inherent low-rank patterns and can be modeled with a third-order sensor×time of day×day tensor structure (see e.g., [41], [42]). Therefore, we would expect the long-term (> 2 hours) structure of traffic time series to be dominated by the global patterns such as daily average. (2) Deep learning models (STGCN, DCRNN, GWNet, ASTGCN) are trained to better characterize the local/micro patterns and variations in the data, using only local information as input and output. This is why these models often provide the best short-term prediction accuracy but fail to compete with HA for long-term forecasting (i.e., > 1 hour). Therefore, we limit the long-term temporal range in our analysis to 2 hours. Given the strong spatiotemporal regularities in traffic time series, we would suggest to use explanatory models (such as historical average or other regression models) instead of local time series forecasting models for prediction beyond 2 hours.

We first compare the base models with their seq2seq “(S)” and universal “+” counterparts. We can see that the proposed framework shows comparable performance to the base model for 30-min-ahead prediction; for long-term (>30 min) prediction, the universal framework with bias block demonstrates substantial improvement, and works well even for 2-hour-ahead prediction. The results clearly show the superiority of using the bias block to reduce the accumulated errors in the base model. It is also interesting to see that the long-term forecasting performance of seq2seq is essentially worse than applying the default base model in an iterative manner. A possible reason is that even with the same input layer and hidden layers, the optimization space of the seq2seq based model is T (i.e., the output channel of the seq2seq-based model) times as large as that of the base model. As a result, seq2seq has a much more complex optimization space, which inevitably causes convergence problem and a much higher possibility of unfitting. This might be the key reason that many current state-of-the-art forecasting models are iterative multi-step models despite the error accumulation issue [17], [18], [23].

As previously discussed, the proposed framework can address this issue by adding a less complex bias bock as the residual connection [33] to the well-trained forecasting models, so as to correct the base model’s forecasting bias. Compared with directly training a large scale

TABLE II

COMPARISON RESULT IN PeMS AND META-LA TO PROVE THE EFFECTIVENESS AND GENERALIZATION OF THE PROPOSED FRAMEWORK. BEST RESULTS ARE HIGHLIGHTED IN BOLD

Model	PeMS (30 / 60 / 90 / 120 min)		
	MAE	MAPE (%)	RMSE
HA	4.60	12.50	8.05
VAR	4.84 / 6.06 / 6.71 / 7.29	11.33 / 14.10 / 15.44 / 16.74	7.04 / 8.63 / 9.38 / 10.08
VAR(S)	**4.43** / 6.13 / 7.01 / 7.54	11.08 / 14.74 / 15.90 / 17.98	**6.97** / 8.71 / 9.85 / 11.56
VAR+	**4.43 / 5.05 / 5.53 / 6.21**	**11.00 / 11.59 / 13.01 / 13.44**	7.02 / **7.75 / 8.94 / 9.71**
STGCN	**3.17** / 4.24 / 5.21 / 6.15	7.70 / 10.59 / 13.45 / 16.45	5.93 / 7.95 / 9.54 / 11.04
STGCN(S)	3.75 / 4.83 / 6.02 / 6.96	8.58 / 11.02 / 15.15 / 16.71	6.43 / 8.68 / 10.15 / 11.82
STGCN+	3.24 / **3.87 / 4.06 / 4.70**	**7.57 / 9.16 / 10.06 / 11.09**	**5.60 / 6.81 / 7.38 / 7.95**
DCRNN	3.35 / 4.41 / 5.62 / 6.33	8.03 / 10.80 / 13.78 / 17.00	6.11 / 8.18 / 9.92 / 11.59
DCRNN(S)	3.32 / 4.38 / 5.77 / 6.51	8.10 / 10.75 / 14.03 / 17.59	5.92 / 8.21 / 10.75 / 12.24
DCRNN+	**3.30 / 3.93 / 4.28 / 4.75**	**7.80 / 9.31 / 10.35 / 11.12**	**5.78 / 6.88 / 7.60 / 8.03**
GWNet	**3.20** / 3.77 / 4.90 / 5.54	**7.67** / 9.28 / 12.17 / 12.16	**5.85** / 6.86 / 8.60 / 9.75
GWNet(S)	3.75 / 4.08 / 4.97 / 5.83	8.03 / 9.65 / 13.10 / 15.35	6.17 / 7.00 / 8.84 / 10.66
GWNet+	3.22 / **3.58 / 4.22 / 4.54**	7.70 / **9.10 / 9.97 / 10.81**	5.90 / **6.75 / 7.30 / 7.85**
ASTGNN	**3.22** / 3.85 / 4.72 / 5.31	**7.78** / 9.35 / 11.06 / 11.85	**5.91** / 6.87 / 8.02 / 9.42
ASTGNN(S)	3.47 / 4.15 / 4.75 / 5.54	7.91 / 9.69 / 11.22 / 11.89	6.10 / 7.07 / 8.14 / 9.73
ASTGNN+	3.25 / **3.63 / 4.18 / 4.39**	7.88 / **9.11 / 9.54 / 10.15**	**5.91 / 6.80 / 7.16 / 7.64**

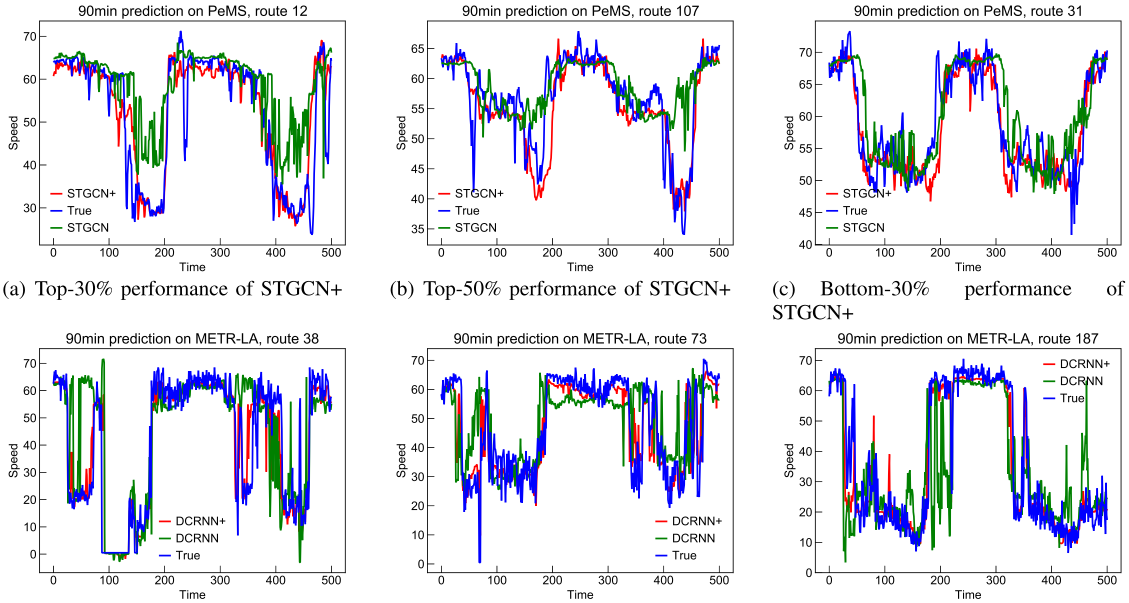
Model	METR-LA (30 / 60 / 90 / 120 min)		
	MAE	MAPE (%)	RMSE
HA	4.16	13.0	7.8
VAR	5.41 / 6.52 / 8.31 / 10.19	12.07 / 15.80 / 18.58 / 21.34	9.13 / 10.11 / 14.40 / 18.23
VAR(S)	5.41 / 6.52 / 8.31 / 10.19	12.07 / 15.80 / 18.58 / 21.34	9.13 / 10.11 / 14.40 / 18.23
VAR+	**5.22 / 5.93 / 6.74 / 7.95**	**11.38 / 14.47 / 16.66 / 18.05**	**8.81 / 9.80 / 12.07 / 15.56**
STGCN	**3.47** / 4.59 / 5.60 / 6.73	9.57 / 12.70 / 14.92 / 17.07	**7.24** / 9.40 / 11.84 / 13.98
STGCN(S)	3.58 / 4.82 / 5.93 / 7.05	9.95 / 12.94 / 15.22 / 17.50	7.65 / 9.91 / 12.33 / 14.45
STGCN+	3.55 / **4.50 / 5.12 / 5.31**	**9.86 / 12.55 / 13.01 / 15.00**	7.33 / **9.38 / 9.94 / 10.86**
DCRNN	3.15 / 3.60 / 4.17 / 5.20	8.80 / 10.50 / 13.11 / 15.95	6.45 / 7.60 / 8.65 / 9.72
DCRNN(S)	**3.11** / 3.58 / 4.23 / 5.44	**8.65** / 10.41 / 13.10 / 16.07	**6.22** / 7.56 / 8.81 / 10.12
DCRNN+	3.22 / **3.57 / 3.96 / 4.73**	8.82 / **10.14 / 12.05 / 13.01**	6.60 / **7.21 / 8.03 / 9.17**
GWNet	**3.07 / 3.53** / 4.10 / 4.75	**8.37** / 10.01 / 12.85 / 14.43	**6.22** / 7.37 / 8.54 / 9.01
GWNet(S)	3.25 / 3.78 / 4.66 / 5.02	8.66 / 10.59 / 13.03 / 14.94	6.63 / 7.85 / 8.85 / 9.71
GWNet+	3.18 / **3.53 / 3.95 / 4.38**	8.58 / **10.00 / 12.02 / 12.85**	6.50 / **7.31 / 7.94 / 8.62**
ASTGNN	**3.15** / 3.59 / 4.01 / 4.56	**8.40** / 10.14 / 12.43 / 13.97	**6.40** / 7.41 / 8.09 / 8.85
ASTGNN(S)	3.21 / 3.64 / 4.22 / 4.68	8.54 / 10.33 / 12.86 / 14.05	6.49 / 7.55 / 8.24 / 9.04
ASTGNN+	3.20 / **3.49 / 3.87 / 4.33**	8.55 / **10.05 / 11.90 / 12.67**	6.47 / **7.34 / 7.90 / 8.49**

seq2seq-based model (see GWNet(S) in Table II), we do not observe any convergence issues when training the bias block thanks to the desirable initialization from the well-trained base model. In addition, the results in Table II also suggest that the proposed framework works well with different base models—including both deep learning models and classical VAR—on different datasets, further demonstrating the superior generalizability of the proposed framework.

Panels (a), (b) and (c) in Figure 6 show examples of 90 min prediction on PeMS, which compare the result of STGCN with/without the proposed framework. Panels (d), (e) and (f) show another set of examples on 90 min prediction on META-LA with DCRNN as the base model. In the two sets of examples, the top 30%, the top 50%, and the last 30% performance of STGCN/DCRNN with the proposed framework are shown. The comparison shows that predictions of STGCN and DCRNN with the proposed framework are considerably more smooth and closer to the ground truth. In addition, even when the proposed framework does not work perfectly (e.g., see bottom-30% performance of Figure 6 (c) and (f)), it can still help the base model outperform the original one. Overall, the results strongly suggest the effectiveness of the proposed framework.

The proposed universal framework is designed to improve long-term forecasting accuracy. We next examine in detail the effect of the proposed framework on short-term forecasting performance (i.e., less than 30 min). We evaluate the MAPE improvements with STGCN as based model for different prediction horizons, from 5 min to 120 min and show the results in Figure 7. As can be seen, the proposed framework actually performs worse than the base model in short-term. The results are not surprising as the seq2seq-based bias block is trained for overall loss instead of the one-step-ahead loss used in the base model. STGCN+ outperforms the base model substantially starting from 60-min-ahead prediction. This clearly demonstrates the effectiveness of the proposed framework.

We next conduct experiments on the PeMS08 traffic occupancy data set [40]. Different from traffic speed data, traffic occupancy often has stronger local variations and lacks of consistent periodicity. Table III records comparison results of different deep learning-based traffic occupancy prediction models. Similar to the conclusion from experiments on traffic speed prediction, we can see that the proposed framework can improve the long-term prediction of different forecasting models. Due to the great variation and lack of consistent periodic patterns in the occupancy data, the HA baseline fails to provide



(a) Top-30% performance of STGCN+ (b) Top-50% performance of STGCN+ (c) Bottom-30% performance of STGCN+

(d) Top-30% performance of DCRNN+ (e) Top-50% performance of DCRNN+ (f) Bottom-30% performance of DCRNN+

Fig. 6. Long-term prediction results of different models for different datasets.

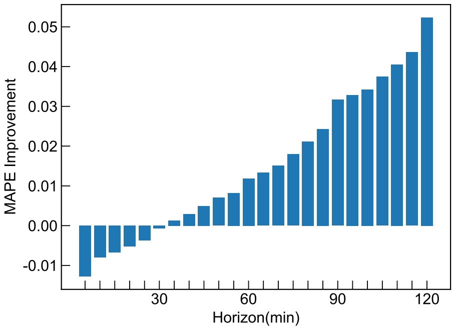


Fig. 7. MAPE improvement for STGCN+ compared with STGCN in PeMS for different prediction horizon, from 5 min to 120 min.

TABLE III
COMPARISON ON TRAFFIC OCCUPANCY PREDICTION IN PEMS08

Model	PeMS08 (60 / 90 / 120 min)	
	MAE	RMSE
HA	85.63	108.45
DCRNN	17.86 / 22.53 / 28.19	27.80 / 32.57 / 36.11
DCRNN(S)	18.13 / 22.68 / 28.35	28.14 / 33.07 / 36.55
DCRNN+	**17.08 / 19.87 / 23.81**	**25.83 / 28.70 / 32.71**
STGCN	18.02 / 23.61 / 29.47	27.83 / 33.92 / 38.58
STGCN(S)	18.90 / 24.77 / 29.59	27.96 / 33.73 / 38.03
STGCN+	**17.22 / 20.87 / 24.23**	**23.17 / 30.50 / 33.48**
ASTGNN	18.61 / 22.35 / 27.74	28.16 / 32.43 / 37.61
ASTGNN(S)	18.90 / 23.18 / 27.95	28.40 / 33.52 / 37.08
ASTGNN+	**17.14 / 18.81 / 20.93**	**26.76 / 30.04 / 32.24**

comparable performance, and it is even worse than copying the most recent observed values. This experiment on occupancy forecasting further confirms the long-term improvement by applying the proposed universal framework.

D. Framework Interpretability

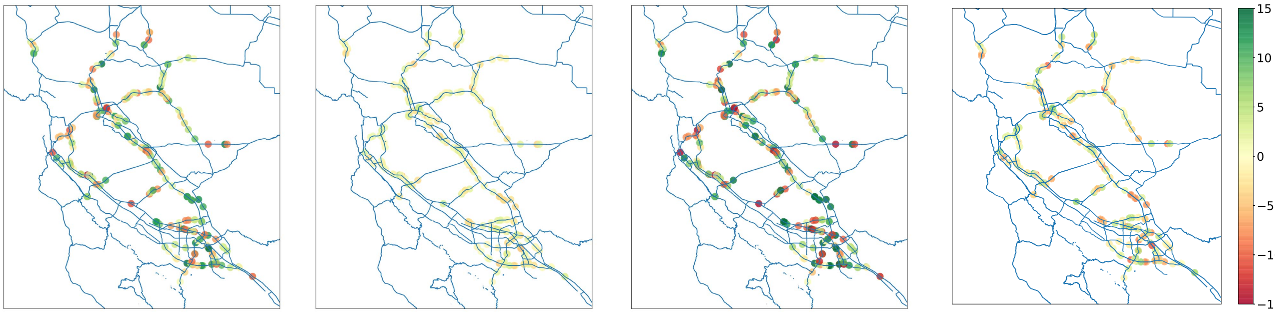
We next present experiments interpreting the proposed framework. In the following experiments, PeMS and STGCN are used as the test dataset and base model, respectively.

Figure 8 shows the output of the bias block for 60-min prediction. Here we show examples on Aug-24-2017 Thu and Aug-26-2017 Sat. For each day, two examples, one for morning peak hour (8:00 am) and the other at 8:00 pm, are shown. The results clearly show that the added bias (i.e., $\mathbb{B}$) varies in different periods. As we can see, at non-peak hours, most nodes in Figure 8(b,d) show an almost-zero bias, while the added biases vary substantially for different sensors during morning peak hours (see Figure 8(a,c)). Naturally, it is much more difficult for the base model to forecast complex traffic conditions during peak periods. However, the proposed bias block is able to mitigate the biases and improve the base model's forecasting under complex traffic conditions.

E. Verification on Different Bias Blocks

In Section III-B, we proposed a bias block composed of dilated convolutional layers and graph convolutional layers. In this section, three different designs are tested and compared with the proposed one. These bias block designs are:

- VAR, a Seq2Seq-based VAR model whose parameters, except the output channel, are as same as the base model in Section IV-C. Its output channel is set to be the same as the proposed bias block in Section III-B, which is equal to the prediction horizon;
- Dilated CNN, a bias block composed of temporal layers. All parameters, links, and structures are as same as the proposed one, except there is not spatial layer;



(a) Aug-24-2017 Thu, 8:00 am (b) Aug-24-2017 Thu, 8:00 pm (c) Aug-26-2017 Sat, 8:00 pm (d) Aug-26-2017 Sat, 8:00 am

Fig. 8. Bias block output in different timestamps.

TABLE IV

COMPARISON ON DIFFERENT BIAS BLOCKS. THE MEASUREMENT IS MAE

Base (Bias Block)	PeMS (30 / 60 / 90 / 120 min)
STGCN (None)	**3.17** / 4.24 / 5.21 / 6.15
STGCN (Proposed)	3.24 / **3.87** / **4.06** / **4.70**
STGCN (VAR)	3.20 / 4.11 / 4.64 / 5.05
STGCN (Dilated CNN)	3.22 / 3.95 / 4.32 / 4.95
STGCN (GNN)	3.24 / 4.14 / 4.77 / 5.11
DCRNN (None)	3.35 / 4.41 / 5.62 / 6.33
DCRNN (Proposed)	**3.30** / **3.93** / **4.28** / **4.75**
DCRNN (VAR)	3.31 / 4.32 / 4.59 / 5.10
DCRNN (Dilated CNN)	3.31 / 4.05 / 4.38 / 4.89
DCRNN (GNN)	3.35 / 4.49 / 4.87 / 5.24

- GNN, a bias block composed of spatial layers only. Similarly with Dilated CNN, all settings are as same as the proposed one, except no temporal layer.

In this experiment, STGCN and DCRNN work as the base model and PeMS dataset is used. The comparative result is shown as Table IV

From Table IV, it is evident that even with a simple design of the bias block, the proposed framework can still improve the long-term forecasting ability of the base model. As discussed in Section III-D, the proposed framework can be regarded as a wider forecasting model with a reasonable initialization. It can provide better representations of hidden spatiotemporal patterns with little extra computation penalty. This experiment proves the framework to be a simple yet extremely effective approach on improving current forecasting model in terms of long-term prediction performance. The design of bias block matters for the performance improvement. This study is intended to focus on highlighting the effectiveness of the proposed framework, how to design the optimal bias block will be studied in the future work.

V. Conclusion

In this paper, we present a universal framework that can enhance existing state-of-the-art short-term traffic models to achieve better long-term forecasting accuracy. The proposed framework consists of a base model and a spatiotemporal neural network-based bias block. Any existing one-step-ahead prediction model (not only deep learning-based models but also classic traffic forecasting models such as VAR) can serve as the base model, and the proposed bias block is comprised of three spatiotemporal modules with both temporal and spatial layers. It should be noted that one does not need to change anything in the base model; in other words, an existing well-trained model can be directly integrated into the proposed framework.

A major advantage of the proposed framework is it does not suffer from convergence problems that undermine seq2seq-based models. Combined with a well-trained base model, the bias block is extended to a large seq2seq-based model with little added complexity. Due to the extended variable space, the proposed framework performs much better in capturing hidden patterns than previous seq2seq-based models. The framework is much easier to train because of its good initialization from the well-trained base model as well as its low complexity, which effectively addresses the bottleneck of seq2seq-based models. In addition, the proposed framework is also more advantageous than iterative multi-step models in that the bias block resolves error accumulation problems by adding biases to offset the errors. Moreover, the base model guarantees a reasonably accurate prediction, and the bias block, which cooperates with the base model by the residual connection [33], can decrease the base model's error and will not lead to the accuracy drop. As a result, the proposed framework can overcome both seq2seq-based models' and iterative multi-step models' problems and thus perform better.

We acknowledge that this framework does not substantially improve short-term prediction, which might be caused by overfitting. Short-term (less than 30 min) prediction is relatively simple and does not need complex structures. The proposed bias block increases the complexity, which is one possible reason why the short-term prediction performance is not increased like the long-term prediction. To achieve the best forecasting performance, a sensible solution is to adopt the default base model for short-term forecasting ($<$ 30 min) and switch to the proposed framework for long-term forecasting (30−120 min).

References

[1] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 3–19, Jun. 2014.

[2] X. Li *et al.*, "Prediction of urban human mobility using large-scale taxi traces and its applications," *Frontiers Comput. Sci.*, vol. 6, pp. 111–121, Feb. 2012.

[3] S. Shekhar and B. M. Williams, "Adaptive seasonal time series models for forecasting short-term traffic flow," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2024, no. 1, pp. 116–125, Jan. 2007.

[4] 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.

[5] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 62, pp. 21–34, Jan. 2016.

[6] L. Li, X. Su, Y. Zhang, Y. Lin, and Z. Li, "Trend modeling for traffic time series analysis: An integrated study," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3430–3439, Dec. 2015.

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

[8] 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. 2020.

[9] Z. Qu, H. Li, Z. Li, and T. Zhong, "Short-term traffic flow forecasting method with M-B-LSTM hybrid network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 1–11, Jan. 2020.

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

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

[12] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transp. Rev.*, vol. 24, no. 5, pp. 533–557, Sep. 2004.

[13] F. Ziel and K. Berk, "Multivariate forecasting evaluation: On sensitive and strictly proper scoring rules," 2019, *arXiv:1910.07325*.

[14] Z. Xiao, X. Fu, L. Zhang, and R. S. M. Goh, "Traffic pattern mining and forecasting technologies in maritime traffic service networks: A comprehensive survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1796–1825, May 2020.

[15] 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.

[16] X. Shi and D.-Y. Yeung, "Machine learning for spatiotemporal sequence forecasting: A survey," 2018, *arXiv:1808.06865*.

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

[18] Y. Huang, Y. Weng, S. Yu, and X. Chen, "Diffusion convolutional recurrent neural network with rank influence learning for traffic forecasting," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2019, pp. 678–685.

[19] F. Diehl, T. Brunner, M. T. Le, and A. Knoll, "Graph neural networks for modelling traffic participant interaction," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 695–701.

[20] B. Liao *et al.*, "Deep sequence learning with auxiliary information for traffic prediction," in *Proc. SIGKDD*, 2018, pp. 611–618.

[21] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, and X. Zhou, "LC-RNN: A deep learning model for traffic speed prediction," in *Proc. IJCAI*, 2018, pp. 3470–3476.

[22] J. Wang, R. Chen, and Z. He, "Traffic speed prediction for urban transportation network: A path based deep learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 100, pp. 372–385, Feb. 2019.

[23] 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.*, 2019, pp. 890–897.

[24] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan, "Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3913–3926, Oct. 2019.

[25] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 6, pp. 1–10, Apr. 1998.

[26] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proc. IJCAI*, 2019, pp. 550–558.

[27] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Trans. Knowl. Data Eng.*, early access, Feb. 3, 2021, doi: 10.1109/TKDE.2021.3056502.

[28] 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, May 2013.

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

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

[31] S. Zhang, Y. Guo, P. Zhao, C. Zheng, and X. Chen, "A graph-based temporal attention framework for multi-sensor traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, early access, Apr. 16, 2021, doi: 10.1109/TITS.2021.3072118.

[32] 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.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 1–12.

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

[35] S. Karen and Z. Andrew, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.

[36] F. Liu, C. Deng, F. Bi, and Y. Yang, "Dual teaching: A practical semi-supervised wrapper method," 2016, *arXiv:1611.03981*.

[37] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

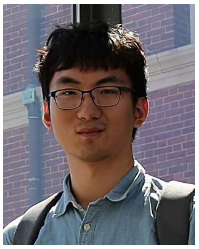
[38] F. Wang *et al.*, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3156–3164.

[39] B. Perozzi, R. Al-Rfou, and S. Skiena, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *Proc. SIGKDD*, 2014, pp. 701–710.

[40] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proc. 35th AAAI Conf. Intell.*, 2021, pp. 4189–4196.

[41] X. Chen, Z. He, and L. Sun, "A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation," *Transp. Res. C, Emerg. Technol.*, vol. 98, pp. 73–84, Jan. 2019.

[42] X. Chen, M. Lei, N. Saunier, and L. Sun, "Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation," 2021, *arXiv:2104.14936*.



Fuqiang Liu (Student Member, IEEE) is currently pursuing the Ph.D. degree with the Department of Civil Engineering, McGill University, Montreal, Canada. His research interests include spatio-temporal data analysis, adversarial and defense studies of deep learning, the robustness of intelligent transportation systems, and efficient design of deep neural networks.

Jiawei Wang (Student Member, IEEE) is currently pursuing the Ph.D. degree with the Department of Civil Engineering, McGill University, Montreal, Canada. His research interest focuses on traffic control with machine learning techniques.

Jingbo Tian is currently pursuing the Ph.D. degree with the Department of Civil Engineering, McGill University, Montreal, Canada. His research interest centers around fusing advanced data-driven models with transportation domain knowledge in various transportation problems.

Dingyi Zhuang received the bachelor's degree in mechanical engineering from Shanghai Jiao Tong University and the master's degree in civil engineering from McGill University. His research interests include urban computing, graph neural networks, spatio-temporal data mining, and Bayesian probabilistic factorization models.

Luis Miranda-Moreno received the Ph.D. degree from the University of Waterloo, ON, Canada. He is currently an Associate Professor with the Department of Civil Engineering, McGill University. His research interests include the development of crash-risk analysis methods, the integration of emergency technologies for traffic monitoring, the impact of climate on transportation systems, the analysis of short- and long-term changes in travel demand, the impact of transport on the environment, the evaluation of energy efficiency measures, and non-motorized transportation.

Lijun Sun (Member, IEEE) received the B.S. degree in civil engineering from Tsinghua University, Beijing, China, in 2011, and the Ph.D. degree in civil engineering (transportation) from the National University of Singapore in 2015. He is currently an Assistant Professor with the Department of Civil Engineering, McGill University, Montreal, QC, Canada. His researches center on intelligent transportation systems, machine learning, spatio-temporal modeling, travel behavior, and agent-based simulation.