IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 October 2023, accepted 23 November 2023, date of publication 30 November 2023,
date of current version 8 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3338223

RESEARCH ARTICLE

Performance Evaluation of Building Blocks of Spatial-Temporal Deep Learning Models for Traffic Forecasting

YUYOL SHIN AND YOONJIN YOON, (Member, IEEE)
Department of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea
Corresponding author: Yoonjin Yoon (yoonjin@kaist.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Basic Research Laboratory under Grant 2021R1A4A1033486, and in part by the Midcareer Research Grant by the South Korean Government under Grant 2020R1A2C2010200.

ABSTRACT The traffic forecasting problem is a challenging task that requires spatial-temporal modeling and gathers research interests from various domains. In recent years, spatial-temporal deep learning models have improved the accuracy and scale of traffic forecasting. While hundreds of models have been suggested, they share similar modules, or building blocks, which can be categorized into three temporal feature extraction methods of recurrent neural networks, convolution, and self-attention and two spatial feature extraction methods of convolutional graph neural networks (GNN) and attentional GNN. More importantly, the models have been mostly evaluated for their entire architectures with limited efforts to characterize and understand the performance of each category of building blocks. In this study, we conduct an extensive, multi-faceted experiment to understand the influence of building block selection on traffic forecasting accuracy, considering environmental characteristics and dataset distributions. Specifically, we implement six traffic forecasting models using three temporal and two spatial building blocks. When we evaluate the models on four datasets with diverse characteristics, the results show each building block demonstrates distinguishable characteristics depending on study sites, prediction horizons, and traffic categories. The convolution models demonstrate higher overall forecasting performance than other models, whereas self-attention models show competitiveness in less frequent traffic categories, transition states, and the presence of outliers. Based on the results, we also suggest an adaptive model evaluation framework for category-wise predictions of test sets based on the performance of the models on validation sets. The results of this evaluation framework demonstrate improved forecasting accuracy at most by 3.7% without further sophistication in existing model architectures. The results enhance the utility of existing models and suggest guidelines for researchers building traffic forecasting model architectures and for practitioners implementing these state-of-the-art techniques in real-world applications.

INDEX TERMS Comparative study, deep learning, graph neural networks, spatial-temporal representation, time-series prediction, traffic forecasting.

I. INTRODUCTION

Traffic forecasting is a complex problem that requires modeling spatial-temporal features of traffic data such as speed, density, and flow, to accurately predict future traffic states. As stated in [1], traffic forecasting aims to make predictions on from few seconds to possibly a few hours of future traffic states based on current and past traffic information. The accurate prediction of future traffic states is a crucial technical capability in intelligent transportation systems (ITS) [1], [2], [3], enabling applications such as network capacity evaluation [4], travel time estimation [5], signal optimization [6], and carbon emission reduction [7]. It is a long-studied problem which dates back to 1930s with efforts from various domains of science and engineering [8]. Owing to advancements in sensor technologies such as GPS

The associate editor coordinating the review of this manuscript and approving it for publication was Vlad Diaconita.

© 2023 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

and loop detectors and deep learning techniques to learn from abundance of data, traffic forecasting problems garnered much research interests in recent years.

In traditional approaches to the traffic forecasting problem, conventional time series models such as autoregressive integrated moving average (ARIMA) [9] and vector autoregressive (VAR) [10] have gained popularity. Other data-driven machine learning algorithms such as support vector regression (SVR) [11] and k-nearest neighbor (kNN) [12] have also been utilized. Some other studies implemented simulation [13], [14] and physical modeling [8]. Although these approaches all demonstrated promising results, their applications have had limitations in accuracy, spatial-temporal range, or computation time.

The recent surge of deep learning algorithms offered methods to fit a wide variety of functions with a larger number of parameters while avoiding overfitting problems, and researchers have been able to leverage these advanced techniques to capture the complex spatial and temporal features of transportation networks in traffic forecasting problems [3]. Recurrent neural networks (RNN) have gained popularity in capturing temporal features with their intrinsic ability to handle sequential data [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38]. The RNN methods, however, have suffered from the vanishing gradient problem, and the convolution-based temporal feature extraction has been suggested to overcome this intrinsic problem of RNN [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60]. More recently, the self-attention mechanism [61] has demonstrated meaningful advances in traffic forecasting [22], [40], [41], [43], [46], [51], [58], [60], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72].

Convolutional neural networks (CNN) and graph neural networks (GNN) provide key spatial feature extraction capabilities. Although they constitute the pioneering efforts to adopt deep learning architectures to traffic forecasting [20], [23], [25], [42], [48], [49], [50], [54], CNN models have limitations in modeling the complex topology of the underlying transportation networks. In contrast, GNN takes advantage of the node-link structure to incorporate the underlying transportation network topology. By modeling traffic sensors and road segments as graph nodes, the hidden representation of a target node is learned by aggregating information from the neighboring nodes connected by edges [15], [16], [17], [21], [22], [24], [26], [27], [28], [29], [31], [32], [34], [35], [36], [37], [38], [40], [41], [43], [44], [45], [46], [47], [51], [52], [53], [55], [57], [58], [59], [60], [63], [64], [66], [67], [68], [71], [72], [73], [74], [75].

Despite the success of deep learning models in processing large datasets with high accuracy, efforts to understand each of theses components, or building blocks, of the models are limited. For spatial feature extraction, Li et al. [16] proposed diffusion convolution, a convolutional GNN layer that modeled traffic flow as a diffusion process on a graph and compared its performance with ChebNet [76] on a traffic flow dataset. Cui et al. [15] suggested traffic graph convolution (TGC) and compared it with spectral CNN [77] and ChebNet [76] in terms of the number of parameters, computation time, and ability to extract localized features. In addition, they showed that the TGC model outperformed the spectral GCN-based models in overall performance. Although these studies provide comparative studies between new and existing GNN layers, they only discuss the performance in terms of overall accuracy and efficiency. In the temporal dimension, Reza et al. [70] present the overall performance comparison between SVR, LSTM, GRU, and transformer without consideration of spatial features. Therefore, an investigation beyond overall performance to characterize each building block is necessary to understand and justify traffic forecasting model architecture.

This study addresses this gap by conducting an extensive and multi-faceted experiment to characterize the building blocks of spatial-temporal deep learning models for traffic forecasting. First, we define the five categories of the building blocks through an extensive literature review. They are RNN, convolution, and self-attention for temporal feature extractions, and convolutional GNN and attentional GNN for spatial feature extractions. Subsequently, we implement six traffic forecasting models, each incorporating distinct combination of three temporal and two spatial building blocks. To construct the models, we draw three models from previous literature, each representing a temporal building block. Through replacement of spatial building blocks in selected models with GCN [78] and GAT [79], we assemble six traffic forecasting models for the experiment. Finally, we evaluate the performance of the models on four real-world datasets with diverse characteristics. In the experiment, we assess the influence of building block selections, and analyze the performance across different traffic categories and presence of outliers.

As the results, we find that the convolution and self-attention-based models demonstrate advantages over the RNN-based counterparts in extracting temporal features for traffic forecasting. In the overall performance evaluation, the convolution models tend to outperform the self-attention models in overall performance. However, the self-attention models show a smaller performance discrepancy in performances between 15-min and 60-min predictions, indicating a potential advantage in long-term forecasting. In addition, the self-attention provides more accurate results in low-frequency traffic categories, and shows higher robustness against outliers than the convolution models. Furthermore, we suggest the adaptive model evaluation framework that flexibly selects models to conduct prediction based on the category-wise performance evaluation. Using this framework, traffic predictions with higher accuracy can be achieved without further sophistication in model architectures. In summary, our main contribution is fourfold:

- Categorize the building blocks for spatial-temporal deep neural networks for traffic forecasting through an extensive literature review. The categorization includes three temporal feature extraction methods - RNN, convolution, and self-attention - and two spatial feature extraction methods - convolutional GNN, and attentional GNN.
- Conduct an extensive, multi-faceted experiment using six traffic forecasting models each representing a distinct combination of three temporal and two spatial building blocks on four different datasets. The overall performance evaluation discovers building block pairs that generally yield higher accuracy: convolutional GNN & convolution and attentional GNN & self-attention.
- Discover the characteristics of each building block. The convolution-based temporal feature extraction maximizes the performance gain in frequent traffic categories, whereas the self-attention and attentional GNN have increased robustness in infrequent conditions, such as low-frequency traffic categories, traffic transitions, and the presence of outliers.
- Propose an adaptive evaluation framework for traffic forecasting, which makes predictions using multiple models based on the performance on distinct traffic categories. The framework increases the previous state-of-the-art performance by 3.7% in a highway traffic speed prediction task, without further sophistication in previous model architectures.

The remaining paper is organized as follows. Section II investigates the literature on deep learning models in traffic forecasting studies. The preliminaries for this study and definitions are in Section III. The methods and data are explained in Section IV, along with the experimental setting. In Section V, we present the results and discussion of the experiment. Finally, Section VI provides the conclusion and future study.

II. LITERATURE REVIEW

Deep learning models have proven effective in various research fields such as image classification [80], object recognition [81], and machine translation [82]. With their ability to process huge data and model non-linear relationships, deep learning has also become cutting-edge in traffic forecasting studies. Following earlier works on stacked autoencoders [83] and deep belief networks [84], many studies suggested deep learning models that capture the spatial-temporal correlation of traffic data.

A. TEMPORAL FEATURE EXTRACTION

To model time-series traffic data, *recurrent neural networks* (RNN) and their variants, such as long-short term memory (LSTM) [85] and gated recurrent unit (GRU) [86] have gained attention in extracting temporal features for traffic forecasting models. Implementation of vanilla LSTM has shown improved performance compared to traditional models such as auto-regressive integrated moving average (ARIMA), support vector machine (SVM), and Kalman filtering [18], [30]. When traffic data were categorized into congestion levels, the LSTM model combined with the restricted Boltzmann machine (RBM) showed at most 93.8% accuracy for congestion prediction tasks [19]. The sequence-to-sequence framework has been adopted in many models for multiple prediction horizons [16], [17], [21], [27], [31], [32], [36]. In Bai et al. [33], a linear transformation layer was implemented to conduct multi-step traffic prediction. Wang et al. [34] suggested a model that utilized GRU to produce aggregated spatial-temporal representations. Several models have employed multiple layers of RNN [16], [21], [33], [35], whereas others have used the attention mechanism [27], [28], [36], [37] to capture the long-term relationship in traffic data.

Another building block to extract temporal features is *convolution*. In the absence of sequential computation, convolution have been able to efficiently train the models and overcome the vanishing gradient problem of RNNs. Originally suggested to process image data, earlier CNN approaches have processed traffic data into an image with each row and column representing each node of the transportation network and time step, respectively [49], [54]. Although these models have demonstrated higher forecasting power than traditional machine learning algorithms and vanilla LSTM, the CNN structure is limited as it represents only 1D spatial complexity. To model time series more appropriately, temporal convolutions such as the gated 1D causal convolution [45], [47], [48], [56], [59], [60] applied convolution operation only along the temporal dimension. By limiting the usage of future information during the temporal feature extraction stage, causal convolutions have become applicable to traffic time-series modelling problems. The dilated causal convolution [87] that applies dilation to 1D causal convolution to increase the reception field size with a limited number of layers has shown improved performance [40], [41], [43], [44], [46], [51], [53], [55], [57], [58], [73].

Recently, *self-attention* has also been widely adopted in traffic forecasting studies. Reza et al. [70] demonstrated the advantage of the transformer architecture over RNN models. To impose sequential information of traffic data, self-attention have been implemented with various positional encoding methods. While the original Transformer [61] implemented the sinusoid to encode the position information of word sequences, Cai et al. [63] and Wen et al. [69] implemented the transformer architecture with variations in the embedding of traffic data and positional encoding. Guo et al. [64] modified the self-attention score to reflect trends in traffic data and implemented a dynamic graph convolution module to replace the position-wise feed-forward layer of the transformer. TrafficBERT [65] used the transformer encoder as in Devlin et al. [88] to retain the forecasting power when training using data from multiple sources. Wang et al. [72] proposed an approach in which the parameters for the self-attention layer is generated using regional distribution of Point-of-Interests (PoI). Self-attention in conjunction with

other temporal feature extraction methods such as GRU [22], [34], [62] and dilated causal convolution [40], [46], [51], [58] have also been proposed. GMAN [68], and AI-GFACN [71] adopted self-attention for both spatial and temporal feature extractions. In addition, Zheng et al. [68] also introduced a transform attention layer that generated spatial-temporal embedding representations for the positional embedding of future time steps. Xu et al. [67] proposed a model in which a temporal attention block followed a spatial attention block. The two attention blocks of the model shared similar structures, except that the graph convolution operation was skip-connected to the output of the spatial attention block to reflect the static structure of the transportation network.

B. SPATIAL FEATURE EXTRACTION WITH GRAPH NEURAL NETWORKS

Earlier efforts have adopted CNN to extract spatial features of traffic data. However, they operate in Euclidean space and fail to represent the complex topology of transportation networks [20], [23], [25], [26], [39], [42], [48], [49], [50], [54], [89], [90], [91].

GNNs have become a popular choice in traffic forecasting since the early adoptions by Li et al. [16] and Yu et al. [56]. The core idea of GNN is to process the data into graph structures and extract the spatial feature of each node by aggregating the information from neighboring nodes. Most GNN methods for supervised learning, such as classification and regression, can be grouped into *convolutional, attentional,* and *message-passing GNNs* based on how they aggregate neighborhood information [92].

Convolutional GNNs multiply fixed weights to the source node features and conduct aggregating operations, such as summation, pooling, and averaging, to extract target node spatial features. The most widely used methods under convolutional GNN are the group of spectral graph convolutions [76], [77], [78], which approximates the filters in the spectral domain. GraphSAGE [93] and diffusion convolution [16] are other examples of convolutional GNNs. *Attentional GNNs* resemble convolutional GNNs in that they multiply the source node features with scalar weights. The difference, however, lies in that the attentional GNNs assign the weights through a function of the source and target node features. Graph attention networks (GAT) [79] and Gated Attention Networks (GaAN) [26] are popular attentional GNN models that implement self-attention mechanisms [61]. Finally, *message-passing GNNs* compute output representations of a target node using a function of the target node and its neighbors. Gilmer et al. [94] is an example of message-passing GNN, which computes the message using hidden representations of source and target nodes and edges. The aggregated messages and the target node features are passed through a neural network to generate output representations. For more explanations on GNNs taxonomy, see Bronstein et al. [92].

As transportation networks are inherently equipped with graph structures, the GNNs have become the most popular spatial feature extraction method for traffic forecasting. Convolutional GNNs have pioneered GNN-based traffic forecasting research, and have been widely used in concurrent models [15], [16], [17], [22], [28], [29], [31], [32], [33], [35], [37], [40], [43], [44], [47], [55], [56], [57], [58], [59], [60], [62], [63], [64], [67], [71], [75], [95]. Several studies [29], [37], [56], [60] adopted spectral graph convolutions [76], [78] that showed higher forecasting power over the basic deep learning models such as feed-forward neural networks and FC-LSTM. Li et al. [16] suggested diffusion convolution, which expanded the application of graph convolution to directed graphs, and has been applied in many traffic forecasting studies [44], [55], [57], [63]. Cui et al. [15] suggested traffic graph convolution (TGC), using element-wise multiplication between learnable parameters and adjacency matrices. Zhang et al. [28] implemented traffic graph convolution with an attention mechanism [96] to capture the dependencies in the time steps regardless of distances. Using a matrix factorization technique, Bai et al. [33] suggested a convolutional GNN module that can apply node specific parameters. Attentional GNNs also have been widely used in traffic forecasting research [21], [26], [27], [36], [41], [45], [52], [66], [73]. The gated attention networks (GaAN) [26] outperformed diffusion convolution in short-term traffic forecasting when combined with GRU. GAT [79] has also been adopted in many studies [21], [27], [36], [41], [52], [73]. Park et al. [66] constructed a new attentional GNN layer that adopts the scaled dot-product attention [61] with sentinel vectors to control the information from neighbor nodes. A few studies have implemented convolutional and attentional GNNs in one model [46], [51], [72]. Message-passing GNN traffic forecasting models have also been suggested using a dual graph that predicts node and edge features [74], and using bidirectional graphs in extracting aggregated spatial-temporal features [34]. Gupta et al. [38] proposed a message-passing GNN-based model with a spatial embedding and attention mechanism based on shortest-paths on graphs. Outside the existing taxonomy of GNNs, graph embedding techniques such as DeepWalk [97], LINE [98], and node2vec [99] have also been adopted to incorporate graph structures [24], [66], [68], [71], [89].

While these studies have achieved significant performance improvements, there have not been sufficient efforts to understand the performance of individual building blocks that constitute these models. Li et al. [16] introduced diffusion convolution as a convolutional GNN layer, employing it to conceptualize traffic flow as a diffusion process occurring on a graph. This approach was then compared with the more traditional ChebNet [76] for their performance. Similarly, Cui et al. [15] conduct a comparative analysis between the proposed traffic graph convolution (TGC) and traditional convolutional GNNs such as spectral GNN [77] and ChebNet [76] for their number of parameters, computational efficiency, feature localization ability, and overall performance. For the temporal feature extraction blocks, Reza et al. [70] evaluates the performances of the transformer compared to other machine learning algorithms

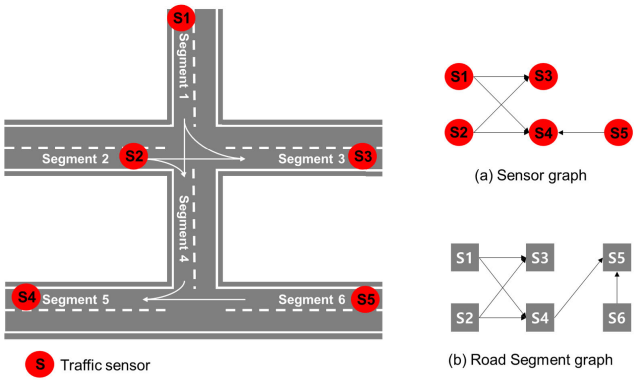

FIGURE 1. Graph construction from a transportation network. The transportation network on the left consists of 6 road segments and 5 traffic sensors. The network can be represented as (a) a sensor graph, or (b) a road segment graph considering the locations and traffic directions.

TABLE 1. Summary of the literature review by spatial and temporal building blocks.

		Temporal			
		RNN	Convolution	Attention	Misc.
Spatial	Conv. GNN	[15]–[17], [22], [28], [29], [31], [32], [33], [35], [37], [62]	[40], [43], [44], [46], [47], [51], [55]–[60], [64]	[22], [40], [43], [46], [51], [58], [62]–[64], [67], [71], [72]	[75], [95]
	Att. GNN	[21], [26], [27] [36]	[41], [45], [46] [51], [52], [73]	[27], [41], [46], [51], [66]	
	Misc.	[24], [34], [38]	[53]	[34], [68], [71] [72]	[74]

architecture with the absence of spatial feature extraction. Although these studies evaluate the performance of traffic forecasting models on overall performance and computation efficiency, a comprehensive building block-wise analysis needs to be conducted considering characteristics of datasets, traffic categories, and robustness. In this study, we address this research gap through an extensive and multi-faceted experiment to reveal the inherent characteristics of the building blocks.

In Table 1, spatial-temporal traffic forecasting models are categorized by the implemented building blocks. Although several studies fall under the miscellaneous category, most studies can be categorized using the five building blocks of spatial and temporal feature extraction. Note that several studies [46], [51], [62], [71], [72] use more than two building blocks to extract the features. For more in-depth reviews of traffic forecasting studies using deep learning models, please refer to Lee et al. [3], Ye et al. [100], and Jiang et al. [101].

III. DEFINITIONS AND PROBLEM STATEMENT

This section explains the preliminaries of our study, which include the mathematical definition of the transportation network graph, graph signal, and traffic forecasting problem.

A. NOTATIONS AND DEFINITIONS

Definition 1: Transportation network graph We represent the transportation network graph as a directed graph $\mathcal{G} = (V, E)$, where V is a set of $|V| = N$ nodes and E is a set of edges representing pairwise connections between the nodes. As defined in Ye et al. [100], a node can represent a sensor, road segment, or road intersection. In this study, we used sensor and road segment graphs depending on the dataset. The hypothetical construction of each type of graph is in Fig. 1. An adjacency matrix $\boldsymbol{A} = (A_{ij}) \in \mathbb{R}^{N\times N}$ is a square Boolean matrix, where the nodes $v_i, v_j \in V$ are connected by an edge $(v_i, v_j) \in E$.

Definition 2: Graph Signal The signal from node v_i at time t is denoted as $\boldsymbol{x}_t^i \in \mathbb{R}^C$, where C is the number of features of the signal. The graph signal is a matrix containing all node signals at time t, denoted as $\boldsymbol{X}_t = [\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, \ldots, \boldsymbol{x}_t^N] \in \mathbb{R}^{N\times C}$.

B. TRAFFIC FORECASTING PROBLEM

The traffic forecasting problem defined on the transportation network graph $\mathcal{G}$ predicts future traffic states for T' time steps based on historical traffic information such as speed, flow, and occupancy. Given historical graph signals for past T time steps on the graph, $\mathcal{G}$, the traffic forecasting problem is defined as finding a function H that maps the historical data to future traffic states:

$$H : [\boldsymbol{X}_{t-T+1}, \ldots, \boldsymbol{X}_t; \mathcal{G}] \rightarrow \left[\hat{\boldsymbol{Y}}_{t+1}, \ldots, \hat{\boldsymbol{Y}}_{t+T'}\right] \quad (1)$$

where $\hat{\boldsymbol{Y}}_t \in \mathbb{R}^{N\times 1}$ is the predicted traffic state at time t.

IV. METHODS AND MATERIALS

This section explains the methods and materials used in this study. First, we explain the GNN-based spatial building blocks, graph convolutional networks (GCN) [78] and graph attention networks (GAT) [79], and three base models with different temporal building blocks. Then, we introduce the datasets and settings for the experiments. The study outline is shown in Fig. 2.

A. SPATIAL FEATURE EXTRACTION WITH GRAPH NEURAL NETWORKS

To investigate the differences between convolutional and attentional GNNs in traffic forecasting research, we implemented one module from each category. Specifically, we implemented the GCN model [78] from the convolutional GNNs category. The GCN model uses the first-order Chebyshev polynomials to approximate the filter in the Fourier-transformed space and incorporates spatial relationships between nodes by aggregating information from neighboring nodes. A GCN layer with input $X_t \in \mathbb{R}^{N\times d}$ on graph $\mathcal{G}$ and d-dimensional feature space at time t can be expressed as follows:

$$\mathrm{GCN}(\boldsymbol{X_t}, \boldsymbol{A}) = \sigma(\hat{\boldsymbol{A}}\boldsymbol{X_t}\boldsymbol{W}), \quad (2)$$

where $\sigma(\cdot)$ is an activation function, and $\boldsymbol{W} \in \mathbb{R}^{d\times h}$ is the weight parameter matrix where h is the output dimension. Whereas GCN originally used the normalized Laplacian matrix $\hat{\boldsymbol{A}} = \tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2}$ where $\tilde{\boldsymbol{A}} = \boldsymbol{I}_N + \boldsymbol{A}$, and $\tilde{\boldsymbol{D}}_{ii} = \Sigma_j \tilde{a}_{ij}$, we use $\hat{\boldsymbol{A}} = \tilde{\boldsymbol{D}}^{-1}\tilde{\boldsymbol{A}}$ to apply GCN on directed graphs.

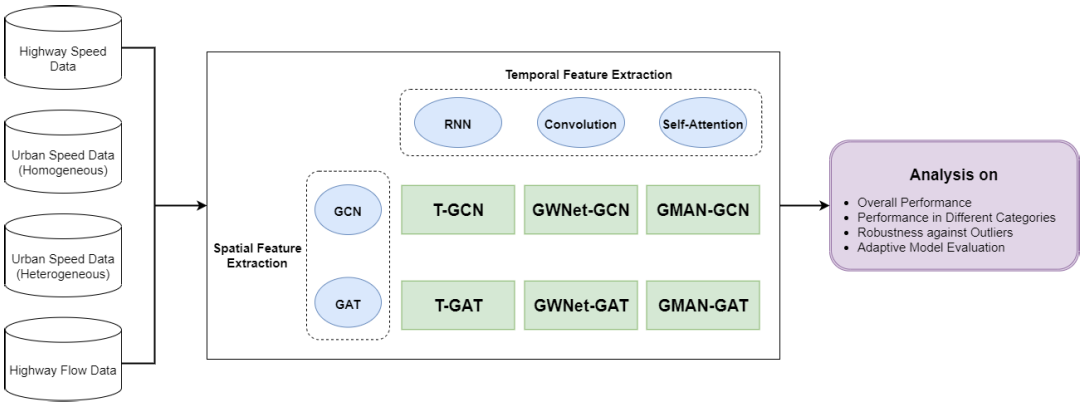

FIGURE 2. Overview of this study. We first define three categories for temporal and two for spatial building blocks. Combining a spatial and a temporal building block, we implement six models and conduct an extensive and multi-faceted experiment using four different real-world traffic datasets. Finally, we analyze the results on overall performance, performance in different traffic categories, performance on outliers, and adaptive model evaluation.

Information from further nodes can be aggregated by staking multiple GCN layers.

From the family of attentional GNNs, we implemented **GAT** [79]. GAT uses self-attention mechanisms [61] to weigh the importance of each neighbor node and aggregates the information from neighbors accordingly. For each node pair v_i and v_j connected by edge (v_i, v_j), the attention score for the k-th head $\alpha_{ij}^{t(k)}$ at time t is defined as follows:

$$\alpha_{ij}^{t(k)} = \frac{\exp\left(\sigma\left(\left\langle \boldsymbol{a}^{(k)}, \left[\boldsymbol{x}_t^i \boldsymbol{W}^{(k)}, \boldsymbol{x}_t^j \boldsymbol{W}^{(k)}\right]\right\rangle\right)\right)}{\sum_{v_l \in \mathcal{N}_i} \exp\left(\sigma\left(\left\langle \boldsymbol{a}^{(k)}, \left[\boldsymbol{x}_t^i \boldsymbol{W}^{(k)}, \boldsymbol{x}_t^l \boldsymbol{W}^{(k)}\right]\right\rangle\right)\right)}, \quad (3)$$

where $x_t^i \in \mathbb{R}^d$ is the signal of node v_i at time t, $\boldsymbol{a}^{(k)} \in \mathbb{R}^{2h'}$ is a learnable parameter vector for the k-th attention head with h' dimension, $\boldsymbol{W}^{(k)} \in \mathbb{R}^{d \times h'}$ is a learnable weight parameter matrix for the k-th attention head, $\langle \cdot, \cdot \rangle$ is the dot product operator, $[\cdot, \cdot]$ concatenates the vectors inside the bracket and $\mathcal{N}_i$ is the neighbor set of node v_i. The GAT layer with K heads applied on the node v_i with graph signal $\boldsymbol{X}_t$ observed from graph $\mathcal{G}$ at time t can be expressed as follows:

$$\mathrm{GAT}(v_i; \boldsymbol{X}_t, \mathcal{G}) = \mathrm{CAT}_{k=1}^{K}\left[\sigma\left(\sum_{v_l \in \mathcal{N}_i} a_{ij}^{t(k)} \boldsymbol{x}_t^l \boldsymbol{W}_v^{(k)}\right)\right], \quad (4)$$

where $\mathrm{CAT}_{k=1}^{K}[\cdot]$ concatenates the outputs of the equation in the bracket for $k = 1$ to K, $\boldsymbol{W}_v^{(k)} \in \mathbb{R}^{d \times h'}$ is a learnable parameter matrix for the k-th attention head. If the output dimension for GAT $h' \times k$ is equal to that of GCN, replacing one with the other becomes possible for any traffic forecasting model.

B. BASE MODELS WITH DIFFERENT TEMPORAL BUILDING BLOCKS

We studied the temporal building block characteristics using RNN-based T-GCN [29], convolution-based Graph WaveNet [55], and self-attention-based GMAN [68] and compared the results by replacing spatial building blocks of these base models with GCN and GAT. We first briefly explain the three base models used in this study.

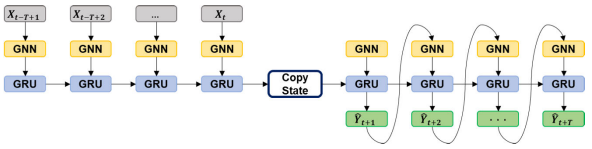

FIGURE 3. Architecture of T-GCN. The model extracts spatial features from input graph signal using GNN layers. Then, the extracted features are fed into GRU units to extract temporal features. The encoder-decoder framework is implemented for generating multiple time-step predictions. The GNN operation is GCN for T-GCN and GAT for T-GAT.

T-GCN [29] is a spatial-temporal traffic forecasting model combining GRU [86] and 2-layer GCN for temporal and spatial feature extraction, respectively. The update gate u_t, reset gate r_t, and outputs h_t of the GRU units at time t on input $\boldsymbol{X}_t \in \mathbb{R}^{N \times C}$ are defined as follows:

$$\boldsymbol{u}_t = \sigma\left(\boldsymbol{W}_u\left[f(\boldsymbol{A}, \boldsymbol{X}_t), \boldsymbol{h}_{h-1}\right] + \boldsymbol{b}_u\right), \quad (5)$$

$$\boldsymbol{r}_t = \sigma\left(\boldsymbol{W}_r\left[f(\boldsymbol{A}, \boldsymbol{X}_t), \boldsymbol{h}_{h-1}\right] + \boldsymbol{b}_r\right), \quad (6)$$

$$\boldsymbol{c}_t = \tanh\left(\boldsymbol{W}_c\left[f(\boldsymbol{A}, \boldsymbol{X}_t), (\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})\right] + \boldsymbol{b}_c\right), \quad (7)$$

$$\boldsymbol{h}_t = \boldsymbol{u}_t \odot \boldsymbol{h}_{t-1} + (1 - \boldsymbol{u}_t) \odot \boldsymbol{c}_t, \quad (8)$$

where $\odot$ is the element-wise Hadamard product and $\sigma(\cdot)$ is the sigmoid activation function, $f(\boldsymbol{A}, \boldsymbol{X}_t) = \sigma(\hat{\boldsymbol{A}}\mathrm{ReLU}(\hat{\boldsymbol{A}}\boldsymbol{X}_t\boldsymbol{W}_0)\boldsymbol{W}_1)$ is the 2-layer GCN model with learnable parameters $\boldsymbol{W}_0 \in \mathbb{R}^{C \times p}$ and $\boldsymbol{W}_1 \in \mathbb{R}^{p \times d}$, $\boldsymbol{W}_u$, $\boldsymbol{W}_r$, and $\boldsymbol{W}_c \in \mathbb{R}^{d \times d_{gru}}$ are learnable parameters, and $\boldsymbol{b}_u$, $\boldsymbol{b}_r$, and $\boldsymbol{b}_c$ are biases. Although the original TGCN adopted a many-to-one structure, we implemented the encoder-decoder framework for the multi-step prediction. In the following discussions, we denote the encoder-decoder T-GCN model as T-GCN, and T-GCN with GAT as T-GAT. Fig. 3 shows the T-GCN and T-GAT architecture.

Graph WaveNet [55] model combines dilated causal convolution [87] and convolutional GNN layers. Since the convolution-based temporal feature extraction requires no sequential computation, the model could overcome the vanishing gradient problem. The Graph WaveNet adopts

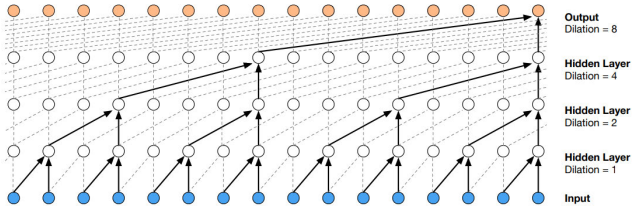

FIGURE 4. Illustration of dilated causal convolution (figure is adapted from Fig. 3 of [87].

the Gated Activation Unit (GAU) [102] for dilated causal convolution. The convolution with input $\boldsymbol{H}^i_{t-T\mathcal{C}1\mathcal{V}t} \in \mathbb{R}^{N\times T\times d}$ at time t with T historical graph signals can be defined as:

$$\begin{aligned}\boldsymbol{H}'_{t-T+1:t} &= \mathrm{GAU}\left((\Gamma_1, \Gamma_2) *_{conv} \boldsymbol{H}_{t-T+1:t}\right)\\ &= \tanh(\Gamma_1 *_{conv} \boldsymbol{H}_{t-T+1:t} \odot \sigma(\Gamma_2 *_{conv} \boldsymbol{H}_{t-T+1:t}),\end{aligned} \quad (9)$$

where $*_{conv}$ is the dilated convolution operation with convolution kernels Γ_1, and $\Gamma_2 \in \mathbb{R}^{p\times d\times d}$. For a node input $\boldsymbol{h}^i_{t-1+T:t} \in \mathbb{R}^{T\times d}$ for node v_i at time t with T historical node signals, the dilated convolution with kernel for one output channel $\gamma \in \mathbb{R}^{p\times d}$ is defined as follows:

$$\begin{aligned}&\gamma *_{conv} \boldsymbol{h}^i_{t-T+1:t}\\ &\quad= \sum_{b=1}^{d}\sum_{p=1}^{p} \gamma(p,b)\boldsymbol{h}^i_{t-T+1:t}(t - s\times p, b),\end{aligned} \quad (10)$$

where the p and b inside the parenthesis in $\gamma(p, b)$ are the indices of the elements of kernel γ, and s is the dilation factor. A dilated causal convolution layer is illustrated in Fig. 4. The output of the dilated causal convolution and gated activation unit is then fed to a spatial building block to generate the layer output with dimension $\mathbb{R}^{N\times(T-s\times(p-1))\times d}$. Note that the temporal lengths of inputs for the later layers are shorter than T.

Fig. 5 shows Graph WaveNet structure with the original spatial building block replaced by GNN. In [55], the GNN layer is implemented with a self-adaptive adjacency matrix term added to diffusion convolution [16]. The dilated causal convolution and GNN operation form a spatial-temporal layer, with residual and skip connections added to prevent information loss from stacking multiple spatial-temporal layers. For a more detailed description of Graph WaveNet, please refer to the original study [55]. This study replaces the spatial building block with GCN and GAT. Hereinafter, we denote the Graph WaveNet implemented with GCN and GAT as GWNet-GCN and GWNet-GAT, respectively.

GMAN [68] is a self-attention-based model using spatial and temporal attention modules to model traffic data. The model extracts spatial and temporal features separately and combines them using a gated fusion module. To impose positional information on the nodes and time steps, the model suggests spatial-temporal embedding, using time indicator vectors and node embedding vectors obtained by node2vec [99]. The temporal attention module of GMAN with input is

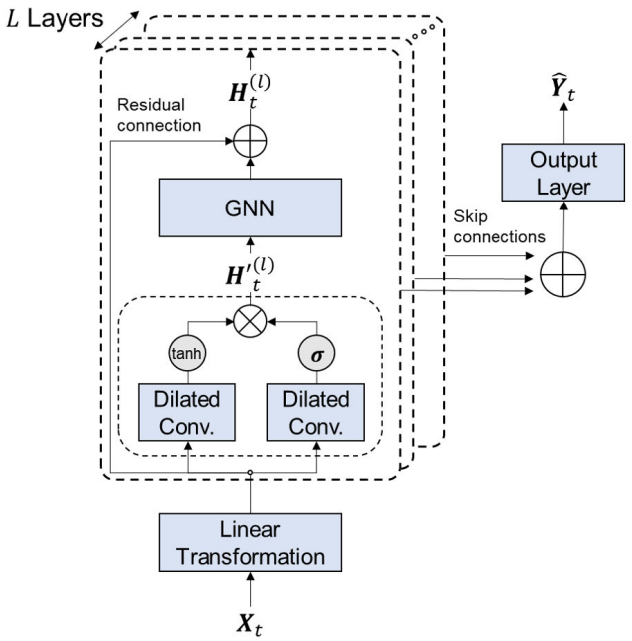

FIGURE 5. Architecture of GWNet. The model extracts temporal features using dilated causal convolution and gated activation unit. Then, a GNN module is implemented after the convolution to extract spatial features. A spatial-temporal (ST) layer consists of a dilated causal convolution with gated activation and a GNN module, and multiple ST-layers are stacked to extract the final representation. The subscripts t in this figure indicate the graph signals from time step $t - T + 1$ to t.

defined as:

$$ht^{(l)}_{i,t} = \mathrm{CAT}_{k=1}^{K}\left[\sum_{\tau\in\mathcal{N}_t} \alpha^{(k)}_{t,\tau}\cdot f_0^{(k)}\left(\left[h^{(l-1)}_{i,\tau}, e_{i,\tau}\right]\right)\right] W_o + b_o, \quad (11)$$

where $ht^{(l)}_{i,t}$ is the temporal feature vector of the l-th layer for node v_i at time t, $h^{(l-1)}_{i,\tau} \in \mathbb{R}^d$ is the output of the previous layer for node v_i at time t, $e_{i,\tau} \in \mathbb{R}^d$ is the spatial-temporal embedding vector, K is the number of head for multi-head attention, $\alpha^{(k)}_{t,\tau}$ is the attention score between the time step t and τ for head k, $\mathcal{N}_t$ is a set of input time steps, and $f_0^{(k)}$ is a non-linear projection defined as $f_0^{(k)}(x) = \mathrm{ReLU}(x\boldsymbol{W} + \boldsymbol{b})$ with learnable parameters $W \in \mathbb{R}^{2d\times d'}$ and $b \in \mathbb{R}^{d'}$, and $W_o \in \mathbb{R}^{Kd'\times d}$ and $b_o \in \mathbb{R}^d$ are learnable parameters. Here, the attention score $\alpha^{(k)}_{t,\tau}$ can be obtained as

$$\alpha^{(k)}_{t,\tau} = \frac{\exp\left(s^{(k)}_{t,\tau}\right)}{\sum_{t'\in\mathcal{N}_t}\exp\left(s^{(k)}_{t,t'}\right)}, \quad (12)$$

where

$$s^{(k)}_{t,\tau} = \frac{\left\langle f_1^{(k)}\left(\left[h^{(l-1)}_{i,t}, e_{i,t}\right]\right), f_2^{(k)}\left(\left[h^{(l-1)}_{i,\tau}, e_{i,\tau}\right]\right)\right\rangle}{\sqrt{d'}} \quad (13)$$

where $f_1^{(k)}$ and $f_2^{(k)}$ are non-linear projections, and d' is the dimension of each head. The gated fusion is implemented to

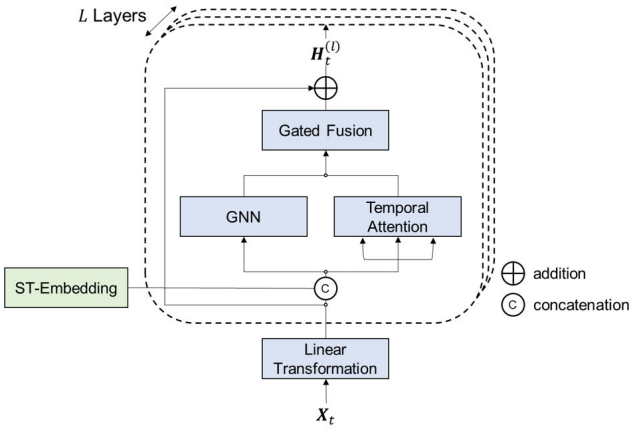

FIGURE 6. The GMAN-GNN encoder. The model extracts temporal features using the self-attention module and spatial features using GNN modules. It can be regarded as a transformer [61] encoder expanded on spatial dimension. By excluding the GNN module and making a residual connection between the input and output of the attention layer, the GMAN encoder can be transformed into a transformer encoder. The subscripts t in this figure indicate the graph signals from time step $t-T+1$ to t.

combine the spatial features $H_S^{(l)}$ and temporal features $H_T^{(l)}$ from the attention modules:

$$\boldsymbol{H}^{(l)} = z \odot H_S^{(l)} + (1-z) \odot H_T^{(l)}, \tag{14}$$

with

$$z = \sigma\left(H_S^{(l)}\boldsymbol{W}_{z,1} + H_T^{(l)}\boldsymbol{W}_{z,2} + \boldsymbol{b}_z\right), \tag{15}$$

where $\boldsymbol{W}_{z,1}, \boldsymbol{W}_{z,2} \in \mathbb{R}^{d\times d}$, and $\boldsymbol{b}_z \in \mathbb{R}^d$ are learnable parameters, and $\sigma(\cdot)$ is the sigmoid activation. While the spatial attention layer to obtain spatial features $H_S^{(l)}$ is implemented in a similar manner to the temporal attention layer in the original work, we replaced the spatial attention module with GCN and GAT, denoted GMAN-GCN and GMAN-GAT (Fig.6). The transform attention layer is implemented between the encoder and decoder to enable the multi-step prediction and reduce error propagation in the prediction task. GMAN can be regarded as a 2-dimensional expansion of the original transformer [61]. Two parallel self-attention modules are employed to extract features from both spatial and temporal dimensions, whereas transformer only considers a single dimension. To merge representations from two self-attention modules, GMAN replaces the feedforward layer in transformer with a feature fusion layer and makes one residual connection between the input and output of an encoder layer. For a more detailed description of GMAN, please refer to the original study [68].

C. DATA

To analyze the performance of each model, we select four real-world datasets with diverse characteristics, namely, PeMS-Bay, METR-LA [16],[1] Urban-core, and Urban-mix [31].[2]

[1]PeMS-Bay and METR-LA datasets are available at https://github.com/liyaguang/DCRNN

[2]Urban-core and Urban-mix datasets are available at https://github.com/yuyolshin/SeoulSpeedData

PeMS-Bay is a widely used speed dataset for traffic forecasting collected by California Transportation Agencies (CalTrans) Performance Measurement System (PeMS). The dataset contains six months of data ranging from January 1, 2017, to June 30, 2017, with a data frequency of 5 min. Spatially, 325 sensors in the Bay Area are included. The dataset examines the model performances for loop detector-based highway speed forecasting.

METR-LA traffic flow dataset contains data collected from loop detectors on Los Angeles County highways and is frequently used in traffic forecasting studies. The dataset contains 5-min traffic flow data from 207 sensors, from March 1, 2012, to June 30, 2012. The dataset analyses differences in model performances on traffic speed and flow datasets.

For the PeMS-Bay and METR-LA, we followed the procedures in Li et al. [16] to process the dataset and generate edges between traffic sensors. We construct the graphs and build adjacency matrices based on the distances between nodes and the threshold Gaussian kernel [103]:

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

where d_{ij} is the distance between sensor v_i and v_j, σ is the standard deviation, and $\epsilon = 0.1$ is the threshold value.

Urban-core and Urban-mix are 5-min speed data for road segments in the Seoul traffic network. Both contain information for one month ranging from April 1, 2018, to April 30, 2018. Urban-core includes 304 records of road segments in Gangnam, Seoul, one of the regions with the highest traffic and economic activities in the country. The road segments have similar structural features, such as speed limit, degree, and length.

Urban-mix is a spatial expansion of Urban-core and has road segments with more heterogeneous characteristics. It contains the inner-city highway connecting the East and West ends of the city, urban arterials, alleys, bridges, and a few intercity highway segments. The transportation network graph of Urban-mix has 1,007 road segments. The edges of transportation network graphs are set between road segments that share endpoints.

When the four datasets are compared in terms of complexity, the highway flow shows higher complexity than highway speed and urban speed demonstrate higher complexity than highway data as in Fig. 7. The approximate entropy values [104] on average are 0.52, 1.20, 1.40, and 1.41 for PeMS-Bay, METR-LA, Urban-core, and Urban-mix, respectively. Table 2 summarizes the datasets.

D. EXPERIMENTAL SETTINGS

We adopt mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE)

TABLE 2. Summary of the datasets.

	PeMS-Bay	METR-LA	Urban-core	Urban-mix
# Nodes	325	207	304	1,007
# Edges	2,694	1,722	1,696	5,597
Resolution	5-min	5-min	5-min	5-min
Duration	6 months	4 months	1 month	1 month
Site	Highway	Highway	Urban	Urban
Sensor Type	Loop Detector	Loop Detector	GPS	GPS
Train/Valid/Test	0.7 / 0.1 / 0.2 (Proportion)	0.7 / 0.1 / 0.2 (Proportion)	21 / 2 / 7 (Days)	21 / 2 / 7 (Days)

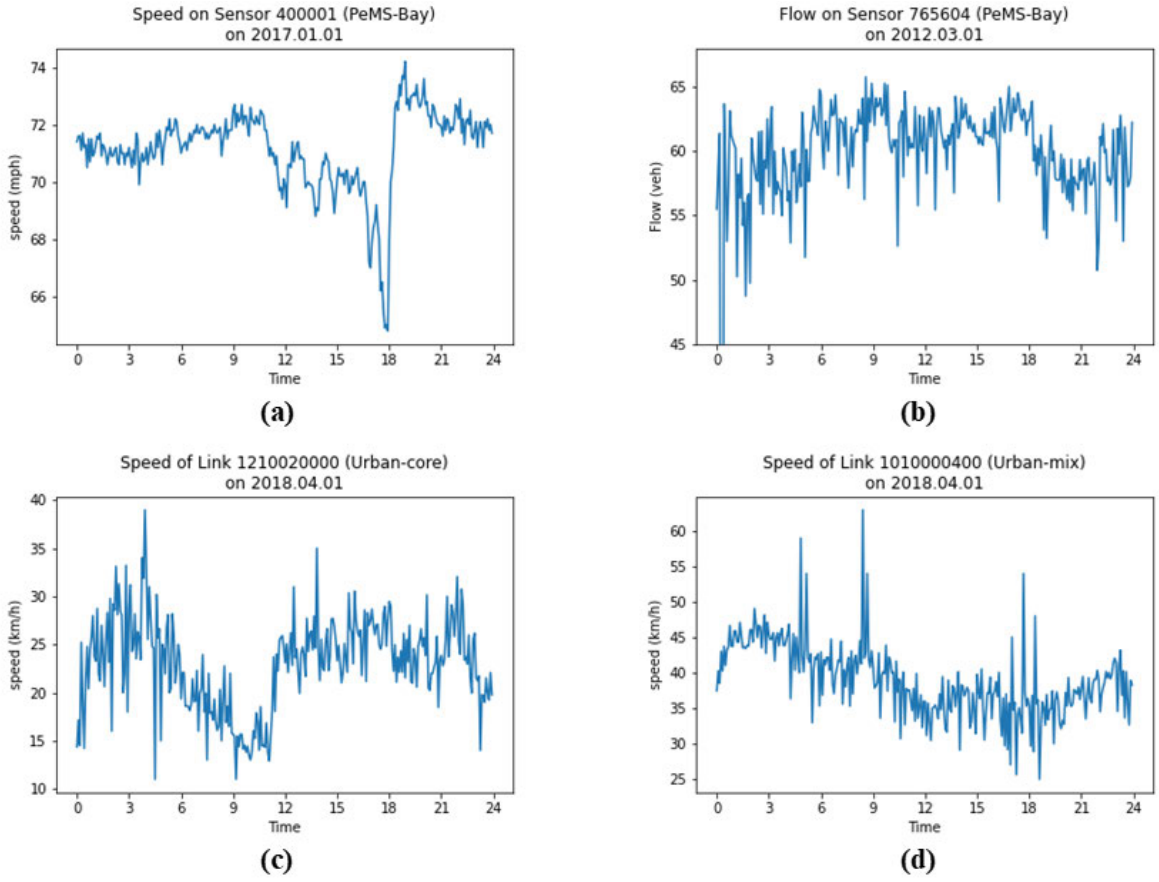

FIGURE 7. One day sample data from different datasets. (a) PeMS-Bay, (b) METR-LA, (c) Urban-core, and (d) Urban-mix. The traffic flow data (METR-LA) shows higher entropy than traffic speed data (PeMS-Bay), and urban data (Urban-core and Urban-mix) show higher entropy than highway data (PeMS-Bay).

as evaluation metrics for model performances.

$$\text{MAE} = \frac{1}{T'N}\sum_{i=1}^{N}\sum_{j=1}^{T'}\left|\hat{y}_j^i - y_j^i\right|, \tag{17}$$

$$\text{RMSE} = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{T'}\frac{\left(\hat{y}_j^i - y_j^i\right)^2}{T'N}}, \tag{18}$$

$$\text{MAPE} = \frac{1}{T'N}\sum_{i=1}^{N}\sum_{j=1}^{T'}\frac{\left|\hat{y}_j^i - y_j^i\right|}{y_j^i}, \tag{19}$$

where T' is the total number of predicted time steps, N is the number of nodes (sensors or road segments), and $\hat{y}_j^i$ and y_j^i are the predicted and actual values.

We calibrated each model hyperparameters as closely as that in the original works [29], [55], [68]. We set the number of hidden units to 64 for GMAN-GCN and GMAN-GAT and 32 for T-GCN, T-GAT, GWNet-GCN, and GWNet-GAT, batch size to 32, and learning rate to 0.001. For GAT, the number of heads and dimensions of each head are 8. The number of layers for GMAN models was 3 except for those in Urban-mix because of memory limitation and GMAN-GAT in METR-LA because the model failed to converge with 3 layers. A 2-layer model was used in these cases. We trained the models using the Adam optimizer, and L1 loss function. The experiment was conducted on a single NVIDIA TITAN RTX with 24 GB memory (GPU) and Intel(R) Xeon(R) CPU ES-2630 v4 @ 2.20 GHz (CPU).[3]

[3]The source codes are available at https://github.com/yuyolshin/STTFEvaluation

TABLE 3. Model performance on traffic datasets.

Prediction	Metric	T-GCN	T-GAT	GWNet-GCN	GWNet-GAT	GMAN-GCN	GMAN-GAT
PeMS-Bay							
15 min	MAE (mph)	2.25	1.58	**1.33**	1.38	1.47	1.44
	RMSE (mph)	3.78	2.96	**2.81**	2.90	3.02	2.92
	MAPE (%)	4.74	3.40	**2.78**	2.90	3.19	3.05
30 min	MAE (mph)	2.51	1.98	**1.67**	1.77	1.77	1.75
	RMSE (mph)	4.46	3.95	**3.78**	3.97	3.88	3.79
	MAPE (%)	5.54	4.47	**3.79**	4.06	4.01	3.93
60 min	MAE (mph)	2.91	2.46	**2.00**	2.18	2.07	2.05
	RMSE (mph)	5.25	5.05	4.60	4.94	4.55	**4.45**
	MAPE (%)	6.69	5.90	**4.77**	5.28	4.82	**4.77**
METR-LA							
15 min	MAE (veh/h)	4.08	3.57	**2.78**	2.88	3.07	3.12
	RMSE (veh/h)	7.79	5.97	**5.41**	5.57	5.74	5.79
	MAPE (%)	10.01	9.21	**7.37**	7.74	8.33	8.53
30 min	MAE (veh/h)	5.08	3.99	**3.17**	3.33	3.42	3.46
	RMSE (veh/h)	9.47	6.88	**6.44**	6.67	6.64	6.66
	MAPE (%)	12.77	10.65	**8.97**	9.55	9.68	9.84
60 min	MAE (veh/h)	6.46	4.66	**3.63**	3.86	3.86	3.89
	RMSE (veh/h)	11.29	8.08	**7.53**	7.83	7.66	7.57
	MAPE (%)	16.68	12.91	**10.58**	11.56	11.42	11.43
Urban-core							
15 min	MAE (km/h)	2.91	2.59	**2.42**	2.44	2.72	2.67
	RMSE (km/h)	4.12	3.79	**3.72**	3.75	4.06	3.98
	MAPE (%)	11.85	10.51	**9.42**	9.51	10.84	10.73
30 min	MAE (km/h)	3.10	2.84	**2.65**	2.69	2.77	2.72
	RMSE (km/h)	4.38	4.08	**4.00**	4.04	4.13	4.05
	MAPE (%)	12.85	11.89	**10.56**	10.80	11.07	10.99
60 min	MAE (km/h)	3.46	3.11	2.87	2.93	2.89	**2.83**
	RMSE (km/h)	4.81	4.42	4.27	4.34	4.25	**4.17**
	MAPE (%)	14.57	13.23	11.57	11.96	11.60	**11.52**
Urban-mix							
15 min	MAE (km/h)	3.37	2.79	**2.59**	2.62	2.91	2.90
	RMSE (km/h)	4.90	4.28	4.21	**4.18**	4.53	4.50
	MAPE (%)	13.29	10.69	**9.78**	9.98	11.14	11.17
30 min	MAE (km/h)	3.63	3.15	**2.92**	2.96	3.09	3.08
	RMSE (km/h)	5.43	4.91	4.83	**4.73**	4.88	4.86
	MAPE (%)	14.42	12.11	**11.17**	11.47	11.90	11.99
60 min	MAE (km/h)	4.90	3.55	**3.24**	3.29	3.38	3.38
	RMSE (km/h)	6.12	5.61	5.46	**5.28**	5.40	5.38
	MAPE (%)	16.42	13.70	**12.51**	12.94	13.15	13.26

V. RESULTS AND DISCUSSION

A. OVERALL PERFORMANCE

Table 3 shows the model performances in the four traffic datasets for 15 min (3 steps), 30 min (6 steps), and 60 min (12 steps) cases. When combined with the RNN model, GAT-based spatial feature extraction yields more accurate results than GCN, except for MAE on the 15-min forecast in METR-LA. The convolution shows improved predictions when combined with GCN except for RMSE in Urban-mix for all prediction horizons. When using self-attention for temporal feature extraction, GMAN-GAT consistently yields improved results than the GCN counterpart on at least one performance metric in all datasets except 15-min and 30-min predictions in METR-LA. Overall, the convolution models yield the best performance among the comparative models except in long-term (60-min) prediction in PeMS-Bay and Urban-core. Although T-GAT produces fair prediction outcomes, RNN shows no clear advantage over the other building blocks for temporal feature extraction.

The three temporal building blocks methods show differences in the gap between the forecasting accuracy on the 15-min and 60-min predictions. The RMSE differences between the two prediction horizons in PeMS-Bay are 70.6%, 63.7%, and 52.4% for the T-GAT, GWNet-GCN, and GMAN-GAT, respectively. The differences in RMSE in all datasets are presented in Table 4. The self-attention shows robust performance against the increase in prediction horizon, yielding a smaller gap between the 15-min and 60-min prediction outcomes. This indicates possible advantages for prediction horizons longer than one hour.

B. PERFORMANCE IN DIFFERENT TRAFFIC CATEGORIES

In this subsection, we analyze the performance of each model in different traffic categories. We divided the traffic states into unequal intervals, considering the range and distribution of each dataset. In PeMS-Bay, we initially divided the speed data with equal intervals of 10 mph. However, we merged the five lower speed intervals because each interval contained few observations, and merged the two higher speed intervals for the same reason. Since the 60∼70 mph interval included nearly 80% of the data, we divided the interval into two intervals of 5 mph. Finally, we have five speed categories in PeMS-Bay: 0∼50 mph, 50∼60 mph, 60∼65 mph, 65∼70 mph, and 70∼90 mph.

Table 5 presents the results of the traffic forecasting models in PeMS-Bay, across different traffic speed categories and

TABLE 4. RMSE gap between the 15-min and 60-min prediction for all datasets. The attention-based GMAN models show smaller gaps compared to the other models.

		T-GCN	T-GAT	GWNet-GCN	GWNet-GAT	GMAN-GCN	GMAN-GAT
PeMS-Bay	Δ	1.47	2.09	1.79	2.04	1.53	1.53
	%	38.9%	7.06%	63.7%	70.3%	50.7%	52.4%
METR-LA	Δ	3.50	2.11	2.12	2.26	1.92	1.78
	%	44.9%	35.3%	39.2%	40.6%	33.4%	30.7%
Urban-core	Δ	0.69	0.63	0.55	0.59	0.19	0.19
	%	16.7%	16.6%	14.8%	15.7%	4.7%	4.8%
Urban-mix	Δ	1.22	1.33	1.25	1.10	0.87	0.88
	%	24.9%	31.1%	29.7%	26.3%	19.2%	19.6%

$\Delta = \mathrm{RMSE}_{60} - \mathrm{RMSE}_{15}$
$\% = \frac{\mathrm{RMSE}_{60} - \mathrm{RMSE}_{15}}{\mathrm{RMSE}_{15}} \times 100\%$

prediction horizons. The best performance is observed in the 65∼70 mph category, which contains the most observations. In contrast, the largest errors are observed in the 0∼50 mph category, which is furthest from the high-speed, high-frequency 65∼70 mph category. In categories with over 60 mph, the category-wise errors are smaller than the overall performance. Similar to the overall performance evaluation, the two models outperform the RNN model. The convolution-based GWNet-GCN achieved high performances in the high-frequency categories. For 60-min prediction, GWNet-GCN produces more accurate predictions than GMAN-GAT in 60∼65, 65∼70, and 70∼90 mph categories. In contrast, GMAN-GAT shows more robust performance across different traffic categories than GWNet-GCN. In PeMS-Bay, the 0∼50 mph category MAE is 9.9 times larger than the 65∼70 mph category MAE for GWNet-GCN on 60-min prediction. In contrast, the ratio is 7.9 for GMAN-GAT. The ratios are 6.9 and 6.4 on 15-min predictions for GWNet-GCN and GMAN-GAT, respectively. Similar trends are observed in other datasets. In METR-LA, GWNet-GCN performs better in high-frequency categories (60∼65 and 65∼75 veh/h), while GMAN-GAT shows higher performance in low-frequency categories (30∼50 and 50∼60 veh/h). In the 0∼30 veh/h category, the self-attention model performance decreases, and the convolution model performance improves. For Urban-core, the distributions are right-skewed as opposed to highway datasets. Therefore, convolution models are more effective at low-speed categories, whereas self-attention models are better suited for high-speed categories. In Urban-mix, GWNet-GCN achieves the highest performance across all speed categories and prediction horizons. The category-wise performances for METR-LA, Urban-core, and Urban-mix are presented in Fig. 8.

We also analyze model performances in conditions where traffic states experience transitions. We denote the condition where the speed increases or decreases more than 30 mph in 90 min (18 time steps) in PeMS-Bay as speed increase and decrease transitions, respectively, and compare the 60-min prediction results. During transitions, the model performances differ from the overall performances. Whereas GWNet-GCN yielded low MAE and MAPE overall, GMAN-GAT outperformed GWNet-GCN in all performance metrics in increasing and decreasing transitions. Table 6 presents

TABLE 5. Performance in MAE by traffic speed categories in PeMS-Bay. GWNet-GCN presents high performances in high-frequency categories, while GMAN-GAT performs better in low-frequency categories.

Category (Ratio)	Model	15 min	30 min	60 min
0 ∼ 50 mph (7.52%)	T-GAT	6.45	7.82	9.11
	GWNet-GCN	**5.00**	7.13	9.04
	GMAN-GAT	5.23	**6.96**	**8.40**
50 ∼ 60 mph (10.19%)	T-GAT	2.74	3.72	4.64
	GWNet-GCN	**2.56**	**3.24**	3.84
	GMAN-GAT	2.67	3.27	**3.74**
60 ∼ 65 mph (31.34%)	T-GAT	1.21	1.62	2.09
	GWNet-GCN	**0.97**	**1.17**	**1.39**
	GMAN-GAT	1.06	1.27	1.50
65 ∼ 70 mph (47.28%)	T-GAT	1.01	1.23	1.57
	GWNet-GCN	**0.72**	**0.81**	**0.91**
	GMAN-GAT	0.82	0.92	1.06
70 ∼ 90 mph (3.67%)	T-GAT	1.78	2.21	2.82
	GWNet-GCN	**1.22**	**1.46**	**1.69**
	GMAN-GAT	1.43	1.63	1.88

TABLE 6. Prediction results of 60-min during traffic transitions.

PeMS-Bay						
	T-GAT		GWNet-GCN		GMAN-GAT	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Increase	12.61	16.04	10.79	14.61	**9.95**	**13.39**
Decrease	9.23	12.43	8.21	11.70	**7.13**	**10.31**
METR-LA						
	T-GAT		GWNet-GCN		GMAN-GAT	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Increase	**10.46**	**11.05**	11.05	15.45	15.45	41.14
Decrease	**11.85**	**10.70**	15.85	15.12	36.11	30.95
Urban-core						
	T-GAT		GWNet-GCN		GMAN-GAT	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Increase	4.33	5.98	4.14	5.88	**4.03**	**5.80**
Decrease	5.32	7.63	5.52	8.10	**5.03**	**7.45**
Urban-mix						
	T-GAT		GWNet-GCN		GMAN-GAT	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Increase	**10.50**	**14.24**	10.72	15.54	10.63	15.01
Decrease	8.73	12.19	**7.76**	**11.98**	8.37	12.63

the performance of 60-min forecasting outcomes during traffic transitions for all datasets. The results on the other datasets show similar trends as for PeMS-Bay. The RNN and self-attention models show advantages over the convolution model except in the Urban-mix. The traffic transition conditions in the other datasets are defined if the states change by 30 veh/h, 10 km/h, and 20 km/h for METR-LA, Urban-core, and Urban-mix, respectively.

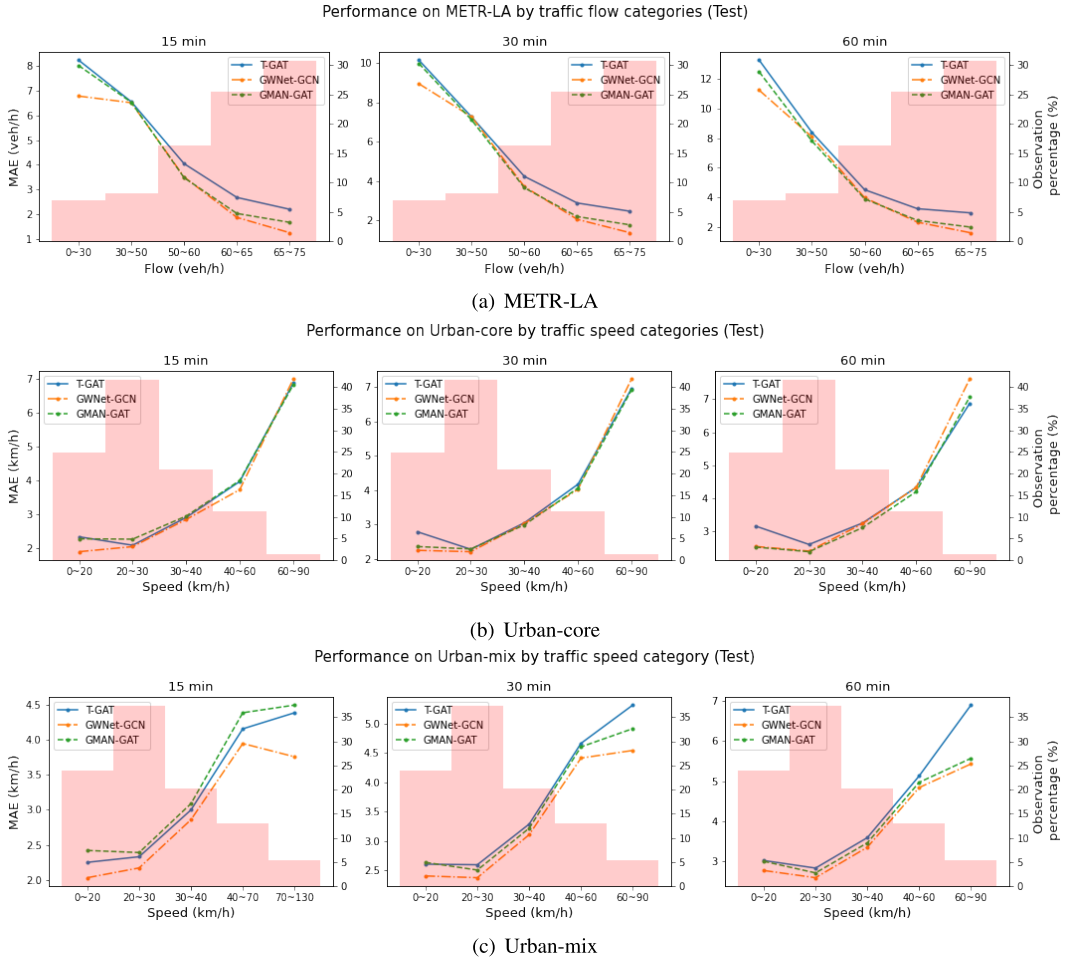

FIGURE 8. Performance by traffic flow and speed categories on (a) METR-LA, (b) Urban-core, and (c) Urban-mix. The line graphs are the MAE of each model in each traffic category, and the red histogram in the background is the ratio of each category in each dataset. Among the three models, GWNet-GCN achieves the best performance in categories with high observation percentage, and GMAN-GAT generally achieves the best performance in categories with low observation percentage.

C. ROBUSTNESS AGAINST OUTLIERS

Another characteristic observed is robustness against outliers in the labels. As in Figs. 9(a), (d), 10 (a) and (d), RNN and convolution-based temporal feature extractions show delayed reactions to outliers, causing large errors within a few time steps. While GWNet-GCN shows the highest overall accuracy in most datasets and prediction horizons as presented in previous sections, the self-attention model is more robust against outliers than the other models. In addition, the attention-based GAT models also show more robustness than GCN models for spatial feature extraction, as shown in Fig. 11.

D. ADAPTIVE MODEL EVALUATION

The model performance was found to change by traffic state categories. In this section, we evaluate the models adaptively by selecting the model depending on the prediction horizon and traffic category-wise performance on validation sets. For adaptive evaluation, we first make pseudo-labels $\hat{\mathcal{Y}}^{(p)^l}$ for each prediction horizon l by averaging the two candidate models, G_1 and G_2, as follows:

$$\hat{\mathcal{Y}}^{(p)^l} = \frac{1}{2}\left(G_1(\mathcal{X})^l + G_2(\mathcal{X})^l\right), \tag{20}$$

where $\mathcal{X}_{val}$ is the input data of validation sets, and $(G(\mathcal{X}_{val})^l$ is the output of the model G for prediction horizon l. The pseudo-labels are necessary to distribute the test sets in which category they should be evaluated. For each traffic category s, we compare the loss for the two models and make predictions $\hat{\mathcal{Y}}_{test^{l,s}}$ as follows:

$$\hat{\mathcal{Y}}_{test^{l,s}} = \begin{cases} \alpha * G_1(\mathcal{X}^s_{test})^l + (1-\alpha) * G_2(\mathcal{X}^s_{test})^l \\ \textit{if } L(G_1(\mathcal{X}^s_{val})^l, \mathcal{Y}^{(p)^{l,s}}_{val}) > L(G_2(\mathcal{X}^s_{val})^l, \mathcal{Y}^{(p)^{l,s}}_{val}) \\ (1-\alpha) * G_1(\mathcal{X}^s_{test})^l + \alpha * G_2(\mathcal{X}^s_{test})^l \\ \textit{if } L(G_1(\mathcal{X}^s_{val})^l, \mathcal{Y}^{(p)^{l,s}}_{val}) < L(G_2(\mathcal{X}^s_{val})^l, \mathcal{Y}^{(p)^{l,s}}_{val}) \end{cases} \tag{21}$$

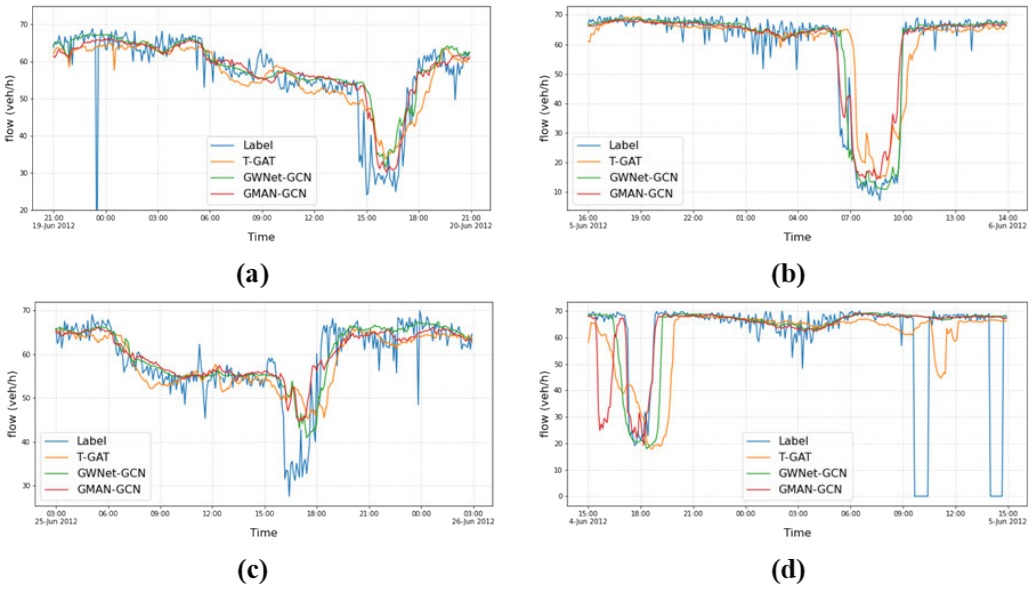

FIGURE 9. 60-min prediction labels (blue) and outcomes of T-GAT (orange), GWNet-GCN (green), and GMAN-GCN (red) in METR-LA for sample nodes. In (a) and (d), the RNN and convolution models show delayed reactions to outliers. While the convolution model shows the highest overall accuracy for 60-min prediction in METR-LA, the attention model shows more robustness against outliers.

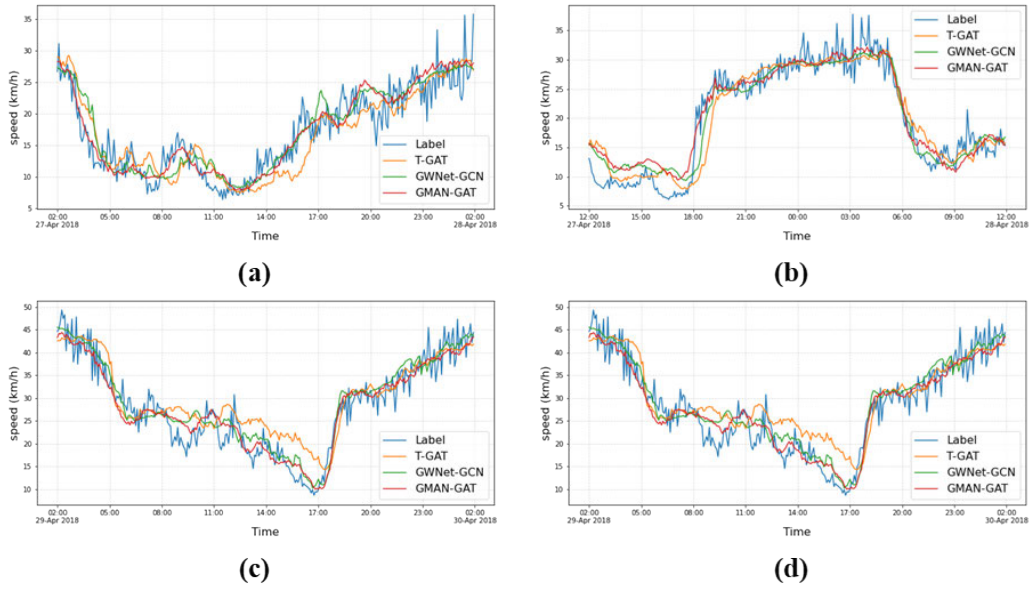

FIGURE 10. 60-min prediction labels (blue) and outcomes of T-GAT (orange), GWNet-GCN (green), and GMAN-GAT (red) in Urban-core for sample nodes. In (a) and (d), the RNN and convolution models show delayed reactions to outliers. In Urban-core, the attention model achieves more robustness against outliers compared to the other models along with the highest accuracy for 60-min prediction.

where α is a predefined value between 0.5 and 1, $\mathcal{Y}^{(p)^{l,s}}$ is the pseudo-label for prediction horizon l included in category s, and $\mathcal{X}^s$ is the corresponding pseudo-label $\mathcal{Y}^{(p)^{l,s}}$. For the final prediction, we calculate Eq. (21) for all categories and aggregate the category-wise results. In this experiment, α is set to 0.7. The concept of this adaptive model evaluation framework is visualized in Fig. 12. The adaptive evaluation framework achieved higher performance on at least two performance metrics in all datasets and prediction horizons as shown in Table 7. For 60-min prediction in PeMS-Bay, the performance gain is the largest, outperforming the previous state-of-the-art GMAN and GWNet by 3.7%. When the Diebold-Mariano test is conducted for 60-min forecasts, forecasts on 57.5%, 44.4%, 31.3%, and 56.8% of nodes are statistically significant ($\alpha = 0.1$) in PeMS-Bay, METR-LA, Urban-core, and Urban-mix, respectively.

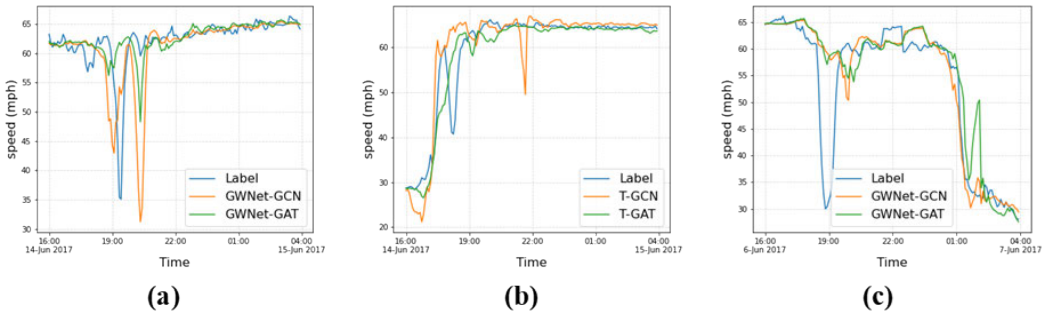

FIGURE 11. Robustness against outliers by different spatial feature extraction methods. GAT models show more robustness against GCN models for all T-GCN, GWNet, and GMAN models.

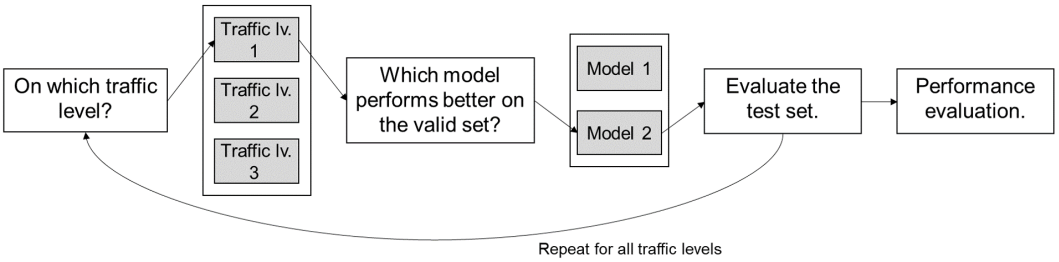

FIGURE 12. Adaptive model evaluation framework. It adaptively selects which model to conduct prediction for different traffic categories based on the performance on validations sets. As a result, the prediction can be made with multiple models, improving the utility of each model.

TABLE 7. Performance of adaptive model evaluation framework.

Category	15 min			30 min			60 min		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
PeMS-Bay									
T-GAT	1.58	2.96	3.40	1.98	3.95	4.47	2.46	5.05	5.90
GWNet-GCN	1.33	2.81	2.78	1.67	3.78	3.79	2.00	4.60	4.77
GMAN-GAT	1.44	2.92	3.05	1.75	3.79	3.93	2.05	4.45	4.77
Adp. Eval.	**1.31**	**2.75**	**2.75**	**1.65**	**3.67**	**3.76**	**1.95**	**4.33**	**4.59**
METR-LA									
T-GAT	3.57	5.97	9.21	3.99	6.88	10.65	4.66	8.08	12.91
GWNet-GCN	**2.78**	5.41	**7.37**	3.17	6.44	**8.91**	3.63	7.53	10.58
GMAN-GAT	3.12	5.79	8.53	3.46	6.66	9.84	3.89	7.57	11.43
Adp. Eval.	**2.78**	**5.32**	7.43	**3.15**	**6.28**	8.92	**3.59**	**7.29**	**10.54**
Urban-core									
T-GAT	2.59	3.79	10.51	2.84	4.08	11.89	3.11	4.42	13.23
GWNet-GCN	2.42	3.72	**9.42**	2.65	4.00	10.56	2.87	4.27	11.57
GMAN-GAT	2.42	3.70	**9.42**	**2.62**	3.97	10.62	2.84	4.18	11.55
Adp. Eval.	**2.41**	**3.69**	9.43	**2.62**	**3.94**	**10.48**	**2.78**	**4.10**	**11.29**
Urban-mix									
T-GAT	2.79	4.28	10.69	3.13	4.91	12.11	3.55	5.61	13.70
GWNet-GCN	2.59	4.21	9.78	2.92	4.83	11.17	3.24	5.46	12.51
GMAN-GAT	2.90	4.50	11.17	3.08	4.86	11.99	3.38	5.38	13.26
Adp. Eval.	**2.58**	**4.13**	**9.75**	**2.88**	**4.69**	**11.03**	**3.18**	**5.25**	**12.30**

E. DISCUSSION

An extensive and multi-faceted evaluation of six traffic forecasting models was conducted to characterize and understand the deep learning model building blocks for traffic forecasting. The convolution models showed the highest forecasting power overall among the three temporal feature extraction methods. This supports the current practice in which most deep learning-based traffic forecasting models are built with convolution-based temporal feature extraction. For temporal building blocks, the self-attention models demonstrate competitive long-term predictions, but the RNN models show no advantages in any task. The results do not imply that convolution and self-attention are superior to RNN but that they have clear advantages over RNN in traffic forecasting. When pairing the spatial and temporal feature extraction methods, improved performances are noticed when convolution is combined with convolutional GNN and self-attention with the attentional GNN except in METR-LA.

We infer that these paired methods are similar in extracting information from input data.

Further assessments reveal that the models show different performance sensitivity to traffic state changes. The convolution model performed well in high-frequency traffic categories, and the self-attention model showed robust performances even in low-frequency traffic categories and with outliers. In addition, during the traffic transitions, the self-attention, and RNN models show advantages in long-term prediction. The attention-based methods in spatial and temporal dimensions demonstrated improved robustness with outliers. Overall, the convolution model achieves more performance gain for the short-term (15-min) prediction and high-frequency traffic categories. In contrast, the self-attention model has more advantages in prediction for less-informed conditions such as longer prediction horizons, low-frequency traffic categories, and outliers.

In addition, we suggest a framework that adaptively selects a model for each category to make predictions based on the validation set performance. The results reveal that the simple implementation of an adaptive evaluation framework could improve the performance of the previous state-of-the-art by 3.7% at most. This framework enhances traffic forecasting performance using the existing models rather than developing more sophisticated models.

VI. CONCLUSION

In this study, we investigated the characteristics and evaluated the performance of building blocks of spatial-temporal deep learning models for traffic forecasting. We implemented six spatial-temporal models using two spatial building blocks and three temporal building blocks and conducted a multi-faceted experiment analyzing the overall performance, category-wise performance, and robustness against outliers. The models were tested on four real-world datasets with diverse transportation networks. While GWNet-GCN demonstrated the most accurate overall performance in most datasets and prediction horizons, GMAN-GAT showed a similar performance level with GWNet-GCN for 60-min prediction in PeMS-Bay and outperformed GWNet-GCN for 60-min prediction in Urban-core. Further investigations revealed that the self-attention model had stronger robustness against data imbalance and outliers than the RNN and convolution models. GAT models showed more robustness than GCN models amongst spatial feature extraction methods. Finally, an adaptive model evaluation framework demonstrated the enhanced performance of the existing models without sophistication in model architecture.

In the future study, we aim to expand the scope of this comparative study on building blocks to include sensitivity analysis of the hyperparameters and impact analysis of different input features such as daily and weekly trends, and multi-channel inputs. Also, revealing characteristics of additional spatial feature extraction methods such as diffusion convolution [16], traffic graph convolution [15], and adaptive graph convolution [44], [55], [57] can be another objective. Sophisticated state-of-the-art models could be investigated to discover whether the model characteristics would persist. Explainable artificial intelligence techniques [105], [106] could also be adopted to explore the deep learning-based traffic forecasting model characteristics. These techniques have been rarely used in traffic forecasting studies [107] and could give a new direction if implemented appropriately. Moreover, the adaptive model evaluation framework will be refined to include predictions during transition states and against time-series anomalies.

GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES

During the preparation of this work the authors used ChatGPT and Grammarly in order to check the grammar. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

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

[3] K. Lee, M. Eo, E. Jung, Y. Yoon, and W. Rhee, "Short-term traffic prediction with deep neural networks: A survey," *IEEE Access*, vol. 9, pp. 54739–54756, 2021.

[4] A. Sumalee, P. Luathep, W. H. K. Lam, and R. D. Connors, "Evaluation and design of transport network capacity under demand uncertainty," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2090, no. 1, pp. 17–28, Jan. 2009.

[5] X. Fang, J. Huang, F. Wang, L. Zeng, H. Liang, and H. Wang, "ConSTGAT: Contextual spatial–temporal graph attention network for travel time estimation at Baidu Maps," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 2697–2705.

[6] C.-Y. Jiang, X.-M. Hu, and W.-N. Chen, "An urban traffic signal control system based on traffic flow prediction," in *Proc. 13th Int. Conf. Adv. Comput. Intell. (ICACI)*, May 2021, pp. 259–265.

[7] D. Rolnick et al., "Tackling climate change with machine learning," *ACM Comput. Surv.*, vol. 55, no. 2, pp. 1–96, Feb. 2022.

[8] M. Jusup, P. Holme, K. Kanazawa, M. Takayasu, I. Romic, Z. Wang, S. Gecek, T. Lipic, B. Podobnik, L. Wang, W. Luo, T. Klanjscek, J. Fan, S. Boccaletti, and M. Perc, "Social physics," *Phys. Rep.*, vol. 948, pp. 1–148, Feb. 2022.

[9] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.

[10] S. R. Chandra and H. Al-Deek, "Predictions of freeway traffic speeds and volumes using vector autoregressive models," *J. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 53–72, May 2009.

[11] C.-H. Wu, J.-M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004.

[12] G. A. Davis and N. L. Nihan, "Nonparametric regression and short-term freeway traffic forecasting," *J. Transp. Eng.*, vol. 117, no. 2, pp. 178–188, Mar. 1991.

[13] R. Chrobok, J. Wahle, and M. Schreckenberg, "Traffic forecast using simulations of large scale networks," in *Proc. IEEE Intell. Transp. Syst. (ITSC)*, Aug. 2001, pp. 434–439.

[14] S. Jeon and B. Hong, "Monte Carlo simulation-based traffic speed forecasting using historical big data," *Future Gener. Comput. Syst.*, vol. 65, pp. 182–195, Dec. 2016.

[15] 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.

[16] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. 6th Int. Conf. Learn. Represent.*, May 2018, pp. 1–16.

[17] B. Liao, J. Zhang, C. Wu, D. McIlwraith, T. Chen, S. Yang, Y. Guo, and F. Wu, "Deep sequence learning with auxiliary information for traffic prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 537–546.

[18] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.

[19] X. Ma, H. Yu, Y. Wang, and Y. Wang, "Large-scale transportation network congestion evolution prediction using deep learning theory," *PLoS ONE*, vol. 10, no. 3, Mar. 2015, Art. no. e0119044.

[20] X. Ma, J. Zhang, B. Du, C. Ding, and L. Sun, "Parallel architecture of convolutional bi-directional LSTM neural networks for network-wide metro ridership prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2278–2288, Jun. 2019.

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

[22] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, "Traffic flow prediction via spatial temporal graph neural network," in *Proc. Web Conf.* New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1082–1092.

[23] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transp. Res. C, Emerg. Technol.*, vol. 90, pp. 166–180, May 2018.

[24] D. Xu, H. Dai, Y. Wang, P. Peng, Q. Xuan, and H. Guo, "Road traffic state prediction based on a graph embedding recurrent neural network under the SCATS," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 29, no. 10, Oct. 2019, Art. no. 103125.

[25] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, p. 1501, Jun. 2017.

[26] 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," in *Proc. 34th Conf. Uncertain Artif. Intell.*, Aug. 2018, 2018.

[27] 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.

[28] Z. Zhang, M. Li, X. Lin, Y. Wang, and F. He, "Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies," *Transp. Res. C, Emerg. Technol.*, vol. 105, pp. 297–322, Aug. 2019.

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

[30] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, Mar. 2017.

[31] Y. Shin and Y. Yoon, "Incorporating dynamicity of transportation network with multi-weight traffic graph convolutional network for traffic forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2082–2092, Mar. 2022.

[32] X. Huang, Y. Ye, X. Yang, and L. Xiong, "Multi-view dynamic graph convolution neural network for traffic flow prediction," *Expert Syst. Appl.*, vol. 222, Jul. 2023, Art. no. 119779.

[33] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Dec. 2020, pp. 17804–17815.

[34] Y. Wang, J. Zheng, Y. Du, C. Huang, and P. Li, "Traffic-GGNN: Predicting traffic flow via attentional spatial–temporal gated graph neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18423–18432, Oct. 2022.

[35] Y. Chen, I. Segovia, and Y. R. Gel, "Z-GCNETs: Time zigzags at graph convolutional networks for time series forecasting," in *Proc. 38th Int. Conf. Mach. Learn.*, Jul. 2021, pp. 1684–1694.

[36] H. Dong, P. Zhu, J. Gao, L. Jia, and Y. Qin, "A short-term traffic flow forecasting model based on spatial–temporal attention neural network," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 416–421.

[37] Q. Zhang, M. Tan, C. Li, H. Xia, W. Chang, and M. Li, "Spatio-temporal residual graph convolutional network for short-term traffic flow prediction," *IEEE Access*, vol. 11, pp. 84187–84199, 2023.

[38] M. Gupta, H. Kodamana, and S. Ranu, "Frigate: Frugal spatio-temporal forecasting on road networks," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2023, pp. 649–660.

[39] C. Chen, K. Li, S. G. Teo, G. Chen, X. Zou, X. Yang, R. C. Vijay, J. Feng, and Z. Zeng, "Exploiting spatio-temporal correlations with multiple 3D convolutional neural networks for citywide vehicle flow prediction," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 893–898.

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

[41] Y. Fang, Y. Qin, H. Luo, F. Zhao, B. Xu, C. Wang, and L. Zeng, "Spatio-temporal meets wavelet: Disentangled traffic flow forecasting via efficient spectral graph attention network," 2021, *arXiv:2112.02740*.

[42] 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.

[43] L. Ge, S. Li, Y. Wang, F. Chang, and K. Wu, "Global spatial–temporal graph convolutional network for urban traffic speed prediction," *Appl. Sci.*, vol. 10, no. 4, p. 1509, Feb. 2020.

[44] L. Han, B. Du, L. Sun, Y. Fu, Y. Lv, and H. Xiong, "Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 547–555.

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

[46] B. Lu, X. Gan, H. Jin, L. Fu, and H. Zhang, "Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1025–1034.

[47] K. Lee and W. Rhee, "DDP-GCN: Multi-graph convolutional network for spatiotemporal traffic forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 134, Jan. 2022, Art. no. 103466.

[48] Q. Liu, B. Wang, and Y. Zhu, "Short-term traffic speed forecasting based on attention convolutional neural network for arterials," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 11, pp. 999–1016, Nov. 2018.

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

[50] G. Shen, C. Chen, Q. Pan, S. Shen, and Z. Liu, "Research on traffic speed prediction by temporal clustering analysis and convolutional neural network with deformable kernels (May, 2018)," *IEEE Access*, vol. 6, pp. 51756–51765, 2018.

[51] Q. Song, R. Ming, J. Hu, H. Niu, and M. Gao, "Graph attention convolutional network: Spatiotemporal modeling for urban traffic prediction," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.

[52] K. Tian, J. Guo, K. Ye, and C.-Z. Xu, "ST-MGAT: Spatial–temporal multi-head graph attention networks for traffic forecasting," in *Proc. IEEE 32nd Int. Conf. Tools with Artif. Intell. (ICTAI)*, Nov. 2020, pp. 714–721.

[53] C. Tian and W. K. V. Chan, "Spatial–temporal attention Wavenet: A deep learning framework for traffic prediction considering spatial–temporal dependencies," *IET Intell. Transp. Syst.*, vol. 15, no. 4, pp. 549–561, Apr. 2021.

[54] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong, "Traffic speed prediction and congestion source exploration: A deep learning method," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 499–508.

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

[56] 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.

[57] Y. Shin and Y. Yoon, "PGCN: Progressive graph convolutional networks for spatial–temporal traffic forecasting," 2022, *arXiv:2202.08982*.

[58] J. Zhao, Z. Liu, Q. Sun, Q. Li, X. Jia, and R. Zhang, "Attention-based dynamic spatial–temporal graph convolutional networks for traffic speed forecasting," *Expert Syst. Appl.*, vol. 204, Oct. 2022, Art. no. 117511.

[59] K. Zhu, S. Zhang, J. Li, D. Zhou, H. Dai, and Z. Hu, "Spatiotemporal multi-graph convolutional networks with synthetic data for traffic volume forecasting," *Expert Syst. Appl.*, vol. 187, Jan. 2022, Art. no. 115992.

[60] B. Chen, K. Hu, Y. Li, and L. Miao, "Hybrid spatio-temporal graph convolution network for short-term traffic forecasting," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 2128–2133.

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017, pp. 1–11.

[62] J. Bai, J. Zhu, Y. Song, L. Zhao, Z. Hou, R. Du, and H. Li, "A3T-GCN: Attention temporal graph convolutional network for traffic forecasting," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 7, p. 485, Jul. 2021.

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

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

[65] K. Jin, J. Wi, E. Lee, S. Kang, S. Kim, and Y. Kim, "TrafficBERT: Pre-trained model with large-scale data for long-range traffic flow forecasting," *Expert Syst. Appl.*, vol. 186, Dec. 2021, Art. no. 115738.

[66] C. Park, C. Lee, H. Bahng, Y. Tae, S. Jin, K. Kim, S. Ko, and J. Choo, "ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1215–1224.

[67] M. Xu, W. Dai, C. Liu, X. Gao, W. Lin, G.-J. Qi, and H. Xiong, "Spatial–temporal transformer networks for traffic flow forecasting," 2020, *arXiv:2001.02908*.

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

[69] Y. Wen, P. Xu, Z. Li, W. Xu, and X. Wang, "RPConvformer: A novel transformer-based deep neural networks for traffic flow prediction," *Expert Syst. Appl.*, vol. 218, May 2023, Art. no. 119587.

[70] S. Reza, M. C. Ferreira, J. J. M. Machado, and J. M. R. S. Tavares, "A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks," *Expert Syst. Appl.*, vol. 202, Sep. 2022, Art. no. 117275.

[71] X. Xu, X. Hu, Y. Zhao, X. Lü, and A. Aapaoja, "Urban short-term traffic speed prediction with complicated information fusion on accidents," *Expert Syst. Appl.*, vol. 224, Aug. 2023, Art. no. 119887.

[72] K. Wang, L. Liu, Y. Liu, G. Li, F. Zhou, and L. Lin, "Urban regional function guided traffic flow prediction," *Inf. Sci.*, vol. 634, pp. 308–320, Jul. 2023.

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

[74] L. Wei, Z. Yu, Z. Jin, L. Xie, J. Huang, D. Cai, X. He, and X.-S. Hua, "Dual graph for traffic forecasting," *IEEE Access*, early access, Dec. 9, 2019, doi: 10.1109/ACCESS.2019.2958380.

[75] S. Li, L. Ge, Y. Lin, and B. Zeng, "Adaptive spatial–temporal fusion graph convolutional networks for traffic flow forecasting," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 4189–4196.

[76] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Proces. Syst.*, vol. 29, 2016, pp. 1–9.

[77] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and deep locally connected networks on graphs," in *Proc. 2nd Int. Conf. Learn. Represent.*, Apr. 2014, pp. 1–14.

[78] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent.*, Apr. 2017, pp. 1–14.

[79] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Represent.*, May 2018, pp. 1–12.

[80] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, Dec. 2012, pp. 1–9.

[81] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[82] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014.

[83] 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.

[84] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

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

[86] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," 2014, *arXiv:1409.1259*.

[87] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.

[88] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[89] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial–temporal network for taxi demand prediction," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 2588–2595.

[90] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial–temporal similarity: A deep learning framework for traffic prediction," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 5668–5675.

[91] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, "DeepSTN+: Context-aware spatial–temporal neural network for crowd flow prediction in metropolis," in *Proc. 33rd AAAI conf. Artif. intell.*, 2019, vol. 33, no. 1, pp. 1020–1027.

[92] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," 2021, *arXiv:2104.13478*.

[93] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[94] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, Aug. 2017, pp. 1263–1272.

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

[96] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[97] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.

[98] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 1067–1077.

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

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

[101] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *Expert Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 117921.
[102] A. van den Oord, N. Kalchbrenner, L. Espeholt, k. kavukcuoglu, O. Vinyals, and A. Graves, "Conditional image generation with PixelCNN decoders," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, Dec. 2016, pp. 1–9.
[103] 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.
[104] S. M. Pincus, "Approximate entropy as a measure of system complexity," *Proc. Nat. Acad. Sci. USA*, vol. 88, no. 6, pp. 2297–2301, Mar. 1991.
[105] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?' Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Aug. 2016, pp. 1135–1144.
[106] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Sys.*, vol. 30, Dec. 2017, pp. 1–10.
[107] A. Barredo-Arrieta, I. Laña, and J. Del Ser, "What lies beneath: A note on the explainability of black-box machine learning models for road traffic forecasting," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 2232–2237.

YUYOL SHIN received the B.S. and Ph.D. degrees in civil and environmental engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2016 and 2022, respectively. He is currently a Postdoctoral Researcher in civil and environmental engineering with KAIST. During his postdoctoral appointment, he has visited UC Berkeley in civil and environmental engineering as a Visiting Scholar, from October 2022 to June 2023. His research interests include spatial-temporal data mining, graph neural networks, artificial intelligence applications, and transportation network analysis.

YOONJIN YOON (Member, IEEE) received the B.S. degree in mathematics from Seoul National University, Seoul, South Korea, in 1996, the dual M.S. degree in computer science and in management science and engineering from Stanford University, Stanford, CA, USA, in 2000 and 2002, respectively, and the Ph.D. degree in civil and environmental engineering from the University of California at Berkeley, Berkeley, CA, USA, in 2010. Since 2011, she has been an Assistant Professor with the Department of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. Before arriving at KAIST, she was a Graduate Student Researcher with the National Center of Excellence in Air Transportation Operations Research (NeXTOR), University of California at Berkeley, from 2005 to 2010. Her previous research experience includes as a Research Assistant with the Artificial Intelligence Center, SRI International, Menlo Park, CA, USA, in 1999, and the Center of Reliability Computing, Stanford University, Stanford, CA, USA, from 2000 to 2002. Her research interests include the traffic management of both manned and unmanned vehicles using stochastic optimization. Her recent research efforts include data-driven driving behavior analysis, and autonomous vehicle traffic flow management using large-scale driving data.