Spatio-Temporal Graph Convolutional Networks: Spatial Layers First or Temporal Layers First?

Yuen Hoi LAU, Raymond Chi-Wing WONG
{yhlauai,raywong}@cse.ust.hk
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Hong Kong, China

ABSTRACT

Traffic forecasting is an important and challenging problem for intelligent transportation systems due to the complex spatial dependencies among neighboring roads and changing road conditions in different time periods. Spatio-temporal graph convolutional networks (STGCNs) are usually adopted to forecast traffic features in a road network. Existing STGCN models involve spatial layers and temporal layers. Some models involves spatial layers first and then temporal layers and some other models involves these layers in a reverse order. This creates an interesting research question on whether the ordering of involving the spatial layers (or temporal layers) first in an existing STGCN model could improve the prediction performance. To the best of our knowledge, we are the first to study this interesting research problem, which creates a deep insight as a guideline to the research community on how to design STGCN models. We conducted extensive experiments to study a number of representative STCGN models for this research problem. We found that these models with spatial layers constructed before temporal layers has a higher chance to outperform that with temporal layers constructed first, which suggests the future design principle of STGCN models.

KEYWORDS

spatio-temporal graph convolutional networks, traffic forecasting, sequence of modeling, spatial dependencies, temporal dependencies

ACM Reference Format:
Yuen Hoi LAU, Raymond Chi-Wing WONG. 2021. Spatio-Temporal Graph Convolutional Networks: Spatial Layers First or Temporal Layers First?. In *Proceedings of ACM Conference (SIGSPATIAL'21).* ACM, Beijing, China, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Traffic forecasting aims to predict future traffic features including volume, speed, occupancy, demand and travel time of each road segment of a road network. It is useful in many applications such as transportation management, navigation systems, order dispatching and ride sharing [43]. Good traffic forecasting is important in achieving higher efficiency and accuracy of these applications. This task is very challenging due to the complex spatial dependencies among irregular road segments, temporal information from their own and external conditions such as weather and holidays. Spatio-temporal graph neural networks have captured tremendous attention and are usually adopted to accomplish these tasks in recent years thanks to their capability to capture the spatial and temporal dependencies among road segments [41].

There are some assumptions regarding spatio-temporal graph modeling. Road segments are in irregular shapes and regarded as nodes in a graph while traffic sensors gather data on road segments. An adjacent matrix represents the nodes' proximities. A node's future value depends on its own historical values as well as neighboring nodes' information. How to capture spatial and temporal dependencies well is the primary goal. Graph convolution networks (GCN) usually form *spatial layers* while recurrent neural networks (RNN) and temporal convolution neural networks (TCN) become *temporal layers.* Recent studies on spatio-temporal graph convolutional networks (STGCNs) are divided into two streams. The first one is to incorporate GCN into RNN [24, 45] and the other is into TCN [42, 40]. GCN is assumed to reflect the spatial dependency relationship among nodes while RNN or TCN entails the temporal information for each node.

Different state-of-the-art STGCNs have different modeling sequence for spatial layers and temporal layers. Some are constructed with spatial layers first [24, 45], some with temporal layers first[42] and a few with both layers gated and fused. Attention mechanisms including additive and scaled dot-product attention are put into some of these models to improve prediction performance. However, none of these studies gives a satisfactory explanation for the modeling sequence of spatial and temporal layers. In view of this, we propose a new research problem: is there a preferable modeling sequence for spatial and temporal layers in STGNN to forecast traffic variables more accurately? To put it succinctly, does a spatio-temporal model constructed with spatial layers first must outperform itself with temporal layers first? To the best of our knowledge, we are the first one to study this research problem.

In this paper, we study this problem by swapping the sequence of the spatial and temporal layers in different STGCNs and comparing their forecasting performance with spatial layers first and temporal layers first respectively for each model by using evaluation metrics including Mean Absolute Errors (MAE) and Root Mean Squared Errors (RMSE).

The notable contributions of our paper are summarized as follows:

Unpublished working draft. Not for distribution.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGSPATIAL'21, June 2021, Beijing, China

© 2021 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

- To the best of our knowledge, we are the first to explore whether there is a preferable modeling sequence for STGCNs (e.g., spatial layers first vs. temporal layers first).
- We propose to swap the order of spatial and temporal layers for each type of STGCNs if applicable. By doing so, original models constructed with spatial layers first are modified with temporal layers first. Original models having temporal layers before spatial ones are modified conversely.
- We conduct experiments on real-world traffic datasets to compare the forecasting performance of each selected state-of-the-art STGCNs with the original modeling sequence and the modified one.

2 RELATED WORK

2.1 Early Forecasting Approaches

Early approaches utilize the historical information of traffic features of a road segment to predict its future features. Auto-regressive Integrated Moving Average (ARIMA) [5], Support Vector Regression (SVR) [34], Recurrent Neural Networks including Long Short-Term Memory (LSTM) models [18] and Gated Recurrent Unit (GRU) [11], and Hidden Markov Model (HMM) [9] are among the early techniques. However, these techniques can only capture temporal characteristics to some extent while spacial features are ignored. In addition, they have to be applied or trained separately for each road segment, leading to a longer training time. Furthermore, their prediction performance is not satisfactory as pointed out by [42, 24, 40, 45].

2.2 Graph Convolution Networks

There are many different Graph Neural Networks (GNN) proposed in recent years. The fundamental idea is to generate a node's high-level representation by utilizing its own features and neighbors' features in non-Euclidean space. Survey papers including [35, 44, 47] introduced a large amount of variants of GNN models. These studies mainly focus on several important tasks of graph data mining: node embedding, node classification, node clustering, edge prediction and graph classification by incorporating spatial characteristics. Graph convolutional networks is one of the GNN branches and divided into two mainstreams: spectral-based and spatial-based. Spectral-based approaches use graph spectral filters to remove noises from node input features [6, 13, 22, 23, 48]. Spatial-based approaches aggregate information from neighbors' features to generate a node's high-level representation [27, 1, 16, 33, 14, 8, 39, 38, 10]. Since the GCN in [22] bridged the gap between spectral-based approaches and spatial-based approaches, it is widely used as the spatial layers in STGCNs due to its efficiency and prediction performance.

2.3 Spatio-Temporal Graph Convolutional Networks

State-of-the-art STGCNs are proposed in recent years to forecast traffic features and the results are promising. However, different STGCNs have different modeling sequences for spatial and temporal layers. GCN usually forms spatial layers while RNN or TCN becomes temporal layers. Some models place spatial layers before temporal layers while the converse is true for some other models. A few recent models fuse the two layers concurrently. Recent related studies are summarized below.

2.3.1 *Spatial Layers First.*

A multitude of related studies using GCN for capturing the dependencies in the spatial domain obtained promising results by virtue of GCN's intrinsic advantages for modeling non-Euclidean data. [24] introduced Diffusion Convolutional Recurrent Neural Network (DCRNN) which models traffic flow as a diffusion process and captures the spatial dependency using bidirectional random walks in GCN and the temporal dependency using the GRU encoder-decoder architecture. [15] proposed Attention Based Spatial-Temporal Graph Convolutional Networks (ASTGCN) with spatial and temporal attention first to capture spatial and temporal dependencies. The second layer of the block is a GCN plus temporal convolution. Adaptive Graph Convolutional Recurrent Network (AGCRN) [3] proposed a node adaptive parameter learning module and a data adaptive graph generation module for enhancing graph convolutional network to capture spatial dependencies while GRU acts as the temporal layers to capture temporal dependencies. Temporal GCN (T-GCN) [45] placed GCN first to capture spatial dependencies, followed by GRU to capture temporal dependencies. Spatio-temporal Graph Attention Networks (ST-GRAT) [28] adapted self-attention mechanisms to dynamically capture both spatial and temporal dependencies. MTGNN [36] designed a graph learning module to learn hidden spatial dependencies and TCN to capture temporal dependencies.

2.3.2 *Temporal Layers First.*

[42] proposed Spatio-Temporal Graph Convolutional Networks (STGCN) which are composed of complete convolutional structures to enable much faster training speed with fewer parameters and a temporal gated convolutional architecture comes before the Chebyshev Spectral Graph Convolution [13]. Graph WaveNet [37] captured the hidden spatial dependency by constructing a self-adaptive adjacency matrix and a gated temporal convolution layer (Gated TCN) before GCN. [20] came up with Long Short-term Graph Convolutional Networks (LSGCN) to use gated linear unit (GLU) to capture dynamic behaviors of temporal features first, which is similar to STGCN. After it, the hidden features are passed to the spatial gated block.

2.3.3 *Spatial and Temporal Layers Fused Concurrently.*

The Graph Multi-Attention Network (GMAN) [46] employed an encoder-decoder architecture with multiple spatio-temporal attention blocks to capture the impact of the spatio-temporal factors on traffic conditions. The Unified Spatio-Temporal Graph Convolution Network (USTGCN) proposed in [29] has spectral graph convolution on a spatio-temporal graph for both spatial and temporal aggregation through direct information propagation. Spatial-temporal Synchronous Graph Convolutional Networks (STSGCN) [30] designed the spatial-temporal synchronous graph convolution module that directly integrates the complex localized spatial-temporal correlations. A very recent work Spatial-Temporal Fusion Graph Neural Networks (STFGNN) [26] is built based on framework of STSGCN to effectively learn hidden spatial-temporal dependencies by fusing various spatial and temporal graphs.

Unpublished working draft. Not for distribution.

3 PROBLEM STATEMENT

The problem we are trying to solve can be abstracted as node feature predictions over time in a spatial-temporal graph using traffic features only and determine whether a model constructed with spatial layers first would outperform that with temporal layers first.

A road network is modeled as a directed graph $G = (V, E, A)$, where each vertex $v \in V$ denotes a road segment and each edge $e \in E$ denotes the proximity between two vertices. An adjacent matrix $A \in \mathbb{R}^{N \times N}$ is derived from the graph, where N is the number of vertices. If $(v_i, v_j) \in E$, then A_{ij} is one, otherwise zero. At each time step t, the graph G has a feature matrix $X^{(t)} \in \mathbb{R}^{N \times D}$, where D is the number of features.

We denote h as a function for a data processing layer before the spatial and temporal layers, g_S as a function for the spatial layers and f_T as the function for the temporal layers in STGCNs. Generally speaking, $g_S(f_T(X)) \neq f_T(g_s(X))$.

Problem: Given the road network $G = (V, E, A)$ and all historical traffic features $X \in \mathbb{R}^{T \times N \times D}$, our problem is to learn and determine whether the composite function $f_T \cdot g_s \cdot h$ representing spatial layers first or $g_S \cdot f_T \cdot h$ representing temporal layers first forecasts P future traffic graph features more accurately given P' historical traffic graph features :

$$[X^{(t-P'+1):t}, G] \xrightarrow{g_T \cdot f_S \cdot h} [X^{(t+1):(t+P)}], \qquad (1)$$

$$[X^{(t-P'+1):t}, G] \xrightarrow{f_S \cdot g_T \cdot h} [X^{(t+1):(t+P)}], \qquad (2)$$

where $X^{(t-P'+1):t} \in \mathbb{R}^{P' \times N \times D}$ and $X^{(t+1):(t+P)} \in \mathbb{R}^{P \times N \times D}$.

4 METHODOLOGY

4.1 Overview of Our Proposed Framework

Various state-of-the-art STGCNs have different modeling sequences for spatial and temporal layers. However, does the sequence of modeling matter for each model? To the best of our knowledge, we are the first to propose this problem. We try to find an answer from swapping the spatial and temporal layers for each selected model to examine whether a model with spatial layers first outperforms that with temporal layers first. Figure 1 shows the concept of models constructed with spatial layers first. When the input data are fed into the model, the data processing layers project the lower dimensional input data into higher dimensional traffic features. The layer of GCN captures the spatial dependencies of hidden features. The spatial post-processing layer can be attention mechanisms, diffusion convolutions, residual networks or simply dense layers. These layers form the spatial layers. The layer of TCN or RNN including LSTM and GRU captures the dynamic behaviors of hidden temporal features. Before the output layer, the temporal post-processing layer transforms the hidden features to predicted values. The same concept applies to models constructed with temporal layers first as illustrated in Figure 2. The data processing layers remain unchanged while the sequence of spatial layers and temporal layers is swapped.

4.2 Data Processing Layers

The first block of layers are data processing layers, which can be spatial-temporal embedding layers [46], attention mechanisms including Bahdanau attention [2], Luong attention [25] and multi-head scaled dot-product attention mechanism [32], or dense layers, projecting the raw traffic features such as speeds and flows into higher dimensional hidden features.

4.3 Spatial Layers

4.3.1 Graph Convolutional Networks. GCNs are the building blocks for the spatial layers to learn spatial dependencies among non-Euclidean data. We briefly describe the graph convolution operator applied in our proposed framework. Given a graph $G = (V, E, A)$, input signals $H^{(t)} \in \mathbb{R}^{N \times D}$ at time t, the adjacent matrix with self-connection $\tilde{A} = A + I_N$, and the diagonal degree matrix $\tilde{D} \in \mathbb{R}^{N \times N}$, a widely used graph convolution [22] is defined as follows:

$$\theta_{*G} H^{(t)} = \sigma(\tilde{D}^{-1}\tilde{A}H^{(t)}\theta_1), \qquad (3)$$

where $\theta_1 \in \mathbb{R}^{D \times F}$ is the learnable parameter matrix, D is the number of features before the GCN operation, F is the number of hidden features after the GCN operation, $\tilde{D}^{-1}\tilde{A}$ is the normalized adjacent matrix, and σ is a non-linear activation function. We can expand the receptive neighborhood range by stacking GCNs. For example, $\sigma(\tilde{D}^{-1}\tilde{A}\ \sigma(\tilde{D}^{-1}\tilde{A}H\theta_1)\theta_2)$ aggregates the information of two hops of neighbors where $\theta_2 \in \mathbb{R}^{F \times F}$. Since there is a multitude of variants of GCNs, we generalize the operation of GCNs as follows:

$$h_v^{(i)} = GCN(h_v^{(i-1)},\ f(\{h_u^{(i-1)}, u \in N(v)\})). \qquad (4)$$

where (1) $h_v^{(i)}$ denotes the feature vector of node v at the i^{th} layer of GCNs, (2) GCN is an operator to aggregate the information of the node v and its neighbors $N(v)$, (3) u is a neighbor of v and (4) f is a function to capture the information of neighbors.

4.3.2 Spatial Post-Processing Layer. After the GCN layer, there is a spatial post-processing layer, which can be attention mechanisms, diffusion convolutions [1, 24], residual networks [17, 37], or dense layers to extract more spatial characteristics. There are a number of attention mechanisms summarized in survey papers [19, 7]. In our modified models, we employ the scaled dot-product attention [32] after the GCN layer to fine-tune spatial dependencies because of its efficiency and comparable performance to other attention mechanisms. The scaled dot-product attention is formulated as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V. \qquad (5)$$

This is a spatial self-attention in our modified models with the same hidden features as the query $Q \in \mathbb{R}^{N \times H}$, the key $K \in \mathbb{R}^{N \times H}$ and the value $V \in \mathbb{R}^{N \times H}$. d_k is the number of hidden features. N is the number of nodes and H is the number of hidden features.

4.4 Temporal Layers

The building blocks of temporal layers are Recurrent Neural Networks (RNNs) or Temporal Convolutional Networks (TCNs) [4]. RNNs including GRU [11] and LSTM [18] have shown the capability of modeling the temporal dependency to a large extent. GRU

Unpublished working draft. Not for distribution.

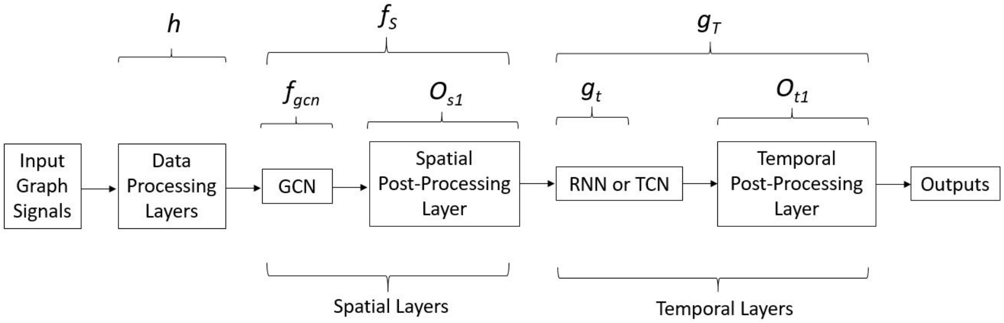

Figure 1: A concept of models with spatial layers first.

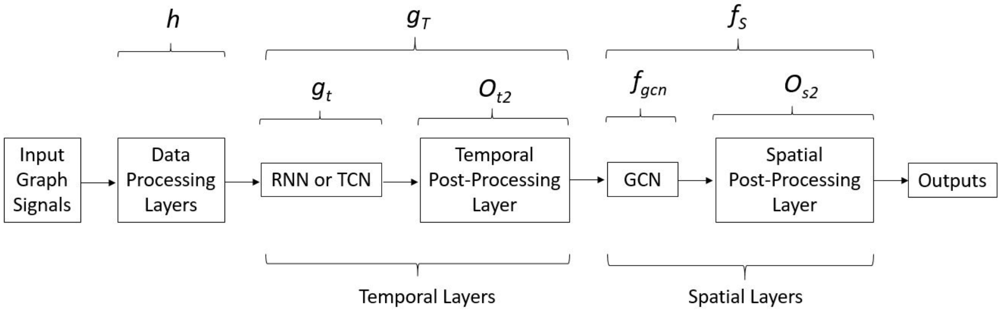

Figure 2: A concept of models with temporal layers first.

Unpublished working draft. Not for distribution.

is preferred over LSTM because of its efficiency and comparable forecasting performance. TCNs are also widely adopted due to its efficiency.

4.4.1 Gated Recurrent Unit. GRU is formulated as follows. At a given time step t, let the input $H^{(t)} \in \mathbb{R}^{P' \times F}$ (where P' is the number of sequences and F is the number of features) denote hidden features projected by previous layers, $U^{(t)} \in \mathbb{R}^{P' \times H}$ denote the output (where P' is the number of sequences and H is the number of hidden features), $U^{(t-1)} \in \mathbb{R}^{P' \times H}$ denote the hidden state of the previous time step, Θ denote learnable parameters, $R^{(t)} \in \mathbb{R}^{P' \times H}$, $Z^{(t)} \in \mathbb{R}^{P' \times H}$ and $C^{(t)} \in \mathbb{R}^{P' \times H}$ denote the reset gate, the update gate and candiate hidden gate, respectively. ϕ is a dense layer. σ is the Sigmoid function and $\odot$ is the Hadamard product.

$$\begin{aligned} R^{(t)} &= \sigma(\phi([H^{(t)}, U^{(t-1)}]; \Theta_R)), \\ Z^{(t)} &= \sigma(\phi([H^{(t)}, U^{(t-1)}]; \Theta_Z)), \\ C^{(t)} &= tanh(\phi([H^{(t)}, (R^{(t)} \odot U^{(t-1)})]; \Theta_C)), \\ U^{(t)} &= Z^{(t)} \odot U^{(t-1)} + (1 - Z^{(t)}) \odot C^{(t)}. \end{aligned} \tag{6}$$

We can stack multiple GRU layers and employ the sequence to sequence architecture [31] for multiple step ahead forecasting.

4.4.2 Temporal Convolutional Networks. Another method to model temporal layers is through TCNs, which are constructed by dilated causal convolution. Compared to RNNs, TCNs have the advantages of being able to handle long-range sequences in a non-recursive manner, enabling parallel computation and alleviating the gradient explosion problem. Let $F = (f_1, f_2, ..., f_K)$ be filters, $H^{(t)} \in \mathbb{R}^{P' \times F}$ be hidden features from previous layers, and $h^{(t)} \in \mathbb{R}^F$ be a node in $H^{(t)}$. The operation of TCNs at time t is defined as

$$F_{*d} H^{(t)} = \sum_{k=1}^{K} f_k h^{(t)}_{P' - (K-k) \times d} \tag{7}$$

where d is the dilation factor which controls the skipping distance. Stacking multiple TCNs captures the information of longer range sequences as illustrated in Figure 3. If $d = 1$, it becomes causal convolution.

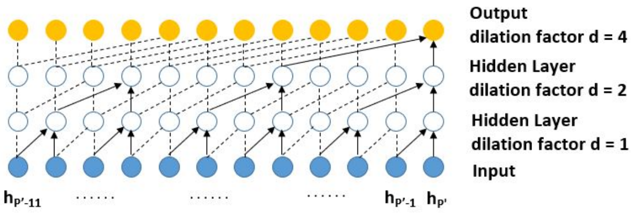

Figure 3: An example of stacking 3 TCNs with kernel size 2 for each 1D convolution, length of sequence 12, dilation factor d = 1, 2 and 4

4.4.3 Temporal Post-Processing Layer. The layer after RNN or TCN layers can be attention mechanisms, gated linear unit [12, 42, 37] or fully connected layers to capture temporal dependencies of hidden features. In our modified models, we employ the same self-attention mechanism as that in spatial layers.

4.5 Proposed Framework

We decompose each STGCN into several blocks of layers and reconstruct the model with spatial layers first and temporal layers first, respectively, as shown in Figure 1 and Figure 2. Given input signals $X^{(t-P'+1):t}$ and a corresponding graph $G(V, E, A)$, models with spatial layers first and temporal layers first can be, respectively, generalized as:

$$X^{(t+1):(t+P)} = O_{t1}(\, g_t(\, O_{s1}(\, f_{gcn}(\, h(X^{(t-P'+1):t}, G))))), \tag{8}$$

$$X^{(t+1):(t+P)} = O_{s2}(\, f_{gcn}(\, O_{t2}(\, g_t(\, h(X^{(t-P'+1):t}, G))))). \tag{9}$$

where $h : \mathbb{R}^{P' \times N \times D} \rightarrow \mathbb{R}^{P' \times N \times H}$ is the function for the data processing layers in both Equation 8 and Equation 9. In Equation 8, $f_{gcn} : \mathbb{R}^{P' \times N \times H} \rightarrow \mathbb{R}^{P' \times N \times H}$ is the GCN layer, $O_{s1} : \mathbb{R}^{P' \times N \times H} \rightarrow \mathbb{R}^{P' \times N \times H}$ is the function for spatial post-processing layer after the GCN layer, $g_t : \mathbb{R}^{P' \times N \times H} \rightarrow \mathbb{R}^{P' \times N \times H}$ is the function for the layer of RNNs or TCNs and $O_{t1} : \mathbb{R}^{P' \times N \times H} \rightarrow \mathbb{R}^{P \times N \times C}$ is the function for the temporal post-processing layer to predict future P periods of C-dimensional traffic features for N nodes. It is assumed that H is the number of hidden units. Similar formulations also apply to Equation 9 while the last layer is O_{s2} instead of O_{t1} to project hidden features to predict future traffic features. Note that although O_{s1} and O_{s2} correspond to spatial post-processing layers, they are just different in projecting inputs into outputs with different number of dimensions (due to the structure of the layers following O_{s1} (or O_{s2})). Similar arguments could be made to O_{t1} and O_{t2}.

Equation 8 corresponds to Equation 1 and Equation 9 corresponds to Equation 2. The function for spatial layers $f_S \equiv O_{si} \cdot f_{gcn}$ while the function for temporal layers $g_T \equiv O_{ti} \cdot f_{gcn}$ for $i \in \{1, 2\}$.

The models are trained by minimizing the errors between predicted values and ground truths through a loss function, which can be Mean Absolute Error, Mean Squared Error or Huber loss.

5 EXPERIMENTS

We set the scope of our research on the models which take only traffic features including speeds and/or flows as input. It is shown by many papers that additional data such as weather and holidays can enrich traffic conditions to improve the forecasting capability of STGCNs. Therefore, we would like to focus on finding out a better architecture for STGCNs which have a clear spatio-temporal modeling sequence without these additional data. Whether a model is influential or not is our major consideration for being selected in our experiments. There are a number of papers [24, 15, 3, 45, 28, 36] proposing spatial-layer-first models. We just chose DCRNN [24], ASTGCN [15] and T-GCN [45] as representative models in our experiments for comparison since they had at least 192 citations within 3 years but other models did not. We also select AGCRN [3] due to an influential work regarding a similar idea of the adaptive graph convolution [23]. There are a few papers [42, 37, 20] proposing temporal-layer-first models. We just chose STGCN [42] and Graph WaveNet [37] as representative models in our experiments for comparison since they had at least 144 citations within 3 years but LSGCN [20] did not.

It is worth mentioning that hybrid models proposed in [29, 30, 46, 26] are not under our framework considering spatial/temporal layers first, followed by other types of layers, since hybrid models consider both spatial and temporal layers at the same time in the form of a graph including spatial and temporal nodes. In [29], it is possible that the hybrid model USTGCN could perform worse than Graph WaveNet which follows our framework in PeMSD4 datasets. In [30], the hybrid model STSGCN could perform worse than DCRNN which follows our framework in PeMS07 datasets. In [46], the hybrid model GMAN could perform worse than DCRNN and Graph WaveNet in Xiamen and PEMS datasets. However, it is interesting to explore what exact forms of graphs in hybrid models could perform better, which could be regarded as a future work.

By swapping the spatial layers and temporal layers and compare the forecasting performance of the different modeling sequence for each of the following models. The first four models were models originally designed with the spatial layer first but the remaining models were models originally designed with the temporal layer first.

- DCRNN [24]: Diffusion convolution which is combined with GRU sequence-to-sequence architecture.
- ASTGCN [15]: An attention-based spatio-temporal graph convolutional network, which further integrates spatial and temporal attention mechanisms to the STGCNs that consist of GCNs and temporal convolution.
- T-GCN [45]: A temporal graph convolutional network model, which is in combination with GCNs and GRU.
- AGCRN [3]: A node adaptive parameter learning module and a data adaptive graph generation module are proposed for enhancing graph convolutional networks while GRU acts as the temporal layers to capture temporal dependencies.
- STGCN [42]: An entire convolution architecture which comprises GCNs and temporal gated convolution.
- Graph WaveNet [37]: A convolution network which introduces a self-adaptive graph convolution for the spatial layers and dilated convolution to capture temporal dependencies.

5.1 How Selected Models Are Mapped to Our Proposed Framework

In this section, we describe how the selected models are mapped to our proposed framework.

- **DCRNN:** The original DCRNN shown in Figure 4a has a clear modeling sequence with input signals directly fed into the diffusion convolution before going into GRU. There are no data processing layers before the spatial layers. A dense layer being the spatial post-processing layer is used between the diffusion convolution and GRU layer. Before the output, a dense layer being the temporal post-processing layer is employed to project higher dimensional hidden features into lower dimensional predicted values. The modified DCRNN shown in Figure 4b has a different sequence for the spatial and temporal layers while other things remain unchanged. In our experiments, we further add attention mechanisms to be the spatial and temporal post-processing layers shown in Figure 4c and Figure 4d to find out whether forecasting performance improves or not. The post-processing layers can be gradually reduced to empty layers.
- **ASTGCN:** There are both spatial attention and temporal attention layers just after the input signals are fed into the original ASTGCN in Figure 5a. The two attention layers act as the data processing layers in our framework. Afterwards, the GCN layer is followed by the TCN layer. There is no

Unpublished working draft. Not for distribution.

Table 1: Performance Comparison of models with spatial layers first and temporal layers first using original dataset

Dataset	Models	15 min		30 min		60 min	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
METR-LA	DCRNN-s (original)	15.71	29.12	16.10	29.34	16.67	29.70
	DCRNN-t (ours)	15.89	28.90	16.51	29.42	17.44	30.10
	DCRNN-s-att (ours)	**14.96**	**27.79**	**15.40**	**28.07**	**16.07**	**28.52**
	DCRNN-t-att (ours)	20.71	32.96	20.68	32.93	20.68	32.93
PeMSD4	ASTGCN-s (original)	**20.2**	**31.58**	**22.24**	**34.46**	**26.75**	**40.64**
	ASTGCN-t (ours)	20.36	31.86	22.55	35	27.38	41.55
Los-loop	T-GCN-s (original)	**3.3027**	5.2563	3.8175	6.2737	4.6255	7.5929
	T-GCN-t (ours)	3.6570	5.4929	3.9915	6.2942	4.7994	7.4709
	T-GCN-s-att (ours)	3.3306	**5.2548**	**3.8167**	**6.2728**	**4.6242**	**7.5908**
	T-GCN-t-att (ours)	3.4204	5.4182	3.8790	6.3025	4.6014	7.6418
PeMSD4	AGCRN-s (original)	18.96	31.10	19.72	32.45	21.28	35.13
	AGCRN-t (ours)	18.75	**30.31**	**19.51**	**31.62**	21.19	**34.19**
	AGCRN-s-att (ours)	**18.72**	30.64	**19.51**	32.22	**20.88**	34.67
	AGCRN-t-att (ours)	20.06	31.63	21.66	34.07	25.37	39.22
PeMSD7(M)	STGCN-s (ours)	2.726	5.057	3.925	7.518	3.613	6.844
	STGCN-t (original)	2.737	5.058	3.894	7.316	3.663	6.862
	STGCN-s-att (ours)	**2.708**	**4.989**	3.453	6.293	**3.402**	**6.395**
	STGCN-t-att (ours)	2.711	4.992	**3.416**	**6.176**	3.448	6.411
METR-LA	Graph WaveNet-s (ours)	**2.8554**	**5.3355**	**3.3451**	**6.5428**	**4.0114**	**8.0558**
	Graph WaveNet-t (original)	2.9865	5.7408	3.6237	7.2001	4.5517	9.0554
	Graph WaveNet-s-att (ours)	3.1229	5.8169	3.7285	7.2734	4.5744	9.0018
	Graph WaveNet-t-att (ours)	3.0474	5.8191	3.6862	7.2772	4.5995	9.0280

Unpublished working draft. Not for distribution.

spatial post-processing layer after the GCN layer. A fully connected layer functions as the temporal post-processing layer. The spatial layers and the temporal layers in the modified ASTGCN illustrated in Figure 5b are swapped. In our experiments, we do not add attention mechanisms to both the original and the modified ASTGCN since it has included attention mechanisms.

- **T-GCN:** Input signals are directly fed into GCN first in the original T-GCN and into GRU afterwards as illustrated in Figure 6a. There are no data processing layers, no spatial post-processing layer and no temporal post-processing layer. The sequence of the spatial and temporal layers is swapped in the modified T-GCN shown in Figure 6b. In our experiments, attention mechanisms are added after the GCN layer and the GRU layer to act as spatial and temporal post-processing layers shown in Figure 6c and Figure 6d. The post-processing layers can be reduced to empty ones.
- **AGCRN:** The adaptive GCN layer first appears after the input layer in the original AGCRN and the GRU layer follows as shown in Figure 7a. We swap the sequence of the adaptive GCN and GRU layers in our experiments as illustrated in Figure 7b. There are no data processing layers, no spatial post-processing layer and no temporal post-processing layer. In our experiments, we add scaled dot-product attention mechanisms after the GCN and GRU layers to be the spatial and temporal post-processing layers as shown in Figure 7c and Figure 7d.
- **STGCN:** The original STGCN shown in Figure 8b consists of two spatio-temporal blocks with temporal gated convolution before the GCN layer. There are no data processing layers after the input signals are fed into the model. The output layer in STGCN is a fully connected layer which projects higher dimensional hidden features into lower dimensional predicted values. We swap the sequence of temporal gated convolution and the GCN layer in our experiments as shown in Figure 8a. We further add attention mechanisms to act as the spatial post-processing layer and the temporal post-processing layer in our experiments as illustrated in Figure 8c and Figure 8d.
- **Graph WaveNet:** There is a linear layer after the input signals enter the original Graph WaveNet shown in Figure 9b and it acts as data processing layers. After the TCN layer, a gated TCN operation is considered as the temporal post-processing layer in our proposed framework. The two linear layers before the output can be considered the spatial post-processing layer. In our experiments, we swap the GCN and TCN layers as illustrated in Figure 9a. Furthermore, we add attention mechanisms after the GCN and TCN layers to be the spatial and temporal post-processing layers as shown in Figure 9c and Figure 9d .

5.2 Experimental Settings

Our experiments are conducted under a computer environment with one Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz and one NVIDIA GeForce RTX-3060 Ti GPU card. The task is to learn the two functions $g_T \cdot f_S : \mathbb{R}^{P' \times N \times D} \rightarrow \mathbb{R}^{P \times N \times D}$ and $f_S \cdot g_T : \mathbb{R}^{P' \times N \times D} \rightarrow \mathbb{R}^{P \times N \times D}$ representing the models with spatial layers first and temporal layers first, respectively, and compare the forecasting results

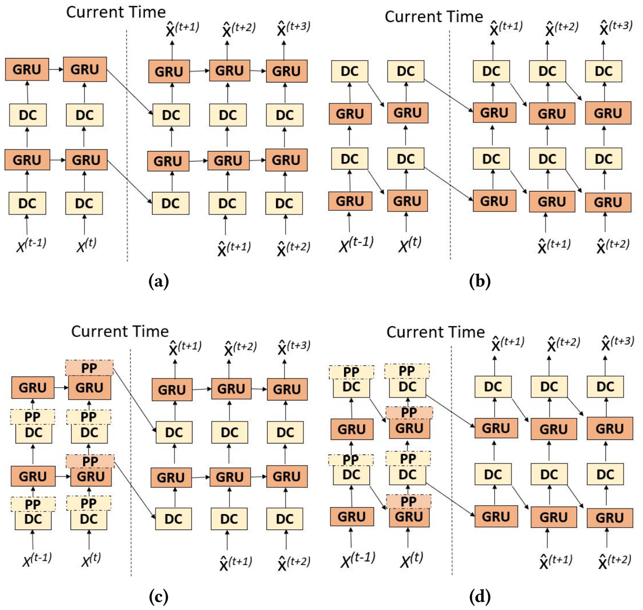

Figure 4: (a) DCRNN (original) with spatial layers first (b) DCRNN (ours) with temporal layers first (c) DCRNN (ours) with spatial layers first and with post-processing layers (d) DCRNN (ours) with temporal layers first and with post-processing layers

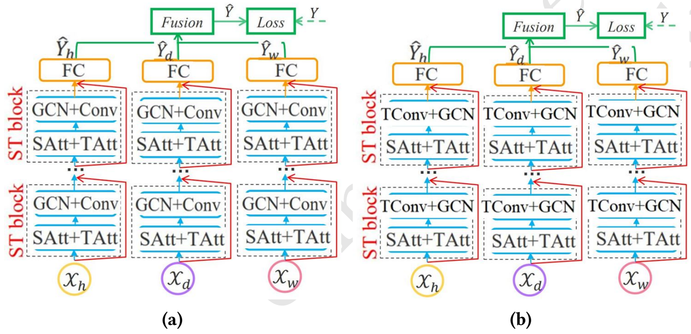

Figure 5: (a) ASTGCN (original) with spatial layers first (b) ASTGCN (ours) with temporal layers first

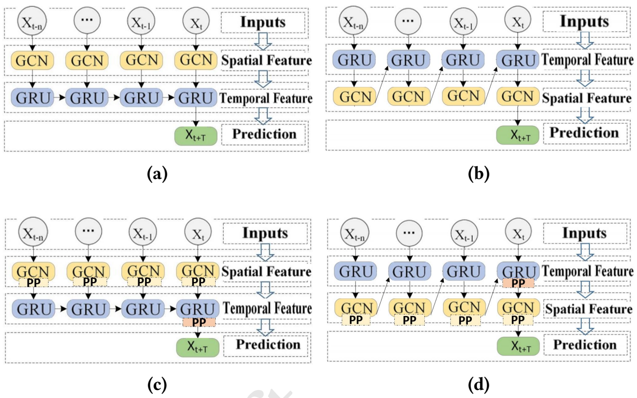

Figure 6: (a) T-GCN (original) with spatial layers first (b) T-GCN (ours) with temporal layers first (c) T-GCN (ours) with spatial layers first and with post-processing layers (d) T-GCN (ours) with temporal layers first and with post-processing layers

for each selected model. We keep the original parameter settings for each model and use the original datasets employed by each corresponding paper for all experiments. We adopt two common metrics to measure the forecasting performance of different models, including Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

5.2.1 *Hyperparameter Settings.*

We use the original set of hyperparameters for each model.

- **DCRNN:** There are two layers of GRU and diffusion convolution (DC) with 2 maximum diffusion steps and 64 hidden units. We train the models by using Adam optimizer [21] to minimize MAE for 100 epochs with batch size as 64. The base learning rate is 0.01 with a decay rate of 0.6 per 10 epochs.
- **ASTGCN:** ASTGCN incorporates both spatial and temporal attention mechanisms. The number of the terms of Chebyshev polynomial is 3 and the kernel size along the temporal dimension is set to 3. All the graph convolution layers and temporal convolution layers use 64 convolution kernels. The batch size is 64 and the learning rate is 0.0001 for training.
- **T-GCN:** The number of hidden units is 64. The learning rate is 0.001, the batch size is 64 and the training epoch is 3000 for the training phase.
- **AGCRN:** The Cheb order is 2 for graph convolution. The embedding dimension for the node adaptive parameter learning is 10. There are two layers of GRU with 64 hidden units. The models are trained using Adam optimizer with an initial learning rate of 0.003 and batch size of 64.
- **STGCN:** The channels of spatial layers and temporal layers in ST-Conv block are 16 and 64, respectively. The kernel size for graph convolution and temporal convolution is set to 3. We train the models by minimizing the MSE using RMSprop for 50 epochs with batch size as 50. The initial learning rate is 103 with a decay rate of 0.7 per 5 epochs.
- **Graph WaveNet:** There are 2 diffusion steps for graph convolution. Dropout rate of 0.3 is applied to the outputs of the graph convolution layer. We randomly initialize node embeddings by a uniform distribution with a size of 10. The models are trained using Adam optimizer with an initial learning rate of 0.001. Eight layers of Graph WaveNet are employed.

5.2.2 *Datasets.*

We employ original datasets for each model.

We collected four sets of data from GitHub released by authors of each selected paper. All the datasets are aggregated every 5 minutes. Thus, each model is tested with its original dataset in its original

Unpublished working draft. Not for distribution.

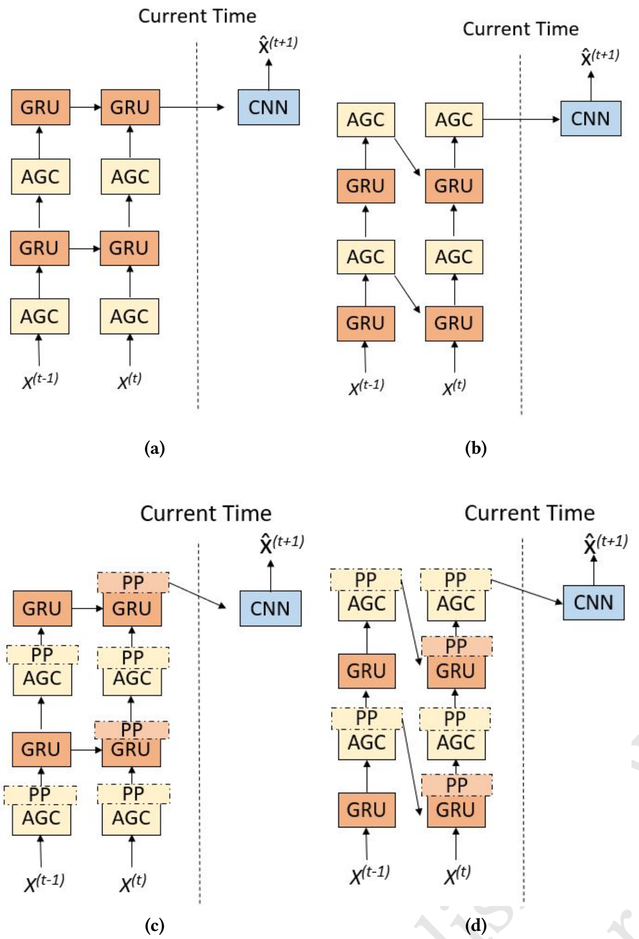

Figure 7: (a) AGCRN (original) with spatial layers first (b) AGCRN (ours) with temporal layers first (c) AGCRN (ours) with spatial layers first and with post-processing layers (d) AGCRN (ours) with temporal layers first and with post-processing layers

Unpublished working draft. Not for distribution.

paper for the best tuned parameters. (1) **METR-LA** [24] is used to test DCRNN and Graph WaveNet. The dataset contains traffic speed collected from 207 loop detectors in the highway of Los Angeles County ranging from Mar 1st 2012 to Mar 31st 2012 for our experiments. 70% of data is used for training, 10% for validation and the remaining 20% for testing.(2) **PeMSD4** [15] is employed to test AGCRN and AGCRN. It contains 307 detectors in San Francisco Bay Area from Jan to Feb in 2016. The first 50 days are used for training while the remaining for testing. (3) **Los-loop** [45] is employed to test T-GCN. It contains traffic speed from Mar 1st 2012 to Mar 7th 2012. 80% of data is used for training while the remaining 20% is for testing. (4) A medium dataset **PeMSD7(M)** [42] among the District 7 of California containing 228 stations from Weekdays of May and June of 2012 is employed to test STGCN. The first month of traffic speed is used as the training set and the rest serves as validation and test set.

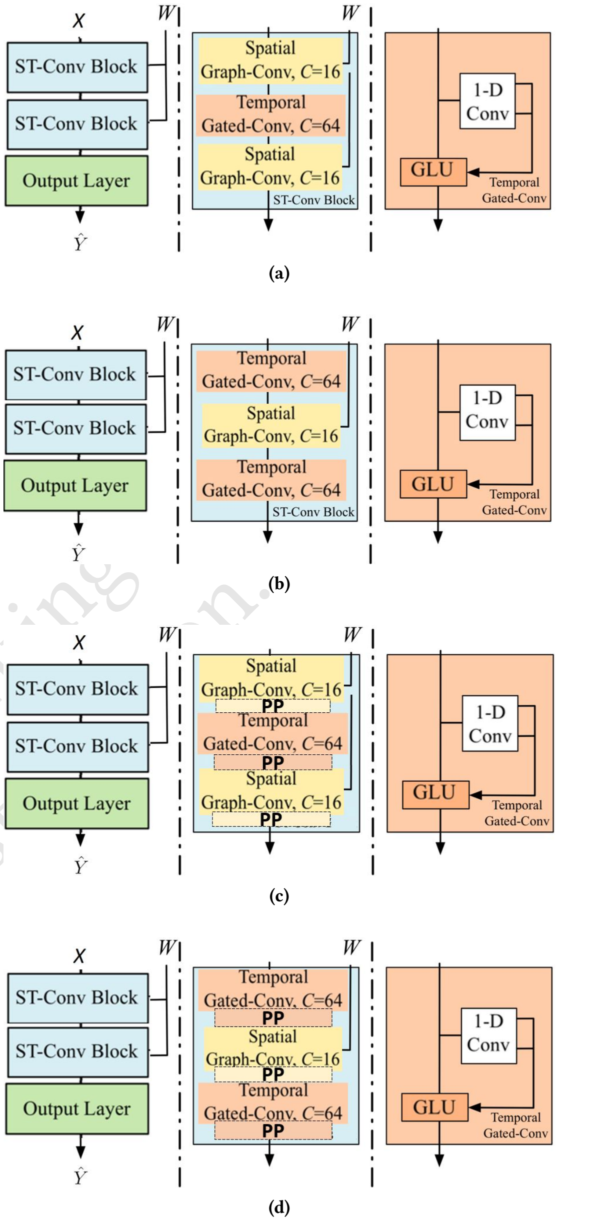

Figure 8: (a) STGCN (ours) with spatial layers first (b) STGCN (original) with temporal layers first (c) STGCN (ours) with spatial layers first and with post-processing layers (d) STGCN (ours) with temporal layers first and with post-processing layers

5.3 Forecasting Performance Comparison

The forecasting performance for each model is shown in Table 1. The model name appended with suffix “s” is the model constructed with spatial layers first and appended with suffix “t” is with temporal

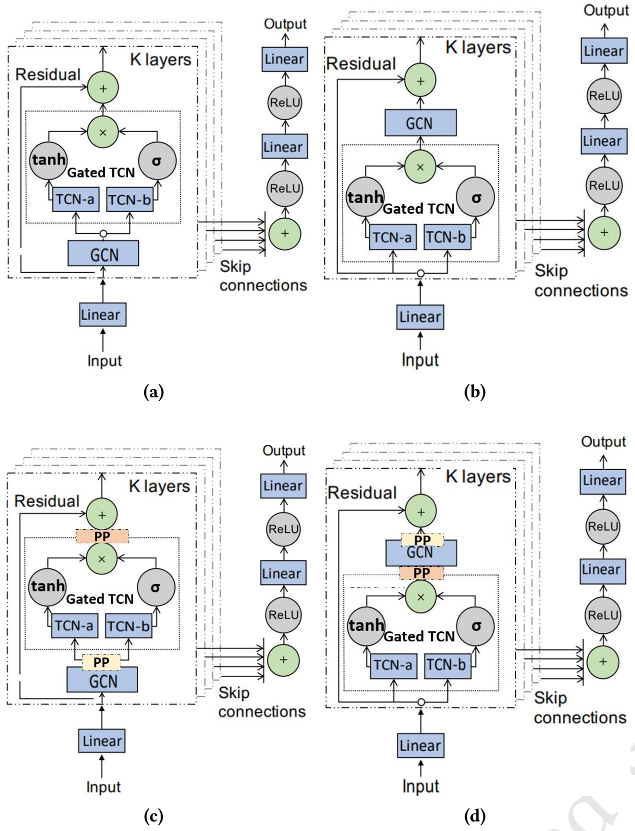

Figure 9: (a) Graph WaveNet (ours) with spatial layers first (b) Graph WaveNet (original) with temporal layers first (c) Graph WaveNet (ours) with spatial layers first and with post-processing layers (d) Graph WaveNet (ours) with temporal layers first and with post-processing layers

layers first. The suffix "att" means that a single head scaled dot-product attention is added to spatial layers or to temporal layers to capture more spatio-temporal dependencies.

DCRNNs constructed with spatial layers first has better performance than those constructed with temporal layers first in terms of MAE and RMSE in the prediction of traffic features in future 15th, 30th and 60th minutes. With the spatial layers constructed first, DCRNN-s performs better than DCRNN-t. With the spatial layers constructed first and the attention layer added, DCRNN-s-att performs better than DCRNN-s in almost all aspects while attention layers cannot make DCRNN-s-att perform better.

ASTGCN is built with both spatial and temporal attention mechanisms before the GCN and TCN layers. We do not further add attention mechanisms to it in our experiments because of the existence of attention mechanisms in the model. With the GCN layer before the TCN layer, ASTGCN-s outperforms its alternative for all prediction tasks.

T-GCN has a clear and simple modeling sequence. GCN is the spatial layer while GRU is the temporal layer. The prediction results of T-GCN-s-att are the best. It may imply that with spatial layers constructed before temporal layers and with attention added, the model is likely to outperform its alternatives.

AGCRN-s-att performs the best in terms of MAE while AGCRN-t performs the best regarding RMSE as the evaluation metric for all the forecasting tasks. The attention mechanisms improve the results of AGCRN-s while they make AGCRN-t perform worse.

The observation for STGCN is similar. STGCN-s-att outperforms other variants except the future 30th-minute forecasting. With spatial layers constructed first, STGCN-s has lower prediction errors than STGCN-t except forecasting 30th-minute traffic features.

Graph WaveNet-s has the best forecasting performance. Graph WaveNet-s-att performs worse than Graph WaveNet-s, meaning that attention layers disturb the forecasting process. WaveNet-s-att performs a bit better than WaveNet-s-att in four out of six results. These results also imply that with spatial constructed first, the forecasting results are likely to be better.

5.4 Discussion of Results

It is observed that for each type of model with spatial layers constructed before temporal layers, it has a relative advantage in forecasting traffic features in almost all time spans from 15 to 60 minutes. With a single head scaled dot-product attention added after both spatial layers and temporal layers, it may improve the forecasting performance. From our experiments, it gives a hint to the modeling sequence of spatio-temporal graph convolutional networks. With spatial layers constructed before temporal layers, spatial layers may better capture complex spatial dependencies among neighbor nodes in those datasets.

There are two effects influencing the forecasting performance of the models and they are originated from the characteristics of datasets: spatial dependencies among neighbor nodes and temporal dependencies for each node for different time periods. A possible reason for the observed results is that the effect of spatial dependencies outweighs that of temporal ones in those datasets. For the STGCNs constructed with spatial layers first, hidden features of each node have incorporated those of neighbors after the GCN operation at each time period. If datasets have strong spatial dependencies, spatial layers can generate meaningful hidden features among neighbor nodes to reflect more accurate traffic conditions. Since it is very likely that the traffic of neighbor nodes at the current timestamp. may affect the traffic of the current node at the next timestamp, considering the spatial layers first somehow already captures some temporal information, which could explain why using the spatial layers first could improve the performance. Afterwards, temporal layers capture the temporal dependencies of those meaningful hidden features from spatial layers for each node and make predictions, which could improve the prediction performance. Another case is that spatial layers may not generate meaningful hidden features for each node if spatial dependencies are not strong enough. Afterwards, temporal layers will process less meaningful hidden features and predictions are made less accurately. If the effect of temporal dependencies is stronger than that of spatial ones, STGCNs constructed with temporal layers first may generate more meaningful hidden features and more accurate predictions.

Unpublished working draft. Not for distribution.

6 CONCLUSION

In this paper, we proposed a new problem for the field of spatio-temporal graph convolutional networks, which states that whether the modeling sequence for spatial layers and temporal layers matters. We tried to give an answer by swapping the sequence of spatial and temporal layers for each of the selected models which have clear modeling sequences. By conducting experiments, for most of the selected models, if constructed with spatial layers before temporal layers and with attention added, they have a higher chance of beating their alternatives. This indicates that STGCNs constructed with spatial layers first may have an advantage over that with temporal layers first. A possible reason is that the effect of spatial dependencies outweighs that of temporal ones in those datasets, leading to a relative advantage in models constructed with spatial layers first. More experiments and mathematical proof help to explore how the modeling sequence of STGCNs matters.

REFERENCES

[1] James Atwood and Don Towsley. "Diffusion-convolutional neural networks". In: *NIPS*. 2016, pp. 1993–2001.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *ICLR*. 2015.

[3] Lei Bai et al. "Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting". In: *NIPS*. 2020.

[4] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*. 2018. arXiv: 1803.01271 [cs.LG].

[5] George E. P. Box and Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. San Francisco : Holden-Day, 1970.

[6] Joan Bruna et al. "Spectral networks and locally connected networks on graphs". In: *ICLR*. 2014.

[7] Sneha Chaudhari et al. *An Attentive Survey of Attention Models*. 2020. arXiv: 1904.02874 [cs.LG].

[8] Jie Chen, Tengfei Ma, and Cao Xiao. "Fastgcn: fast learning with graph convolutional networks via importance sampling". In: *ICLR*. 2018.

[9] Zhitang Chen, Jiayao Wen, and Yanhui Geng. "Predicting future traffic using Hidden Markov Models". In: *2016 IEEE 24th International Conference on Network Protocols (ICNP)*. 2016, pp. 1–6. DOI: 10.1109/ICNP.2016.7785328.

[10] Wei-Lin Chiang et al. "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 257–266.

[11] Junyoung Chung et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[12] Yann N Dauphin et al. "Language modeling with gated convolutional networks". In: *International conference on machine learning*. PMLR. 2017, pp. 933–941.

[13] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering". In: *NIPS*. 2016.

[14] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. "Large-scale learnable graph convolutional networks". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 1416–1424.

[15] Shengnan Guo et al. "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 922–929.

[16] Will Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1024–1034.

[17] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 770–778.

[18] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[19] Dichao Hu. *An Introductory Survey on Attention Mechanisms in NLP Problems*. 2018. arXiv: 1811.05544 [cs.CL].

[20] Rongzhou Huang et al. "LSGCN: Long Short-Term Traffic Prediction with Graph Convolutional Networks". In: *IJCAI*. 2020, pp. 2355–2361.

[21] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *ICLR*. 2014.

[22] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *ICLR*. 2017.

[23] Ruoyu Li et al. "Adaptive graph convolutional neural networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[24] Yaguang Li et al. "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting". In: *International Conference on Learning Representations (ICLR '18)*. 2018.

[25] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. "Effective approaches to attention-based neural machine translation". In: *EMNLP*. 2015.

[26] Li Mengzhang and Zhu Zhanxing. "Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting". In: *Proceedings of the AAAI conference on artificial intelligence*. 2021.

[27] Alessio Micheli. "Neural network for graphs: A contextual constructive approach". In: *IEEE Transactions on Neural Networks* 20.3 (2009), pp. 498–511.

[28] Cheonbok Park et al. "ST-GRAT: A Novel Spatio-temporal Graph Attention Networks for Accurately Forecasting Dynamically Changing Road Speed". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 1215–1224.

[29] Amit Roy et al. "Unified Spatio-Temporal Modeling for Traffic Forecasting using Graph Neural Network". In: *IJCNN*. 2021.

[30] C. Song et al. "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 2020, 914–921. DOI: 10.1609/aaai.v34i01.5438.

[31] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks". In: *NIPS*. 2014.

[32] Ashish Vaswani et al. "Attention is all you need". In: *NIPS*. 2017, 5998–6008.

[33] Petar Veličković et al. "Graph attention networks". In: *ICLR*. 2017.

[34] Chun-Hsin Wu, Jan-Ming Ho, and Der-Tsai Lee. "Travel-time prediction with support vector regression". In: *IEEE transactions on intelligent transportation systems* 5.4 (2004), pp. 276–281.

[35] Zonghan Wu et al. "A Comprehensive Survey on Graph Neural Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (2021), pp. 4–24. DOI: 10.1109/TNNLS.2020.2978386.

[36] Zonghan Wu et al. "Connecting the dots: Multivariate time series forecasting with graph neural networks". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 753–763.

[37] Zonghan Wu et al. "Graph WaveNet for Deep Spatial-Temporal Graph Modeling". In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*. 2019.

[38] Keyulu Xu et al. "How Powerful are Graph Neural Networks?" In: *ICLR*. 2019. URL: https://openreview.net/forum?id=ryGs6iA5Km.

[39] Keyulu Xu et al. "Representation Learning on Graphs with Jumping Knowledge Networks". In: *ICML*. 2018.

[40] Sijie Yan, Yuanjun Xiong, and Dahua Lin. "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition". In: *AAAI*. 2018.

[41] Xueyan Yin et al. *A Comprehensive Survey on Traffic Prediction*. 2021. arXiv: 2004.08555v1 [eess.SP].

[42] Bing Yu, Haoteng Yin, and Zhanxing Zhu. "Spatio-temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*. 2018.

[43] H. Yuan and G. Li. "A Survey of Traffic Prediction: from Spatio-Temporal Data to Intelligent Transportation". In: *Data Sci. Eng.* (2021), 63–85. DOI: https://doi.org/10.1007/s41019-020-00151-z.

[44] Ziwei Zhang, Peng Cui, and Wenwu Zhu. "Deep Learning on Graphs: A Survey". In: *CoRR* abs/1812.04202 (2018).

[45] Ling Zhao et al. "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 21.9 (2020), 3848–3858. ISSN: 1558-0016. DOI: 10.1109/tits.2019.2935152. URL: http://dx.doi.org/10.1109/TITS.2019.2935152.

[46] Chuanpan Zheng et al. "GMAN: A Graph Multi-Attention Network for Traffic Prediction". In: *AAAI*. 2020, pp. 1234–1241.

[47] Jie Zhou et al. "Graph neural networks: A review of methods and applications". In: *AI Open* 1 (2020), pp. 57–81. ISSN: 2666-6510. DOI: https://doi.org/10.1016/j.aiopen.2021.01.001. URL: https://www.sciencedirect.com/science/article/pii/S2666651021000012.

[48] Chenyi Zhuang and Qiang Ma. "Dual graph convolutional networks for graph-based semi-supervised classification". In: *Proceedings of the 2018 World Wide Web Conference*. 2018, pp. 499–508.

Unpublished working draft. Not for distribution.