Spatio-Temporal Pivotal Graph Neural Networks for Traffic Flow Forecasting

Weiyang Kong[1], **Ziyu Guo**[1], **Yubao Liu**[1, 2]
[1]Sun Yat-Sen University,Guangzhou, China
[2]Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China
kongwy3@mail2.sysu.edu.cn, guozy28@mail2.sysu.edu.cn, liuyubao@mail.sysu.edu.cn

Abstract

Traffic flow forecasting is a classical spatio-temporal data mining problem with many real-world applications. Recently, various methods based on Graph Neural Networks (GNN) have been proposed for the problem and achieved impressive prediction performance. However, we argue that the majority of existing methods disregarding the importance of certain nodes (referred to as pivotal nodes) that naturally exhibit extensive connections with multiple other nodes. Predicting on pivotal nodes poses a challenge due to their complex spatio-temporal dependencies compared to other nodes. In this paper, we propose a novel GNN-based method called Spatio-Temporal Pivotal Graph Neural Networks (STPGNN) to address the above limitation. We introduce a pivotal node identification module for identifying pivotal nodes. We propose a novel pivotal graph convolution module, enabling precise capture of spatio-temporal dependencies centered around pivotal nodes. Moreover, we propose a parallel framework capable of extracting spatio-temporal traffic features on both pivotal and non-pivotal nodes. Experiments on seven real-world traffic datasets verify our proposed method's effectiveness and efficiency compared to state-of-the-art baselines.

Introduction

Traffic flow forecasting is a classical spatio-temporal data mining problem. The problem has been found useful in many real-world applications such as intelligent route planning, dynamic traffic management, smart location-based applications, and so on (Wu and Tan 2016). The purpose of the problem is to predict the traffic flows of several future times based on the historical traffic observations (e.g. recorded by sensors of traffic networks). The challenges of traffic prediction mainly stem from the intricate spatio-temporal correlations between sensors. With the rapid advancement of neural networks, deep learning methods have become capable of capturing complex spatio-temporal features (Yu, Yin, and Zhu 2018; Yan, Xiong, and Lin 2018; Seo et al. 2018). Graph Neural Networks (GNNs) have shown promising results in modeling transportation networks (Li et al. 2018; Wu et al. 2019; Li and Zhu 2021; Han et al. 2021; Lan et al. 2022; Wu et al. 2022). Typically, GNNs use graph nodes to represent sensors in transportation networks, while the edges between the nodes signify the connections between the various sensors (Tedjopurnomo et al. 2022; Jin et al. 2023).

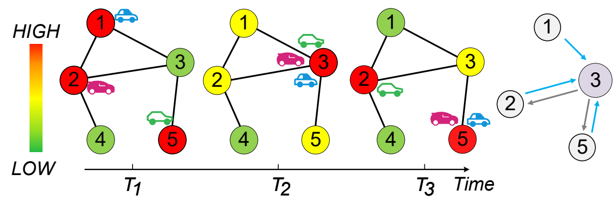

Figure 1: A toy traffic network example with 5 nodes. The nodes with different colors denote their traffic flow changes.

Although significant improvements have been made in traffic forecasting with spatio-temporal data, we argue that the majority of existing methods overlook a common phenomenon in traffic: certain nodes in traffic network, either due to their geographical location (in the city center) or characteristics (next to large interchanges), exhibit extensive connections with multiple other nodes, forming a complex spatio-temporal network centered around these nodes. In this paper we refer to these nodes as pivotal nodes. For example, a toy traffic network comprising five nodes (sensors) is given in Figure 1. The traffic flow is first aggregated from nodes 1, 2, and 5 to node 3 during time interval $[T_1,T_2]$. Then, the traffic flow is subsequently distributed from node 3 to nodes 2 and 5 during time interval $[T_2,T_3]$. Compared to other nodes, node 3 exhibits stronger abilities in both aggregating and distributing traffic flow. We can select the node 3 as a pivotal node. Such rich inter-node spatio-temporal relationships endow the pivotal nodes with intricate interdependencies and potential multiplicity of roles within the entire network structure, making them challenging to accurately predict. The existing methods fell short in accurately represent the spatio-temporal dependencies centered around these pivotal nodes. Most existing methods learned spatial and temporal dependencies separately, which fail to effectively capture the accurate spatio-temporal dependencies. Few methods (Song et al. 2020; Li and Zhu 2021) tried to synchronously capture spatio-temporal relationships. But these methods are constrained to represent the simplified spatio-temporal dependencies among all nodes since accurately considering the spatio-temporal dependencies among all nodes leads to a high time and space complexity. For example, STSGCN (Song et al. 2020) employed a one-hot adjacency matrix to represent local spatio-temporal dependencies.

In this work, we focus on modeling the traffic phenomenon around pivotal nodes, which involve the following two challenges: The first challenge is how to identify pivotal nodes, and the second challenge lies in how to precisely extract spatio-temporal features at these pivotal nodes. To tackle the above challenges, we propose a novel GNN-based method called Spatial-Temporal Pivotal Graph Neural Networks (STPGNN). For the first challenge, we propose a Pivotal node Identification Module (PIM), which decompose the process of traffic flow propagation into two components: aggregation and distribution. We design a scoring mechanism that evaluates the aggregation and distribution capabilities of each node, thus identifying pivotal nodes based on the scoring results. For the second challenge, we construct a pivotal graph with the identified pivotal nodes, and propose a Pivotal Graph Convolution Module (PGCM) to capture the intricate spatio-temporal dependencies around pivotal nodes with the pivotal graph. Similar with the existing method, we adopt graph convolution and linear unit to capture the spatial and temporal dependencies on non-pivotal nodes respectively. A parallel framework is designed to fuse the results from the two graph convolution. By integrating pivotal nodes, our model effectively establishes extensive correlations, and significantly improves the model's inferential capabilities for accurate prediction tasks. The main contributions of the work are as follows:

- We address the traffic prediction problem of certain pivotal nodes with complex spatio-temporal dependencies. We propose a novel method for identifying the pivotal nodes in traffic network. We construct a pivotal graph and introduce a pivotal graph convolution module to accurately capture the spatio-temporal dependencies on pivotal nodes.
- A parallel framework is proposed to capture the spatio-temporal dependencies among all nodes. This framework can integrate the spatio-temporal features captured on both pivotal and non-pivotal nodes.
- We conducted experiments on seven real-world datasets from different sources, and the results demonstrate the effectiveness and efficiency of our approach.

Related Work

Traffic Flow Forecasting

Traffic flow forecasting is a spatio-temporal data forecasting problem. Similar problems include shared bicycle demand forecasting, bus and taxi demand forecasting, crowd flow forecasting, etc (Li et al. 2015; Chai, Wang, and Yang 2018; Hu et al. 2021; Zhao et al. 2019). Traditional statistical methods like ARIMA (Williams and Hoel 2003) and SVM (Drucker et al. 1996) are widely used in time series prediction. Since they ignore the spatial information, it is difficult for them to handle complex spatio-temporal data. Recently, deep learning methods are often used for handling the non-linearity and complexity of traffic data. Convolutional Neural Networks (CNNs) have been regularly applied to traffic flow prediction (Zhang et al. 2016; Zhang, Zheng, and Qi 2017; Ouyang et al. 2022; Yao et al. 2018). Each cell in the set records the number of vehicles passing in that cell in a time period. In order to capture the spatial correlations between the grid regions, methods with CNNs model the traffic flow readings as an image, and similar techniques developed for image recognition can be easily applied (Tedjopurnomo et al. 2022). For better investigation of sequence data, Recurrent Neural Networks (RNNs) were proposed. With the memorization capability to sequences, methods with RNNs were soon applied to traffic flow forecasting (Ye et al. 2019; Shi et al. 2015; Zonoozi et al. 2018). More recently, methods with Graph Neural Networks are proposed to handle spatio-temporal correlations in traffic flow data and obtain impressive results (Pan et al. 2022; Shen et al. 2022; Sun et al. 2022; Guo et al. 2022). DCRNN (Li et al. 2018) proposes a bi-directional process of diffusion to simulate actual road conditions, and uses gated recurrent units to capture temporal information. ASTGCN (Guo et al. 2019) uses two attention layers to capture the dynamics of spatial dependencies and temporal correlations. STGCN, Graph WaveNet, LSGCN and AGCRN (Yu, Yin, and Zhu 2018; Wu et al. 2019; Huang et al. 2021; Bai et al. 2017) follow and improve the GCN methods to extract spatio-temporal information. In particular, Graph WaveNet designs a self-adaptive matrix to take the influence between nodes and their neighbors into account while LSGCN uses an attention layer to do similar work. STSGCN, STFGNN and STGODE (Song et al. 2020; Li and Zhu 2021; Fang et al. 2021) propose GCN methods based on similar characteristics that can capture spatio-temporal information synchronously. MTGNN (Wu et al. 2020) proposes a graph learning module that constructs a dynamic graph by computing the similarity between learnable node embeddings. DMSTGCN, TPGNN, and DSTAGNN (Han et al. 2021; Wu et al. 2022; Lan et al. 2022) capture the spatio-temporal characteristics by constructing dynamic associations between nodes. SGP (Cini et al. 2023)propose a scalable architecture that exploits an efficient encoding of both temporal and spatial dynamics. However, most existing methods have not taken into account the significance of certain pivotal nodes in traffic network, thereby failing to accurately extract the spatio-temporal features of these nodes. Consequently, this limitation hampers the improvement of the performance of the model.

Graph Neural Networks

Graph Neural networks (GNN) are originally designed to study the structure of the graph and are widely used in node embedding (Pan et al. 2019), node classification (Kipf and Welling 2017), etc. In recent years, to model the graph structures in transportation systems, GNNs such as graph convolutional and graph attention networks have been used for the problem and achieved SOTA performance. Bruna *et al.* (Bruna et al. 2014) proposes GCN based on the spectral graph theory, which can use a filter to smooth the input graph signal and aggregate the information of neighbor nodes. Defferrard *et al.* (Defferrard, Bresson, and Vandergheynst 2016) proposes a Chebyshev extension to reduce the complexity of laplacians computation of GCN. Kipf and

Welling *et al.* (Kipf and Welling 2017) simplifies the Chebyshev extension method. Velickovic *et al.* (Velickovic et al. 2018) proposes GAT, which introduces attention mechanisms into graph to update the weight of a node's neighbors. Monti *et al.* (Monti et al. 2017) applies Gaussian kernels to learn the weight of a node's neighbors. Hamilton, Ying, and Leskovec *et al.* (Hamilton, Ying, and Leskovec 2017) proposes GraphSAGE, which aggregates the features of nodes' neighbors and themselves through a fixed sampling method.

Preliminary

Traffic network is represented by an undirected graph $G = (V, E)$, where V is the set of nodes (sensors), $N = |V|$ denotes the number of nodes, and E is the set of edges between two nodes. The adjacency matrix derived from G is denoted by $U \in \mathbb{R}^{N\times N}$. U is obtained based on the Euclidean distance between nodes. In our problem, we assume that each node records its traffic flow data as graph signal. A graph signal is $X^t \in \mathbb{R}^N$, where t denotes the t-th time step. The graph signal represents the traffic flow values at the t-th time step. Given a traffic network G and its historical S step graph signal matrix $X^{1:S} = (X^1, X^2, ..., X^S) \in \mathbb{R}^{N\times S}$, our problem is to predict its next T step graph signals, namely $X^{S+1:S+T} = (X^{S+1}, X^{S+2}, ..., X^{S+T}) \in \mathbb{R}^{N\times T}$. We formulate the problem as finding a function $\mathcal{F}$ to forecast the next T steps data based on the past S steps historical data:

$$(X^{S+1}, X^{S+2}, ..., X^{S+T}) = \mathcal{F}((X^1, X^2, ..., X^S)). \quad (1)$$

Proposed Model

As shown in Figure 2, the main framework of our proposed method consists of an input layer, a Pivotal node Identification Module (PIM), M stacked spatio-temporal layers (ST-Layer) and an output layer. The input layer contains one linear units convolution layer. The PIM identifies pivotal nodes with a scoring function and generates a pivotal graph accordingly. For each spatio-temporal layer, it incorporates parallel structures, namely the pivotal graph convolution module (PGCM) and the graph convolution module with linear unit. The pivotal graph convolution module is designed for feature extraction on pivotal nodes. Meanwhile, in the second branch, the graph convolution module with linear unit is mainly employed for feature extraction on non-pivotal nodes. The output of each spatio-temporal layer is concatenated and then sent to the output layer. The output layer contains two activation layers and two linear units convolution layers.

Pivotal node Identification Module

In traffic network, certain nodes, referred to as pivotal nodes, possess more complex node relationships. We introduce the definition of pivotal nodes and the method devised to identify them. Intuitively, pivotal nodes are expected to exhibit stronger capabilities in both aggregating traffic flow from other nodes and distributing it to them. Therefore, our initial focus lies in quantifying these capabilities. We consider the following approach: Let $H \in \mathbb{R}^{N\times S}$ be the input, where $H_i \in \mathbb{R}^S$ represents the traffic features of node i during the S time period, then we calculate $E = (e_{i,j}) \in \mathbb{R}^{N\times N}$ as follow:

$$e_{i,j} = \frac{\sum_{k=1+d}^{S}(H_{i,k}H_{j,k-d}^{\top})w_{i,j}}{\sqrt{\sum_{k=1+d}^{S}(H_{i,k}^2)}\cdot\sqrt{\sum_{k=1}^{S-d}(H_{j,k}^2)}}, \quad (2)$$

here, k denotes the index of time step and d represents the time required for traffic flow to propagate between nodes. In this paper, we set $d = 1$ which corresponds to 5 minutes in the context of the commonly used public dataset PEMS (detailed in the experimental section). $W = (w_{ij}) \in \mathbb{N}\times\mathbb{N}$ is a trainable parameter matrix. The term $e_{i,j}$ is obtained by calculating the similarity between the traffic flow at node i during the current time step and this at j during the previous d time step, where higher similarity indicates a greater influence of the traffic features propagate from node j to node i. The entries in i-th row of matrix E represents the node dependencies from other nodes to i with time d, thereby the summation of each row in matrix E yields a representation of the aggregation capability of each node in traffic graph. Similarly, the summation of the i-th column of matrix E can represent the distribution capability of node i. Henceforth, we can propose a scoring function capable of measuring the aggregation and distribution capabilities of nodes i with the generated E as follow:

$$Score(i) = \sum_{j=1}^{N}(e_{i,j} + e_{j,i}). \quad (3)$$

The set of the pivotal nodes C is defined as follows:

$$C = \{i|Score(i) \in TopK(Score)\}, \quad (4)$$

where, K is a hyperparameter to control the number of pivotal nodes. We empirically set $K = \frac{1}{5}N$, and we further analyze the hyperparameter setting in the experimental section.

With the identification of pivotal nodes established, the subsequent step involves constructing a graph that incorporates spatio-temporal dependencies of pivotal nodes. For pivotal nodes, it is reasonable to use the corresponding entries in matrix E as the spatio-temporal dependencies between nodes at adjacent time steps. However, for non-pivotal nodes, it is no longer suitable to use E due to their lack of strong aggregation and distribution capabilities. Hence, the adjacency matrix $A = (a_{i,j}) \in \mathbb{R}^{N\times N}$ is defined as follows:

$$a_{i,j} = \begin{cases} sigmoid(e_{i,j}), & \text{if } i \in C \vee j \in C \\ u_{i,j}, & \text{others} \end{cases}, \quad (5)$$

here, u denotes the entry in the given adjacency matrix U, and the $sigmoid$ function is used to normalize the entries in matrix E. Figure 2 (a) depicts a pivotal graph with five nodes in Figure 1.

Pivotal Graph Convolution Module

The Graph Convolutional Network is a powerful method to extract nodes' features with its neighbors' information. Most existing methods employed it to capture spatial relationships between nodes and were unable to synchronously capture the spatio-temporal dependencies. Few methods such

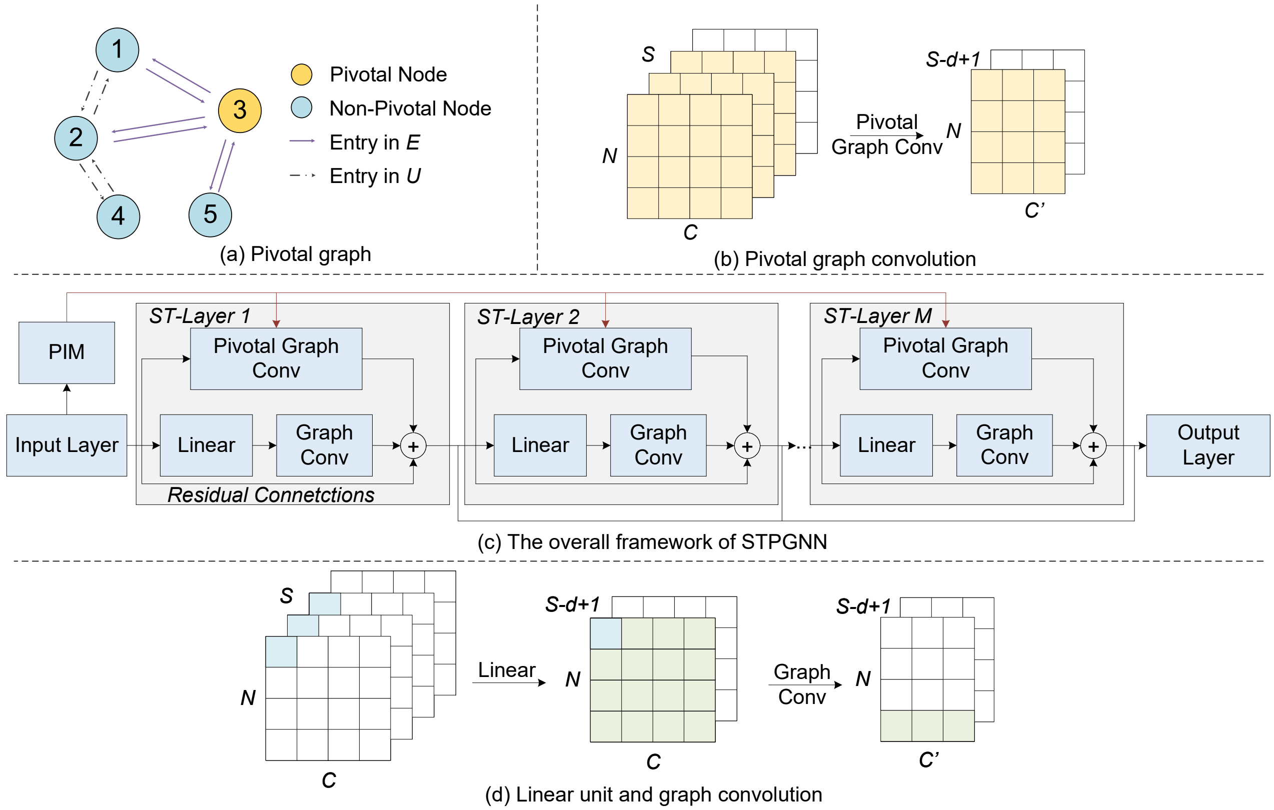

Figure 2: Detailed framework of STPGNN.

as STSGCN (Song et al. 2020) designed synchronous graph convolution to capture the spatio-temporal dependencies, but these methods can only consider simplified and local spatio-temporal dependencies with a one-hot adjacency matrix since considering the spatio-temporal dependencies among all nodes leads inevitably to the problem of high time and space complexity with $O(T_d^2 N^2)$. where T_d is the number of time steps span of the spatio-temporal dependencies. In this paper, we can focus on the spatio-temporal dependencies around pivotal nodes, which significantly reduces the complexity to $O(T_d K N)$, where K is the number of pivotal nodes. We build a pivotal graph convolution module where the convolutional operation is designed to extract pivotal nodes' features with matrix A. The graph convolutional operation is defined in the vertex domain, which means it can fuse node features with its neighbors without requiring spectral filter such as graph Laplacian. Let $H^l \in \mathbb{R}^{S\times N\times C}$ be the input graph signal of the l-th ST-Layer where C is the number of channel. The pivotal convolutional operation can be formulated as follow:

$$H^l = [H_1^l, H_2^l, ..., H_{S-d+1}^l], \tag{6}$$

$$H_i^{l+1} = \sum_{k=1}^{d} (\sigma(AH_{[i,:,:]}^l W + B)), \tag{7}$$

here, $H^{l+1} \in \mathbb{R}^{(S-d+1)\times N\times C'}$ represent the output. C' is the number of output channels. d denotes the length of kernel size in the temporal dimension, and k denotes the index of kernel. A is the adjacency matrix of pivotal graph. σ denotes the activation function, such as $sigmoid$ or $tanh$. $W \in \mathbb{R}^{C\times C'}$ and $B \in \mathbb{R}^{N\times C'}$ are all model parameters. The output H^{l+1} incorporate spatio-temporal features from d time steps. Figure 2 (b) illustrates the computation process of the pivotal graph convolutional operation.

A Parallel Structure for Non-pivotal Nodes

Although PGCM addresses the spatio-temporal dependencies on pivotal nodes, we still need to consider the extraction of features on non-pivotal nodes. Similar to existing methods such as GraphWavenet (Wu et al. 2019), We adopt graph convolution and linear convolution to capture the spatial and temporal correlations among non-pivotal nodes respectively, and we propose a parallel structure to simultaneously extract features from both pivotal and non-pivotal nodes. The format of graph convolution is as follows:

$$O = \sum_{q=0}^{Q} U^q X L_q, \tag{8}$$

where U^q represents the $q-th$ power series of the diffusion matrix U, $X \in \mathbb{R}^{N\times S}$ denotes the input signals, $O \in \mathbb{R}^{N\times S}$ denotes the output, and $L_q \in \mathbb{R}^{S\times S}$ denotes the matrices of learnable parameters. The linear unit (Yu and Koltun 2016) is designed to capture long-range behaviors of temporal features. Mathematically, given an input $x \in \mathbb{R}^{\mathbb{T}}$ and a filter

Dataset	Node	Samples	Sample Rate	Data Type
PEMS-03	358	26208	5min	Traffic Flow
PEMS-04	307	16992	5min	Traffic Flow
PEMS-07	883	28224	5min	Traffic Flow
PEMS-08	170	17856	5min	Traffic Flow
England	314	17353	15min	Traffic Flow
TaxiBJ	1024	5596	30min	Taxi GPS
PEMS-BAY	325	52116	5min	Traffic Speed

Table 1: Datasets description

$f \in \mathbb{R}^{\mathbb{K}}$, the linear unit operation of x with f at time step t is represented as

$$x \star f(t) = \sum_{s=0}^{K-1} f(s)x(t - d \times s), \tag{9}$$

where $\star$ represents the convolution operation. Figure 2 (d) illustrate how graph convolution and linear unit collaborate with each other.

Loss Function

Mean absolute error (MAE) is chosen as loss function. The objective function is shown below:

$$L(\hat{X}^{(t+1):(t+T)}; \Theta) = \frac{1}{TN} \sum_{i=1}^{i=T} \sum_{j=1}^{j=N} |\hat{X}_j^{(t+i)} - X_j^{(t+i)}|, \tag{10}$$

here $\hat{X}^{(t+1):(t+T)}$ is the prediction value, Θ denotes all learnable model parameters, and $X_j^{(t+i)}$ denotes the ground truth.

Experiments

In this section, we evaluated our proposed model by empirically examining on seven real-world datasets with the state-of-the-art models for traffic forecasting. We not only examine on traffic flow datasets but also choose traffic speed and GPS datasets to showcase the generality of our method in addressing traffic prediction problem.

Datasets and Baseline Methods

We conduct experiments on seven widely used real-world public traffic datasets from: (1)TaxiBJ (Junbo Zhang 2017)(2)England[1](3)PEMS[2](PEMS03, PEMS04, PEMS07, PEMS08, and PEMS-BAY). A brief description are given in Table 1. The baselines are ARIMA (Williams and Hoel 2003), DCRNN (Li et al. 2018), GWNet (Wu et al. 2019), STSGCN (Song et al. 2020), MTGNN (Wu et al. 2020), DMSTGCN (Han et al. 2021), DSTAGNN (Lan et al. 2022), TPGNN (Wu et al. 2022), and SGP (Cini et al. 2023).

Experiment Settings

To make a fair comparison, we follow existing experimental settings, and use the same evaluation metrics as the original publications in each dataset. All the tests adopt 60 minutes as the history time window except for the England dataset, for which we employed a 3-hour window. The forecasting time window is set to be the same as history time window. We adopt Mean Absolute Errors (MAE), Mean Absolute Percentage Errors (MAPE), and Root Mean Squared Errors (RMSE) to measure the performance of different methods. Every experiment is repeated 5 times and the average performance is reported.

Experiment Results

Table 2 summarizes the experimental results. It shows the comparison of different approaches for the traffic flow forecasting tasks. From the table 2, we can see our method outperforms other baseline methods on seven datasets except that our method performs sub-optimally in metric MAPE on the PEMS-BAY dataset. We can easily observe that the traditional statistical methods such as ARIMA often have poor performance since they cannot efficiently handle the complex spatio-temporal data. By utilizing the multi-head attention mechanism, DSTAGNN, has achieved sub-optimal performance on most datasets. Constructing dynamic graph structures and incorporating auxiliary information, TPGNN and DMSTGNN have achieved sub-optimal results on the remaining datasets. In comparison to these suboptimal methods, our model has shown significant improvements on the TAXI-BJ and PEMS08 datasets, with a **14.15%** increase in MAPE on the TAXI-BJ dataset and a **10.32**% increase on the PEMS08 dataset. On the PEMS03, TAXI-BJ, PEMS08, and PEMS-BAY datasets, our method has outperformed the suboptimal results by 9.05%, 7.60%, 6.98% and 6.21%, in terms of MAE, respectively. Moreover, on the TAXI-BJ and PEMS04 datasets, our method has achieved 6.54%, and 6.03% improvement in terms of RMSE, respectively. In general, the results in Table 2 verify the effectiveness of our model.

In the pivotal nodes identify module, we employ a scoring mechanism to identify pivotal nodes, which includes a $TopK$ function where K is a hyperparameter controlling the number of pivotal nodes. We conduct experiments to validate the appropriate value for K. As shown in Table 3, we represent the performance of STPGNN under different size of K on PEMS07 and PEMS08 datasets. Initially, the performance of model consistently improves as K increases. However, after surpassing a certain threshold, the performance gradually declined. This threshold is approximately one-fifth of the total number of nodes.

Component Analysis

To further verify effectiveness of different modules of STPGNN, we conduct experiments on PEMS08 (as shown in Figure 3 (a)) and England (as shown in Figure 3 (b)). The results on other datasets show similar results. In particular, we design three variants of the STPGNN model: (1) $RemPG$: This variable removes the pivotal node identification module and randomly selects nodes as pivotal nodes. (2) $RemPGCN$: This variable removes the pivotal graph convolution module and solely utilizes graph convolution module and linear units to extract features from all nodes. (3)

[1]http://tris.highwaysengland.co.uk/detail/trafficflowdata

[2]http://pems.dot.ca.gov/

Datasets	Metric	ARIMA	DCRNN	GWNet	STSGCN	MTGNN	DMSTGCN	DSTAGNN	TPGNN	SGP	STPGNN	Improve(%)
PEMS03	MAE	26.33	18.18	19.85	17.48	17.23	16.82	15.67	16.88	15.82	**14.37**	9.05
	MAPE(%)	22.90	18.18	19.85	16.78	17.35	16.71	14.74	16.53	15.74	**14.23**	3.58
	RMSE	33.05	30.31	32.94	29.21	25.89	25.81	27.21	25.78	25.92	**24.62**	4.71
PEMS04	MAE	28.55	24.70	25.45	21.19	19.98	19.75	19.53	19.63	19.57	**18.34**	6.49
	MAPE(%)	19.55	17.12	17.29	13.90	14.13	13.91	12.97	13.04	13.13	**12.49**	3.84
	RMSE	40.36	38.12	39.70	33.65	31.92	31.43	31.46	31.44	31.52	**29.64**	6.03
PEMS07	MAE	33.89	25.30	26.85	24.26	23.92	23.73	21.42	23.52	23.66	**20.52**	4.39
	MAPE(%)	17.60	11.66	12.12	10.21	12.43	12.21	9.08	11.20	9.92	**8.75**	3.77
	RMSE	46.38	38.58	42.78	29.03	35.86	36.01	34.51	35.20	34.97	**33.38**	3.39
PEMS08	MAE	31.23	17.86	19.13	17.13	15.03	14.87	15.67	14.92	14.96	**13.90**	6.98
	MAPE(%)	19.25	11.45	12.68	10.96	10.23	10.11	9.94	10.11	10.27	**9.01**	10.32
	RMSE	33.47	27.83	31.05	26.80	23.89	23.86	24.77	23.76	24.03	**23.05**	3.08
England	MAE	4.23	3.59	3.12	3.02	3.03	2.98	2.97	3.07	3.05	**2.87**	3.48
	MAPE(%)	5.72	4.90	4.53	4.48	4.42	4.37	4.39	4.41	4.52	**4.19**	4.30
	RMSE	7.68	7.42	7.17	7.03	7.05	6.99	7.02	7.11	7.25	**6.81**	2.64
TaxiBJ	MAE	25.32	19.81	18.77	17.69	18.07	17.59	16.85	17.23	17.03	**15.66**	7.60
	MAPE(%)	59.35	34.19	33.52	31.04	31.98	31.79	29.76	30.89	30.12	**26.07**	14.15
	RMSE	51.54	31.68	30.66	28.30	29.97	27.71	27.53	28.19	27.86	**25.84**	6.54
PEMS-BAY	MAE	2.33	1.74	1.64	1.67	1.68	1.78	1.71	1.65	1.54	**1.45**	6.21
	MAPE(%)	5.40	3.90	3.85	3.75	3.69	4.10	3.60	3.47	**3.44**	3.57	*
	RMSE	4.76	3.97	3.75	3.82	3.74	3.97	3.71	3.65	3.52	**3.46**	1.73

Table 2: Performance comparison of different approaches on seven datasets.

Dataset	PEMS07(with 883 sensors)					
K	100	125	150	175	200	225
MAE	21.32	21.13	20.85	**20.52**	20.56	20.63
MAPE(%)	9.32	9.13	8.86	**8.75**	8.79	8.91
RMSE	34.25	33.96	33.51	**33.38**	33.42	33.63
Dataset	PEMS08(with 170 sensors)					
K	5	15	25	35	45	55
MAE	15.21	14.63	14.02	**13.77**	13.96	14.38
MAPE(%)	10.17	9.53	9.22	**8.96**	9.31	9.74
RMSE	23.97	23.25	23.03	**22.90**	23.11	23.68

Table 3: The performance of STPGNN with different K.

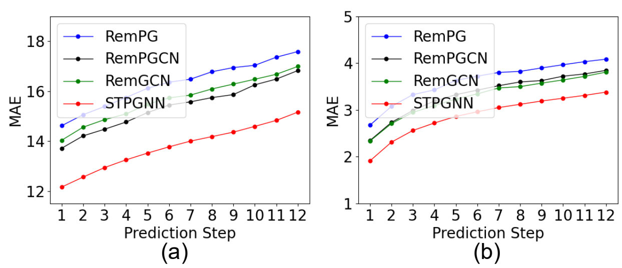

Figure 3: Performance comparison of different variant on PEMS08 and England datasets.

Figure 4: An illustrative case study conducted on the PEMS-BAY dataset. (a) The geographical positions of all nodes. Pivotal node 90 has been highlighted in red. (b) Node 90 is situated on an overpass. (c) The top 25 nodes aggregating towards node 90. (d) Similar to (c), the top 25 nodes which are distributed from node 90.

$RemGCN$: This variable removes the graph convolution module and linear units and exclusively employs the pivotal graph convolution module to construct the ST-Layers. From the results in Figure 3, we have the following findings. When removing the pivotal node identification module, the method fails to capture the pivotal nodes, resulting in a significant decrease in prediction accuracy. The performance of the model declines after the removal of PGCM or GCN. However, comparing $RemPGCN$ and $RemGCN$ reveals that the effectiveness of using $PGCN$ varies across different datasets. We speculate that this might be due to the more traffic flow propagation around pivotal nodes in the England dataset. In summary, identifying pivotal nodes and extracting spatio-temporal dependencies on these nodes can substantially enhance the precision of model predictions.

Effect of the Pivotal Nodes

We present a case study to gain a better understanding of our proposed approach. we investigate the pivotal graph from our method trained on the PEMS-BAY dataset to verify if our model has truly identified the pivotal nodes. We have plotted all the sensors in the PEMS-BAY dataset on a map,

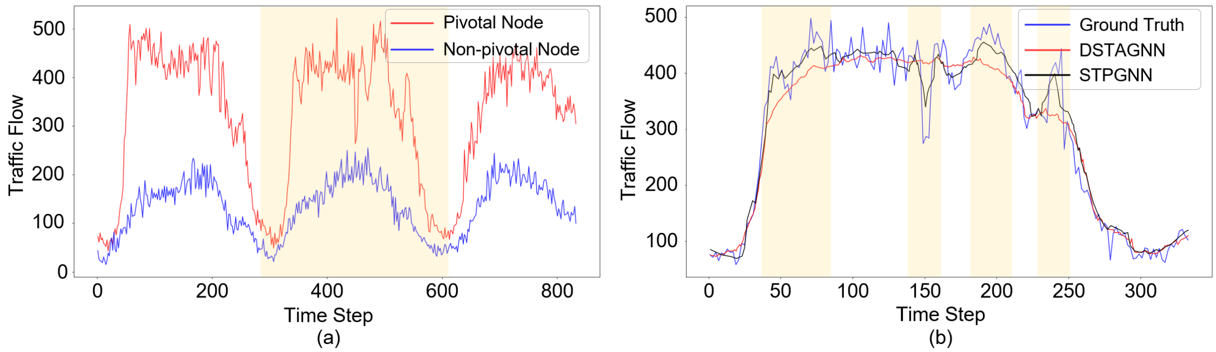

Figure 5: An illustrative case study conducted on the PEMS08 dataset. (a) The ground truth for pivotal and non-pivotal nodes. (b) The prediction results of STPGNN and DSTAGNN on pivotal node 0.

as shown in Figure 4 (a), with node 90 highlighted in red. In the scoring mechanism, node 90 has received the highest score due to its exceptional performance in both aggregation and distribution capabilities. Figure 4 (b) illustrates the precise location of node 90. As evident, node 90 is located on a overpass with high traffic flow, which validates the suitability of our chosen pivotal nodes in real-world scenarios. In Figure 4 (c) and (d), we present highly connected sensors selected from the 90-th row and the 90-th column in matrix E respectively. It can be observed that the majority of nodes are situated near node 90, with a few nodes at a greater distance. Additionally, the nodes involved in aggregation and distribution are located on opposite sides of the pivotal node. This demonstrates that our method is indeed capable of capturing the process of traffic aggregation and distribution at pivotal nodes.

To validate the effectiveness on modeling the pivotal nodes of our method, We evaluate the predictive performance of our method on pivotal nodes. we choose an identified pivotal node (Node 0) and a non-pivotal node (Node 113) from the PEMS08 dataset. We present the traffic data records of two nodes over a span of three days, as shown in Figure 5 (a). It can be observed that the pivotal node exhibits higher traffic flow compared to the non-pivotal node, and its traffic flow undergoes significant variations throughout each day, indicating the challenge in predicting its behavior. Furthermore, in Figure 5 (b), we compare our method with the sub-optimal method DSTAGNN in terms of predictions for Node 0 on the second day. It is evident that our method successfully captures the abrupt changes in traffic flow (highlighted by the colored region), demonstrating its effectiveness in predicting the traffic variations of pivotal nodes. These results provide strong evidence that our proposed method outperforms alternative approaches in accurately forecasting the traffic flow on pivotal nodes.

Computation Time

To demonstrate the efficiency of our method, we compare STPGNN with DMSTGCN, DSTAGNN, and TPGNN, which have achieved suboptimal results on multiple datasets. The training time refers to the duration it takes for the model to complete one epoch under the same batch. Every experiment is repeated 10 times and the average performance is reported. Table 4 show the computation cost on the PEMS04 and PEMS08 datasets. Similar conclusions are drawn from the testing results on other datasets. STPGNN and DMSTGNN is faster than DSTAGNN and TPGNN in both training and inference. Although DSTAGNN achieves significant improvements in prediction accuracy through the use of multi-head attention mechanisms, it requires a substantial computational cost, leading to significantly increased time in training and inference.

Dataset	PEMS04			
Model	DSTAGNN	DMSTGCN	TPGNN	STPGNN
Training(s)	150.37	40.36	70.28	**32.36**
inference(s)	10.21	5.53	8.32	**3.93**
Dataset	PEMS08			
Model	DSTAGNN	DMSTGCN	TPGNN	STPGNN
Training(s)	117.41	35.37	58.25	**27.32**
inference(s)	8.94	4.28	7.53	**3.16**

Table 4: The computation cost on the PEMS04 and PEMS08 datasets.

Conclusion

In this paper, we propose a novel GNN-based method for traffic flow forecasting problems, in which we address a common phenomenon about pivotal nodes. We design a scoring mechanism to identify these pivotal nodes in traffic network and propose a novel pivotal graph convolution module to extract spatio-temporal features at these nodes. Furthermore, we introduce a parallel framework to concurrently capture spatio-temporal dependencies on both pivotal and non-pivotal nodes. Extensive experiments demonstrate the effectiveness and efficiency of our method.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported by the NSFC 61572537, and the CCF-Huawei Populus Grove Challenge Fund 202305. Yubao Liu is the corresponding author.

References

Bai, L.; Yao, L.; Li, C.; Wang, X.; and Wang, C. 2017. Adaptive graph convolutional recurrent network for traffic forecasting. In *NIPS*.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2014. Spectral Networks and Locally Connected Networks on Graphs. In *ICLR*.

Chai, D.; Wang, L.; and Yang, Q. 2018. Bike Flow Prediction with Multi-Graph Convolutional Networks. In *SIGSPATIAL*.

Cini, A.; Marisca, I.; Bianchi, F. M.; and Alippi, C. 2023. Scalable Spatiotemporal Graph Neural Networks. *Proceedings of the 37th AAAI Conference on Artificial Intelligence*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *NIPS*.

Drucker, H.; Burges, C. J.; Kaufman, L.; Smola, A.; and Vapnik, V. 1996. Support vector regression machines. *NIPS*.

Fang, Z.; Long, Q.; Song, G.; and Xie, K. 2021. Spatial-temporal graph ode networks for traffic flow forecasting. In *SIGKDD*.

Guo, S.; Lin, Y.; Feng, N.; Song, C.; and Wan, H. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *AAAI*.

Guo, S.; Lin, Y.; Wan, H.; Li, X.; and Cong, G. 2022. Learning Dynamics and Heterogeneity of Spatial-Temporal Graph Data for Traffic Forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 34(11): 5415–5428.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *NIPS*.

Han, L.; Du, B.; Sun, L.; Fu, Y.; Lv, Y.; and Xiong, H. 2021. Dynamic and Multi-Faceted Spatio-Temporal Deep Learning for Traffic Speed Forecasting. In *SIGKDD*.

Hu, Q.; Ming, L.; Xi, R.; Chen, L.; Jensen, C. S.; and Zheng, B. 2021. SOUP: A Fleet Management System for Passenger Demand Prediction and Competitive Taxi Supply. In *ICDE*.

Huang, R.; Huang, C.; Liu, Y.; Dai, G.; and Kong, W. 2021. LSGCN: long short-term traffic prediction with graph convolutional networks. In *IJCAI*.

Jin, G.; Liang, Y.; Fang, Y.; Huang, J.; Zhang, J.; and Zheng, Y. 2023. Spatio-Temporal Graph Neural Networks for Predictive Learning in Urban Computing: A Survey.

Junbo Zhang, D. Q., Yu Zheng. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. *AAAI*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Lan, S.; Ma, Y.; Huang, W.; Wang, W.; Yang, H.; and Li, P. 2022. DSTAGNN: Dynamic Spatial-Temporal Aware Graph Neural Network for Traffic Flow Forecasting. In *ICML*.

Li, M.; and Zhu, Z. 2021. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *AAAI*.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR*.

Li, Y.; Zheng, Y.; Zhang, H.; and Chen, L. 2015. Traffic Prediction in a Bike-Sharing System. In *SIGSPATIAL*.

Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; and Bronstein, M. M. 2017. Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs. In *CVPR*.

Ouyang, K.; Liang, Y.; Liu, Y.; Tong, Z.; Ruan, S.; Zheng, Y.; and Rosenblum, D. S. 2022. Fine-Grained Urban Flow Inference. *IEEE Transactions on Knowledge and Data Engineering*, 34(6): 2755–2770.

Pan, S.; Hu, R.; Fung, S.-f.; Long, G.; Jiang, J.; and Zhang, C. 2019. Learning graph embedding with adversarial training methods. *IEEE transactions on cybernetics*, 50(6): 2475–2487.

Pan, Z.; Zhang, W.; Liang, Y.; Zhang, W.; Yu, Y.; Zhang, J.; and Zheng, Y. 2022. Spatio-Temporal Meta Learning for Urban Traffic Prediction. *IEEE Transactions on Knowledge and Data Engineering*, 34(3): 1462–1476.

Seo, Y.; Defferrard, M.; Vandergheynst, P.; and Bresson, X. 2018. Structured Sequence Modeling with Graph Convolutional Recurrent Networks. In *ICONIP*.

Shen, Y.; Jin, C.; Hua, J.; and Huang, D. 2022. TTPNet: A Neural Network for Travel Time Prediction Based on Tensor Decomposition and Graph Embedding. *IEEE Transactions on Knowledge and Data Engineering*, 34(9): 4514–4526.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.; Wong, W.; and Woo, W. 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *NIPS*.

Song, C.; Lin, Y.; Guo, S.; and Wan, H. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *AAAI*.

Sun, J.; Zhang, J.; Li, Q.; Yi, X.; Liang, Y.; and Zheng, Y. 2022. Predicting Citywide Crowd Flows in Irregular Regions Using Multi-View Graph Convolutional Networks. *IEEE Transactions on Knowledge and Data Engineering*, 34(5): 2348–2359.

Tedjopurnomo, D. A.; Bao, Z.; Zheng, B.; Choudhury, F. M.; and Qin, A. K. 2022. A Survey on Modern Deep Neural Network for Traffic Prediction: Trends, Methods and Challenges. *IEEE Transactions on Knowledge and Data Engineering*, 34(4): 1544–1561.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. GRAPH ATTENTION NETWORKS. *stat*, 1050: 4.

Williams, B. M.; and Hoel, L. A. 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6): 664–672.

Wu, Y.; and Tan, H. 2016. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. *arXiv preprint arXiv:1612.01022*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *SIGKDD*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2022. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *NIPS*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph wavenet for deep spatial-temporal graph modeling. In *IJCAI*.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *AAAI*.

Yao, H.; Tang, X.; Wei, H.; Zheng, G.; Yu, Y.; and Li, Z. 2018. Modeling spatial-temporal dynamics for traffic prediction. *arXiv preprint arXiv:1803.01254*.

Ye, J.; Sun, L.; Du, B.; Fu, Y.; Tong, X.; and Xiong, H. 2019. Co-Prediction of Multiple Transportation Demands Based on Deep Spatio-Temporal Neural Network. In *SIGKDD*.

Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *IJCAI*.

Yu, F.; and Koltun, V. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. In *ICLR*.

Zhang, J.; Zheng, Y.; and Qi, D. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In *AAAI*.

Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; and Yi, X. 2016. DNN-Based Prediction Model for Spatio-Temporal Data. In *SIGSPATIAL*.

Zhao, B.; Xu, P.; Shi, Y.; Tong, Y.; Zhou, Z.; and Zeng, Y. 2019. Preference-aware task assignment in on-demand taxi dispatching: An online stable matching approach. In *AAAI*.

Zonoozi, A.; Kim, J.-J.; Li, X.; and Cong, G. 2018. Periodic-CRN: A Convolutional Recurrent Model for Crowd Density Prediction with Recurring Periodic Patterns. In *IJCAI*.