Predicting Traffic Speed Under the Impact of Maintenance Downtime with Graph Convolutional Networks

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

By

Yuanjie Lu
Bachelor of Science
JiangXi Normal University, 2018

Director: Dr. David Lattanzi, Professor
Department of Computer Science

Fall Semester 2020
George Mason University
Fairfax, VA

Dedication

I dedicate this thesis to my family and my friends. First of all, I thank my parents for supporting me to study in the United States and giving me support and encouragement. Besides, I thank my friends for being able to guide me to prepare something before writing the paper.

Acknowledgments

The work presented in this thesis was supported by funds from the Virginia Transportation Research Council. Since graduating from the INTO Mason program which is related to English pathway program, I have spent one and a half years on master's research. I not only improved my English and academic skils, but also became a persistent, diligent and innovative researcher. I am very grateful to Dr. Amarda Shehu and Dr. David Lattanzi, who provided me with opportunities to conduct artificial intelligence research and provided me with long-term teaching. Special thanks to Dr. David Lattanzi, who supported me and funded me to work on an application project. He is a consultant who taught me to think critically and independently. When I started out as a master's student, my English skills were poor and I didn't have research experience on artificial intelligence, but he didn't worry about me, he gave me the respect and encouragement and told me that whatever the experiment was successful or failed, all I did was to learn the process and gain new knowledge. In the research, he endowed me freedom to chose a topic and patiently gave me instructions. Even though I didn't come up with some experimental value in my first year, he still believed I could do it in the future. In his team, I felt the harmonious atmosphere of the each people, everyone is hardworking and enthusiastic. Appreciate to Dr. Amarda Shehu, she always encouraged me to be a smart and excellent student. In class, she taught me how to think and how to express my thought, even though I couldn't describe my thoughts accurately in class, she still gives me enough patience and guidance to help me complete the task. Thanks to my friend, Parastoo Kamranfar, Reza, Chris and Mozghan, they are all excellent PhD students who showing me how to become a good student. Also appreciate for all committee members, Dr. Amarda Shehu, Dr. David Lattanzi and Dr. Zoran Duric, who help me graduate in master.

Table of Contents

List of Tables

List of Figures

Abstract

PREDICTING TRAFFIC SPEED UNDER THE IMPACT OF MAINTENANCE DOWNTIME WITH GRAPH CONVOLUTIONAL NETWORKS

Yuanjie Lu

George Mason University, 2020

Thesis Director: Dr. David Lattanzi

In this paper, a traffic forecasting approach based on graph convolutional networks is proposed to learn the effect of maintenance downtime on the surrounding area. Since inappropriate construction can lead to traffic congestion, disturbance and accidents, it is important to evaluate the work zone downtime effect. Furthermore, learning the correlation between each roadway is essential because if construction is carried out, the upstream area will not only block the downstream area, but also affect the surrounding areas. In order to evaluate maintenance downtime effect on traffic roads, traffic speed predictions are used by many researchers as the first experiment to quantify the impact of construction work on traffic states. However, in early studies, there are many machine learning models and time-series models used to predict the abrupt changes in the traffic speed and quantify the impact of maintenance work, but these models cannot explain nonlinear relationships between the speed with traffic incidents, nor can they dynamically describe the causes of traffic speed changes, thus it is difficult to assess the impact of work zones in more complex traffic environments. In addition, few studies based on the deep learning approaches are used to measure the impact of traffic construction on the surrounding area, thus it is hard to reference the latest literature based on deep learning to tackle the problem. To predict

traffic speed under the impact of construction work, we design a novel model based on the graph convolutional neural network (GCN) to accommodate the spatial-temporal dependencies among traffic states, differentiate the intensity of connecting to neighboring roads and predict the speed under the road maintenance condition. The advantage of using a graph convolutional network is that it can transfer real-time traffic information between each road segment and adjacent segments, and automatically learn the non-structure features under the impacts of traffic incidents. The more nodes and layers the network has, the more information is involved in the calculation because each road segment represents a node and shares its parameters throughout the network. If we use the graph model to construct a node and find the correlation of its adjacent nodes, and then compare the predictive speed according to historical data, we can quantify the influence of traffic incidents in the area, making it possible to assess the economic losses in the event of traffic accidents. In the experiment, we compare our model with four baselines on two real-world datasets which are collected from the road sensor network around Tyson's Corner and Los Angeles. The result shows that our model is better than other benchmarks and proves that it is feasible to provide more traffic information for graph-based models to improve traffic flow prediction.

Chapter 1: Introduction

1.1 Background

Road maintenance and restoration activities often involve roadway downtime with impacts on traffic. This usually requires lane closures, disrupts traffic operation and not only causes road congestion but also creates significant safety hazards. According to an urban mobility report released in 2019 [1], the economic toll of traffic congestion has increased by nearly 48% over past ten years. In 2017, the average auto commuter spent extra 56 hours in the car and wasted 21 gallons of fuel due to traffic congestion. These wasted time and fuel costs add up to $1,080 per commuter. Moreover, the losses caused by road work zones are nearly 10% of the overall loss and 24% of losses on the freeway in the US, which has become the second-largest contributor to nonrecurring delay congestion [2]. Due to the economic loss of traffic incidents, it is necessary to quantify the associated mobility impact of road maintenance and restoration activities in order to enhance developed traffic management to help civic engineers and research to set dot grid planner in smart transportation and help drivers make informed choices to reduce traffic congestion.

1.2 Research problem

There are various challenges in traffic forecasting based on road maintenance. For instance, how to distinguish between normal and abnormal traffic conditions, how to calculate the complexity of traffic networks, how to form an efficient and intelligent methodology to reflect the temporal and spatial correlation. Unfortunately, most traditional traffic speed predictions are based on linear regression models since they are only characterized by traffic speed, but fail to consider whether the speed is affected by normal traffic conditions or abnormal conditions. In addition, even if the normal traffic data is provided to models, these models can only be used to predict traffic speeds that are not affected by any incidents. Therefore, it is hard to find the correlation between traffic speed with other traffic factors such as weather, temperature, human factors and traffic accidents, making it impossible to provide accurate traffic information in more complex environments.

To predict the future speed under the impact of construction work, it is critical to choose the speed data based on the construction conditions. Although there are many studies used to predict traffic speed in the work area, they are mainly focused on traditional machine learning methods to solve time-series problems, which require researchers to have some expertise to perform data mining, data modeling and data analysis. In addition, these studies only focus on the construction of the active area and upstream segments, while ignoring the intersecting roads and the downstream surrounding area, so that they fail to evaluate the influence of construction work in a region. Nowadays, more and more research is based on deep learning approaches rather than traditional machine learning, however, still few studies are used to measure the influence of construction work in regions. If we extend the latest approach based on deep learning to measure the influence of construction work, then in the future, we can improve the model to evaluate the influence under multiple traffic incidents, so that traffic will be more efficient and convenient.

1.3 Contribution

We choose a model named ASTGCN as the basic model, which is the most advanced benchmark for predicting traffic speeds in 2019. To tackle the challenges in maintenance downtime research, we extend the basic model and propose a framework called STGCN-WZ (spatio-temporal graph convolution network for traffic forecasting under the work zone impact) to evaluate the influence of construction work at every location on the traffic network. We first identify several ideas about construction work that are not considered in the previous literature, then we set up a new dataset that record the speed under the impact of maintenance downtime. After that, we use a sequence to sequence architecture based on a new deep learning approach that uses historical time steps and construction work information to predict future speed. Specifically, we feed traffic speed under the impact of work zones into the multi-head attentions mechanisms, making our model capture spatio-temporal correlations, then we utilize a graph convolution operation and a standard convolution operation to process the speed, so that the model can learn the topological characteristics and the temporal correlation. Finally, we make a comparison between predictive speed and the label speed, to evaluate how much impact the work zones exert on the traffic speed. The main contributions of this project are summarized as follows:

- We build a new real-world traffic dataset around Tyson's Corner in Fairfax, Virginia to record all traffic speed during each maintenance downtime event. We also have maintenance downtime information recording the location and time of each construction work. This dataset is helpful for us to evaluate our model and other baselines performance under the work zone condition.

- We consider several issues not involved in the construction literature. For example, how to set the scope of the impact of construction on the surrounding area, how to avoid the sparse data in the construction work feature map, and how to avoid insufficient data in speed feature map and construction work feature map. To address those issues, we build a formulation that sets the scope of impact of construction on the construction work feature map and use a window sliding approach to obtain the time steps during each construction work. As a result, our model proves that the formulation is suitable for solving the problem of data sparseness and the sliding window approach is helpful to solve the problem of insufficient data.

- We present a new learning objective to fuse multi-feature maps on the traffic network. By setting the learnable weights for each feature map, STGCN-WZ is able to assess how construction work affects traffic speed. In addition, we build a comparative model that uses only historical speed to predict future speed to assess whether our model exceeds other baselines of unused construction information. The experiment shows that adding a new structural feature matrix can help the model gain more knowledge on traffic network. If we continue to expand this approach, we also can capture other traffic attributes such as traffic congestion, traffic accidents and human-factors.

1.4 Related work

In the existing literature, research on traffic speed prediction is allocated into three categories: traffic simulation approaches, statistical analysis approaches and machine learning approaches [3].

Firstly, traffic simulation approaches are mainly used to simulate real-time traffic environment provided by the area-wide online traffic data to provide an estimation of traffic time, cost of travel and short-term traffic predictions [4]. This method is beneficial to identify the levels of congestion of certain roads and find alternative routes. The approaches are grouped into three modellings, which are microscopic modelling, mesoscopic modelling and macroscopic modelling [5]. In fact, the real-time simulation solutions based in the context of heterogeneous road networks such as urban, interurban, and rural have many challenges [6]. For example, due to the lack of detail and of flexibility in traffic simulation, researchers need to be highly experienced with the most advanced simulation technology to make accurate predictions. Furthermore, building this model requires a significant computing capability under a large number of parameters such as location, traffic flow, density and shock waves. Thus, traffic simulation approaches fail to predict the complex road networks.

Secondly, in the statistical analysis model, most of the traffic forecasting are based on the regression functions, such as linear regression models, time series model and Kalman filter models. Various models such as ARIMA model [7], seasonal ARIMA model [8], and ARIMAX model[9] are used in successive time sequences of traffic variables to forecast the traffic parameters for short-term periods. Markov chain model [10] and Kalman filter model [11] also have been used to predict the traffic speed for short-term periods. The above models consider the dynamic variation in the traffic state and have a convenient calculation.

However, none of these models can overcome the influence of abnormal factors or reflect the nonlinearities and uncertainties of traffic data. Hence, it is powerless in large-scale and abnormal data of traffic prediction.

Thirdly, in the machine learning model, most of the literature is based on the statistical regularity of historical data, learning from the dynamic behavior of traffic system to predict and evaluate the traffic state of future data. K-nearest neighbors algorithm (KNN) is one of the most influential and popular methods among learning-based behavior, which uses traffic speed as a prediction value to detect the traffic condition [12]. More advanced models such as support vector machines(SVM) [13], online Support Vector Machine (SVM) [14] are used to predict the accuracy according to the high dynamics and sensitivity of traffic flow. Artificial neural network (ANNs) is also used for traffic predictions, because it can handle multi-dimensional data and has the ability to work with flexibility and generalizability [15]. In the literature, Huang and Ran used an ANN model to predict the traffic speed during the adverse weather conditions [16]. Moretti et al. [17] use a hybrid model combing statistical and ANN to predict urban traffic flow . Another model called ANN-SVM, which integrates two different machine learning methods, is used to improve the prediction accuracy further. The ANN model is used to predict delays while the SVM model is used to predict work zone capacities [15]. Although these traditional models only need enough historical data to learn the statistical regularity automatically, researchers need to manually create the traffic features, which means they need to have professional knowledge to select representative data. In addition, these methods are limited to less complex traffic scenarios; if the traffic data is influenced by abnormal factors, it is hard to collect and manipulate the data into a usable and desired form.

To tackle the issues of traditional models, researchers have deployed a more advanced and powerful model named deep learning into traffic prediction. It enables machines to learn characteristic value automatically, not only freeing them from the limitations of human-made manipulation features but also building the area-wide spatiotemporal dependencies.

At the beginning of research in deep learning, some methods only consider temporal dependence. In [18], Huang et al. proposed a network architecture combined with DBN and a regression model, which is used to capture the random features from multiply traffic dataset. In the long short-term memory (LSTM) neural network and recurrent neural network (RNN), Ma et al. [19] proposes a neural network named LSTM NN to capture the nonlinear traffic dynamically and they also combine deep restricted Boltzmann machines (RBM) with a recurrent neural network (RNN) to improve the model performance. The above models only reflect the correlation between traffic forecasting and time but fail to reflect the spatial structure; they cannot accurately predict the change of conditions in the traffic network. In addition, once the spatial structure of the research area changes, the results are not representative. To capture the spatial correlation, many researchers have developed more efficient ways to incorporate traffic information into a spatial structure. For example, Jo et al. [20] convert traffic datasets into an enhanced physical map as both input and output, using a convolutional neural network to forecast the speed based on image-to-image learning. Genders et al. [21] propose a state space to encode the discrete traffic states as inputs to a deep convolutional neural network and abbreviate the topology of the transportation network to a simple grid structure. Although these conversion methods can capture the spatial relationship, before implementing the methods, complex data preprocessing is required. Besides, simplifying the transportation network will cause a distortion of the actual network shape, which results in a biased spatial correlation.

The best plausible way to incorporate the topology of a traffic network into a deep neural network is a graph neural network (GNN), which was initially outlined in Gori et al. [22] and further elaborated in Scarselli et al. [23] and Gallicchio et al. [24]. In road transportation, data is mainly collected from traffic sensors, which are devices that provide data or information to support traffic management applications such as signal control, gathering of vehicle volume and incident detection [25]. Because sensors provide their geographic location, multiple sensors can be linked to a matrix to effectively reflect spatial dependencies. Compared with the previous deep learning methods, GNN requires a smaller number of weight parameters and has a superior performance [26]. Other powerful aspects of the graph neural network are that it redefines the irregular structure data on the graph, stores the features in each node, and then extracts the spatial features of the topological graph. There are many studies on graph neural networks, which are divided into three main categories according to historical development, namely network embedding, recurrent graph neural networks and graph convolutional network [26]. First, network embedding is used for mapping graph nodes to vectors in a low-dimensional and more discriminative space [27]. The advantage of this is that the data can be compressed, thereby it can reduce the computational complexity and increase the calculation speed. However, it is difficult to find the best dimension in actual embedding. Lower dimensions can reduce the complexity of time and space but undoubtedly lose the information in the original image. In addition, the model mainly lacks the generalization ability. Whenever a new node is added in the data, it needs to retrain the model to represent this node. Therefore, this model is not suitable for a dynamic graph. To solve the limitation, Kang et al. [28] introduce a model that use graph self-attention layers with Gumbel-Softmax technique to learn graph-level spatial embedding. Zheng et al. [29] propose a graph multi-attention network based on embedding the original data to learn spatial correlations and non-linear temporal correlations. Second, recurrent graph neural networks (RecGNNs) are essential to the development of graph neural networks. The design is based on traditional recurrent neural networks as models of sequential data and the purpose is to continuously learn a target node's representation by propagating information with neighboring nodes around a central node until stable equilibrium is reached [26]. Implementations of graph recurrent architecture can be found in [30], [23], focusing on several expansion methods of recurrent graph neural and process general types of graphs such as acyclic, cyclic, directed, and undirected, etc. However, using RecGNNs is computationally expensive; it only transmits the information of each node clustering and updates the state of its own node, which cannot capture the spatial relationships in the traffic network. Inspired by the success of convolutional neural networks in computer vision, many researchers have attempted to find the operation of convolution in graph data. Cui et al. [31] propose a framework that using CNN to capture spatial structures and using LSTM to capture temporal structures to learn the interactions between each road and predict the traffic state. Li et al. [32] introduce a framework named DCRNN to incorporates both spatial and temporal dependencies in the traffic flow, where the spatial dependency is obtained by bidirectional random walks on the graph. Those methods mentioned above belong to the most notable branch of the GNN methods, called graph convolution networks (GCNs).

Instead of iterating states and propagating information from a sequence of nodes, GCNs attempt to support a graph with a fixed structure and build convolutional layers to extract essential features. As GNNs are more convenient to extract the spatial feature from the topological map, many new frameworks such as graph attention network [33], Variational Graph Auto-Encoders [34] and spatio-temporal graph neural networks [35] are based on this theory for graph modeling. GCNs fall into two types: spectral-domain and spatial-domain. Firstly, spatial-based approaches inherit the theory from RecGNNs, it is used to sum the neighboring vertex around a center vertex, in other words, the embedding of nodes is the aggregation result of all the neighbor nodes embedding including self-embedding. In 2009, Micheli et al. [36] use the idea of message passing from RecGNNs, using a constructive neural network named NN4G to address the graph dependency, which is the pioneer for subsequent spatial domain research. Secondly, spectral-based approaches use algebraic and spectral graph theoretic concepts to process the signal on graphs[37]. In 2013, Bruna et al. [38] first introduce a graph convolution based on spectral graph theory. In fact, most of the researches in the traffic prediction are based on the spectral domain, because the spatial-based domain needs to initialize a central vertex as the starting point of the sequence, then propagate and update the vertex state based on its neighbors. However, the neighborhood of each node is structured differently and therefore it is unfeasible to define a standard convolution in the graph.

To design a convolution in the graph, researchers turn to spectral-domain approaches, which relies on a graph Fourier basis generalize to new undirected graphs and study the influence of the eigenvalues of the Laplacian matrix and the corresponding eigenvectors on the topological properties of the graph. Since the Laplacian matrix is a symmetric matrix, it can be decomposed and its spatial characteristics can be obtained through the Laplacian operator. More details of graph convolutional network theory will be described in a later section. In spectral research, Yu et al. [35] pioneered a model that predicts traffic speeds by combining the GCN and the GRU model, where the GCN is used to learn topological structures for capturing the spatial correlation and the GRU is used to learn variations of each tensor for capturing the temporal dependencies. Guo et al. [39] first attempt to apply attention method into the model to reflect the dynamic spatio-temporal correlations of traffic data. Zheng et al. [29] propose an encoder-decoder architecture and apply a transform attention mechanism, where both the encoder and decoder consist of the multi self-attention in the network. Zhou et al. [40] design a new policy gradient to update the model parameters in order to alleviate the bias in the attention-based graph convolution neural model. Nowadays, most existing methods mainly combine graph convolutional networks with other most influential methods such as recurrent neural networks, natural language processing to solve various traffic flow problems. However, these advanced methods are rarely used to evaluate the spatial and temporal dependencies under the effects of traffic construction downtime.

Based on the background, we propose a new neural network approach based on a state-of-the-art baseline to capture the spatial and temporal dependencies under the impact of construction work in the surrounding region. The rest of the paper is shown as follows. In Chapter 2, we introduce some preliminaries concepts about graph convolutional networks and formalize the traffic prediction problem. In Chapter 3, We describe the framework of the spatio-temporal graph convolution network for traffic forecasting within the work zone (STGCN-WZ). Then in Chapter 4, we discuss the experiment about the three model components. In Chapter 5, we review our related work and summarize our project.

Chapter 2: Preliminaries

This section not only states the definition of a road network structure from graph neural networks by using basic terminology and mathematical expression but also introduces the background about the network such as Laplacian matrix, Laplacian operator and Fourier transform. Table 2.1 summarizes the used notations in this paper.

2.1 Traffic network structure

We denote a road network as graph $G = (V, E)$, given the road segments V and road distance E between the pairs of each sensors. Then, we define a weighted adjacency matrix, represented by:

$$A = (A_{v_1 v_1}, A_{v_1 v_2}, \cdots, A_{v_n v_n}) \in R^{N \times N}, \tag{2.1}$$

where N is the number of road segments and $A_{v_i v_j}$ represents the correlation (usually measured by the road distance) between vertex v_i and vertex v_j.

In traffic, useful information includes traffic speed, traffic flow and some other external conditions like the number of lane closures, weather, etc. For those facts, we can define those we need as feature maps, represented by:

$$X = (X_{v_1 t_1}, X_{v_1 t_2}, \cdots, X_{v_i t_j}, \cdots, X_{v_n t_k}) \in R^{N \times T} \tag{2.2}$$

where N is the number of road segments, T is the length of historical time series, $X_{v_i t_j}$ represents the value in the road i-th and at the j-th time step. In this project, speed is the main feature of our research, which named X^s. We also add an external feature map X^c which is related to the maintenance downtime conditions. We denote that if there is a

Table 2.1: Frequently mathematical notations

Notation	Description
G	directed sensor network
V	road segments/ sensors
E	road distance for each pairs of sensors
$A/W/A_{v_i v_j}$	weighted adjacency matrix of G
$X/X_{v_i t_j}$	feature map
X_T	traffic speeds over past P time steps
U	the matrix of eigenvectors ordered by eigenvalue
D/L	undirected degree matrix / Laplacian matrix
$g_\theta/*$	a learnable convolution kernel / graph convolution operation
N/n	the number of road segments / the n-th road segment
T/k	the length of historical time series / the k-th time step
H/h	the number of time steps for training / h-th time step
P/p	the number of time steps for predicting / p-th time step
v_i	traffic speeds measured in road segment i
t_j	traffic speeds measured at time j
λ	a hyper-parameter of setting the influence of construction work
U_1, U_2, U_3, b	the parameter of temporal attention
W_1, W_2, W_3, b	the parameter of spatial attention

construction work happened in the road N_i and at the time T_j, we have $X^c_{v_i t_j} = 1$, but at the same time, if other roads do not have work zone conditions, we set it to 0.

2.2 Data construction

Speed modeling is based on the time series forecasting, which predicts values over a period of time based on the historical data. In the traffic forecasting, the speed $X^s_{v_i,t_j}$ of vertex v_i during time t_j is related to the speeds $X^s_{v\in V,t_{1,\cdots,j-1}}$ of all road segments including itself during time t_1 to t_{j-1}. Although the task of modeling future speed $X^s_{v\in V,t_i}$ predictions based on entire previous speeds seems feasible, the last term in predicting $X^s_{v\in V,t_i}$ using previous speed $X^s_{v\in V,t_{j-1}}$ still requires conditioning on $j-1$ times, which is impossible to model because of the complicated calculations. To figure out the problem, we use sliding window method over the historical time series to generate the training set and predicting

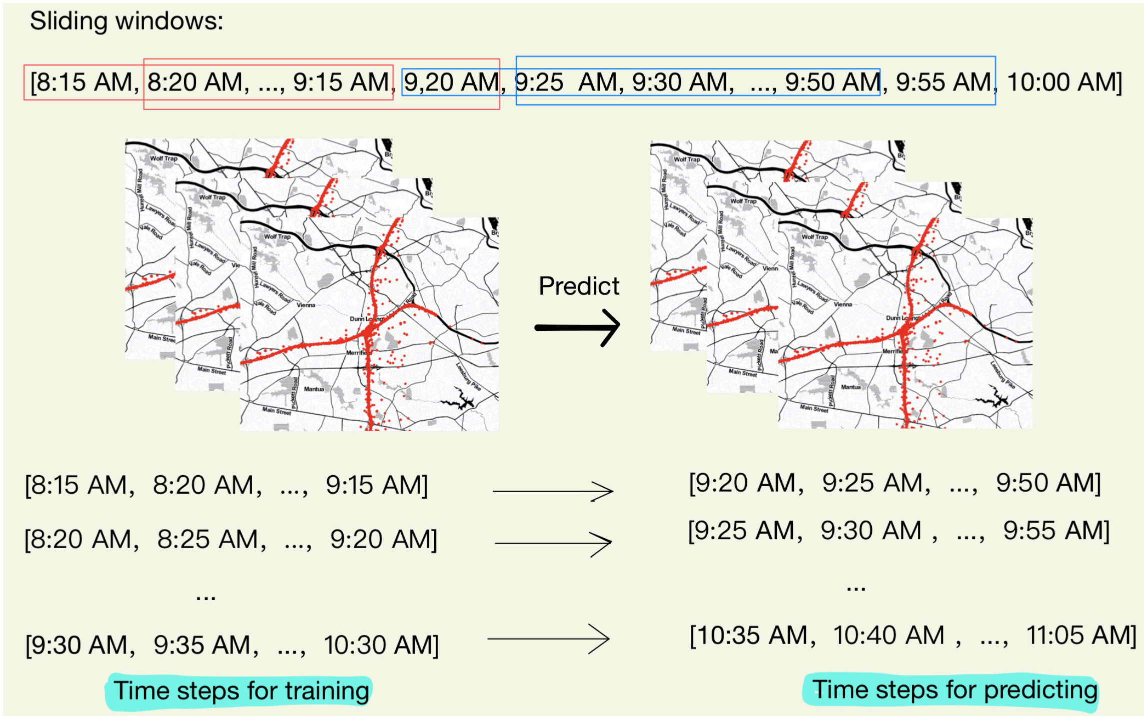

Figure 2.1: Data construction by using sliding window

set. For the entire set of time series $X_T = (t_1, t_2, \cdots, t_k)$, we set two hyper-parameters H and P as the number of time steps for training and the number of time steps for predicting. Similarly, we also use the same approach in the construction feature map X^c. Fig 2.1 shows how to set the input.

2.3 Graph convolution in spectral-domain

This section introduce three concepts in the spectral-domain: Laplacian matrix, Laplacian Operator and Fourier transform. Because the convolution operation in the CNN model cannot directly be applied in the graph, it is necessary to redefine a convolution operation in the spectral-domain and then convert it back to the spatial domain through the convolution theorem. We first design a Laplacian matrix of the graph to derive the Laplacian operator and then perform eigendecomposition by Fourier transform.

2.3.1 Laplacian matrix

In graph theory, the Laplacian matrix is also called the graph Laplacian or discrete Laplacian[41]. It is widely used in the field of machine learning such as dimensional reduction, classification, clustering, etc. There are three kinds of Laplacian matrix, which are simple Laplacian, Symmertric normalized Laplacian and Random walk normalized Laplacian.

Given a simple graph G with n vertices, the simple Laplacian matrix $L_{n\times n}$ is defined as[42]:

$$L = D - W, \tag{2.3}$$

where D is the degree matrix and W is the adjacency matrix of the graph. The Symmertirc normalized Laplacian of a graph is defined as :

$$L^{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}, \tag{2.4}$$

where I is an identity matrix. The Random walk normalized Laplacian is describe as:

$$L^{rw} = D^{-1}L = I - D^{-1}W \tag{2.5}$$

2.3.2 Laplacian operator

Laplacian operator is mainly used to explore the divergence of Euclidean space, however, it is also useful to find the divergence in the graph. The physical meaning of the Laplacian operator ($\triangle f$) is the second-order derivate in the Euclidean space, which is the divergence ($\nabla\cdot$) in the gradient (∇f) of a scalar field, where $\nabla\cdot$ indicates the strength of the divergence of the vector field at each point and f is a twice-differentiable real-valued function. The divergence of the gradient of Laplacian can be defined as:

$$\triangle f = \nabla^2 f = \nabla \cdot (\nabla f)\,, \tag{2.6}$$

where $\nabla = (\frac{\partial}{\partial x_1}, \cdots, \frac{\partial}{\partial x_n})$. Equivalently, $\triangle f$ is the sum of all the unmixed second partial derivatives in the Cartesian coordinate system x_i, represented by:

$$\triangle f = \frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \cdots + \frac{\partial^2 f}{\partial x_n^2} = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2} \tag{2.7}$$

In traffic speed, since we have N road segments and E edges in the graph, the function f defined above is the N-dimensional vector, shown as $f = (f_1, \cdots, f_i, \cdots, f_N)$, where f_i is the function value of f at node i in the graph. Then the Laplacian of f for each node i can be formulated as: $\triangle f_i = \sum_{j \in N_i} W_{ij}(f_i - f_j)$, where N_i denotes the set of points of road segments except node i, W_{ij} represents whether node i and node j are adjacent. If W_{ij}=0, then node j is not a neighbor of node i. To derive the above formula, we can obtain that:

$$\triangle f_i = \sum_{j \in N_i} W_{ij}(f_i - f_j) = \sum_{j \in N_i} W_{ij} f_i - \sum_{j \in N_i} W_{ij} f_j = d_i f_i - w_{i:} f, \tag{2.8}$$

where $d_i = \sum_{j \in N_i} W_{ij}$, $w_{i:} = (w_{i1,\cdots,iN})$, $f = (f_1, \cdots, f_N)^{\mathsf{T}}$ and $w_{i:} f$ denotes the inner product of two vectors. Finally, for all Laplacian operators $\triangle f$, they are represented by:

$$\begin{aligned} \triangle f &= \begin{pmatrix} \triangle f_i \\ \vdots \\ \triangle f_N \end{pmatrix} = \begin{pmatrix} d_1 f_1 - w_{1:} f \\ \vdots \\ d_N f_N - w_{N:} f \end{pmatrix} = \begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_N \end{pmatrix} f - \begin{pmatrix} w_{1:} \\ \vdots \\ w_{N:} \end{pmatrix} f \\ &= diag(d_i) f - W f = (D - W) f = L f \end{aligned} \tag{2.9}$$

Deriving from the above formula, the function $(D - W)$ called the simple Laplacian matrix is used to find the graph divergence. The i row of the matrix actually reflects the gain accumulation of the disturbance generated by the i node to other adjacent nodes. In other words, the equation can infer the relationship between each node and its neighbors. More information about Laplacian operator can be found in [43].

2.3.3 Fourier transform

The key to graph convolutional networks is to use Fourier domain by computing the eigen-decomposion of the graph Laplacian[44]. This require to review the idea of an orthogonality of Fourier transform[45]. According to the Laplacian matrix L, we first define the decomposition as

$$L = D - A = U\Lambda U^{-1} = U \begin{bmatrix} \lambda_n & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} U^{-1}, \tag{2.10}$$

where $U \in R^{N \times N}$ is the matrix of eigenvectors ordered by eigenvalues and Λ is the diagonal matrix of eigenvalues. Next, the graph Fourier transform to x is described as $f(x) = U^T x$ and the inverse Fourier transform to x is defined as $f^{-1}(\hat{x}) = U\hat{x}$, where $\hat{x}$ is the output generated from the original input x by the graph Fourier transform[44]. Then, using the graph Fourier transform with the multiplication of x and a filter $g \in R^N$, the convolutions in the graph are defined as:

$$x * g = f^{-1}(f(x) \odot f(g)) = U(U^\top x \odot U^\top g), \tag{2.11}$$

where $*$ is graph convolution operation and $\odot$ defines the Hadamard product. Finally, we define $U^\top g$ as g_θ, which is a learnable convolution kernel. The graph convolution is written as:

$$(x * g)_G = U g_\theta U^\top x. \tag{2.12}$$

2.3.4 The correlation between Laplacian operator and Fourier

This section explains how the Laplacian operator used in graph Fourier transform. Given a function F, the Fourier transform $\mathcal{F}$ is defined as[46]:

$$F(w) = \mathcal{F}[f(x)] = \int_{-\infty}^{\infty} f(x)\ e^{-iwx} dx, \tag{2.13}$$

where $w \equiv 2\pi v$. By default, the $\mathcal{F}$ obtained by the Fourier transform of the f is actually the integral of the f and the basis function e^{-iwx}. As the Laplacian operator is defined as $\triangle g = \sum \frac{\partial^2 g}{\partial x_i^2}$, which is used to sum the second-order partial derivatives. We substitute e^{-iwx} into the characteristic equation to obtain the function $\triangle e^{-iwx} = -w^2 e^{-iwx}$, then use this function on the convolution of the graph.

Chapter 3: Methodology

In this chapter, we focus on the methodology of the model. Fig 3.1 presents the overall framework of the STGCN-WZ model. Firstly, we build a traffic network structure, set a speed feature map and a construction work feature map using sliding window approach. For each feature map, we establish learnable weights corresponding to each feature map, and then use the fusion method to set these maps as new inputs, called the speed wave. Secondly, we feed the speed wave into the graph convolution network, which is operated by attentions and convolution operations in two layers. Thirdly, the output of the graph convolutional network is sent to a simple convolution neural network, bidirectional RNN and linear operation, making the output has the same dimension as the predicted label.

3.1 Construction Work Feature Map

We introduce more details about feature maps than Chapter 2.1. In terms of a construction feature matrix X^c, if there is a construction work happened in the road N_i and at the time T_j, we can set $X^c_{v_i t_j} = 1$, otherwise we set 0. However, setting 0 and 1 will cause the feature matrix to be sparse. Meanwhile, for those roads that are not under construction work, since they are all set to 0, there is no difference between these roads, especially some roads are very close to the construction road, while other roads are very far away from the construction road.

We build a construction work feature map X^c to help the model learn traffic attributes better. If there is a construction work happened in the road N_i and at the time T_j, we can set $X^c_{v_i t_j} = 1$, otherwise we set 0. However, setting 0 and 1 will cause the feature matrix to be sparse because there are few work zones happened at the same time. In addition, setting roadway to 0 means that even if the roadway is close to the work zone, it will not

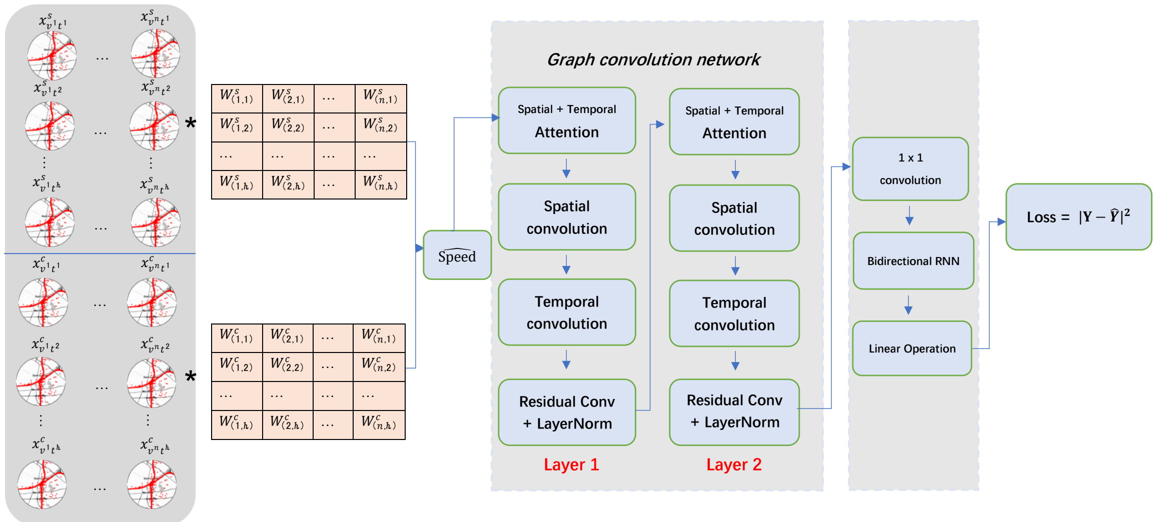

Figure 3.1: The framework of STGCN-WZ

be affected by the work zone, which is wrong fact. Thus, before importing the model, we modify it as:

$$X^c_{v \in V t_j} = \begin{cases} \max(0, 1 - (\frac{dis(v,v_i)}{\lambda})^2), & \text{if there is construction work } X^c_{v_i t_j} \\ 0, & \text{if there is no construction work } X^c_{v_i t_j} \end{cases}, \tag{3.1}$$

where λ is a hyper-parameter, $dis(vi, v_j)$ defines the distance between node v_i and node v_j, which is calculated from the weighted adjacent matrix. For any node $v \in V$ at any time $t \in T$, X^c_{vt} represent a parameter affected by the construction area. If $X^c_{v_i t_j} = 1$, it means the construction work takes place at the road i and at time j.

At the initial stage of model, we create a learnable parameter weight W_c that has the same dimensions as the speed input. Then we use the fusion approach to combine the construction feature map and speed feature map as speed wave $\hat{X}_s = W_s \odot X^s + W_c \odot X^c$, where $\odot$ is Hadamard product.

3.2 Spatial-Temporal Attention

The spatial-temporal attention model is widely used in natural language processing, image recognition and speech recognition, and is one of the important technologies of current deep learning[47]. Some studies such as [48] and [40], have used attention to capture the dynamic spatial and temporal relationships on the traffic road. In this project, we first use the spatial-temporal attention mechanism and the multi-head attention mechanism both. The attention mechanism divides the traffic speed information into shorter pieces of information X, and then integrates the correlation of each piece. In other words, it helps the model to assign different weights to each part of the input X, and then extracts more important information, so that the model can make more accurate judgments. At the same time, it does not bring too much overhead to the calculation and storage of the model. Multi-head attention mechanism is one of the branches of attention approach, it is equivalent to the ensemble of multiple different self-attentions, where each attention does not share weights. The advantage of this approach is that it allows the model to jointly attend to information from different representation subspaces at different positions, so that the model can handle more complex spatial problems in the traffic.

3.2.1 Temporal attention

In the temporal attention, there is a correlation between traffic conditions in different time periods. Usually the current traffic situation is influenced by the situation of previous time. Because the dimension of traffic feature map is $X \in R^{N\times T\times C}$, where N is the number of sensors, T is the number of time step and C is the number of channels, the temporal attention mechanism can be written as[39]:

$$E = V \cdot \sigma\{(X^{\mathsf{T}} \cdot U_1) \cdot U_2 \cdot (X \cdot U_3) + b\}, \tag{3.2}$$

where $U_1 \in R^N, U_2 \in R^{C\times N}, U_3 \in R^C, V \in R^{N\times N}, b \in R^{T\times T}, E \in R^{T\times T}$, sigmoid σ is the activation function. After using the function, then we use softmax to normalize the matrix

E, which is represented as:

$$E_{ij} = \frac{\exp(E_{ij})}{\sum_{j=1}^{T_{r-1}} \exp(E_{ij})} \tag{3.3}$$

where E_{ij} denotes the dependencies between the time i and the time j. Finally, we get a new $\widehat{X}$, which is calculated by $(X_1, X_2, \cdots, X_{T_{r-1}})E$.

3.2.2 Spatial attention

After using temporal attention, we turn to use spatial attention to capture the dynamic correlation between nodes in the spatial dimension. Similar to temporal attention, the spatial attention mechanism is defined as:

$$S = V \cdot \sigma\{(X \cdot W_1) \cdot W_2 \cdot (W_3 \cdot X)^{\mathsf{T}} + b\}, \tag{3.4}$$

where $W_1 \in R^T, W_2 \in R^{C\times T}, W_3 \in R^T, V \in R^{N\times N}, b \in R^{T\times T}, S \in R^{T\times T}$. Then use softmax function to calculate the attention weights:

$$S_{ij} = \frac{\exp(S_{ij})}{\sum_{j=1}^{N} \exp(E_{ij})}, \tag{3.5}$$

where S_{ij} denotes the dependencies between the node i and the node j. Finally, we compare S with the adjacency matrix A to update the weight in the graph.

3.2.3 Multi-head Spatial-temporal attention

Another approach to capture spatial and temporal dependencies is to use multi-head attention [49]. Given the X as the traffic speed, we calculate three matrix, key K, Values V and queries Q. The matrix of outputs is represented as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, head_2, \cdots, head_h)W^O, \tag{3.6}$$

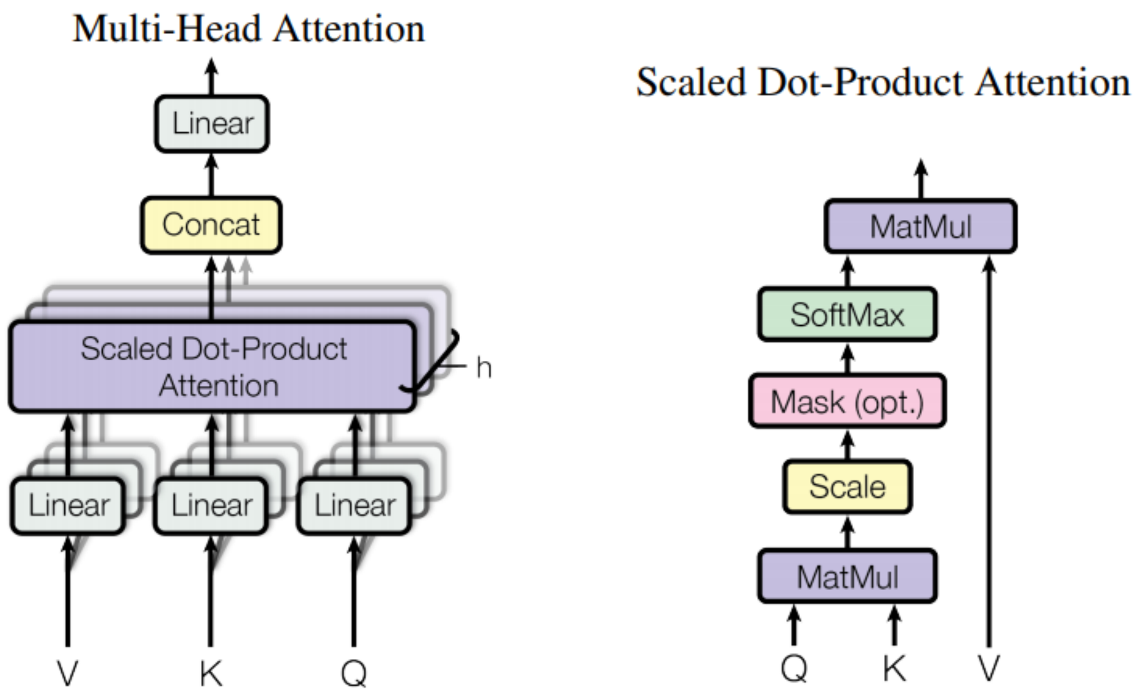

Figure 3.2: The RMSE of the models in Tyson's Corner (left) and Los-loop dataset (right)

where $head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$ and these parameter.

In this project, we mainly use multi-head spatial-temporal attention to capture the correlations. The advantage of using multi-head attention is that it has fewer parameters to train faster and also has good performance, even if the sequence is too long, multi-head attention can grasp the key points well without losing important information. In addition, because it allows models to learn information in different subspaces, it can help models learn spatial relevance in more complex spatial structures. Fig 3.2 shows how to use multi-head attention to get the result. More details of using multi-head approach in the network can be found in [29].

3.3 Spatial-Temporal Convolution

3.3.1 Spatial convolution

To learn the topological properties in the traffic network, we adopt a graph convolution operation in the model. As the equation 2.12 mentioned above, the convolution operator based on spectral and Fourier transformer is defined as $(f * g)_G = U g_\theta U^\top f$. Although this approach is theoretically feasible, the computational cost is very expensive, because each sample needs feature decomposition, and each forward propagation needs to be calculate the product of U, g_θ and $U^\top$. Thus, a new method is proposed in [50], which uses Chebyshev polynomials to fit the convolution kernel to reduce the computational complexity. In [51], Hammond et al. mention that g_θ can be expanded by Chebyshev polynomials, which is defined as:

$$g_\theta(\Lambda) \approx \sum_{k=0}^{K} \theta_k T_K(\widehat{\Lambda}), \tag{3.7}$$

where $\widehat{\Lambda} = \frac{2\Lambda}{\lambda_{max}} - I_N$, λ_{max} is the spectral radius, θ is the vector of Chebyshev coefficient, T_K is defined as $T_k(x) = 2xT_{k-1} - T_{k-2}(x)$, where $T_0(x) = 1$ and $T_1(x) = x$. Thus, the graph convolution operation is denoted as:

$$(x * g)_G = \sum_{k=0}^{K} \theta_k T_K(\widehat{L})x, \tag{3.8}$$

where $\widehat{L} = \frac{2L}{\lambda_{max}} - I_N = U\widehat{\Lambda}U^\top$. Compared with the previous graph convolution operation with complexity of $O(N^2)$, the complexity of this approach is $O(K|E|)$, which greatly improves the calculation speed. To further improve the efficiency of calculation based on Chebyshev model, we use a layer-wise linear model[52], which is also called first-order ChebNet. Based on the previous work, this method has officially become the pioneering work of GCN. In this project, we set $K = 1$ and $\lambda_{max} = 2$, then the function is written as

$$(x * g)_G = \theta_0 x - \theta_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x \tag{3.9}$$

then we set $\theta = \theta_0 = -\theta_1$ to avoid overfitting. Finally, the simple function is defined as:

$$(x * g)_G = \theta(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x) \tag{3.10}$$

Because the range of $I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ is $[0, 2]$, using the discrete Laplacian operation in the network will cause the gradient to explode and disappear. To figure out the problem, we use a renormalization trick, the function is shown as:

$$I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}, \tag{3.11}$$

where $\widetilde{A} = A + I_N$, $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$. In the end, the GCN model can be expressed as:

$$H^{(l+1)} = f(H^l, A) = \sigma((x * g)_G) = \sigma(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} H^l \theta^l) \tag{3.12}$$

where H^l is the output of l layer, θ is a learnable weight, σ is the sigmoid function.

3.3.2 Temporal convolution

In the previous research, zhao et al. [53] capture the temporal dependences based on the traditional recurrent neural network. Zhou et al. [40] use GRUs as the base architecture in the encoder-decoder framework to learn the temporal correlation. In this project, we use a standard convolution operation to merge the information at the time slices, so the output after the spatial-temporal convolution as an example is written as:

$$\bar{X}_H^{(l+1)} = \sigma(\Phi * (\sigma((x * g)_G X_H^{(l)}))) \in R^{C \times N \times T}, \tag{3.13}$$

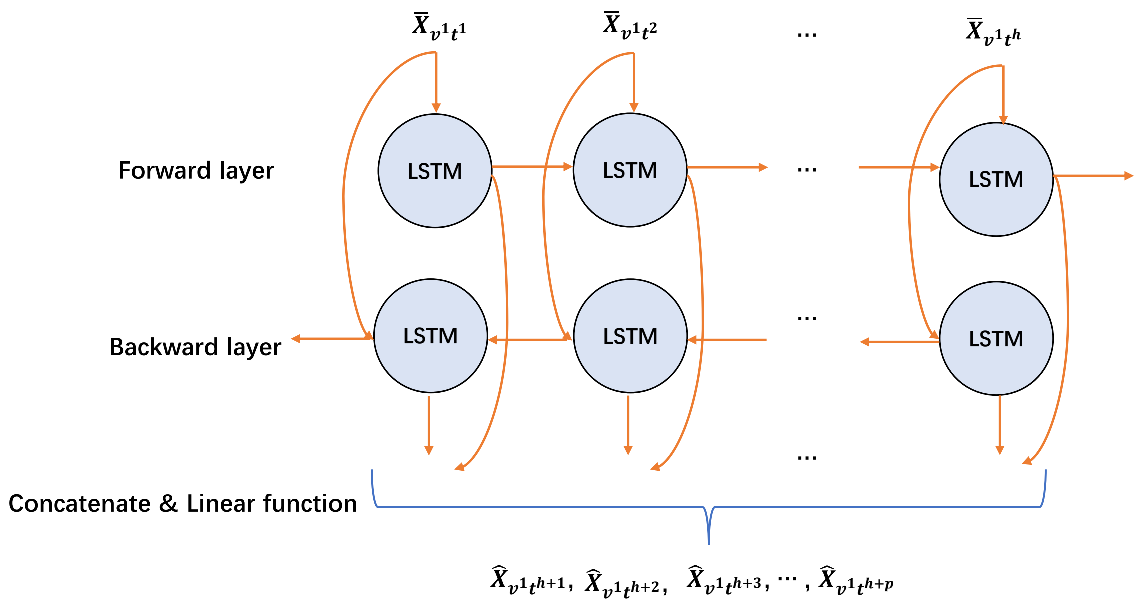

Figure 3.3: After the spatial-temporal convolution, the speed information is fed into Bi-LSTM and Linear function

where Φ is a parameter of temporal convolution kernel, the first $*$ is a standard convolution operation, $(x * g)_G$ represents using graph convolution operation and first σ is an activation function which is ReLU.

After we use two block layers in spatial-temporal convolution, we utilize a 1×1 convolutional layer to reduce the output channel size to 1 in order to merge the temporal features of traffic data. Then we append a bidirectional recurrent neural network to learn the dynamic behavior in the time sequence for each node. Finally, we add a linear function to make sure the output has the same dimension with the prediction. Fig 3.3 shows the process that using Bi-LSTM and a linear function, the input $\bar{X} \in R^{N \times H}$ represents the speed information after merging the traffic features where H is the length of historical time series and $\hat{X} \in R^{N \times P}$ is the speed prediction which has the same dimension P and N as the traffic label.

Chapter 4: Experiment

4.1 Datasets

In our experiment, the time steps that need to be predicted are 3, 6, and 12. If the time interval is 5 minutes, we can predict the future speed in 15, 30 and 60 minutes. We create a new dataset near Tyson's Corner in Fairfax, Virginia, and use a traffic speed dataset named Los-loop which is provided by [53] in Los Angeles County. Fig 4.1 shows the study area of two datasets. The difference between the two datasets is that the traffic speed provided by the Tyson dataset is limited to the construction work at most and the traffic speed in Los Angeles is affected by multiple traffic factors such as traffic congestion and collisions. Besides, compared with Los Angeles dataset, we have to record many cases of construction work, so the speed and construction work feature map information in our dataset are usually discontinuous, while the speed in the Los Angeles loop is not. Moreover, because of the missing data, the dataset in Los Angeles is filled by the linear interpolation method. The advantage of using these two datasets is to test our model's generalization ability for the abnormal traffic speed and to detect whether our model will achieve better experimental results even if we do not use the construction feature map.

The Tyson dataset includes 1) historical traffic speed recorded by traffic sensors in the Tyson's Corner region ranging from January 1st, 2019 to December 31st, 2019 in Tyson's Corner, Fairfax, the United States; 2) Raw data which includes some attributes such as traffic standardization type, location, duration, start date and end date; 3) Tyson's Corner traffic sensor information such as location, the distance on the road segments. Data preprocessing in traffic can be divided into five steps. Firstly, the original has many traffic categories like collision, disable vehicle, traffic congestion, road constructions, etc. We unify the above data into three categories: work zones, collisions and other facts. Secondly, we

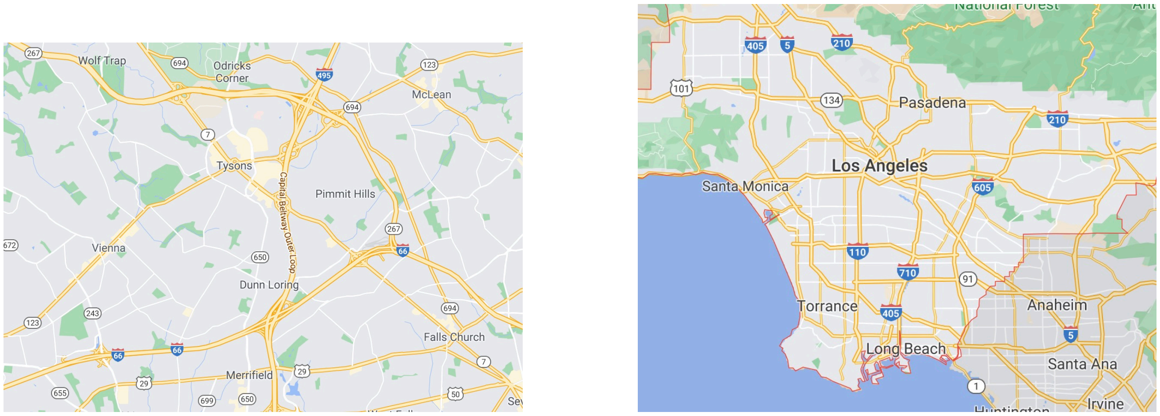

Figure 4.1: Tyson's Corner on google map (left) and Los Angeles County on google map (right)

handle null values, duplicate values and missing values in the original data. Third, we clean up road construction data and keep work zone data that has no traffic accidents in order to make sure our data is only related to construction work. Fourth, we use Min-Max normalization to train traffic speed to prevent data imbalance and collect the traffic speed readings of sensors every 5 minutes. If there is an hour duration of construction work, 12 consecutive fragments are collected. Fifth, for each road construction data, we collect traffic speeds for 131 sensors during road construction. Furthermore, we shuffle the traffic speed data to make sure our data is balanced and use 70% of the data for training, 10% for validation and 20% for testing. Finally, our dataset includes normalized traffic speeds related to the construction work $X^s_{v_i,t_j}$, a construction matrix which is used to record whether there are road construction data in the traffic road $X^c_{v_i,t_j}$ and a road segment matrix $S \in R^{N \times N}$ that records the relationship between each sensor.

The Los Angeles dataset is provided by [53], which is based on the highway of Los Angles County. It has 207 sensors and only provides traffic speed is from Mar.1 to Mar.7, 2012 with the 5 minutes interval of the time step. Thus, we use this dataset to test whether our model has good performance on traffic speed forecasting without using the construction work feature map.

4.2 Baselines

We compare our model with the following baseline methods like traditional machine learning methods and some graph neural network methods that have a good performance on the traffic speed. The difference between our model and these baselines is that our model not only uses the latest methods in transportation such as multi-head attentions, graph convolution operation to capture spatio-temporal correlations, but also uses additional feature map to provide more information to the model. In our project, we divide the model into two parts, STGCN-WZ and STGCN-WZ (No), which means that one part uses historical speed and construction work feature maps to predict speed, while the other part uses only history speed to predict speed.

- **TGCN[53]:** It uses a graph convolutional network to learn complex topological structures and uses gate recurrent unit to learn dynamic traffic flow.

- **STGCN[35]:** A "sandwich" structure that using two gated sequential convolution layers and one spatial graph convolution layer to capture the spatial-temporal correlations.

- **GraphWaveNet[54]:** It uses an adaptive dependency matrix and node embedding to capture the hidden spatial dependency in the data, and then feeds the information to dilated casual graph convolution.

- **ASTGCN[39]:** Based on attention mechanism, it adopts Chebyshev polynomials into graph convolution network to capture the dynamic spatial changes and uses a standard convolution neural network to capture the temporal correlations.

4.3 Evaluation Metrics

We use three metrics to evaluate the traffic speed prediction performance of the STGCN-WZ: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). RMSE and MAE are used to measure the error between the prediction with label speed. If the RMSE in our model is 4, it means that the speed we predict is 4 mph different from the real speed. MAPE considers not only the error between the prediction and the label speed, but also the ratio of the error to the true value. The smaller the value, the better the prediction performance.

4.4 Hyper-parameters

Following the previous work such as [40], [48], [53], the number of historical time steps for training H is 3, 6 and 12, and the number of time steps needed to be predicted P is also 3, 6, 12. In the optimizer, we use Adam optimizer with the initial learning rate of 0.001 in order to improve the efficiency of convergence. In the construction work feature map, we initially set $\lambda = 2$ as the hyper-parameter of setting the influence of construction work.

4.5 Experimental Research

In this section, we will evaluate the model and other baselines performance on both datasets. There are the following five research questions and solutions in the experiment. Q1, for example, is the solution corresponding to the question 1.

- Q1: Which model is the base model and why should we use it?
- Q2: What is the difference between our model and the base model?
- Q3: Compared with the baselines, how does STGCN-WZ and STGCN-WZ(No) perform in Tyson dataset?
- Q4: How does the model perform in other dataset without using the work zone feature map?
- Q5: Can STGCN-WZ provide reasonable explanations about the experiment results?

4.5.1 Basic model(Q1)

The model we use is based on [39], named attention based spatial-temporal graph convolutional network (ASTGCN) model. It mainly uses the attention approach to capture the spatial-temporal dependencies and uses one of the graph convolution operations named Chebyshev and a standard convolution layer to merge the traffic information to predict the speed. The reason we use is that the approach in spatial-temporal attention is the latest and model performance exceeds the state-of-the-art baselines in 2019. Furthermore, it also provides a good method that integrates the output of three components, which is hourly-period component, daily-period component and weekly-period component, with learning parameters to predict traffic speed. For example, it uses three learning parameters W_h, W_d, W_w to reflect the degree of influence of the three time components, which correspond to the speed per hour, the speed from 8:00 AM to 9:00 AM per day, and the speed from 8:00 AM to 9:00 AM per Monday. The final prediction result can be written as:

$$S = W_h \odot S_h + W_d \odot S_d + W_w \odot S_w, \tag{4.1}$$

where $\odot$ is the Hadmard product, W_h, W_d and W_w are learning parameters and S represents the traffic speed. Interestingly, this approach does not improve the traffic forecasting model. However, it provides a way that incorporate multiple attributes in the traffic. In order to evaluate the traffic performance based on the construction work, we integrate multiple attributes as the input to provide more traffic information in the model.

4.5.2 The difference between ASTGCN and STGCN-WZ(Q2)

Differing from the dataset in ASTGCN, the Tyson dataset consists of a speed feature and a construction speed, so historical traffic speed mainly comes from the traffic speed when construction work occurs. In the model structure, there are three main differences between ASTGCN and our mode. First, ASTGCN use spatial-temporal attention described in chapter 3.2 and a traditional graph convolution operation

$$(x * g)_G = \sum_{k=0}^{K} \theta_k T_K(\widehat{L})x, \tag{4.2}$$

where K = 3. While in STGCN-WZ, we set two learnable weights as speed weight W_s and construction weight W_c, using Hadamard product to add these together as speed wave: $\hat{X}_s = W_s \odot X^s + W_c \odot X^c$. Second, ASTGCN model use a simple attention approach in spatial-temporal correlation and use Chebyshev as a graph convolution operation, but our model compares the attention with multi-head attention to find which attention is better. At the same time we use 1stChebNet operation to improve computational efficiency. Third, in the temporal convolution layer, ASTGCN use a standard convolution to fit the predicted dimensions, but we use a 1×1 convolution layer and use bidirectional RNN to decrease the dimension of output and learn the time series of each node, then we use a linear function to fit the same dimension with the prediction. The reason we use bidirectional RNN is that the trend of traffic speed are not only related to the previous state, but also to the future state, we can predict not only the future according to the current speed, but also use the current speed to predict the speed that has occurred before. Unless the speed of traffic is affected by abnormal factors, each road section of traffic has its own certain trend of change. After using Bi-LSTM, we use simple linear function to adjust the output dimension to make it equal to the traffic label.

4.5.3 Performance Comparison between STGCN-WZ and baseline(Q3)

We divide our model into two parts: STGCN-WZ represents that we use speed and work zone feature map to predict the speed and STGCN-WZ(No) denotes that we only use speed as input to predict the speed. For the baselines, it is noted that these only use traffic speed as input to traffic forecasting due to the lack of construction work information. The purpose of comparison is to evaluate whether our model can effectively predict traffic and whether adding new feature maps can improve the results of traffic prediction. The overall performance evaluations of the baselines and our model are reported in Table 4.1. From the results, we can see that our model achieve the best performance in all evaluation metrics. In addition, we also have following findings: First, the longer the predicted time steps t', the worse the model's result, especially in GraphWaveNet model and T-GCN model, because GraphWaveNet is the autoregressive generative model, thus the accuracy of predicting long-term will decrease sharply, and T-GCN uses the traditional GRU model to learn temporal features and use graph convolution operation to learn spatial dependence, so the performance of the model cannot connect the previous time sequence and capture adjacent nodes correlations. Second, compared with the models such as T-GCN, STGCN and GraphWaveNet, our model and ASTGCN model show that the attention mechanism can help the model predict traffic speed more effectively. This is because the attention mechanism is the same as the human visual system, which focus limited attention on key information, thereby it helps the model obtain effective information quickly. Also, it helps the model that each step of the calculation does not rely on the previous calculation results, thus the problem related to sequential data can be handled with recurrent neural network in parallel. Third, compared with ASTGCN and STGCN-WZ(No), the performance of STGCN-WZ(No), which use speed as the input to predict, is slightly better than ASTGCN, this is because STGCN-WZ(No) uses first-order ChebNet as a graph convolution operation and Bi-LSTM in the last step, thereby greatly reducing the parameters and extracting spatial-temporal sequence features in parallel. When we compare ASTGCN and STGCN-WZ, the result shows that our model has significant improvement. For example, at 3 time steps, STGCN-WZ improves over the model ASTGCN by 0.14 and over the model STGCN-WZ(No) by 0.06 with respect to RMSE. Fourth, intuitively, the addition of construction feature map designed from Chapter 3.1 can effectively help the model to analyze analyze the relevance of traffic. But the challenge is that how do we design a suitable and exactly construction feature map. To better predict traffic road conditions, we need not only to involve more appropriate models, but also to collect and process traffic data more accurately.

Table 4.1: Performance of the baselines and our model on the Tyson's Corner dataset

models	15 min($t^{'} = 3$)			30 min($t^{'} = 6$)			60 min($t^{'} = 12$)		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
T-GCN	5.36	3.39	9.90	6.04	3.56	11.21	6.82	4.32	12.57
STGCN	5.10	3.30	10.01	5.64	3.42	11.07	6.58	4.25	13.32
GraphWaveNet	4.53	2.92	9.04	5.82	3.52	11.68	6.90	4.35	13.87
ASTGCN	4.14	2.75	8.65	5.12	3.16	10.37	5.95	3.57	11.21
STGCN-WZ(No)	4.06	2.68	8.37	4.96	3.09	10.79	5.88	3.56	10.93
STGCN-WZ	**3.99**	**2.60**	**8.20**	**4.87**	**3.08**	**9.54**	**5.73**	**3.51**	**10.65**

4.5.4 Performance comparison in Los-loop dataset(Q4)

Table 4.2 shows traffic flow prediction performance in Los-loop dataset. Because the dataset provide the speed information, we use STGCN-WZ(No) to compare other models. Unlikely with Tyson dataset, Los-loop dataset is primarily focused on a short period of traffic speeds, and the traffic speeds provided are basically less affected by natural disasters, car accidents, and human factors than Tyson's environment. Hence, we aims to evaluate our model with baselines to check whether our model well predict the traffic speed under less traffic incidents. According to the results, we find that the ASTGCN model and STGCN-WZ(No) model are better than other baselines due to the spatial and temporal attention. For example, for the 15-min traffic speed, the MAE errors of the ASTGCN model and the STGCN-WZ(No) model are nearly 1.4 lower than T-GCN and STGCN, and for the 60 min, the MAE of the best models are 1.7 lower than the worst models. Compare each model in MAE horizontally, we find that the MAE increased by 0.6 at a time in the T-GCN model when time steps go from 3 to 6 and 6 to 12 while our model only increase 0.4 at a time. In terms of the ASTGCN model and the STGCN-WZ(No) model in Los-loop dataset, we find the performance of our model which only use speed as input is also better than the based model. This proves that our model has excellent performance in predicting traffic speed regardless of the traffic environment.

Table 4.2: Performance of the baselines and our model on the Los-loop dataset

models	15 min($t^{'}$ = 3)			30 min($t^{'}$ = 6)			60 min($t^{'}$ = 12)		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
T-GCN	6.18	4.16	11.04	6.93	4.82	13.74	8.26	5.41	15.04
STGCN	6.27	4.18	11.92	6.88	4.74	12.53	7.99	5.37	14.14
GraphWaveNet	5.37	3.00	09.77	6.65	3.72	12.35	7.90	4.71	14.00
ASTGCN	5.20	3.04	07.87	6.13	3.51	10.11	7.57	4.42	12.89
STGCN-WZ(No)	**4.97**	**2.94**	**07.76**	**5.84**	**3.28**	**09.07**	**7.28**	**4.38**	**12.71**

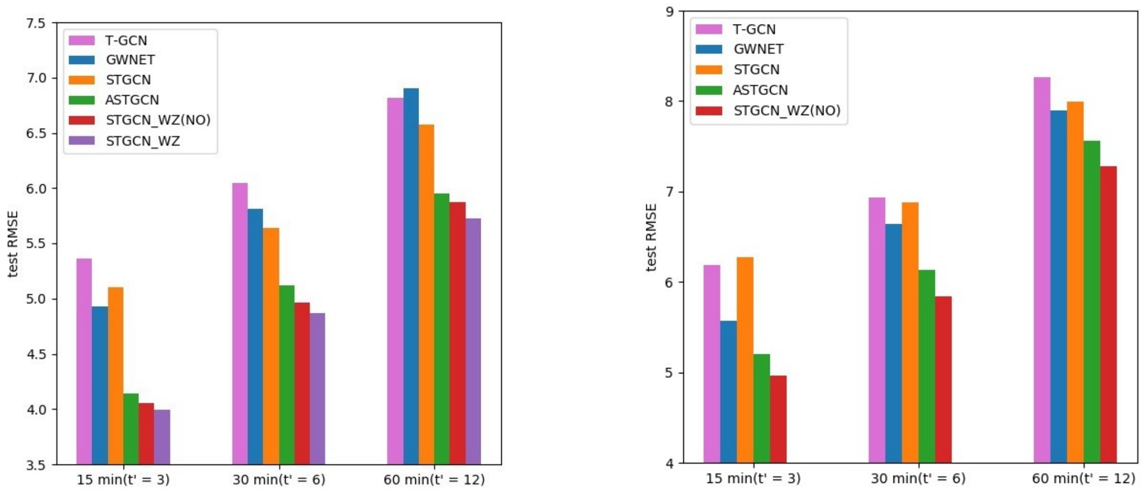

Figure 4.2: The RMSE of the models in Tyson's Corner (left) and Los-loop dataset (right)

4.5.5 Model evaluation and analysis (Q5)

We show several figures to compare our model with other baselines. First, Fig 4.2 shows the RMSE of the models mentioned above at different time steps in Tyson and Los dataset, where STGCN_WZ(NO) means that only the speed is fed into the model to predict the traffic speed. From the bar chart, it is intuitively found that our model in Tyson performs better than others and our model which only using speed to predict speed is similar to the base model in Los dataset. Second, we compare the performance of the model at different time steps. Fig 4.3 shows the results of training RMSE and validation RMSE, where (Val)P = 3 means that the predicted time steps is 3, and the validation RMSE during the training epoch. As we can see, short-term time prediction using STGCN-WZ method is better than long-term time prediction, and our model converges faster than other benchmarks and performs best during training. Next, we show the result of speed prediction in three road segments named '110-0475', '110+04177', '110P04611' on two days, which is from 11:00 AM to 5:00 PM, 2019-11-19 and 8:20 AM to 5:00 PM, 2019-11-22 in Fig 4.4, 4.5, 4.6. From those figures, it is obviously found that our proposal model accurately captures the trend of traffic speed, the shorter the time step, the much similar the traffic speed to the real speed, this means that our model has more accurate results in processing short-term traffic speed prediction. In Figure 4.4, the result shows that, when time step is 3, 6 and 12, there is no much difference between the predicted speed and the real speed. However in rush hours of Figure 4.5 and Figure 4.6, the difference between the predicted speed and the real speed gradually increases when the time step is longer. Since we have 131 road segments in Tyson, we use a heat map to evaluate the performance of each road segments using MSE as evaluation metrics. Due to limited pictures, we only show the heat map of some roads. From Fig 4.7, the y-axis represents the name of a road segment in the real-world and the x-axis represents the prediction of the i-th time period. The far right represents the error line, using MSE as the evaluation metrics. The darker the color and the larger the number, the worse the prediction. In this figure, we find that the performance of highway prediction is not good as the speed prediction of non-highway, especially for some intersections, this is because there are many uncontrollable factors on the highway. For example, the results of the road segments named "110+04174" and "110P04177" on the I-66 highway are worst than the result of road segments named "110+05695" on Virginia's 7th congressional district. Also, different highways have different prediction performance, for example, the performance of I-495 highway is better than I-66 highway, for the reason that the traffic flow of I-66 is much heavy than that of I-495, especially during peak times. Furthermore, the performance of the model gradually weakens as the predicted time period is longer, in other words, the performance of the first period predicted by the model is much better than that of the last time period.

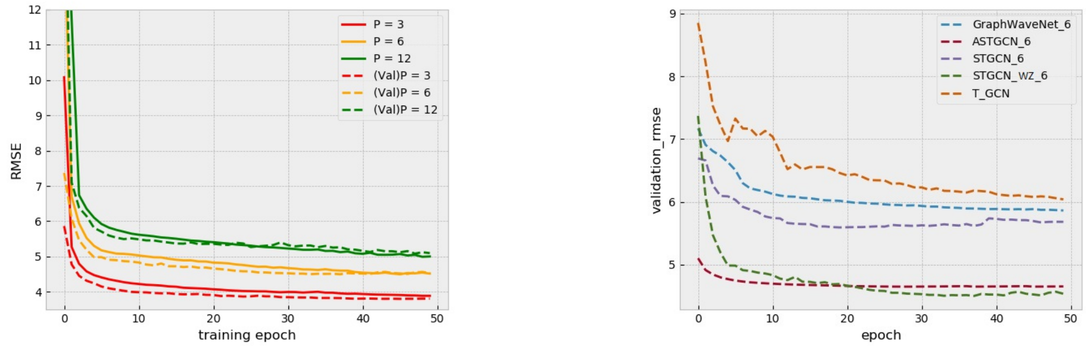

Figure 4.3: RMSE in the STGCN-WZ model training set (left) and RMSE in the STGCN-WZ model validation set (right)

4.6 Ablation Studies

In the ablation studies, we use different parameters to effectively find the best performance in our model. We construct four studies, mainly about setting the construction work feature map, speed wave in, spatial-temporal attention and temporal convolution operation. The studies are summarized as follows:

- **STGCN-WZ-I** use $\lambda = 5,\ 7$ described in the chapter 3.1

We build a function that setting the influence for each construction work X^c, written as:

$$X^c_{v \in V t_j} = \begin{cases} \max(0, 1 - (\frac{dis(v,v_i)}{\lambda})^2), & \text{if there is construction work } X^c_{v_i t_j} \\ 0, & \text{if there is no construction work } X^c_{v_i t_j} \end{cases}, \tag{4.3}$$

where λ is a hyper-parameter, $dis(v_i, v_j)$ defines the distance between two nodes v_i and v_j. This helps the model understand the maximum impact of construction on the road. If λ is 3, it means that the construction will not have any impact beyond two miles. The closest to the construction, the most affected by the construction work. Table 4.3 shows the evaluation of our model using $\lambda = 5, 7$. In the result, we find that $\lambda = 3$ is better

Table 4.3: The ablation of finding the best influence factor on construction work by setting different λ

	RMSE	MAE	MAPE
$\lambda = 3$	**4.87**	**3.08**	**9.54%**
$\lambda = 5$	4.98	3.18	11.83%
$\lambda = 7$	5.12	3.21	12.18%

Table 4.4: The ablation of two speed wave $\hat{X}_s$ function to check whether building a learnable parameter is useful to traffic prediction

	RMSE	MAE	MAPE
$W_s \odot X^s + W_c \odot X^c$	**4.87**	**3.08**	**9.54%**
$X^s + W_c \odot X^c$	4.89	3.08	9.85%

Table 4.5: The evaluation of using a simple convolution method compared with linear function

	RMSE	MAE	MAPE
Linear function	**4.87**	**3.08**	**9.54%**
1×1 Convolution	4.98	3.10	10.99%

than others, which proves that when λ is 3, the effect is more accurate. Thus, a well-design feature map can significantly improve the performance of traffic prediction.

- **STGCN-WZ-II** use a new speed wave function described as $X^s + X^c \odot W_c$

For the speed wave $\hat{X}_s = W_s \odot X^s + W_c \odot X^c$ mentioned in Chapter 4.5.2, we do not use the weight of learning speed W_s and show the result is in Table 4.4. As we can see, speed wave function $\hat{X}_s = W_s \odot X^s + W_c \odot X^c$ is slightly better than another function.

- **STGCN-WZ-III** use Chebyshev polynomials described in chapter 3.3.1 instead of first-order ChebNet.

In Fig 4.8, we show the training time in STGCN-WZ which use first-order ChebNet and the training time in ASTGCN which use Chebyshev polynomials. As we can see, the training time of ASTGCN dropped from nearly 450 seconds per epoch to 300 seconds per epoch, however, the training time of our model only need 38 seconds per epoch to train. It is obviously found that using first-order ChebNet can dramatically speed up the running speed of the model.

- **STGCN-WZ-IV** use 1×1 standard convolution rather than a linear function

In Chapter 3.3.2, we use a linear function to match the dimension of output with the label. Here we use a standard convolution to evaluate the performance, shown as Table 4.5. We find that using a linear function is better than using a 1×1 convolution method.

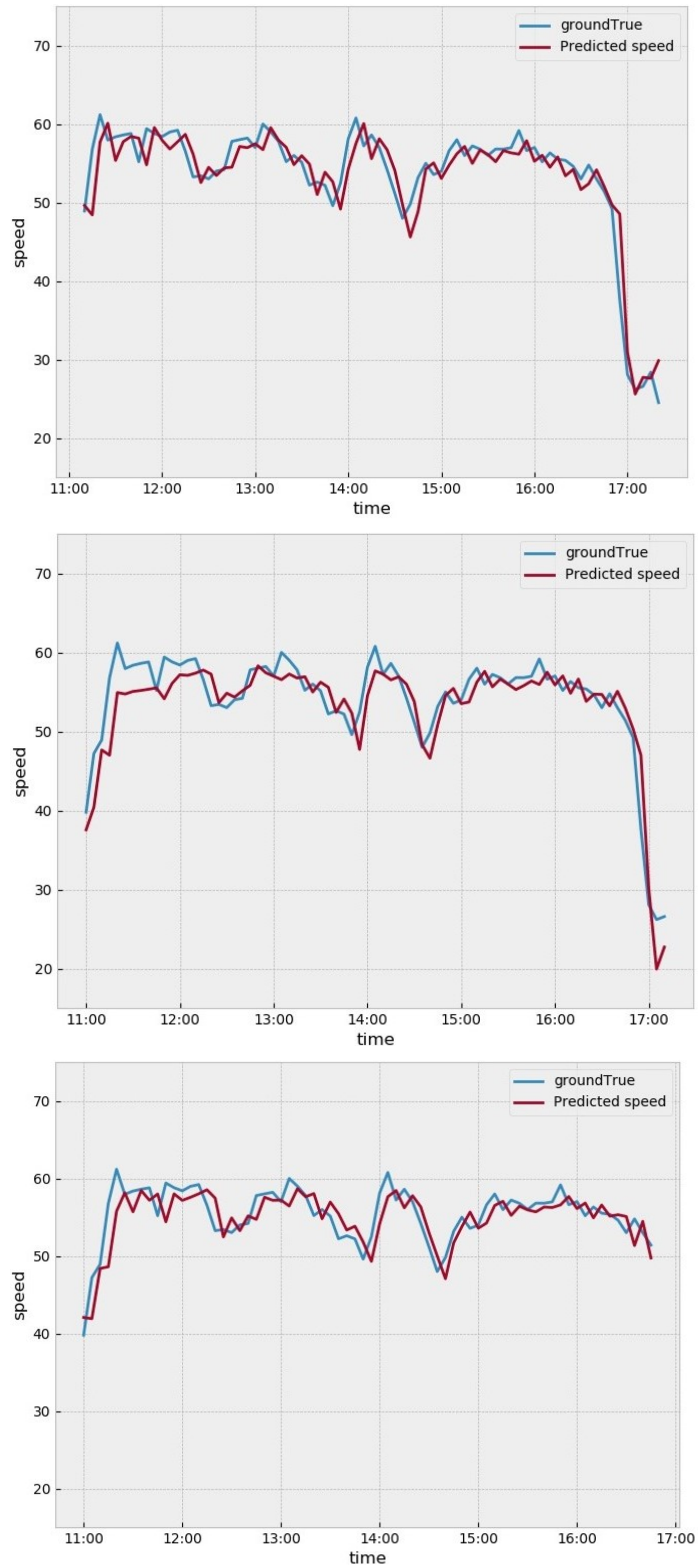

Figure 4.4: Traffic speed on road segment '110-04175' , 11/19/2019, $t' = 3$ (left), 6 (middle), 12 (right)

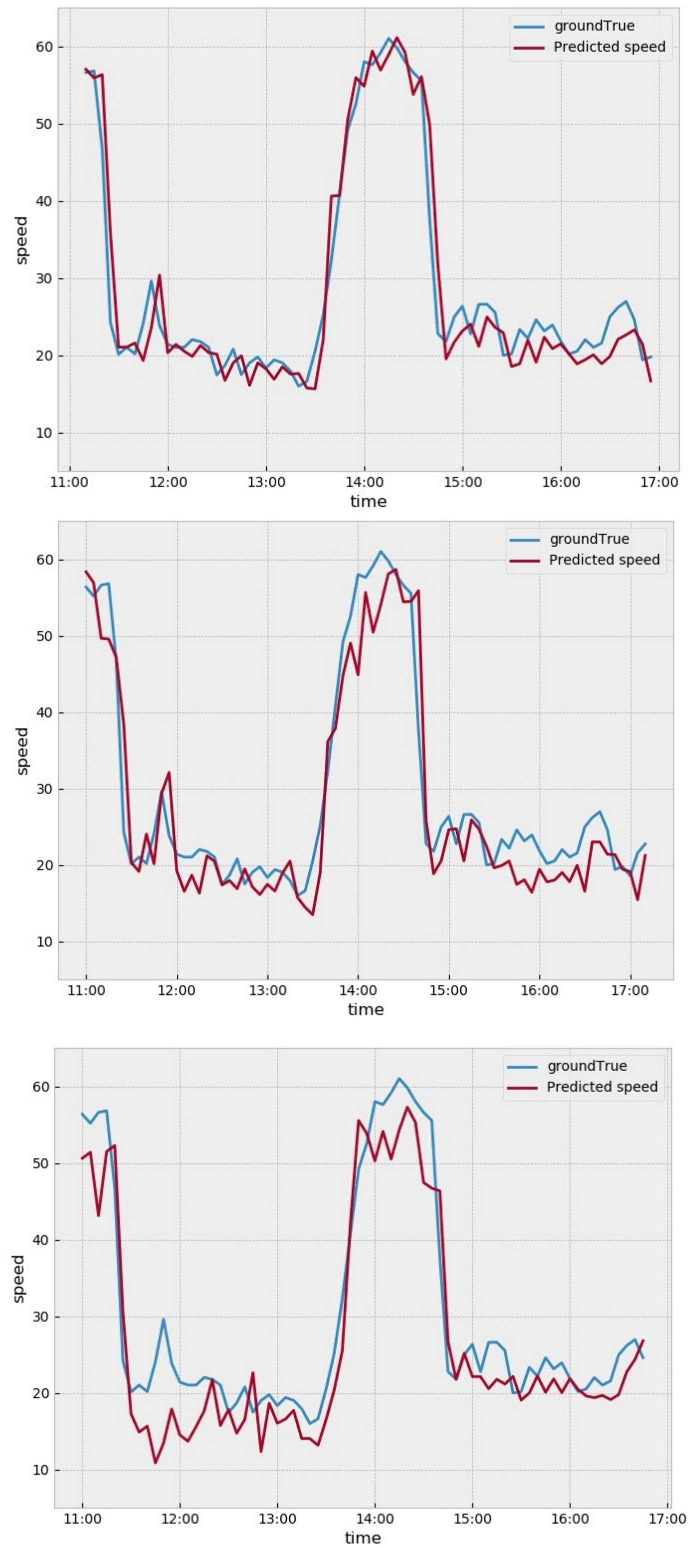

Figure 4.5: Traffic speed on road segment '110+04177', 11/22/2019, $t^{'} = 3$ (left), 6 (middle), 12 (right)

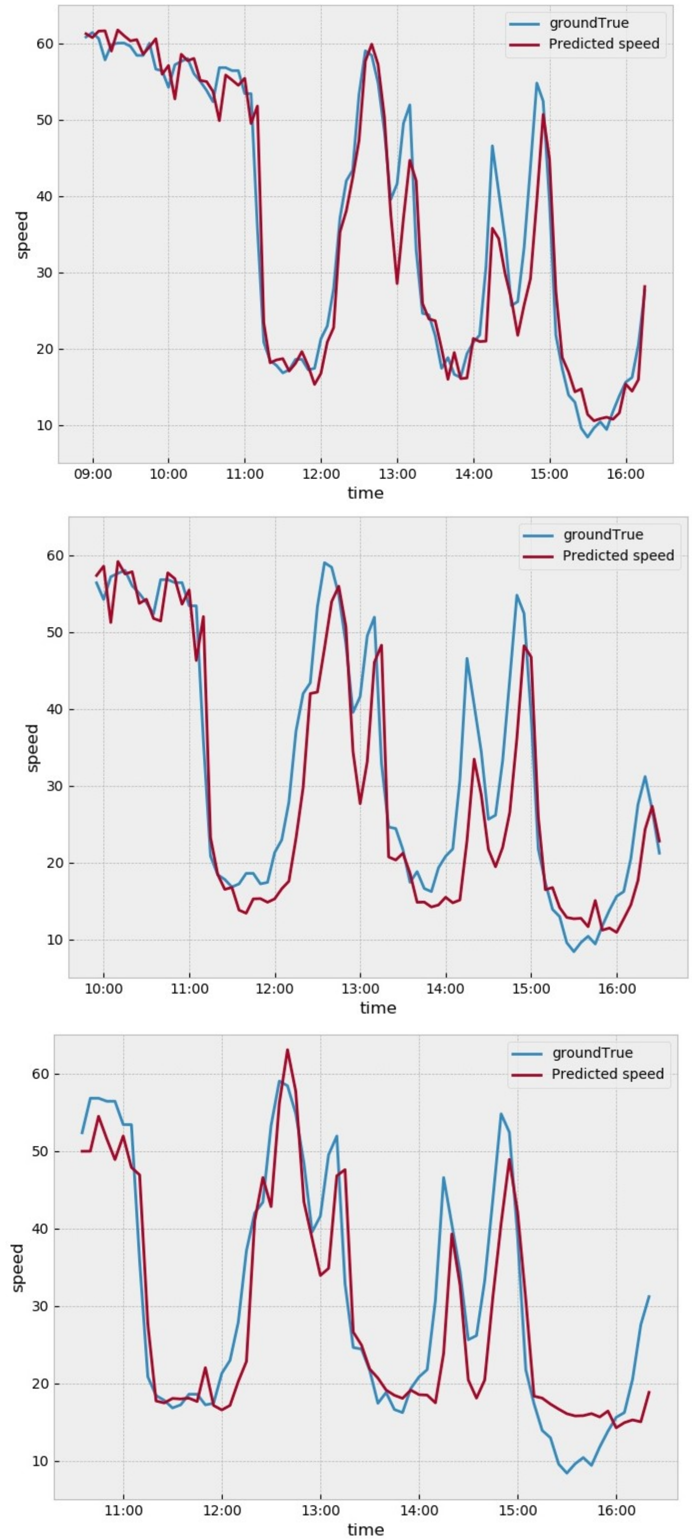

Figure 4.6: Traffic speed on road segment '110P04611', 11/22/2019, $t^{'} = 3$ (left), 6 (middle), 12 (right)

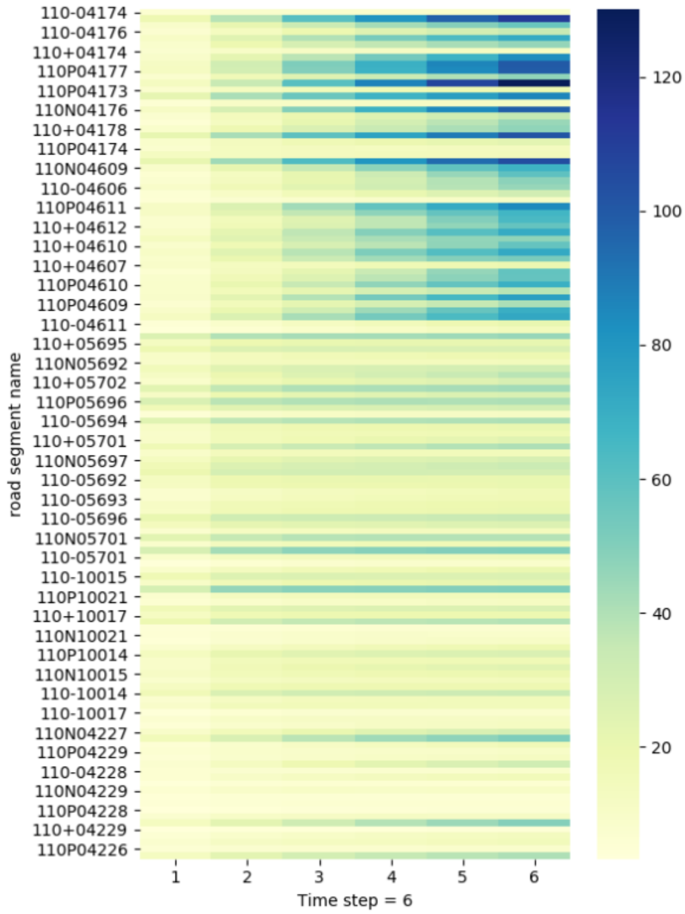

Figure 4.7: The heatmap of speed prediction using MSE at time step = 6 (The x-axis represents the predicted i-th time period, the right side represent the error bar)

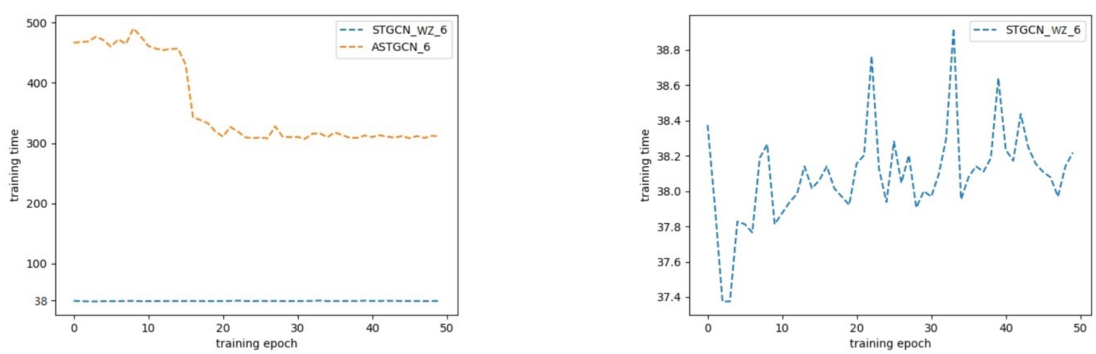

Figure 4.8: The training time between STGCN-WZ and ASTGCN (left) and training time in STGCN-WZ (right)

Chapter 5: Conclusion

In this project, we propose a graph convolutional network based on the impact of construction work to predict traffic speed. Unlike previous traffic speed studies, this project uses multiple regression feature maps to study the impact of construction work on the surrounding area. From the experiment based on two datasets and four baselines, there are four following aspects we find in this research:

First, according to the result on two datasets, we show that our model has better performance than other baselines in terms of predicting traffic speed regardless of the traffic environment and prove that adding a new feature matrix can effectively help the graph neural network learn the traffic speed under the influence of the work zone area better.

Second, according to historical literature and model analysis, graph neural networks have powerful advantages in the transportation field compared to other deep learning methods. It can not only capture the traffic spatial-temporal correlations, but also handle time-series problems. The combination of convolutional neural network and natural language processing approaches can help models predict data in irregular datasets. Third, according to the heap map, the prediction result based on the highway speed is not good as other roads in long-term prediction, which can be explained that highway speed are affected by multi abnormal events, especially at traffic intersections. Furthermore, due to the different results of two highway, we found that not all roads will be affected by construction. Quantifying the impact of construction on the surrounding area depends on whether the data is clean, detailed and whether the traffic characteristics are accurate.

Fourth, although sliding windows is used to cut time series to obtain data input and output, there will inevitably be a phenomenon, that is, the input speed information is normal, but the label speed is abnormal. In other words, when we train a sequence that has T time steps and use P time steps as label speed, the sequence for training is not effected by construction work while the data in the label sequence is effected. However, based on the model, the normal speed is used to obtain the output, which will be very different from the label speed. This explains why each model cannot accurately predict the future traffic speed.

In the future work, we will design a more detailed and representative construction feature map to help the model predict traffic speed. Then we are going to predict traffic speed under multiple factors such as weather conditions and accidents occur simultaneously to find that whether multi feature maps will cause the better results and better simulations of road conditions. In terms of neural network, we will design a better network based on the latest method based on natural language processing to handle the time-series problem better.

Bibliography

Bibliography

[1] D. Schrank, B. Eisele, and T. Lomax, "2019 urban mobility report," p. 50.

[2] B. Du, S. Chien, J. Lee, and L. Spasovic, "Predicting freeway work zone delays and costs with a hybrid machine-learning model," *Journal of Advanced Transportation*, vol. 2017, 2017.

[3] D.-x. SHI, T.-j. DING, B. DING, and H. LIU, "Traffic speed forecasting method based on nonparametric regression," *Computer Science*, no. 2, p. 47, 2016.

[4] A. Pell, A. Meingast, and O. Schauer, "Trends in real-time traffic simulation," *Transportation research procedia*, vol. 25, pp. 1477–1484, 2017.

[5] N. N. N. Azlan and M. M. Rohani, "Overview of application of traffic simulation model," in *MATEC Web of Conferences*, vol. 150. EDP Sciences, 2018, p. 03006.

[6] F. Dion, K. Sivakumaran, and X. J. Ban, "Evaluation of traffic simulation model use in the development of corridor system management plans (csmps)," Tech. Rep., 2012.

[7] M. Tong and H. Xue, "Highway traffic volume forecasting based on seasonal arima model," *Journal of Highway and Transportation Research and Development (English Edition)*, vol. 3, no. 2, pp. 109–112, 2008.

[8] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[9] B. M. Williams, "Multivariate vehicular traffic flow prediction: evaluation of arimax modeling," *Transportation Research Record*, vol. 1776, no. 1, pp. 194–200, 2001.

[10] S. Sun, G. Yu, and C. Zhang, "Short-term traffic flow forecasting using sampling markov chain method with incomplete data," in *IEEE Intelligent Vehicles Symposium, 2004*. IEEE, 2004, pp. 437–441.

[11] P. Nie, Y. Zhi, and H. Zhao-Cheng, "Constrained kalman filter combined predictor for short-term traffic flow," *Journal of Traffic & Transportation Engineering*, vol. 8, no. 5, pp. 86–90, 2008.

[12] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 62, pp. 21–34, 2016.

[13] C.-H. Wu, J.-M. Ho, and D.-T. Lee, “Travel-time prediction with support vector regression,” *IEEE transactions on intelligent transportation systems*, vol. 5, no. 4, pp. 276–281, 2004.

[14] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, “Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions,” *Expert systems with applications*, vol. 36, no. 3, pp. 6164–6173, 2009.

[15] B. Du, “An artificial neural network model for predicting freeway work zone delays with big data,” 2016.

[16] S.-H. Huang and B. Ran, “An application of neural network on traffic speed prediction under adverse weather condition,” Ph.D. dissertation, University of Wisconsin–Madison, 2003.

[17] F. Moretti, S. Pizzuti, S. Panzieri, and M. Annunziato, “Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling,” *Neurocomputing*, vol. 167, pp. 3–7, 2015.

[18] X. Ma, H. Yu, Y. Wang, and Y. Wang, “Large-scale transportation network congestion evolution prediction using deep learning theory,” *PloS one*, vol. 10, no. 3, p. e0119044, 2015.

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

[20] D. Jo, B. Yu, H. Jeon, and K. Sohn, “Image-to-image learning to predict traffic speeds by considering area-wide spatio-temporal dependencies,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1188–1197, 2018.

[21] W. Genders and S. Razavi, “Using a deep reinforcement learning agent for traffic signal control,” *arXiv preprint arXiv:1611.01142*, 2016.

[22] M. Gori, G. Monfardini, and F. Scarselli, “A new model for learning in graph domains,” in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2. IEEE, 2005, pp. 729–734.

[23] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[24] C. Gallicchio and A. Micheli, “Graph echo state networks,” in *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2010, pp. 1–8.

[25] D. Gibson, M. K. Mills, and L. A. Klein, “A new look at sensors,” *Public roads*, vol. 71, no. 3, 2007.

[26] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, “A comprehensive survey on graph neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[27] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[28] Z. Kang, H. Xu, J. Hu, and X. Pei, "Learning dynamic graph embedding for traffic flow forecasting: A graph self-attentive method," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 2570–2576.

[29] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

[30] L. Ruiz, F. Gama, and A. Ribeiro, "Gated graph recurrent neural networks," *arXiv preprint arXiv:2002.01038*, 2020.

[31] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[32] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[33] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[34] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[35] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[36] A. Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009.

[37] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[38] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[39] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "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.

[40] F. Zhou, Q. Yang, K. Zhang, G. Trajcevski, T. Zhong, and A. Khokhar, "Reinforced spatio-temporal attentive graph neural networks for traffic forecasting," *IEEE Internet of Things Journal*, 2020.

[41] W. N. Anderson Jr and T. D. Morley, "Eigenvalues of the laplacian of a graph," *Linear and multilinear algebra*, vol. 18, no. 2, pp. 141–145, 1985.

[42] E. W. Weisstein, “Laplacian matrix,” *https://mathworld.wolfram.com/LaplacianMatrix.html*, 1999.

[43] ——, “Laplacian operation,” *https://mathworld.wolfram.com/Laplacian.html*, 1999.

[44] Y. Ma, J. Hao, Y. Yang, H. Li, J. Jin, and G. Chen, “Spectral-based graph convolutional network for directed graphs,” *arXiv preprint arXiv:1907.08990*, 2019.

[45] D. I. Shuman, B. Ricaud, and P. Vandergheynst, “A windowed graph fourier transform,” in *2012 IEEE Statistical Signal Processing Workshop (SSP)*. Ieee, 2012, pp. 133–136.

[46] E. Weisstein, “Fourier transform,” https://mathworld.wolfram.com/FourierTransform.html, accessed Oct 5, 2020.

[47] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[48] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, “Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 890–897.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[50] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[51] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[52] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.

[53] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, “T-gcn: A temporal graph convolutional network for traffic prediction,” *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[54] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph wavenet for deep spatial-temporal graph modeling,” *arXiv preprint arXiv:1906.00121*, 2019.