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

by

Yuanjie Lu
A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Master of Science
Computer Science

Committee:

_______________ Dr. Amarda Shehu, Thesis Director

_______________ Dr. David Lattanzi, Committee Member

_______________ Dr. Zoran Duric, Committee Member

_______________ Dr. Stu Vwx, Chairman, Department of Computer Science

_______________ Dr. Kenneth S.Ball, Associate Dean for Research and Graduate Studies

Date: _______________ Fall Semester 2020
George Mason University
Fairfax, VA

Predicting traffic speed based on graph convolutional networks under the impact of maintenance downtime

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. Amarda Shehu, Professor
Department of Computer Science

Fall Semester 2020
George Mason University
Fairfax, VA

Dedication

I dedicate this thesis to ...

Acknowledgments

I spent one and a half years in master research since I graduate from INTO Mason program which is related to the English pathway program. After more than two years at George Mason University, I not only improved my English and academic capability, but also become a persistent, diligent and innovative researcher. I am really appreciate Dr. Amarda Shehu and Dr. David Lattanzi, who provided me with an opportunity to do a research in Artificial Intelligence and taught me in a long process. Especially appreciate for Dr. David Lattanzi, who supported me and funded me to work in an application project. He is an advisor who teaches 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, but he never cared about it, 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 BASED ON GRAPH CONVOLUTIONAL NETWORKS UNDER THE IMPACT OF MAINTENANCE DOWNTIME

Yuanjie Lu

George Mason University, 2020

Thesis Director: Dr. Amarda Shehu

This paper proposes a graph convolutional network (GCN)-based method that learns the effect of maintenance downtime on the surrounding area using traffic speed as performance metrics. It is important to evaluate the work zone downtime effect because construction at an inappropriate time will cause congestion on the traffic road and also cause disturbance to the drivers, trigger traffic accidents, and further affect traffic in a vicious circle. It is also essential to learn the correlation between each neighbor road, because the construction may not affect the surrounding areas, but in other places, the construction work effect is vast. However, in early traffic forecasting studies, there are many time-series models used to accommodate autocorrelations between traffic states, but these models do not involve spatial and temporal structures, which causes the problem of finding the correlation between each road segment is hard. In addition, few studies are used to measure the impact of traffic construction on the surrounding area. To predict traffic speed under the impact of construction work, we design a novel graph-based neural network based on the graph convolutional neural network (GCN) to accommodate the Spatial-temporal dependencies among traffic states, differentiate the intensity of connecting to neighbor roads and predict

the speed under the road maintenance condition. The advantage of using a graph convolutional network is that it can transfer the information between each road segment and adjacent segments, and automatically learn the features and structure of the graph. In the traffic network, each road segment represents a node and shares its parameters throughout the network. The more nodes, the more layers, and the more information involved in the calculation. Use the novel GCN model to construct in a node, and to find the speed of adjacent nodes, then compare the predictive speed according to historical data, we can find out the influence of setting maintenance in the node. In the experiment, we evaluate our model with four baselines on two real-world dataset collected from the road sensor network in Tyson's Corner and Los Angles. The results show that our model is better than other benchmarks in predicting traffic speed regardless of whether the traffic environment is affected by the traffic incidents.

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 the ten years. In 2017, the average auto commuter spent 56 hours in the traffic congestion and wasted 21 gallons of fuel due to 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 the nonrecurring delay congestion [2]. Due to the economic loss of traffic incidents, it is necessary to quantify the associated mobility impact of construction work scheduled maintenance in order to enhance developed traffic management and help drivers make informed choices to reduce traffic congestion.

1.2 Research problem

However, there are various challenges in traffic forecasting based on road maintenance, such as distinguishing between the normal with abnormal traffic conditions, calculating the complexity of traffic networks, forming an efficient and intelligent methodology based on the latest technology and reflecting the temporal and spatial correlation under the influence of traffic maintenance. Unfortunately, many traditional traffic prediction models are only characterized by traffic speed, which makes the model unable to distinguish whether the input data is affected by incidents, so that the model cannot provide accurate traffic information due to the outliers and noisy data. In addition, even if clean data is provided to models, these models can only be used to predict traffic speed that are not affected by any incidents. To predict the future speed under the impact of construction work, it is critical to clean the data based on the construction conditions. Although there are many studies used to predict the work zone traffic speed, they mainly focus on traditional machine learning methods or traditional traffic simulation and they only study the impact of work construction on the work zone activity area and upstream segments but ignore further impacts on intersecting roads, and in the surrounding region. Nowadays, more and more people use deep learning methods to predict traffic speed, but still few studies use the method to measure the impact of traffic construction on the surrounding area. If we extend the latest approach to provide work zone traffic data, then in the future, we can also improve it to handle the traffic speed under multiple incidents, so that traffic will be more efficient and convenient.

1.3 Contribution

In this paper, we spend a long time cleaning and organizing a Tyson' Corner dataset which is based on the traffic speed under the influence of the construction work area. In aiming to predict the speed based on am external environment, we propose a framework based on the graph neural network, called STGCN-WZ (Spatial-temporal graph convolution network for traffic forecasting within the work zone). We add a construction work feature map to predict the traffic speed under various road geometric in work zone area. Differing from traditional approaches of studying downtime, the proposed model is based on work zone conditions to predict the spatial and temporal dependence of all adjacent roadways.

STGCN-WZ follows the sequence to sequence architecture, using traffic speed and construction work features to predict the feature speed sequence. This model requires information related to the work zone, including locations, initial time, and end time to establish feature matrix. The main contributions of this paper are summarized as follows:

- We develop a spatial and temporal attention mechanism into the traffic prediction to capture the spatial dependencies and temporal dependencies under the work zone downtime conditions.

- We use a window sliding approach to obtain traffic flow based on construction conditions and combine multiple feature matrices to capture spatial and temporal features from the original graph-based traffic network.

- We build a new real-world highway traffic dataset named Tyson's Corner and attempt to predict the traffic speed based on the Tyson dataset using the existing latest deep learning methods. The experiment demonstrates that adding a new construction feature matrix has better traffic-related knowledge than those methods without any traffic conditions.

1.4 Related work

In the existing literature, the field of research was 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 modelings, 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. Besides, building this model requires a significant computing capability under a large number of parameters such as location, traffic flow, density, shock waves, etc. 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 such as traffic accidents and reflect the nonlinearities and uncertainties of traffic data. Hence, it is powerless in large-scale and unclean 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-neighbor neighbors (KNN) approach 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-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 have 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 to predict delays while the SVM model is to predict work zone capacity [15]. Although these traditional models only need enough historical data to learn the statistical regularity automatically, researchers need to manually operate the traffic features, which requires them to have professional knowledge. In addition, these methods are limited to less complex traffic scenarios; they struggle to filter the data for traffic abnormalities.

To figure out the problem, 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. In the initial stages of 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, providing information such as location, traffic flow, etc. 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 [25]. 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 [25]. First, network embedding is used for mapping graph nodes to vectors in a low-dimensional and more discriminative space [26]. 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. [27] introduce a model that uses self-attention layers Gumbel-Softmax technology to learn graph-level spatial embedding. Zheng et al. [28] 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 [25]. Implementations of graph recurrent architecture can be found in [29], [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. [30] 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. [31] 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 [32], Variational Graph Auto-Encoders [33] and Spatial-temporal graph neural networks [34] 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. [35] 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[36]. In 2013, Bruna et al. [37] 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. [34] 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. [38] first attempt to apply attention method into the model to reflect the dynamic spatial-temporal correlations of traffic data. Zheng et al. [28] 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. [39] 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 Spatial-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

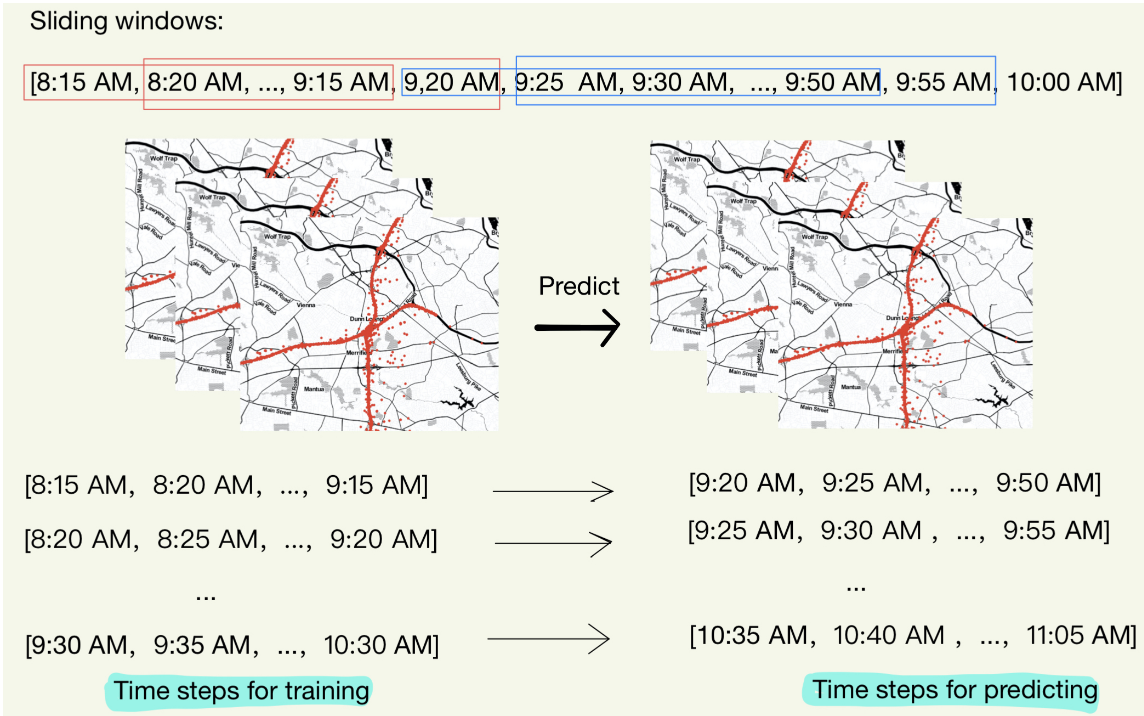

Figure 2.1: Data construction by using sliding window

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[40]. 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[41]:

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

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:

$$\triangle f = \begin{pmatrix} \triangle f_i \\ \vdots \\ \triangle f_N \end{pmatrix} = \begin{pmatrix} d_1 f_1 - w_{1:} f \\ \vdots \\ d_N f_N - w_{N:} f \end{pmatrix} = \begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_N \end{pmatrix} f - \begin{pmatrix} w_{1:} \\ \vdots \\ w_{N:} \end{pmatrix} f \tag{2.9}$$

$$= diag(d_i) f - W f = (D - W) f = L f$$

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 [42].

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[43]. This require to review the idea of an orthogonality of Fourier transform[44]. 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[43]. 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:

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

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

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

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[45]:

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

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 section, Fig 3.1 present the overall framework of the STGCN-WZ model in the paper. More details will be introduced later.

3.1 Construction Work Feature Map

In this part, we introduce more detailed structures about feature maps than Chapter 2. In the construction feature matrix $X^c_{v_i t_j}$, 0 and 1 represent whether construction has occurred in the road i and time j. In fact, this feature matrix is a sparse matrix because there is no much construction work that occurred at the same time. 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. 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} represents 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

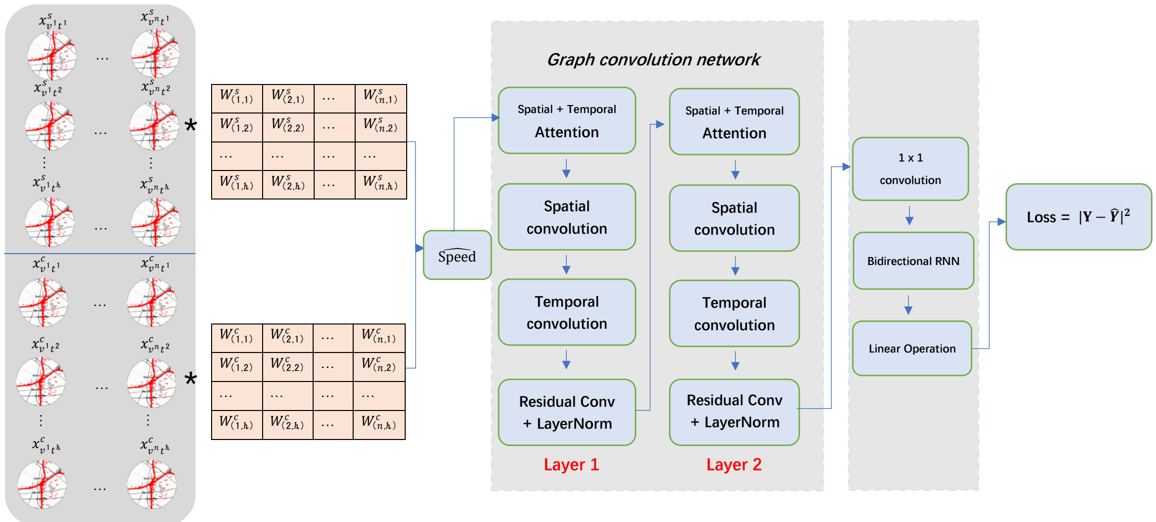

Figure 3.1: The framework of STGCN-WZ

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[46]. Some studies such as [47], [39], have used attention to capture the dynamic spatial and temporal relationships on the traffic road. In this project, we compare and use the spatial-temporal attention mechanism and the multi-head attention mechanism both, finally we choose multi-head attention.

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[38]:

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

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 company S with the adjacency matrix A to update the weight in the graph.

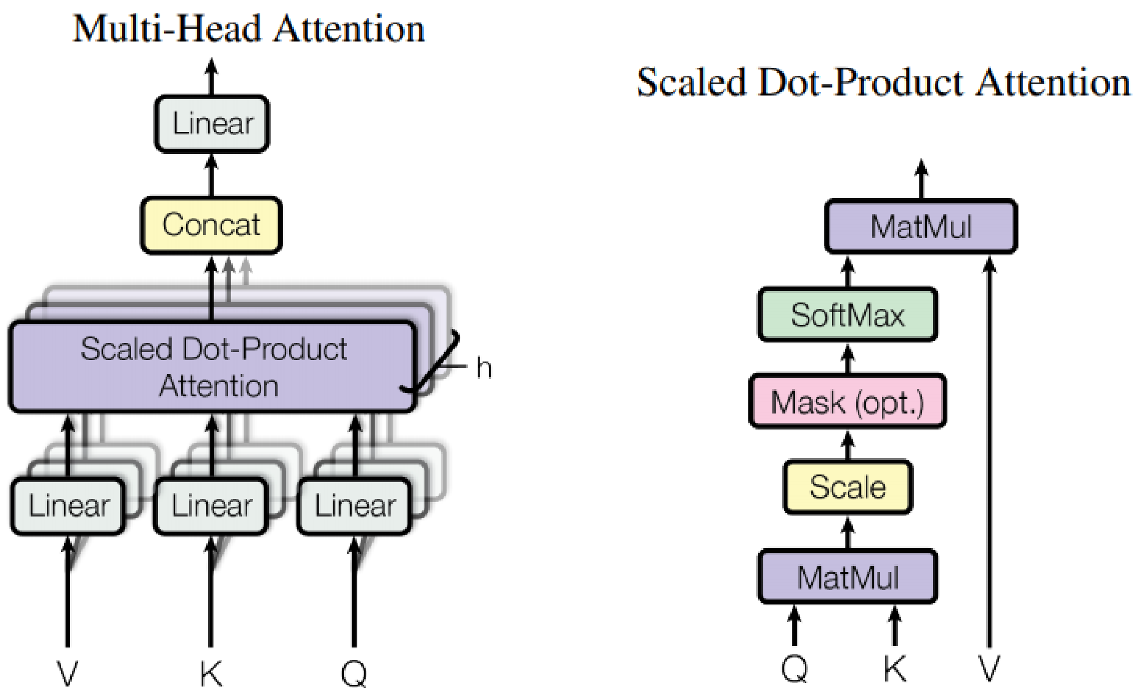

Figure 3.2: The RMSE of the models in Tyson's Corner (left) and Los-loop dataset (right)

3.2.3 Multi-head Spatial-temporal attention

Another approach to capture spatial and temporal dependencies is to use multi-head attention [48]. 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:

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

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 this 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. 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 [28].

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 $(f * g)_G = U g_\theta U^\intercal f$. 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 $U^\intercal$. Thus, a new method is proposed in [49], which uses Chebyshev polynomials to fit the convolution kernel to reduce the computational complexity. In [50], 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 $\widehat{L} = \frac{2L}{\lambda_{max}} - I_N = U\widehat{\Lambda}U^\intercal$. 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[51], 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

We turn to use temporal convolution after using spatial convolution. In [52], zhao et al. capture the temporal dependences based on the traditional recurrent neural network. In [39], Zhou et al. 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 extract the features in the temporal. After we use 2 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

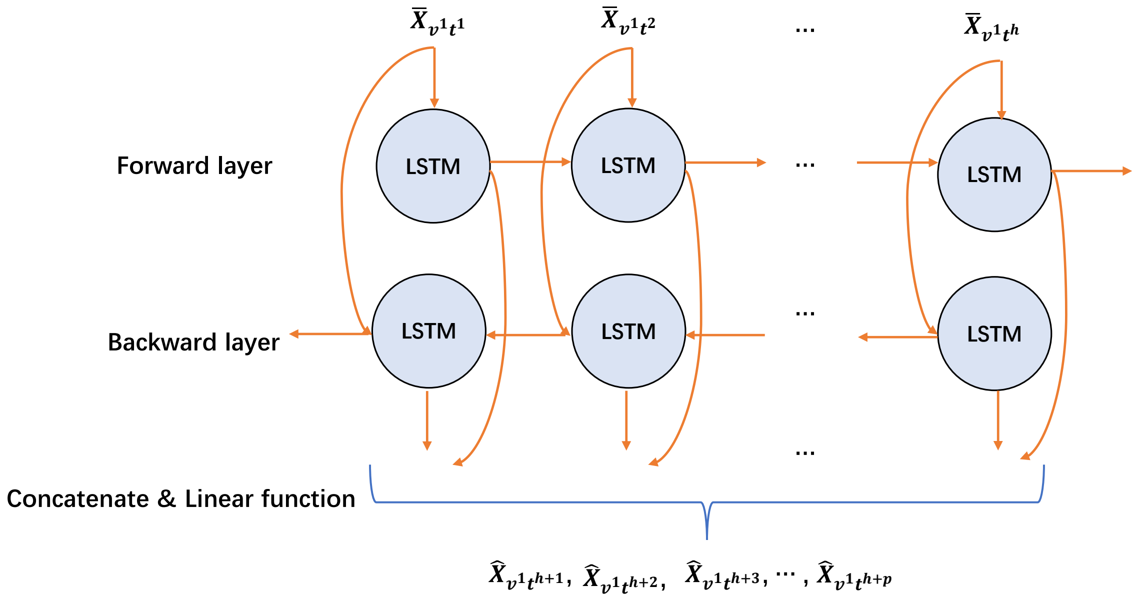

Figure 3.3: After the spatial-temporal convolution, the speed information is fed into Bi-LSTM and Linear function

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 entire process after the spatial and temporal convolution.

Chapter 4: Experiment

4.1 Datasets

In our experiment, we predicted the traffic speed of 3 time steps, 6 time steps and 12 time steps in the future. Because the time interval is 5 minutes, we can predict the future 15, 30 and 60 minutes. Fig 4.1 shows the study area of two dataset.

4.1.1 Tyson's Corner dataset

Since we get the raw data from INRIX, VDOT, We spend 60% of their time on cleaning and organizing data. Finally, our 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 State; 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 features such as location, the distance on the road segments, etc. 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 unified the above data into three categories: Work zone, Collision and other facts. Secondly, we 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

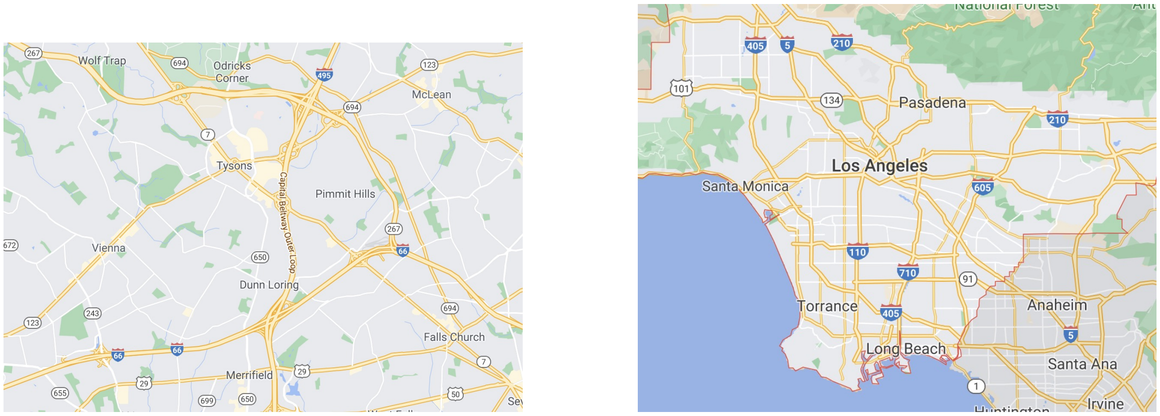

Figure 4.1: Tyson's Corner on google map (left) and Los Angles County on google map (right)

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 include normalized traffic speeds related to the construction work $X^s_{v_i,t_j}$, a construction matrix which 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.

4.1.2 Los-loop dataset

Next, we use a dataset in [52]: Los-loop dataset, which is based on the highway of Los Angles County. It has 207 sensors and traffic speed is from Mar.1 to Mar.7, 2012 with the 5 minutes interval of the time step.

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:

- **TGCN[52]:** It uses a graph convolutional network to learn complex topological structures and uses gate recurrent unit to learn dynamic traffic flow.
- **STGCN[34]:** A ”sandwich” structure that using two gated sequential convolution layers and one spatial graph convolution layer to capture the spatial-temporal correlations.
- **GraphWaveNet[53]:** 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[38]:** 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

Several widely used metrics are used in our model: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). The benefit of using these metrics is that it can intuitively compare the difference between the predicted speed and the real speed.

4.4 Hyper-parameters

Following the previous work such as [39], [47], [52], etc, we use 3, 6 and 12 historical times as input sequence. Then we use 3, 6 and 12 historical times as the length of output. 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 feature map, we set $\lambda = 2$ to evaluate the impact of construction on different distances as much as possible.

4.5 Experimental Research

In this section, we evaluate the performance of our model with comparing existing models and two datasets in traffic speed predictions. There are the following five research questions and answer in the experiment.

- Which model do we implement and why use this model?
- What is the difference between our model and the base model?
- Compared with the baseline traffic model, how does STGCN-WZ perform in the tyson database?
- How does the model perform in other datasets without using the work zone feature map?
- Can STGCN-WZ provide reasonable explanations about the experiment results?

4.5.1 Basic model($Q1$)

The model we use is based on [38], 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 it is that in 2019, the model performance exceeds the state-of-the-art baselines. Furthermore, it also divides the time segment into three parts and uses multi-component fusion to fuse them. Interestingly, this method makes the result worse. In order to evaluate the traffic performance based on multi-factors, we propose a new model based on ASTGCN model.

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

Differing from the dataset in ASTGCN, Tyson's 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. Frirst, ASTGCN use spatial-temporal attention described in chapter 3.2 and a traditional graph convolution operation $(x * g)_G = \sum_{k=0}^{K} \theta_k T_K(\widehat{L})x,$, 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 only use simple attention 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.

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

We divied 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 baseline models, 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 baseline models and our model are reported in Table 4.1. From the results, we can see that our model, which combines speed and construction work feature map, achieves the best perforamce 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 the GraphWaveNet model and T-GCN, because GraphWaveNet are the autoregressive generative model, thus the accuracy of predicting long-term will decrease sharply, and TGCN model only uses the traditional GRU model in temporal and

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

models	15 min($t' = 3$)			30 min($t' = 5$)			60 min($t' = 12$)		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
T-GCN	5.3574	3.3912	0.0990	6.0414	3.5641	0.1121	6.8192	4.3231	0.1257
STGCN	5.1046	3.2957	0.1001	5.6391	3.4207	0.1107	6.5761	4.2532	0.1332
GraphWaveNet	4.5261	2.9173	0.0904	5.8157	3.5207	0.1168	6.9043	4.3532	0.1387
ASTGCN	4.1365	2.7490	0.0865	5.1173	3.1641	0.1037	5.9516	3.5749	0.1121
STGCN-WZ(No)	4.0584	2.6805	0.0837	4.9594	3.0880	0.1079	5.8751	3.5622	0.1093
STGCN-WZ	**3.9938**	**2.6023**	**0.0820**	**4.8719**	**3.0844**	**0.0954**	**5.7287**	**3.5138**	**0.1065**

graph convolution operation in spatial, so the performance of the model cannot connect the previous time sequence and capture adjacent nodes correlations. Second, the models based on attention approach include ours, has better performance than others. This is because the attention method can well capture the historical temporal and spatial correlations. Third, compared with ASTGCN and STGCN-WZ(No), we find that the performance of STGCN-WZ(No), which only 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. Moreover, when we add an external feature map, set a learnable weight, and combine the speed feature map into the STGCN-WZ model so that the model fully gets the correlations between the data, we find that the model has significant improvement. In terms of results, STGCN-WZ improves over the model ASTGCN by 2% and over the model STGCN-WZ(No) by 19% with respect to RMSE for 3-steps. 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 4.2: 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	6.1833	4.1618	0.1104	6.9281	4.8187	0.1374	8.2604	5.4124	0.1504
STGCN	6.2704	4.1800	0.1192	6.8814	4.7357	0.1253	7.9887	5.3654	0.1414
GraphWaveNet	5.3654	3.0042	0.0977	6.6461	3.7182	0.1235	7.8951	4.7148	0.1400
ASTGCN	5.2021	3.0443	0.0787	6.1271	3.5061	0.1011	7.5650	4.4192	0.1289
STGCN-WZ(No)	**4.9676**	**2.9429**	**0.0776**	**5.8438**	**3.2782**	**0.0907**	**7.2826**	**4.3831**	**0.1271**

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 only 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.

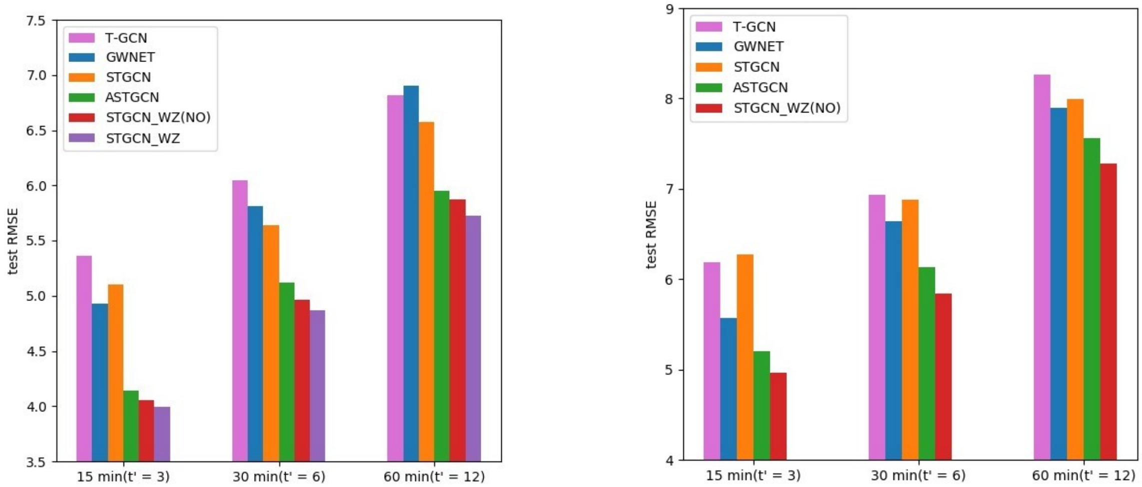

Figure 4.2: The RMSE of the models in Tyson's Corner (left) and Los-loop dataset (right)

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 char, it is intuitive to see 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 this figure, we can find that our model has achieved great results in predicting traffic speed based on the influence of construction. The result shows that when the time step = 3, the

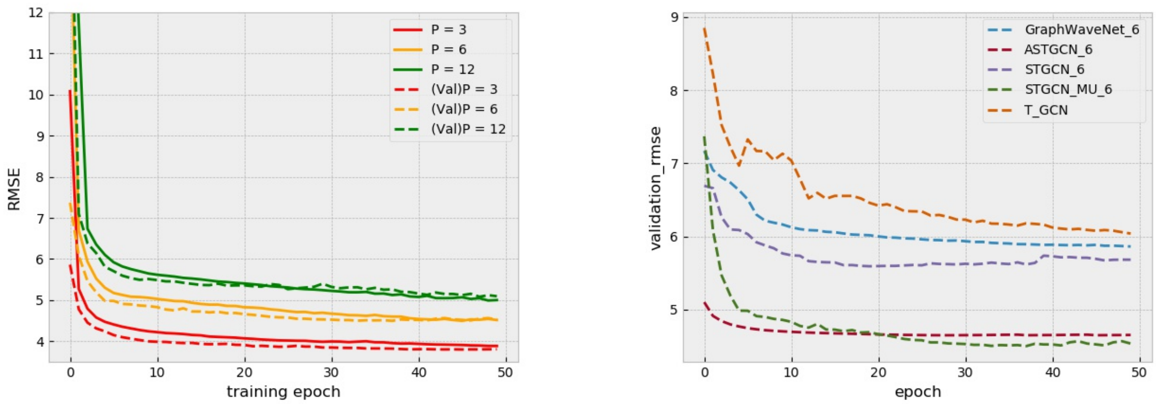

Figure 4.3: The training RMSE and validation RMSE of STGCN-WZ model (left) and the validation RMSE versus the training epoch (right)

predicted speed is very similar to the real speed; when the time step = 12, the difference between the predicted speed and the real speed gradually increases. 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 discovery in the heap map is following as: firstly, 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. Secondly, 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. Thirdly, 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.

Table 4.3: The ablation of construction map

	RMSE	MAE	MAPE
$\lambda = 3$	**4.8719**	**3.0844**	**0.0954**
$\lambda = 5$	4.9829	3.1809	0.1183
$\lambda = 7$	5.1254	3.2153	0.1218

Table 4.4: The ablation of different speed wave $\hat{X}_s$

	RMSE	MAE	MAPE
$W_s \odot X^s + W_c \odot X^c$	**4.8719**	**3.0844**	**0.0954**
$X^s + W_c \odot X^c$	4.8925	3.0856	0.0985

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

	RMSE	MAE	MAPE
Linear function	**4.8719**	**3.0844**	**0.0954**
1×1 Convolution	4.9849	3.1036	0.1099

4.6 Ablation Studies

In a complex graph neural network, parameters have a great influence on the performance of the model. We used ablation experiments to effectively find the best parameter solution. We examine our model STGCN-WZ from four main components, which are construction map setting, speed wave setting, spatial-temporal attention, and temporal convolution operation. In particular, we set these components as:

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

Different λ can build different construction maps in our model. Table 4.3 shows the evaluation of our model using $\lambda = 5, 7$, we find that $\lambda = 2$ is better than others, this proves the importance of the construction feature map. A well-design feature map can significantly improve the performance of traffic prediction.

- **STGCN-WZ-II** remove the weight of learning speed, setting speed wave 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 remove the weight of learning speed W_s and the result is in Table 4.4. As we can see, the first method is slightly better than removing the weight method.

- **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 between STGCN-WZ which use first-order ChebNet and ASTGCN which use Chebyshev polynomials. 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.

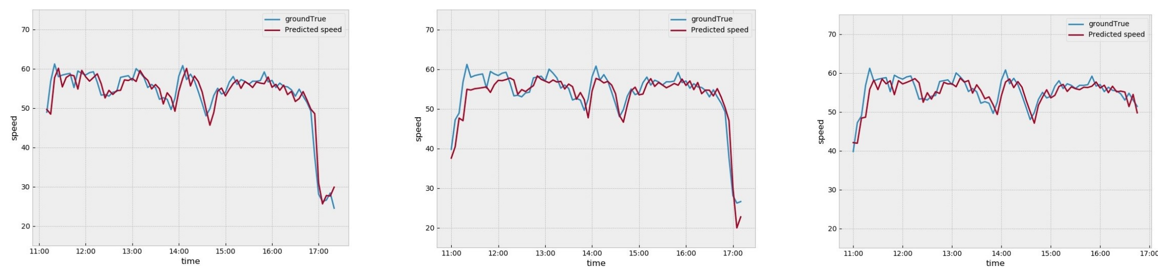

Figure 4.4: Traffic speed on road segment ’110-04175’ , 11/19/2019, $t^{'} = 3$ (left), 6 (middle), 12 (right)

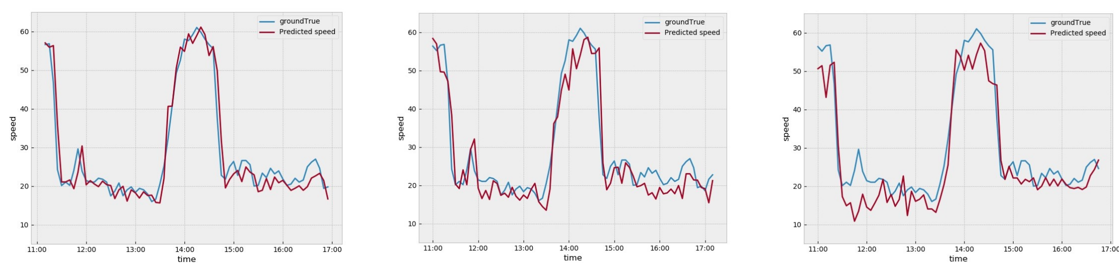

Figure 4.5: Traffic speed on road segment ’110+04177’, 11/22/2019, $t^{'} = 3$ (left), 6 (middle), 12 (right)

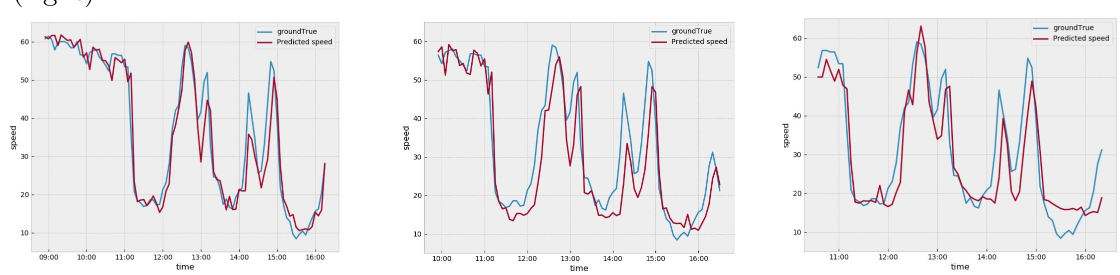

Figure 4.6: Traffic speed on road segment ’110P04611’, 11/22/2019, $t^{'} = 3$ (left), 6 (middle), 12 (right)

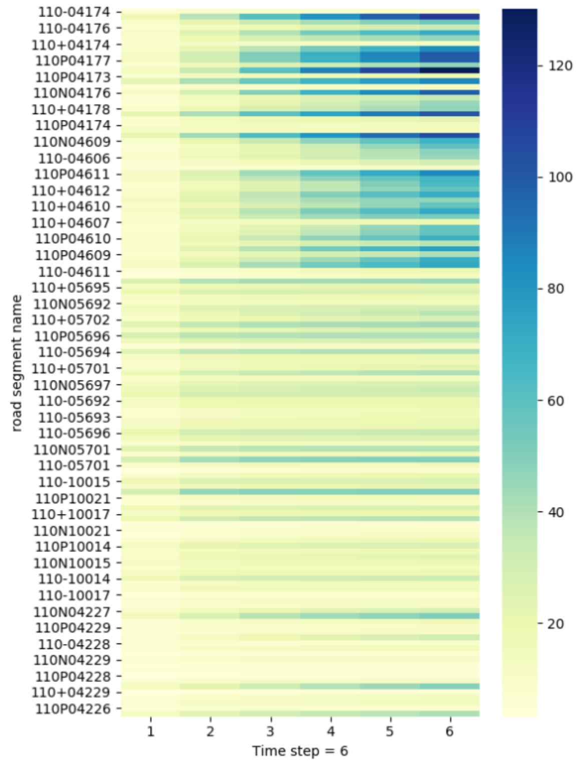

Figure 4.7: The heatmap of speed prediction using MSE at time step = 6 (The x-axis represents the predicted i-th time period)

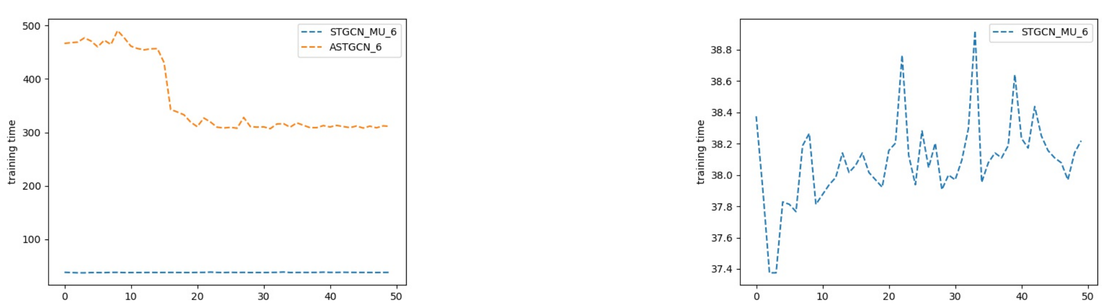

Figure 4.8: The training time between STGCN-WZ and ASTGCN (left) and training time in STGCN-WZ (right)

Chapter 5: Conclusion

In this project, we design a graph convolutional network based on the impact of construction work to predict traffic speed, which consists of the fusion of two feature maps, multi-head attentions, first-order ChebNet and bidirectional RNN. Differing from the previous work studying in traffic speed, our project is to study the impact of public construction on surrounding region by using multiple regression feature maps. From the experimental based on two dataset 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 works 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 correlation of topological space, but also handle time-series problems. The combination of convolutional neural network and natural language processing will help solve the problem of irregular data sets.

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, it will inevitably exist a phenomenon, that is, when the model input a normal (Traffic speed is not affected by abnormal factors) data, it will predict normal traffic speed.

However, many real labels are influenced by abnormal factors, which are different with the output, so the prediction is not accurate. This explains why the model predicts different effects under different data.

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] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

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

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

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

[34] 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.

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

[36] 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.

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

[38] 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.

[39] 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.

[40] 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.

[41] E. W. Weisstein, "Laplacian matrix," *https://mathworld.wolfram.com/LaplacianMatrix.html*, 1999.

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

[43] 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.

[44] 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.

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

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

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

[52] 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.

[53] 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.

Curriculum Vitae

Include your *curriculum vitae* here detailing your background, education, and professional experience.