# Traffic flow prediction for vehicle emission calculation based on graph convolutional networks

**Peng Jiang**[1], **Igor Bychkov**[4], **Jun Liu**[2,3], **Tianjiao Li**[2], **Alexei Hmelnov**[4]

[1]Department of Science and Technology Cooperation, Westlake University, No.18, Shilong Mountain Street, Xihu District, Hangzhou, China
[2]"the Belt and Road" Institute for Information Technology, Hangzhou Dianzi University, No.115, Wenyi Road, Xihu District, Hangzhou, China
[3]School of Automation (Artificial Intelligence), Hangzhou Dianzi University, No.1158, Number Two Street, Jianggan District, Hangzhou, China
[4]Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, 134 Lermontov st. Irkutsk, Russia

E-mail: `jiangpenghz@163.com`

**Abstract.** Monitoring the distribution of vehicle exhaust emissions within the city is a very challenging problem since it is affected by many complex factors, such as spatial-temporal correlation and the other environment conditions. In addition, the technology of using sensors to directly monitor vehicle exhaust emissions is still in the initial stage, and it is hard to implement direct monitoring in a large area. Thus, we use the existing environmental theory to measure the distribution of vehicle exhaust emissions in cities by traffic volume. In this paper, the problem we need to solve is how to use the data of sparse monitoring stations and inherent traffic network to infer the spatial-temporal distribution of traffic volume. In order to solve this problem, we propose a graph convolutional network model to extract the characteristics of traffic data and other features. We have done a lot of experiments on real traffic data sets. The experimental results show that the proposed method performs better than the existing methods.

## 1. Introduction

With the rapid growth of vehicle ownership in China, a mass of $NO_x$, $CO$, $HC$, $PM_x$ and other harmful gases emitted by vehicles have aggravated urban air pollution, resulting in the deterioration of air quality and increasingly frequent haze weather. The precondition of vehicle exhaust pollution control is effective monitoring of them, so we need some monitoring means to quantify vehicle exhaust emissions. However, it is difficult for us to measure the emission of vehicles directly in a large area, so we calculate those emissions by the COPERT model [1], which only needs to provide the urban context data and traffic status of each road section to calculate the vehicle emissions. Urban context data can be obtained through statistics, while traffic information must be obtained through real-time monitoring of stations which cannot be deployed in all road segments.

To determine the optimal location of new monitoring stations it is required to maximize the inference performance of the traffic volume distribution model on the resulting monitoring network. This seems to be a reasonable and practical idea. After all, the layout of monitoring stations is very sparse. It is very important for us to accurately infer the traffic volume distribution on the unobserved road segments using the data monitored by the existing stations. However, without the monitoring data on the unobserved road segments, it is difficult for us to know on which road segments the stations can be placed to maximize the inference accuracy. To approximately achieve this, Hsieh et al propose a two-stage framework on deployment of air quality monitoring stations, which uses the inference model to estimate the distribution of air quality index (AQI), and then obtains the location of $K$ new stations through the location selection model to minimize the assessment uncertainty [2]. However, this novel approach can't be directly applied to our problem, since dividing the traffic network into several grids causes the overlook of spatial correlation.

In order to achieve the above purpose, we use graph convolutional neural network to deal with this problem. It makes the training model of higher prediction accuracy and at the same time of smaller uncertainty.

The contributions of this paper are summarized as follows:

(i) The proposed approach is not only able to forecast the spatial-temporal distribution of traffic volume but also to provide a basis for selecting the location of new stations and maximizing the reliability of traffic inference.

(ii) We entirely use the graph convolution to learn spatial-temporal correlation of structured time series.

(iii) We conduct extensive experiments on two real-world data sets. The MAE (mean absolute error) and RMSE (Root Mean Square Error) of the inference model are 49.82 and 71.74 respectively, which outperforms the baseline methods.

The other parts of this paper are as follows: the second section is the introduction of data and features, the theory of graph convolutional neural network and the problem description of this paper. The third section introduces the structure of spatial-temporal graph convolutional neural network in detail. The fourth section presents the experimental results. Finally, the full text is summarized and the future work is prospected.

## 2. Data and methodology

### *2.1. Data description*

The data utilized in this paper comes from a competition about urban computing. In the data set there are 35 roads with traffic flow records. Of them 27 roads are used to train the prediction model and other 8 roads are utilized to test its performance. The data consists of the following data sets:

(i) Road network features

(ii) Point of interests (POI) features

(iii) Speed pattern features

(iv) Weather features

(v) Time features

(vi) Volume Records

### *2.2. Graph convolution*

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ with $N$ vertices $\nu_i \in \mathcal{V}$, where $\mathcal{E}$ is the edge set and $A \in \mathbb{R}^{N \times N}$ denotes the binary adjacency matrix. Defferrard et al built a graph convolution defined as:

$$g_\theta *_\mathcal{G} x \approx \sum_{k=0}^{K} \theta_k T_k(\widetilde{L})x \tag{1}$$

where $x \in \mathbb{R}^N$ is the signal on the graph, $*_\mathcal{G}$ is the convolution operator, $g_\theta$ denotes the spectral filter, $\widetilde{L} = \frac{2}{\lambda_{\max}} L - I_N$, $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, $D_{ii} = \sum_j A_{ij}$ , $\lambda_{\max}$ denotes the largest eigenvalue of $L$ and $\theta_k$ is the Chebyshev coefficient [3]. The Chebyshev polynomials $T_k(x)$ are recursively defined as $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_1(x) = x$ and $T_0(x) = 1$.

Kipf et al proposed a first-order approximate graph convolution operation [4], which simplified this model by limiting $K$ to 1 and approximating $\lambda_{\max}$ by 2, which allows us to rewrite the convolution the following way:

$$g_\theta *_\mathcal{G} x \approx \theta_0 x + \theta_1 (\frac{2}{\lambda_{\max}} L - I_N) x \approx \theta_0 x - \theta_1 (D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) x \tag{2}$$

Then we constrain the number of parameters: let $\theta = \theta_0 = -\theta_1$ and further apply a normalization trick to the convolution matrix:

$$g_\theta *_\mathcal{G} x \approx \theta (I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) x = \theta (\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}) x \tag{3}$$

which gives the following form of the matrix of the convolution operation:

$$Z = \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} \tag{4}$$

where $\widetilde{A} = A + I_N$ and $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$.

The above definition of graph convolution is extended to data with $C_{in}$ input channels, i.e., $X \in \mathbb{R}^{N \times C_{in}}$ (each vertex is a $C_{in}$-dimensional feature vector), and the propagation rule of this simplified model is given by:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \tag{5}$$

where $H^{(l)}$ is the output and $W^{(l)}$ is the trainable weight matrix of the $l^{th}$ layer, $H^{(0)} = X$ and $\sigma(\cdot)$ is an activation function.

### *2.3. Methodology*

Our real purpose is to monitor the spatial-temporal distribution of vehicle exhaust emissions in urban traffic network. However, the existing measurement technology is difficult to monitor emissions directly in a large scale. Fortunately, given the spatial-temporal distribution of traffic conditions and traffic network data, the distribution of emissions can be calculated by the existing COPERT model, so that our goal turned to volume monitoring. Since the traffic monitoring stations can't cover the whole city, we can only obtain traffic data of partial road segments. Therefore, according to the urban context data, traffic speed and volume acquired by established monitoring stations, we will infer the traffic volume of any road in the city at any time stamp. Thereafter, spatial-temporal distribution of traffic volume can be further employed to estimate the distribution of vehicle exhaust emissions according to the COPERT model.

## 3. Inference of the model

In this section, we describe the structure of proposed model (called STGC-LD) in detail, which includes spatial-temporal learning block, two attributes extraction block and a label distribution learning block, as shown in Figure 1. The spatial-temporal learning block is employed to learn the spatial correlations and temporal dependencies from traffic travel speed. First attribute block is responsible for processing external factors (e.g. time of the day and weather), while second attribute block is used to extract structural features of traffic network. These blocks are all connected by residuals, which makes it easier for them to be added and deleted. Finally, the label distribution learning block estimates the spatial-temporal distribution of traffic volume within the city, but also reveal the confidence of its inference.

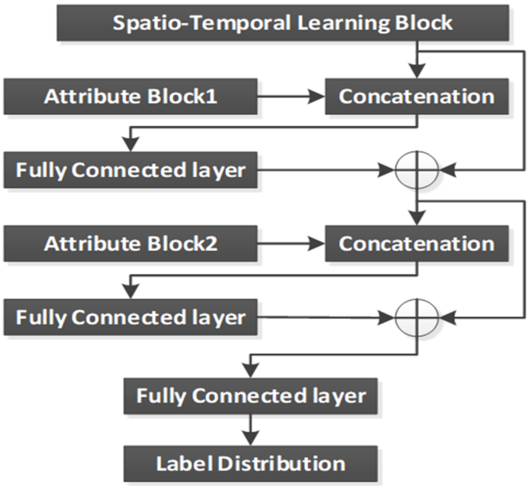


Figure 1: The general structure of the proposed model STGC-LD.

*3.1. Spatial-temporal learning block*

There is a certain correlation between traffic volume and travel speed, and nearby roads with similar travel speed follow the same volume patterns in all probability. Accordingly, we design a spatial-temporal learning block, containing a layer spatial graph convolution (SGC) and a layer temporal graph convolution (TGC), to extract the spatial-temporal properties of travel speed, which is a 3-dimensional structured time series.

*3.1.1. SGC for extracting spatial features.* We deal with the adjacency matrix as:

$$\widehat{A} = \sigma(\widetilde{A} \circ W_{embed}) \tag{6}$$

where $W_{embed}$ is the learnable matrix that can be adjusted to affect the degree of closeness, and $\circ$ denotes the element-wise matrix product. Then we put $\widehat{A}$ and $\widehat{D}_{ii} = \sum_j \widehat{A}_{ij}$ into the graph convolutional network, and get the adaptive graph convolutional network as:

$$H^{(l+1)} = \sigma(\widehat{D}^{-\frac{1}{2}} \widehat{A} \widehat{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \tag{7}$$

The above formula can adjust the weight of edges adaptively based on the graph structure and the attributes of each vertex, and learn the influence of different adjacent vertices.

We set the travel speed to $Att_{speed} \in \mathbb{R}^{t \times n \times C_{speed}}$ and adjacency matrix of traffic network to $A_s \in \mathbb{R}^{n \times n}$, where $t$, $n$, $C_{speed}$ are the number of time steps, the number of road segments in the traffic network and dimension of the speed feature, respectively. The graph convolution described above can only process two-dimensional data, but travel speed is a 3-dimensional tensor. Hence, we share parameters on the time axis, that is, we do the same convolution on each time stamp. After a convolution operation, the output $Z_s \in \mathbb{R}^{t \times n \times C_{out}}$ is defined as:

$$Z_S^i = \widehat{D_S}^{-\frac{1}{2}} \widehat{A_S} \widehat{D_S}^{-\frac{1}{2}} Att_{speed}^i W_S \ , \ i \in \{1, 2, ..., t\} \tag{8}$$

Where $Att_{speed}^i \in \mathbb{R}^{n \times C_{speed}}$, $W_S \in \mathbb{R}^{C_{speed} \times C_{out}}$ is a kernel of spatial graph convolution.

*3.1.2. TGC for extracting temporal features.* Nowadays, although the model based on recurrent neural network is widely used in time series analysis, its application in traffic forecasting task still suffers from the complexity of gate mechanisms, time-consuming iterations and low response to dynamic changes. Such networks cannot simulate very long-range temporal dependencies (e.g. period and trend), and training becomes harder as depth increases. In this paper, graph convolution is employed to encode the temporal correlation directly, avoiding the explicit smoothing regularization in the loss function. Firstly, we need to construct an affinity graph for the time series. Since the traffic volume does not change abruptly on the time axis and follows a strong periodicity, we connect neighbor and periodic timestamps on the time series of each road section to construct the time affinity graph. For a time stamp node $T_i$ of a time series, the time neighbors of the point can be expressed as

$$\begin{aligned}\{&T_{i-p*P_{week}}, ..., T_{i-P_{week}}, T_{i-p*P_{day}}, ..., T_{i-P_{day}}, ..., T_{i-p}, ..., T_{i-1},\\ &T_i, T_{i+1}, ..., T_{i+p}, T_{i+P_{day}}, ..., T_{i+p*P_{day}}, T_{i+P_{week}}, ..., T_{i+p*P_{week}}\}\end{aligned} \tag{9}$$

where $p$ is a super-parameter, $P_{day}$ and $P_{week}$ represent the period of one day and one week respectively. Besides, we set the temporal edge weights as 1.

We transpose the output of the SGC to $Q = Z_S^T \in \mathbb{R}^{n \times t \times C_{out}}$, and set the temporal adjacency matrix as $A_T \in \mathbb{R}^{t \times t}$. Then, we share parameters in the space, that is, we do the same convolution on the time series for each vertex. After the convolution operation, the features are mapped as follows:

$$Z_T^i = \widehat{D_T}^{-\frac{1}{2}} \widehat{A_T} \widehat{D_T}^{-\frac{1}{2}} Q^i W_T \ , \ i \in \{1, 2, ..., n\} \tag{10}$$

where $W_T \in \mathbb{R}^{C_{out} \times C_{out}}$ is a kernel of the temporal graph convolution.

*3.1.3. Spatial-temporal learning.* In order to extract the spatial correlations and temporal dependencies of structured sequences of data simultaneously, we design a spatial-temporal Learning block which stacks a SGC layer and a TGC layer. Too many convolution layers could converge the features of interconnected vertices to the same values [5]. Moreover, layer normalization is equipped with the spatial-temporal Learning block to prevent overfitting. The output of this block is denoted as $X_{ST}$.

### *3.2. Attribute block 1*

In this block, we preprocess and integrate the weather features and time features. The time range 6 : 00-23 : 00 is divided into 17 timeslots, each timeslot corresponds to an hour, namely $TimeAtt \in \{1, 2, ..., 17\}$. Since the dimension of $TimeAtt$ is large, the one-hot coding would lead to a high computing cost, so we adopt the embedding method to transform these categorical features into low-dimensional vectors. Specifically, the embedding method is to multiply each categorical value $\nu \in \mathbb{R}^{1 \times C}$ by a learnable parameter matrix $W \in \mathbb{R}^{C \times O}$. Usually we have $O \ll C$, so that the embedding method can effectively reduce the dimension of input features and make model calculation more efficient. Furthermore, a significant property of embedding method is that the categorical values with similar semantic meaning are usually very close in the embedding space [6]. The output of this block is denoted as $X_{Att}$.

### *3.3. Attribute block 2*

Traffic network attributes mainly include road network structure, road section features, POI features, etc. We utilize the embedding method to process the number of lanes, road grade and other categories of road network features, and normalize the road length, POI features and so on. Then, the preprocessed features are concatenated and fed into SGC to extract spatial correlation. In our model, we connect blocks by residuals to make them easier to add and remove. He et al has shown that training the neural networks with residual connections is easier and more robust [7]. The output of this block is denoted as $X_{Net}$.

### *3.4. Label distribution learning block*

We adopt label distributed learning [8] on a single model. The input of LDL is set to $X \in \mathbb{R}^{t\times n\times d}$, where $t$, $n$, $d$ are the number of time steps, the number of road segments and the number of feature dimensions of each road respectively. The task of LDL is to estimate the traffic volume distribution vector $y(\nu, j) = \{y_0, y_1, ..., y_{q_{\max}}\} \in \mathbb{R}^{q_{\max}+1}$ of a road $v$ at a timestamp $j$, where $q_{\max}$ is determined by the maximum average traffic volume per lane in the training data. In this problem, we quantify the real volume value from existing station as a normal distribution vector, whose expectation is the real value and variance is a super-parameter. Then, the model is learned by minimizing the symmetric Kullback-Leibler divergence of the estimated and the observed label distributions:

$$Loss_L = \min \frac{1}{t}\sum_{j}^{t}\frac{1}{2\,|L|}\sum_{\nu\in L}\sum_{i=0}^{q_{\max}} KL(\nu, j)[i] \tag{11}$$

$$KL(\nu, j)[i] = \mathbf{y}(\nu, j)[i]\log\widehat{\mathbf{y}}(\nu, j)[i] + \widehat{\mathbf{y}}(\nu, j)[i]\log\mathbf{y}(\nu, j)[i] \tag{12}$$

where $L$ is a set of observed roads, $\widehat{\mathbf{y}}(\cdot,\cdot)$ is the estimated label distribution. If we need to know the specific value of traffic volume, we can compute the expectation of the probability distribution vector, namely:

$$\sum_{i=0}^{q_{\max}} i\widehat{\mathbf{y}}(\cdot,\cdot)[i] \tag{13}$$

## 4. Experiments

### *4.1. Inferring performance comparison*

To demonstrate the effectiveness of proposed inference model and deployment model, we further compare them with several existing approaches using the real traffic data described in the Section 2.1. The parameters of all the models are fine-tuned through the grid search. In the following experiments, we repeat each of them 50 times to obtain the average results.

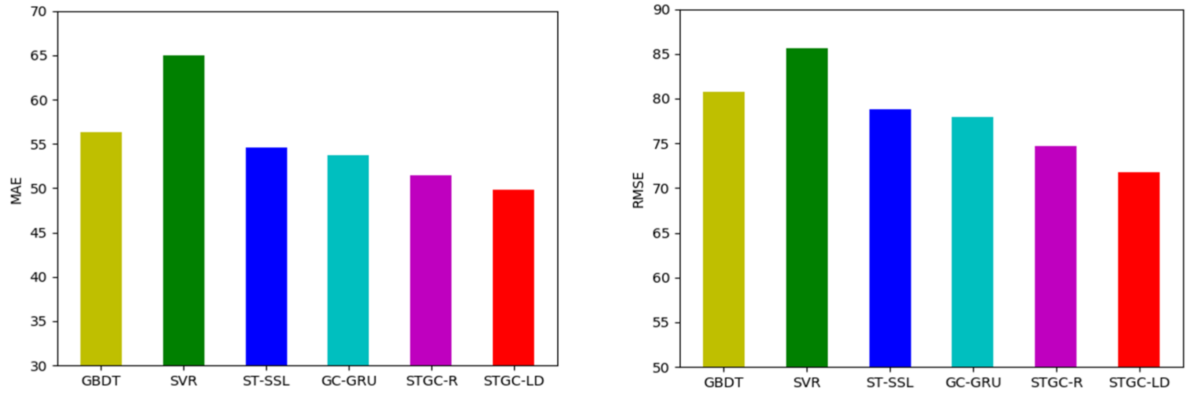
*4.1.1. Training data usage.* The traffic network contains 793 road segments, in which 35 road segments are equipped with loop detectors (i.e. have volume values), while the remaining roads are unknown. The traffic volumes were collected every hour from March 16 to April 1, 2016 (17 days in total) and sampled each day from $6:00$ to $23:00$. In the experiment, we randomly divide the set of 35 road sections into the two subsets of 27 and 8 roads, the former contain $27*17*17$ instances which are used as the training set and the latter contain $8*17*17$ instances to be used as the testing set. All the experiments had been repeated 50 times and the training and testing sets were randomly shuffled in each repetition.

*4.1.2. Model settings.* For the inference model each fully connected layer has 64 channels. The temporal neighbor parameter is set to 3, and the variance of the normal distribution in LDL is set to 2. Besides, we set the initial learning rate as $10^{-3}$ with a decay rate of 0.9 after every 40 epochs.

### *4.2. Competitors*

(i) Gradient Boosting Decision Tree (GBDT). In our problem, we neglect the spatial and temporal correlation of data, simply treats all historical observed data from all stations as the training data to build a supervised learning model.

(ii) Support Vector Regression (SVR). SVR is an important application branch of Support Vector Machine (SVM), and it is used for regression task of traffic volume. The experimental setup of SVR is consistent with GBDT.



(a) mean absolute errors

(b) root mean square errors

Figure 2: Comparison of various traffic volume estimating algorithms.

(iii) Spatial-Temporal Semi-Supervised Learning (ST-SSL) [9]. This method constructs the spatial-temporal affinity graph and determines the spatial and temporal edge weights respectively. Finally, the change rate of the spatial neighbor and the value of the temporal neighbor are smoothed.

(iv) Graph Convolutional Recurrent Neural Network (GC-GRU). Referring to literature [10], we first use one layer graph convolution for feature extraction and put the new feature into GRU for time correlation analysis.

(v) STGC-Regression (STGC-R). In order to verify the effectiveness of LDL, we set output of the proposed network structure to a single node, the other structures remain unchanged. And the corresponding loss function is changed to the loss function of the regression task, namely

$$Loss = \min \frac{1}{t} \sum_{j}^{t} \frac{1}{|L|} \sum_{\nu \in L} (y(\nu, j) - \widehat{y}(\nu, j))^2$$

### *4.3. The obtained errors for different inference models*

The experimental results show that the performance of the proposed algorithm is better than that of other algorithms as shown in the Figure 2. The supervised learning algorithms GBDT and SVR perform worse than the other four semi-supervised learning algorithms, since the scarcity of training samples makes it difficult to train a supervised model with good generalization performance. In addition, we use the same network structure to regress this problem, and find that its performance is not as good as LDL, which indicates that LDL can better overcome the challenge of poor prediction performance of regression method due to insufficient labeled samples.

### *4.4. Evaluating inference models with various time spans*

We experimented with data of various time spans, ranging from 1 day to 17 days. The estimate results of all methods are shown in Table 1 and Table 2, which shows that the prediction accuracy of the proposed algorithm is always better than that of other algorithms. With the increase of the time spans of input data, the inference performance of each algorithm decreases gradually, but the performance of the proposed algorithm is more stable. This is because it uses the graph convolution to extract time features, which is well suited for the periodicity of long time series data and improves the inference performance of the model for long structure sequence data.

Table 1: MAE of volume inference for all methods under various time spans.

| Method | 1d | 2d | 5d | 7d | 10d | 14d | 17d |
|---|---|---|---|---|---|---|---|
| GBDT | 54.33 | 54.22 | 55.08 | 55.23 | 55.45 | 56.11 | 56.35 |
| SVR | 62.52 | 63.29 | 62.41 | 63.31 | 63.98 | 64.50 | 65.01 |
| ST-SSL | 52.47 | 52.40 | 53.22 | 53.36 | 53.65 | 54.35 | 54.55 |
| GC-GRU | 51.64 | 51.55 | 52.38 | 52.54 | 52.82 | 53.58 | 53.76 |
| STGC-R | 50.40 | 50.21 | 51.01 | 51.35 | 51.53 | 51.29 | 51.41 |
| STGC-LD | **48.32** | **49.53** | **48.80** | **49.42** | **49.64** | **50.01** | **49.82** |

Table 2: RMSE of volume inference for all methods under various time spans.

| Method | 1d | 2d | 5d | 7d | 10d | 14d | 17d |
|---|---|---|---|---|---|---|---|
| GBDT | 76.21 | 76.87 | 77.10 | 77.62 | 78.65 | 79.88 | 80.79 |
| SVR | 80.94 | 82.86 | 80.80 | 82.09 | 83.52 | 84.54 | 85.62 |
| ST-SSL | 74.23 | 74.87 | 75.09 | 75.59 | 76.70 | 77.97 | 78.80 |
| GC-GRU | 73.25 | 73.90 | 74.06 | 74.57 | 75.69 | 77.06 | 77.89 |
| STGC-R | 71.65 | 72.21 | 72.34 | 73.02 | 74.03 | 74.44 | 74.67 |
| STGC-LD | **68.24** | **71.37** | **69.19** | **70.38** | **71.06** | **71.90** | **71.74** |

### *4.5. The influence of different experimental settings*

(i) We use one-hot coding and embedding to process categorical features (*TimeAtt*, POI features and so on) respectively. We have found that the embedding method is better than the one-hot method.

(ii) Effect of Layer Normalization: After introducing layer normalization, the performance has been improved.

## 5. Conclusion

In this paper, we propose a spatio-temporal semi-supervised graph convolutional network model. The model can predict the temporal and spatial distribution of traffic flow on the road section without monitors by using urban environmental data and observation data from existing sites. We have carried out experiments on real traffic data, and the results of the suggested model are better than that of the other comparison methods, indicating that our method is more suitable for the inference of traffic volume. In our future work we will further optimize the network structure and parameters to get better results. In addition, the proposed model can also be used in some practical applications, such as urban population monitoring.

**Acknowledgments**

This work was supported in part by the Leading Talents of Science and Technology Innovation in Zhejiang Province 10 Thousands Plan under Grant 2018R52040, in part by the National Key Research and Development Program of China under Grant 2016YFC0201400, in part by the Provincial Key Research and Development Program of Zhejiang Province under Grant 2017C03019, and in part by the International Science and Technology Cooperation Program of Zhejiang Province for Joint Research in High-tech Industry under Grant 2016C54007.

## References

[1] Shang J, Zheng Y, Tong W, Chang E and Yu Y 2014 Inferring gas consumption and pollution emission of vehicles throughout a city *In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM 1027-1036

[2] Hsieh H P, Lin S D and Zheng Y 2015 Inferring air quality for station location recommendation based on urban big data *In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM 437-446

[3] Defferrard M, Bresson X and Vandergheynst P 2016 Convolutional neural networks on graphs with fast localized spectral filtering *In Advances in neural information processing systems* 3844-3852

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

[5] Li Q, Han Z and Wu X M 2018 Deeper insights into graph convolutional networks for semi-supervised learning *In Thirty-Second AAAI Conference on Artificial Intelligence*

[6] Gal Y and Ghahramani Z 2016 A theoretically grounded application of dropout in recurrent neural networks *In Advances in neural information processing systems* 1019-1027

[7] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *In Proceedings of the IEEE conference on computer vision and pattern recognition* 770-778

[8] Gao B B, Xing C, Xie C W, Wu J and Geng X 2017 Deep label distribution learning with label ambiguity *IEEE Transactions on Image Processing* **26(6)** 2825-2838

[9] Meng C, Yi X, Su L, Gao J and Zheng Y 2017 City-wide traffic volume inference with loop detector data and taxi trajectories *In Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.* ACM 1

[10] Cui Z, Henrickson K, Ke R and Wang Y 2018 Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting *arXiv preprint* arXiv:1802.07007