# Fine-Grained Spatial-Temporal Representation Learning with Missing Data Completion for Traffic Flow Prediction

Shiqi Wang[1], Min Gao[1(✉)] Zongwei Wang[1], Jia Wang[1], Fan Wu[1], and Junhao Wen[1]

School of Big Data and Software Engineering, Chongqing University, Chongqing, China
{shiqi, gaomin, zongwei,jiawang,wufan,jhwen}@cqu.edu.cn

**Abstract.** Spatial-temporal traffic flow prediction is beneficial for controlling traffic and saving traffic time. Researchers have proposed prediction models based on spatial-temporal representation learning. Although these models have achieved better performance than traditional methods, they seldom consider several essential aspects: 1) distances and directions from the spatial aspect, 2) the bi-relation among historical time intervals from the temporal aspect, and 3) missing historical traffic data, which leads to an imprecise spatial-temporal features extraction. To this end, we propose Fine-Grained Features learning based on Transformer-encoder and Graph convolutional networks (FGFTG) to improve the performance of traffic flow prediction in a missing data scenario. FGFTG consists of two components: feature extractors and a data completer. The feature extractors learn fine-grained spatial-temporal representations from spatial and temporal perspectives. They extract smoother representation with the information of distance and direction from a spatial perspective based on graph convolutional networks and node2vec and achieve bidirectional learning for temporal perspective utilizing transformer encoder. The data completer simulates the traffic flow data distribution and generates reliable data to fill in missing data based on generative adversarial networks. Experiments on two public datasets demonstrate the effectiveness of our approach over the state-of-the-art methods.

**Keywords:** Traffic Flow Prediction · Generative Adversarial Network · Graph Convolutional Neural Network · Transformer Encoder.

## 1 Introduction

The appearance of cars from the 19th century has brought tremendous changes to people's life. It dramatically helps government agencies to avoid potential catastrophic traffic accidents [48] and brings convenience to daily travel [3]. With the development of technology, the auto industry has been widely used in intellectualization since the 1980s. In recent years, traffic flow prediction is consequently becoming a hot issue for researchers because it can predict the state of road traffic and is beneficial in a wide range of applications [12, 28, 31]. Traffic jams can be significantly reduced due to route planning based on the prediction. The prediction can also provide insights to the regulatory authorities for decision-making, risk assessment, and traffic management.

The traffic flow is easily influenced by multiple factors such as weather, holidays, and traffic accidents, which tremendously aggravates prediction accuracy. Meanwhile, traffic data provided by road sensors are sometimes missing due to sensor damage or network congestion. For example, the missing data in PeMS dataset [44] accounted for 11.3%. Consequently, analyzing and coping with the fast-changing and missing data effectively becomes an urgent problem to solve. Existing studies [47, 23, 43, 13, 39] usually consider spatial and temporal information at the same time. Generally, they usually use Graph Convolutional Network to extract spatial features, which only focus on spatial node's flow change (content information) and neighbor information, ignoring the influence of distance and direction information between nodes. Fig. 1 shows the real change of traffic

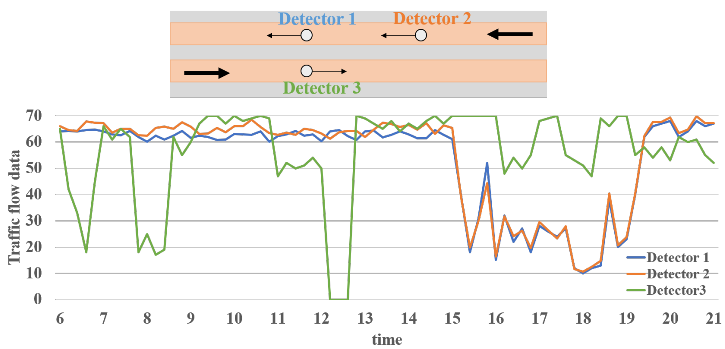


Fig. 1: The line chart of traffic flow (reflect the direction and distance of spatial node).

flow data on the freeway. Detector 1 and detector 2 detect traffic flow data in the same direction at different locations, while detector 1 and detector 3 are in the same location in different directions. The result shows that the distance between detector 1 and detector 3 is closer while the flow distribution is further, indicating that direction is more critical in this region. Likewise, extracting temporal features also has several shortcomings like error accumulation and one-way learning occlusion, resulting in partial temporal information loss. In alleviating the problem of missing data, early studies [21, 5, 14] fill in data according to fixed distribution assumptions, such as Gaussian distribution. Nevertheless, these fixed data distributions usually fail to fit the real flow changes due to the insufficient consideration of the actual traffic flow context. In summary, these kinds of methods are too brutal to ensure the robustness of filling data.

Facing these problems, we propose a fine-grained spatial-temporal representation learning with missing data completion for accurate traffic flow prediction, which is challenging due to: (1) integrating spatial features like distance and direction information and (2) considering bi-relation among historical time intervals from the temporal aspect and (3) generating reliable traffic flow data for missing values under the high complexity and variability distribution. To tackle the aforementioned challenges, we propose FGFTG (Fine-Grained spatial-temporal Features learning based on Transformer encoder and Graph convolutional networks), which contains two parts. The first is temporal-spatial feature extractors. In particular, spatial feature extractor based on graph convolution neural network (GCN [7] ) and node2vec [15] technique is designed for integrating content, neighbour, distances, and directions information. Temporal feature extractor based on transformer encoder [35] is created to consider bi-relations among time intervals. The second part is data completer. We design DCGAN (Data Completion based on Generative Adversarial Networks) to fit the complex distribution of traffic flow data and generate reliable data for missing values. In summary, we present the main contributions as follows:

- We innovatively propose FGFTG, a spatial-temporal traffic flow data prediction framework, to learn fine-grained representations by two feature extractors.
- We design a data completer model DCGAN to simulate the real traffic flow data distribution and generate reliable data to fill in missing data.
- We conduct experiments on two public datasets. The experiments show that our model FGFTG significantly outperforms the state-of-the-art traffic flow prediction models and demonstrate our model's effectiveness in filling in missing data.

## 2 Related Work

Early in the 1960s, traffic flow prediction was regarded as transportation and operational research, which mainly depends on queuing theory and simulation experiments [9]. Later in the 21st century, data-driven based on the statistics is presented. The most popular methods are Auto-Regressive Integrated Moving Average (ARIMA) [36], Kalman filtering [26], Exponential Smoothing model [45], etc [8, 32, 38, 30]. Nevertheless, this type of model is easily influenced by dynamic features such as weather, traffic accidents, and holidays, causing inaccurate results. In 2014, researchers applied deep learning technologies, such as convolution neural network (CNN), recurrent neural network (RNN), and long short term memory (LSTM) to this field [18, 37, 50, 12, 42, 17], which effectively solves the problems of massive data and complex factors [27, 29, 46]. But researchers only focus on a single road or area [1, 10, 16, 24, 33, 40, 41] to reduce the computation process, while ignoring the spatial dependency between roads or areas. In 2017, Zheng et al. [47] proposed the ST-ResNet model, which is the first time proposing the concept of spatial-temporal traffic flow prediction. In their work, not only the temporal features were calculated, spatial features are also considered. Subsequently, many researchers started this study based on spatial and temporal dependency.

In 2018, Li et al. [23] proposed Graph Convolutional Networks based on Recurrent Neural Network (GCRNN) to deal with the complex spatial dependency on road networks and non-linear temporal dynamics with changing road conditions. Yu et al. [44] proposed Spatio-Temporal Graph Convolutional Networks (STGCN) comprising several spatio-temporal convolutional blocks to model spatial and temporal dependencies. Yao et al. [43] proposed a DeepMulti-View Spatial-Temporal Network (DMVST-Net) framework to model both spatial and temporal relations. Later in 2019, DeepSTN [25] chose the ConvPlus structure to model the long-range spatial dependence among crowd flows in different regions and combine PoI (Point of Interest) distributions and time factors to express the effect of location attributes. Cao et al. [4] analyzed seasonal dependencies based on data analysis and extracted different features based on these dependencies for training the prediction model. Geng et al. [13] proposed a spatial-temporal multi-graph convolution network (ST-MGCN) from three aspects of neighborhood graph, functional similarity graph, and transportation connectivity graph to extract temporal features for traffic prediction. In 2020, Sun et al. [34] divided the urban area into different irregular regions by road network and viewed each region as a node that is associated with time-varying inflow and outflow. Auto-ST [22] designs a novel search space tailored for the spatio-temporal domain, which consists of optional convolution operations and learnable skip connections. However, this work neglects the problem of information loss in the temporal dimension and cannot efficiently integrate the content, neighbors, distance, and direction information of nodes in the spatial dimension.

## 3 Methodology

Fig.2 illustrates the framework of FGFTG. It consists of two feature extractors and a data completer, where the latter provides data support for the former. Specifically, the traffic spatial-temporal flow graph containing missing data go through the DCGAN model to fill in the incomplete parts. In this way, we gain the traffic flow graph without missing values at T-2, T-1, T+1, and T+2. Then, feature extractors regard these graphs as input, utilizing a spatial feature extractor and a temporal feature extractor to gain fine-grained representations. At last, we can predict the traffic flow graph at time T.

### 3.1 Feature Extractors

Current approaches for traffic flow prediction are inadequate for extracting spatial features and temporal features. Traditional methods [47, 23, 43, 13] using GCN to extract spatial features fail to account for the distance and direction information between nodes. Likewise, temporal extraction methods based on RNN and GCN also have some problems, such as accumulation of errors, one-way learning occlusion, and limited GCN field of view, which fail to consider

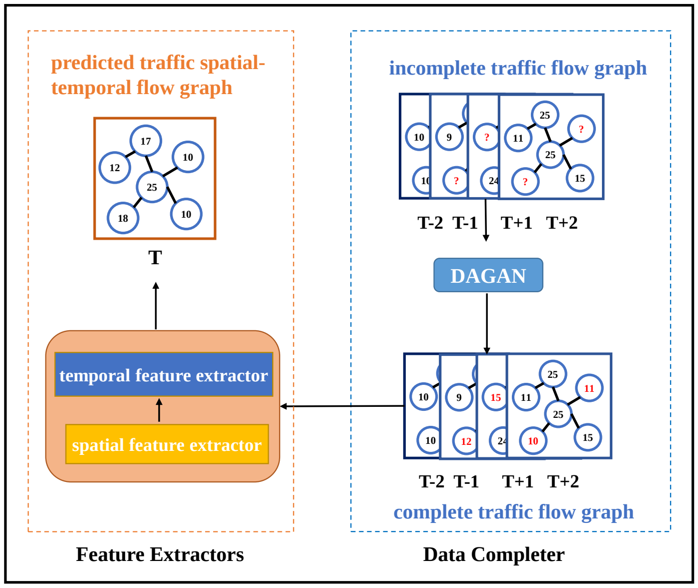


Fig. 2: The framework of FGFTG.

bi-relations among historical time intervals. To fill the gaps in existing research, we propose a feature learning method as shown in Fig. 3. From left to right, each red box represents the traffic spatial-temporal flow graph at different time nodes coming from DCGAN. The yellow part is spatial feature extracting layers, which output vectors with spatial information. The blue part is the temporal feature extracting layer composed of n-layer temporal feature extractors and output vectors with temporal information. Finally, feature vectors are mapped to the predicted flow distribution graph through a fully connected layer. Our model can effectively predict traffic flow data based on the spatial feature extractor and the temporal feature extractor.

**Spatial Feature Extractor** :

In this part, we further utilize GCN [7] and node2vec [15] to obtain spatial features, including the content of spatial nodes, neighbour dependency, distance, and direction. GCN obtains content and neighbor features by combining local graph structure and node features. The inputs are composed of a traffic flow data matrix and neighbor information matrix. According to [18], we consider a multi-layer GCN with the following layer-wise propagation rule:

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

where $\hat{D}$ is the diagonal matrix of traffic flow data, $\hat{A}$ is the adjacency matrix of neighbour information, $W^l$ denotes the trainable weight matrix of layer $l$, $H^l$ is the matrix of activations in the $l^{th}$ layer, and $\sigma$ denotes an activation function.

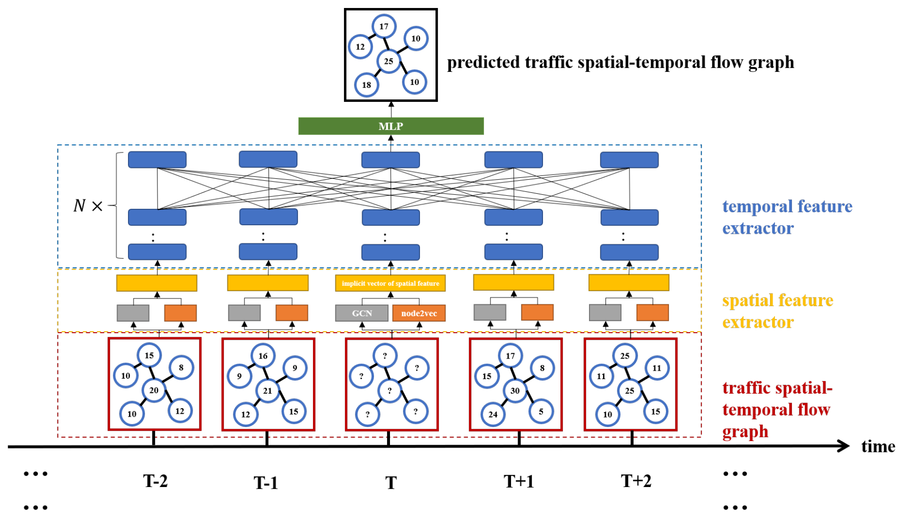


Fig. 3: The framework of feature extractors for traffic flow prediction.

Then, we utilize the node2vec method to obtain distance and direction information by depth-first search and breadth-first search. Finally, we concatenate the two feature vectors, which simultaneously combine the content information, neighbors, distance, and direction. We regard them as the spatial feature, which is the input of the temporal extraction layer.

**Temporal Feature Extractor** : In this part, we select transformer encoder [35] as temporal feature extractor. Compared with CNN, we expand its view field during the whole training phase. Compared with RNN, our model can avoid dependence on time sequence and considers bidirectional learning. The whole

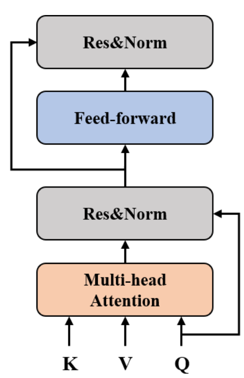


Fig. 4: One layer of temporal feature extractor.

temporal extraction part is composed of multi-layer temporal feature extractors. Fig. 4 demonstrates one layer of the extractor. The first part is Multi-head Attention Networks, which consists of multiple self-attention mechanism networks; intuitively, it helps the network focus on the more critical parts of the prediction task. The second part is Residual Block and Layer Normalization, which can alleviate the problems of over-fitting and gradient disappearing problems caused by the complex structure of the self-attention mechanism. The last part is a feed-forward neural network for further improving the ability of model feature extraction.

**Model Training Based on Self-Supervised Learning** : The traditional training method based on supervised learning is giving the traffic flow data at $t_1, t_2, t_3, t_4$ and predicting the traffic flow data at $t_5$, which mainly focus on regression prediction task. The prior knowledge can improve robustness and performance if the classification task is also considered to assist the prediction task. Our work proposes a training optimization method based on self-supervised learning, which simultaneously considers regression and classification tasks.

To fully utilize data features in the training process, we randomly select one prediction time node and mask or change action in the traffic data. For the masked data, the model performs a regression prediction task. For the changed data, the model performs a classification prediction task which aims to classify and judge whether the data is changed. Compared with the traditional training method, our method holds several advantages. Firstly, it can observe the complete traffic flow data in the training process, which is beneficial for the prediction task. Secondly, the classification sub-task can enhance model robustness and better judge the reliability of predicted traffic data. Thirdly, we make a regression sub-task for the masked data, ensuring the final traffic flow prediction precision.

### 3.2 Data Completer

Traditional data completion methods such as random fill or average fill are too brutal to fit the real data distributions. To this end, we propose a data completer DCGAN based on the idea of GAN to fill in the missing data. It consists of a generator and a discriminator, where the generator tries to generate enhanced data similar to real data distribution. The discriminator aims to judge the authenticity of generated data. Specifically, the generator is constructed by the full connected and deconvolution layers, while the convolution and full connected layers construct the discriminator. We introduce the Kullback-Leibler Divergence (KLD) to evaluate the divergence between real data distribution and generate data distribution. Eq.2 defines the KL divergence of distribution $p(x)$ relative to distribution $q(x)$. Our model can be split into three parts: Convert Data Format, Model Training, and Data Completion Process.

$$\mathrm{KL(p\|q)} = -\int p(x) \ln \left[\frac{q(x)}{p(x)}\right] dx \tag{2}$$

**Convert Data Format** :

In this part, we transform the original data into a spatial-temporal traffic flow matrix for subsequent calculation. As shown in Fig.5, we firstly number the spatial node in a single time point. The six spatial nodes in the figure are numbered 1-6. Its corresponding traffic flow data are [15,10,20,10,8,12], which are regarded as a matrix column. Moreover, each spatial node contains a period of temporal nodes. For example, spatial node 1 has seven temporal nodes whose traffic data are [15,13,12,10,9,7,5]. We take these temporal node traffic data as a row of this matrix. After constructing the matrix, the value "0" of the matrix is considered as the missing data.

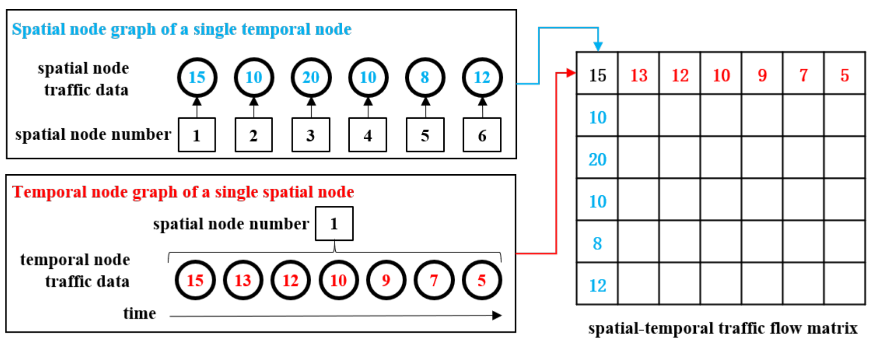


Fig. 5: The transformation of spatial-temporal traffic flow data in matrix form.

**Model Training** : In the training process as shown in Fig.6, the generator produces enhanced data based on Gaussian distribution, and the discriminator judges their authenticity. Both of them train and optimize parameters respectively according to the loss function as Eq.3:

$$\min_G \max_D V(G, D) = E_{x\sim p_{\text{data}}}[D(x)] + E_{x\sim p_G}[1 - D(G(z))] \tag{3}$$

where x denotes real traffic data, z denotes the noise data generated by Gaussian distribution, $G(\cdot)$ is a mapping function of the generator, and $D(\cdot)$ is the neural network function of the discriminator. The specific steps are as follows.

**Discriminator Training** aims to maximize loss under the fixed parameters of the generator. Initially, we sample $\{x^1, x^2, ..., x^n\}$ from the real traffic flow data and obtain enhanced traffic flow data $\{\hat{x}^1, \hat{x}^2, ..., \hat{x}^m\}$ created by generator. Then, we take gradient ascent to optimize parameters $\theta_D$ to maximize the loss function according to the Eq. 4:

$$\begin{aligned} \max_D \ \mathrm{V} &= \frac{1}{m}\sum_{i-1}^{m} D\left(x^i\right) + \frac{1}{m}\sum_{i-1}^{m}\left(1 - D\left(\tilde{x}^i\right)\right), \\ \theta_D &= \theta_D + \mu\nabla\mathrm{V}\left(\theta_D\right). \end{aligned} \tag{4}$$

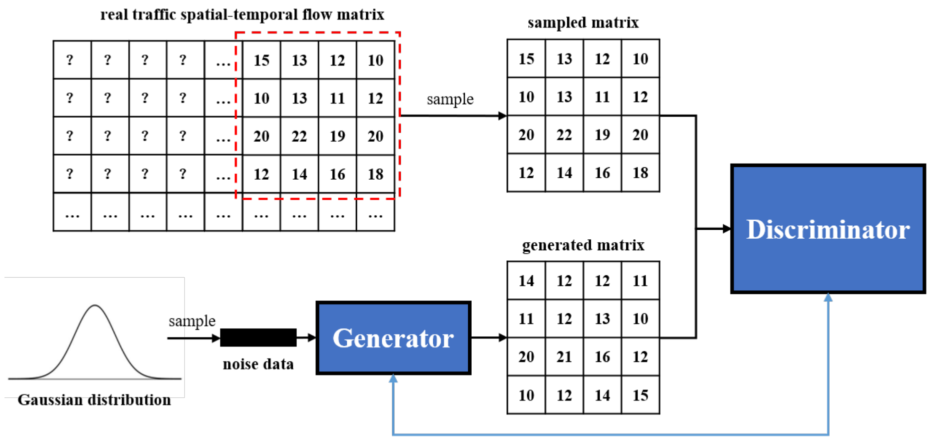


Fig. 6: The process of DCGAN.

**Generator Training** aims to minimize loss under the fix parameters of discriminator. We sample noise data $\{z^1, z^2, ..., z^m\}$ from Gaussian distribution and take gradient descent to optimize parameters $\theta_G$ so as to minimize the loss function according to the Eq. 5:

$$\begin{aligned}\min_{G} \ \mathrm{V} &= \frac{1}{m}\sum_{i-1}^{m}\left(1-D\left(G\left(z^{i}\right)\right)\right), \\ \theta_G &= \theta_G - \mu \nabla \mathrm{V}\left(\theta_G\right).\end{aligned} \tag{5}$$

**Data Completion Process** :

We use the trained generator for the data completion process. Firstly, we locate missing data and construct a real traffic data matrix by its neighbor data. Then, we randomly sample several noise data based on Gaussian distribution and transform them into multiple traffic spatial-temporal flow metrics by the generator. Finally, we calculate the Euclidean distance between the generated metric and the real matrix (except the missing items) and choose the highest similarity matrix to fill in the missing value.

## 4 Experiments and Analysis

### 4.1 Experimental Settings

**Datasets.** Two datasets showing in Table 1 are used in our experiments.

- **PeMS** was collected from Caltrans Performance Measurement System in real-time, which records 59 days of traffic flow data from January to February in 2018 at important sites of California Highway in the United States. There are 307 sensors in the system, corresponding to 307 spatial nodes. The sensors are connected with 341 roads, corresponding to 341 spatial edges. Every five

Table 1: The information of dataset.

| DataSets | spatial node | temporal node | spatial edge | traffic flow Max/Avg | record days |
|---|---|---|---|---|---|
| **PeMS** | 307 | 16,992 | 341 | 336/186 | 59 |
| **XIAMEN** | 95 | 17,856 | 296 | 254/112 | 62 |

minutes is regarded as a time node, and every day corresponds to 288 time nodes. The whole dataset corresponds to 16,992 time nodes in total.
- **XIAMEN** records 62 days of traffic flow data in Xiamen, China, from July to August in 2016. There are 95 sensors corresponding to 95 spatial nodes. The sensors are connected with 296 roads, corresponding to 296 spatial edges. Every five minutes is regarded as a time node, and every day corresponds to 288 time nodes. The whole dataset corresponds to 17,856 time nodes in total.

In particular, we sort the data by time order, and the first 60% is the training set, 20% are used for testing, and the remaining 20% for validation.

**Evaluation Metric** : We use Rooted Mean Square Error (RMSE) and Mean Absolute Errors (MAE) to evaluate the proposed model, which are defined as Eq. 6 and Eq. 7:

$$RSME = \sqrt{\frac{\sum_{i=1}^{n} \left(\hat{y}_i - y_i\right)^2}{n}} \tag{6}$$

$$MAE = \frac{\sum_{i=1}^{n} \left|\hat{y}_i - y_i\right|}{n} \tag{7}$$

where $y_i$ and $\hat{y}_i$ respectively denote the real value and prediction value at the $i$th time interval. $n$ is the total number of samples in the testing data.

### 4.2 Performance of Traffic Flow Prediction

**Methods for Comparison** : We compare our method FGFTG with the following 8 baselines:

- **Linear regression (LR)**: We compare our method with different versions of linear regression methods: Ridge Regression (i.e., with L2-norm regularization) and Lasso (i.e., with L1-norm regularization).
- **History Average (HA)**: Historical average predicts the future traffic flow data using average values of historical records.
- **Vector Auto Regression (VAR) [2]**: VAR can capture the pairwise relationships among all flows but has massive computational costs due to a large number of parameters.

- **Autoregressive Moving Average Model (ARIMA) [36]**: ARIMA is a combination of Auto Regression (AR) and Moving Average (MA) with a different process.
- **XGBoost [6]**: XGBoost is a mighty boosting tree-based method and is widely used in data mining applications.
- **STGCN [44]**: STGCN is a universal framework for processing structured time series. It is able to tackle traffic network modeling and predicting issues and be applied to more general spatio-temporal sequence learning tasks.
- **MSTGCN [11]**: Each module of MSTGCN uses the GCN model to extract spatial features. It uses a one-dimensional convolution method to extract temporal features to capture the spatial-temporal correlation of traffic data effectively.
- **GMAN [49]**: GMAN uses an encoder-decoder structure to simulate the influence of spatial-temporal factors on traffic conditions. The encoder encodes the input traffic characteristics. The decoder converts the encoded traffic characteristics into the traffic feature vector and utilizes the traffic feature vector to predict the output sequence.

**Performance Comparison** :
Table 2 shows the performances of eight baselines and our model on two datasets with missing values. The results show that our FGFTG method performs the best in terms of all measurements on both datasets. The deep learning models achieve a better performance than the traditional model and tree model. On the PeMS dataset where the missing value accounts for 11.3%, compared with the worst deep learning model STGCN, even the best traditional XGBoost model still holds higher RSME and MAE (an increase of 42.0% and 130.7%, respectively). This result powerfully demonstrates the necessity of studying the deep learning model. Compared with deep learning models, our model FGFTG still achieves better performance.

To verify the effectiveness of the FGFTG structure, we firstly use the traditional training method without Self-Supervised Learning $FGFTG^{withoutSSL}$ to compare with the best deep learning method GMAN. Results show our model achieves a 6.9% and 3.8% lower RMSE and MAE, respectively, indicating that our model performs better prediction performance. Later, we add Self-Supervised Learning in the training process to further improve our performance. Compared with GMAN, FGFTG reduces the RMSE and MAE by 11.3% and 6.7%. Moreover, similar results can be seen on the XIAMEN dataset. Consequently, the results demonstrate that our FGFTG has good generalization performance on flow prediction tasks.

**Time Comparison** : Table 3 shows the running time (measured by the second) of different baselines on PeMS dataset. To make the comparison fairly, all the experiments are conducted on the same machine with a 10-core 20-thread CPU (Xeon E5-2630 v4, 2.20GHz) 128G RAM. We can easily observe that Ridge and

Table 2: The comparison of model prediction effects.

| **Method** | **PeMS** | | **XIAMEN** | |
|---|---|---|---|---|
| | **RMSE** | **MAE** | **RMSE** | **MAE** |
| **Ridge** | 91.52 | 62.82 | 66.51 | 45.47 |
| **Lasso** | 90.49 | 64.71 | 64.22 | 43.16 |
| **HA** | 54.14 | 36.76 | 44.03 | 29.52 |
| **ARIMA** | 68.13 | 32.11 | 43.30 | 24.04 |
| **VAR** | 51.73 | 33.76 | 31.21 | 21.41 |
| **XGBoost** | 34.41 | 22.75 | 26.55 | 18.20 |
| **STGCN** | 24.23 | 9.86 | 16.40 | 7.44 |
| **MSTGCN** | 22.87 | 9.67 | 17.01 | <u>7.13</u> |
| **GMAN** | <u>21.83</u> | <u>8.43</u> | <u>15.84</u> | 7.28 |
| $\boldsymbol{FGFTG^{withoutSSL}}$ | 20.33 | 8.11 | 15.23 | 6.87 |
| **FGFTG** | **19.36** | **7.86** | **14.71** | **6.44** |
| **Improvement** | 11.31% | 2.35% | 7.13% | 9.68% |

Lasso regression take the shortest running time but present the worst performance on precision. HA, ARIMA and FGFTG differ by running time of 1-2 seconds, but the prediction precision differs by 2-4 times. Due to the complex structural design, VAR and XGBoost hold extensive calculations and take more running time. As deep learning methods, STGCN, MSTGCN, and GMAN perform worse than FGFTG, whether in prediction precision or time-consuming. Generally speaking, our proposed FGFTG model achieves the best performance. The same results can be seen on the XIAMEN dataset.

Table 3: The comparison of time consumption.

| **Method** | **cost_time(seconds)** | |
|---|---|---|
| | **PeMS** | **XIAMEN** |
| **Ridge** | 1.51 | 0.93 |
| **Lasso** | 1.32 | 0.75 |
| **HA** | 7.45 | 6.86 |
| **ARIMA** | 6.45 | 5.65 |
| **VAR** | 21.45 | 18.65 |
| **XGBoost** | 34.41 | 26.26 |
| **STGCN** | 34.26 | 9.75 |
| **MSTGCN** | 10.88 | 9.11 |
| **GMAN** | 13.10 | 10.65 |
| $\boldsymbol{FGFTG^{withoutSSL}}$ | 8.92 | 8.62 |
| **FGFTG** | 8.90 | 8.61 |

### 4.3 Effect of Data Completer

To further explore the influence of data missing, we set different scenarios to further show their impact on traffic flow prediction. There are three machine learning algorithms been compared: Ridge, XGBoost, and STGCN. It is evident in Fig.7 that the more missing data, the worse the performance of the model.

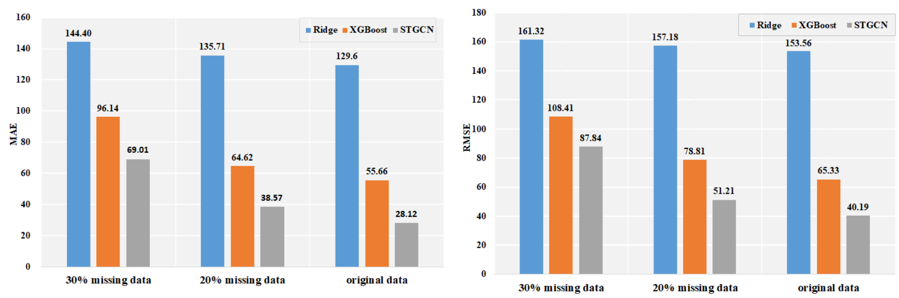


Fig. 7: The effects of missing data in PeMS dataset.

Next, we design two experiments to verify the performance of our proposed data completer: DCGAN.

- **Reliability of Data Generation**: We randomly choose 30% data and set their value to zero, regarding them as missing data. Then, we apply different generation methods to fill the missing data and adopt MAE to evaluate the reliability of different methods.
- **Validity of Promoting Prediction Accuracy**: Different data completion methods are applied to fill the missing data and compare the performance in the spatial-temporal traffic prediction.

**Methods for Comparison** : We compare our DCGAN with the following five baselines:

- **Randomly fill**: Replace missing data with random values taken from the training set.
- **Average fill**: Replace missing data with average values taken from the training set.
- **Moving average fill**: Replace missing data with average values taken from the neighbor of missing data.
- **Matrix Factorization (MF)**[20]: Decompose the user-item interaction matrix into the product of two lower dimensionality rectangular matrices and replace the missing data with the help of the decomposed matrices.
- **Singular Value Decomposition (SVD)**[19]: Decompose the user-item interaction matrix into the product of three matrices: two lower dimensionality rectangular matrices and one non-negative real diagonal matrix.

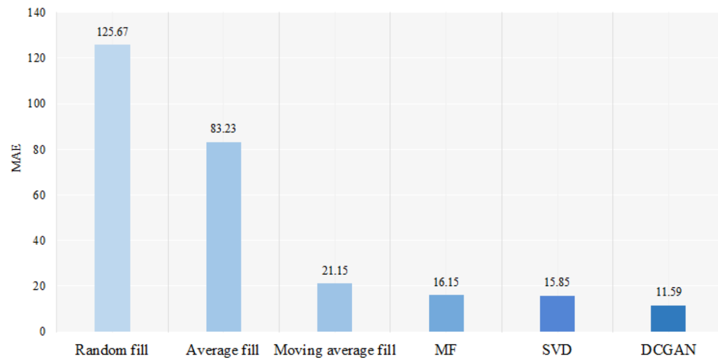


Fig. 8: The results of reliable verification experiment.

**Reliability of Data Generation** : We use the generated data by five baselines to fill in the missing values and calculate the MAE value. As shown in Fig.8, in the existing methods, the reliability of generated data by random filling and average filling is larger than moving average filling. Furthermore, the MAE value of DCGAN is 45.20% lower than moving average filling. Compared with methods based on matrix decomposition, DCGAN also outperforms the MF and SVD methods by 28.23% and 26.88%. Results show that our DCGAN outperforms all baseline and is more reliable.

**Validity of Promoting Prediction Accuracy** : We take five different methods to fill in missing data and regard the results as inputs of three prediction models to promote prediction accuracy. Table 4 shows the performance of the

Table 4: The result of valid promoting prediction accuracy.

| Method | RMSE | | | MAE | | |
|---|---|---|---|---|---|---|
| | Ridge | XGBoost | STGCN | Ridge | XGBoost | STGCN |
| **Origin DataSet** | 153.56 | 65.33 | 40.19 | 129.60 | 55.66 | 28.12 |
| **Random fill** | 158.40 | 67.23 | 50.18 | 134.83 | 59.41 | 31.62 |
| **Average fill** | 144.29 | 58.66 | 39.46 | 100.98 | 46.23 | 26.64 |
| **Moving average fill** | 112.15 | 46.12 | 38.42 | 91.26 | 34.28 | 22.32 |
| **MF** | 115.31 | <u>40.23</u> | 32.62 | 78.41 | 32.16 | 18.62 |
| **SVD** | <u>101.41</u> | 41.48 | <u>31.26</u> | <u>76.45</u> | <u>30.51</u> | <u>14.46</u> |
| **DCGAN** | **91.52** | **34.41** | **24.23** | **62.82** | **22.75** | **9.86** |
| **Improvement** | 9.75% | 14.47% | 22.49% | 17.83% | 25.43% | 31.81% |

proposed method as compared to all other competing methods. Results show that not every data filling method is helpful for traffic flow prediction. Such as the Random filling method, the RMSE, and MAE values increased after filling operation, indicating the instability of this method. Average filling and Moving average filling achieve much better accuracy. Moreover, Moving average filling outperforms Average filling due to its delicate operation. By contrast, our DCGAN achieves the best performance among all the baselines. Taking XGBoost as an example, after DCGAN filling, the RMSE and MAE decrease by 47.33% and 59.13% compared with the origin dataset. Compared with the Moving average filling method, DCGAN reduces the RMSE and MAE by 25.39% and 33.63%, respectively. Additionally, We find that matrix decomposition methods achieve better improvement than data statistics-based methods but are still worse than our DCGAN method. Taking STGCN as an example, compared with the SVD method, the RMSE and MAE of DCGAN decreased by 22.49% and 31.81%, respectively. Consequently, DCGAN significantly outperforms those methods in promoting prediction accuracy.

## 5 Conclusion and Future Work

In this paper, we proposed FGFTG to learn fine-grained spatial-temporal representation for traffic flow prediction. In particular, we first present the spatial-temporal feature extractors to learn better representations, which can fuse content, neighbor, distance, and direction simultaneously and solve the problem of temporal feature loss. Next, to improve the robustness and integrity of data, we propose a novel data completer DCGAN to fill in missing data. Experiments based on two public datasets demonstrate that the proposed FGFTG can lead to better performance than state-of-the-art models. We plan to convert the origin data into graph nodes formula rather than metric to demonstrate the dependency among spatial nodes better for future work. We also plan to explore more temporal dimension features such as weather, holidays, and other factors.

## Acknowledgments

This research was supported by the National Key Research and Development Program of China (2020YFB1712903), the Research Program of Chongqing Technology Innovation and Application Development (CSTC2019jscx-zdztzxX0031 and cstc2020kqjscx-phxm1304), and the Overseas Returnees Innovation and Entrepreneurship Support Program of Chongqing (cx2020097).

## References

1. Abadi, A., Rajabioun, T., Ioannou, P.A.: Traffic flow prediction for road transportation networks with limited traffic data. IEEE Transactions on Intelligent Transportation Systems **16**(2), 653–662 (2015)
2. Ang, A., Piazzesi, M.: A no-arbitrage vector autoregression of term structure dynamics with macroeconomic and latent variables. Journal of Monetary Economics **50**(4), 745–787 (1999)

3. Beirão, G., Cabral, J.S.: Understanding attitudes towards public transport and private car: A qualitative study. Transport policy **14**(6), 478–489 (2007)
4. Cao, L., Ma, K., Cao, B., Fan, J.: Forecasting long-term call traffic based on seasonal dependencies. In: International Conference on Collaborative Computing: Networking, Applications and Worksharing. pp. 231–246. Springer (2019)
5. Chen, C., Wang, Y., Li, L., Hu, J., Zhang, Z.: The retrieval of intra-day trend and its influence on traffic prediction. Transportation research part C: emerging technologies **22**, 103–118 (2012)
6. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international Conference on Knowledge Discovery and Data Mining. pp. 785–794 (2016)
7. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. arXiv preprint arXiv:1606.09375 (2016)
8. Deng, D., Shahabi, C., Demiryurek, U., Zhu, L., Yan, L.: Latent space model for road networks to predict time-varying traffic. In: Acm Sigkdd International Conference (2016)
9. Drew, D.R.: Traffic flow theory and control. McGraw-Hill Series in Transportation **316** (1968)
10. Fan, Z., Xuan, S., Shibasaki, R., Adachi, R.: Citymomentum: an online approach for crowd behavior prediction at a citywide level. In: the 2015 ACM International Joint Conference (2015)
11. Feng, N., Guo, S., Song, C., Zhu, Q., Wan, H.: Multi-component spatial-temporal graph convolution networks for traffic flow forecasting. Journal of Software **30**(3), 759–769 (2019)
12. Fu, R., Zhang, Z., Li, L.: Using lstm and gru neural network methods for traffic flow prediction. In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC). pp. 324–328. IEEE (2016)
13. Geng, X., Li, Y., Wang, L., Zhang, L., Liu, Y.: Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. Proceedings of the AAAI Conference on Artificial Intelligence **33**, 3656–3663 (2019)
14. Ghahramani, Z., Jordan, M.I.: Supervised learning from incomplete data via an em approach. In: Advances in neural information processing systems. pp. 120–127 (1994)
15. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. the 22nd ACM SIGKDD International Conference (2016)
16. Hoang, M.X., Yu, Z., Singh, A.K.: Fccf: forecasting citywide crowd flows based on big data. In: the 24th ACM SIGSPATIAL International Conference (2016)
17. Kang, D., Lv, Y., Chen, Y.y.: Short-term traffic flow prediction with lstm recurrent neural network. In: 2017 IEEE 20th international conference on intelligent transportation systems (ITSC). pp. 1–6. IEEE (2017)
18. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2017), https://openreview.net/forum?id=SJU4ayYgl
19. Klema, V., Laub, A.: The singular value decomposition: Its computation and some applications. Automatic Control IEEE Transactions on **25**(2), 164–176 (1980)
20. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
21. Li, L., Li, Y., Li, Z.: Efficient missing data imputing for traffic flow by considering temporal and spatial dependence. Transportation research part C: emerging technologies **34**, 108–120 (2013)

22. Li, T., Zhang, J., Bao, K., Liang, Y., Li, Y., Zheng, Y.: Autost: Efficient neural architecture search for spatio-temporal prediction. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 794–802 (2020)
23. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting (2017)
24. Li, Y., Zheng, Y., Zhang, H., Chen, L.: Traffic prediction in a bike-sharing system. In: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems. pp. 1–10 (2015)
25. Lin, Z., Feng, J., Lu, Z., Li, Y., Jin, D.: Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. Proceedings of the AAAI Conference on Artificial Intelligence **33**, 1020–1027 (2019)
26. Lippi, M., Bertini, M., Frasconi, P.: Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. IEEE Transactions on Intelligent Transportation Systems **14**(2), 871–882 (2013)
27. Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y.: Traffic flow prediction with big data: A deep learning approach. IEEE Transactions on Intelligent Transportation Systems **16**(2), 865–873 (2015)
28. Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y.: Traffic flow prediction with big data: a deep learning approach. IEEE Transactions on Intelligent Transportation Systems **16**(2), 865–873 (2014)
29. Ma, X., Zhuang, D., He, Z., Ma, J., Wang, Y.: Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. Sensors **17**(4), 818 (2017)
30. Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J.: Predicting taxi–passenger demand using streaming data. IEEE Transactions on Intelligent Transportation Systems **14**(3), 1393–1402 (2013)
31. Polson, N.G., Sokolov, V.O.: Deep learning for short-term traffic flow prediction. Transportation Research Part C: Emerging Technologies **79**, 1–17 (2017)
32. Shekhar, S., Williams, B.M.: Adaptive seasonal time series models for forecasting short-term traffic flow. Transportation Research Record **2024**(1), 116–125 (2007)
33. Silva, R., Kang, S.M., Airoldi, E.M.: Predicting traffic volumes and estimating the effects of shocks in massive transportation systems. Proceedings of the National Academy of Sciences of the United States of America **112**(18), 5643–8 (2015)
34. Sun, J., Zhang, J., Li, Q., Yi, X., Zheng, Y.: Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks. IEEE Transactions on Knowledge and Data Engineering **PP**(99), 1–1 (2020)
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
36. Wei, L., Zheng, Y., Chawla, S., Yuan, J., Xing, X.: Discovering spatio-temporal causal interactions in traffic data streams (2011)
37. Wei, W., Wu, H., Ma, H.: An autoencoder and lstm-based traffic flow prediction method. Sensors **19**(13), 2946 (2019)
38. Xiaolong, L.I., Pan, G., Zhaohui, W.U., Guande, Q.I., Shijian, L.I., Zhang, D., Zhang, W., Wang, Z.: Prediction of urban human mobility using large-scale taxi traces and its applications. Frontiers of Computer Science **6**(001), 111–121 (2012)
39. Xu, J., Zhang, Y., Jia, Y., Xing, C.: An efficient traffic prediction model using deep spatial-temporal network. In: International Conference on Collaborative Computing: Networking, Applications and Worksharing. pp. 386–399. Springer (2018)

40. Xu, Y., Kong, Q.J., Klette, R., Liu, Y.: Accurate and interpretable bayesian mars for traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems **15**(6), 2457–2469 (2014)
41. Xuan, S., Zhang, Q., Sekimoto, Y., Shibasaki, R.: Prediction of human emergency behavior and their mobility following large-scale disaster. ACM (2014)
42. Yang, B., Sun, S., Li, J., Lin, X., Tian, Y.: Traffic flow prediction using lstm with feature enhancement. Neurocomputing **332**, 320–327 (2019)
43. Yao, H., Fei, W., Ke, J., Tang, X., Ye, J.: Deep multi-view spatial-temporal network for taxi demand prediction (2018)
44. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting (2017)
45. Yu, H.F., Rao, N., Dhillon, I.S.: Temporal regularized matrix factorization for high-dimensional time series prediction. In: NIPS. pp. 847–855 (2016)
46. Yu, R., Li, Y., Shahabi, C., Demiryurek, U., Yan, L.: Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting. Proceedings of the 2017 SIAM International Conference on Data Mining (2017)
47. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Thirty-first AAAI conference on artificial intelligence (2017)
48. Zhang, N., Wang, F.Y., Zhu, F., Zhao, D., Tang, S.: Dynacas: Computational experiments and decision support for its. IEEE Intelligent Systems **23**(6), 19–23 (2008)
49. Zheng, C., Fan, X., Wang, C., Qi, J.: Gman: A graph multi-attention network for traffic prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 1234–1241 (2020)
50. Zheng, Z., Yang, Y., Liu, J., Dai, H.N., Zhang, Y.: Deep and embedded learning approach for traffic flow prediction in urban informatics. IEEE Transactions on Intelligent Transportation Systems **20**(10), 3927–3939 (2019)