# Spatial-Temporal Traffic Data Imputation via Graph Attention Convolutional Network

Yongchao Ye[1], Shiyao Zhang[2], and James J.Q. Yu[1*]

[1] Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China

[2] Academy for Advanced Interdisciplinary Studies, Southern University of Science and Technology, Shenzhen 518055, China

12032868@mail.sustech.edu.cn, syzhang@ieee.org, yujq3@sustech.edu.cn

**Abstract.** High-quality traffic data is crucial for intelligent transportation system and its data-driven applications. However, data missing is common in collecting real-world traffic datasets due to various factors. Thus, imputing missing values by extracting traffic characteristics becomes an essential task. By using conventional convolutional neural network layers or focusing on standalone road sections, existing imputation methods cannot model the non-Euclidean spatial correlations of complex traffic networks. To address this challenge, we propose a graph attention convolutional network (GACN), a novel model for traffic data imputation. Specifically, the model follows an encoder-decoder structure and incorporates graph attention mechanism to learn spatial correlation of the traffic data collected by adjacent sensors on traffic graph. Temporal convolutional layers are stacked to extract relations in time-series after graph attention layers. Through comprehensive case studies on the dataset from the Caltrans performance measurement system (PeMS), we demonstrate that the proposed GACN consistently outperforms other baselines and has steady performance in extreme missing rate scenarios.

**Keywords:** Graph Attention · Temporal convolution · Data Imputation

## 1 Introduction

In recent years, many data-driven approaches have been proposed with the development of intelligent transportation system (ITS), such as traffic speed prediction, traffic signal control, and origin-destination prediction [22]. These data-driven approaches heavily rely on high-quality spatial-temporal traffic data. However, due to various natural and human factors, traffic data collected in practice are often incomplete or corrupted, e.g., about 10% traffic data is missing in

* This work is supported by the Stable Support Plan Program of Shenzhen Natural Science Fund No. 20200925155105002, by the General Program of Guangdong Basic and Applied Basic Research Foundation No. 2019A1515011032, and by the Guangdong Provincial Key Laboratory (Grant No. 2020B121201001). James J.Q. Yu is the corresponding author.

Beijing. Some extreme missing scenarios are reported in Alberta, Canada [12]. The average missing rate of traffic data in 7 years was 50%, and the short-term missing rate can reach up to 90%. It is reported that missing data is unfavorable to data-driven models, such as traffic flow prediction [8]. Thus, imputing the missing values by analyzing spatial-temporal traffic features is an urgent issue.

Traditional imputation methods are based on universal interpolation methods such as k-Nearest Neighbor (k-NN) and support vector regression [10]. These methods are inefficient when there are massive missing points. Compared with traditional methods, methods based on deep learning have improved estimation accuracy by capturing temporal or spatial dependency. Denoising stacked autoencoder (DSAE) [5], which combines denoising and stacked autoencoders, is a typical deep model applied to impute traffic data. Subsequent work improved imputation accuracy by using DSAE as a generator and designing a discriminator [4]. However, these methods only focus on isolated road segments and show their limitations in modeling spatial correlation. Further research reconstructed traffic trajectories into a two-dimensional matrix and applied a convolutional neural network (CNN) for encoding and decoding [1].

There is a research gap in the aforementioned imputation methods, especially when they are employed to impute the traffic data in a large region. With a complex network topology, the real-world traffic data is with a non-Euclidean structure. These traditional learning models (such as CNN, etc.) are efficient for data with Euclidean structures, but they are relatively insufficient in modeling non-Euclidean spatial correlation. With a graph structure for relational reasoning, graph neural network (GNN) is a framework that can learn the correlation in topological space [21]. In ITS, GNN and its variants have been applied to traffic forecasting tasks. See [16] for an example. Utilizing graph convolutional layers in a GAN framework, graph convolutional generative autoencoder (GCGA) achieved significant performance in real-time traffic seed estimation [19]. Guo *et al.* proposed ASTGCN, which captured spatial-temporal correlations by attention mechanism and convolution module [6]. GNN has shown its advantages in modeling topological relationships of the traffic network. However, to the best of our knowledge, traffic data imputation method using GNN-based approaches has not been assessed in the literature.

To bridge the research gap, we propose a novel imputation model, namely, graph attention convolutional network (GACN). The primary efforts of this work are summarized as follows.

1. We design a spatial-temporal block with a graph attention layer and a convolutional layer. Compared with the previous work, the graph attention mechanism can better capture non-Euclidean spatial correlations in traffic networks. As far as we are concerned, this is the first time to apply graph attention network (GAT) [13] in traffic data imputation.
2. We propose an end-to-end traffic data imputation model that follows the encoder-decoder structure. Specifically, both the encoder and the decoder consist of two spatial-temporal blocks for spatial-temporal modeling, which are concatenated to obtain an embedding of the input.

3. We conduct comprehensive case studies on real-world traffic datasets. In addition, we investigate a wide missing rate range from 10% to 90% to evaluate all kinds of practical scenarios. The experimental results demonstrate that the proposed model achieves accurate imputation and maintains steady performance under extreme missing scenarios.

The remainder of this paper is structured as follows. Section 2 briefly reviews the development of traffic data imputation methods and attention mechanism. Section 3 presents the problem formulation and introduces the proposed imputation method. Section 4 represents and analyzes experimental results. Finally, the paper is concluded in Section 5.

## 2 Related Work

### 2.1 Traffic Data Imputation

**Traditional Imputation Approaches** In the early traffic data imputation literature, traditional methods can be summarized into three groups, i.e., prediction, interpolation, and statistical learning [10]. Autoregressive integrated moving average (ARIMA) and its variants are typical prediction examples. One distinct shortcoming of the prediction model is that it only uses the foregoing temporal information of the missing points. Commonly used interpolation methods, e.g., k-NN interpolates missing points by averaging neighboring observed points. However, these methods show their limits by mainly focusing on a single traffic sensor or a road section. On the other hand, statistical learning is another line of traffic data imputation research, such as Probabilistic principal component analysis (PPCA) [9]. More advanced method utilized low-rank tensor structures to represent the traffic data and recover missing points [3].

**Deep Learning Based Approaches** In the era of big data, many deep learning based approaches have been proposed for time-series data imputation, but not necessarily traffic data imputation. Generative adversarial networks (GAN), which learn the distribution of training samples, have been applied to create data imputation models. Yoon *et al.* proposed generative adversarial imputation nets (GAIN) [17]. In GAIN, the discriminator was fed with a hint vector which provided auxiliary information about the missing position. Luo *et al.* proposed an end-to-end GAN (E$^2$GAN) framework for multivariate time-series imputation [11]. These universal imputation models usually concentrate on temporal modeling rather than the correlations between different time-series. There are also methods that capture spatial correlations with CNN layers [1, 15]. For example, Yang *et al.* designed a GAN model with CNN layers and bidirectional attention [15]. However, these methods neglect the network topology, which is crucial to spatial correlations modeling.

### 2.2 Graph Attention in ITS

The attention mechanism is first emerged as a method to improve recurrent neural networks' performance on sequence-to-sequence learning tasks. The idea of the attention mechanism is to select features that have a relatively larger impact on the task. In real-world traffic networks, the graph is large-scale and combined with noise. Extracting features from these graphs is difficult and may lead to massive computational cost. A practical solution is to incorporate the attention mechanism on graphs, namely graph attention networks (GAT) [13]. Graph attention mechanism allows a model to focus more on relevant parts of the graph. In ITS-related applications, especially traffic speed prediction, many methods based on graph attention mechanism have been proposed [6, 20]. In the reported results, these methods outperform graph convolution networks (GCN) in many scenarios.

Inspired by the above literature and making use of its outstanding capability in modeling complex and dynamic traffic networks, we employ graph attention mechanism in our model to learn the spatial correlations for traffic data imputation.

## 3 Methodology

The proposed GACN estimates missing values by capturing spatial-temporal correlations from the observations. In this section, we first introduce our data preprocessing techniques, including traffic network representation and traffic data imputation formulation. Then, we present the structure of the proposed GACN. Finally, we give a detailed description of the spatial graph attention and temporal convolutional layers respectively.

### 3.1 Data Preprocessing

In this paper, we define a traffic network as an undirected graph $G = (V, E, \boldsymbol{A})$, where $V$ is the set of vertexes, i.e., the set of sensors in a specific region $|V| = N$. Edges $e(v_i, v_j) \in E$ represent the spatial correlations between sensors. The adjacent matrix of graph $G$ is represented by $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, in which $a_{ij} = 1$ indicates sensor $i$ and sensor $j$ are adjacent and $a_{ij} = 0$ otherwise. Similar to [20], we generate the adjacent matrix by thresholded Gaussian kernel,

$$a_{ij} = \begin{cases} 1, \exp\left(-\frac{\mathrm{dist}(v_i,v_j)^2}{\sigma^2}\right) \geq \varepsilon \text{ or } i = j \\ 0, \text{otherwise} \end{cases}, \tag{1}$$

where $\sigma^2$ and $\varepsilon$ are the thresholds that decide the sparsity of $\boldsymbol{A}$. $\mathrm{dist}(v_i, v_j)$ is the Euclidean distance between sensors $i$ and $j$, which can be derived from their respective locations.

Suppose a sensor records $T$ observations in a day. We denote the ground truth traffic speed in a day as $\boldsymbol{S} = \{s_i^t\} \in \mathbb{R}^{N\times T}$, where $s_i^t$ is the observation from sensor $i$ at time $t$. Fig. 1 depicts a diagram of traffic data imputation problem. We first randomly drop points on a complete observation $\boldsymbol{S}$ as input. A missing mask $\boldsymbol{M} = \{m_i^t\} \in \mathbb{R}^{N\times T}$ that records the missing positions is defined as,

$$m_i^t = \begin{cases} 1, \text{if } x_i^t \text{ is observed} \\ 0, \text{if } x_i^t \text{ is missing} \end{cases}. \tag{2}$$

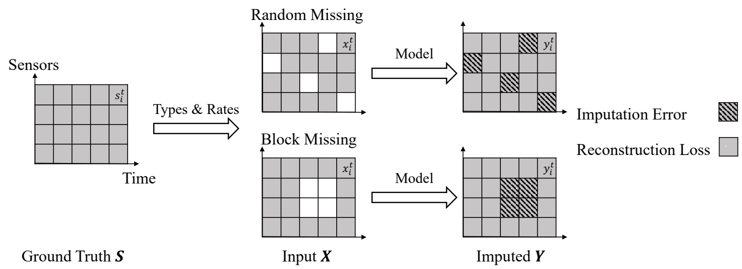



**Fig. 1.** Diagram of traffic data imputation problem.

Then the input $\boldsymbol{X} = \{x_i^t\} \in \mathbb{R}^{N \times T}$ can be represented by,

$$\boldsymbol{X} = \boldsymbol{S} \odot \boldsymbol{M}, \tag{3}$$

where $\odot$ is the element-wise multiplication. $x_i^t = 0$ indicates a missing value and $x_i^t = s_i^t$ indicates an observation. Missing rate is defined as the ratio of missing values to the observations. The output of the model is denoted as $\boldsymbol{Y} = \{y_i^t\} \in \mathbb{R}^N \times T$. The imputation error $E(\boldsymbol{S}, \boldsymbol{Y})$, which is calculated on missing points, is defined as follows,

$$E(\boldsymbol{S}, \boldsymbol{Y}) = \sum_{i,t} m_i^t \|s_i^t - y_i^t\|^2, \tag{4}$$

Take Fig. 1 as an example. The model takes random missing points as input. Then, the imputation error is only calculated on those blocks with diagonal shadow (the missing points).

According to the previous studies, we investigate two traffic data missing types in this study, i.e., random missing (RM) and block missing (BM) [8]. BM is the same as not missing at random (NMR) in [8]. Fig. 1 gives an illustration of two missing types, where the blank blocks are the missing points, and the blocks with dark shadows are observations. In RM, which is typically caused by transmission failure, all the missing points are randomly scattered. BM is more commonly caused by data center errors or power failure. The BM missing points are gregarious in the spatial and temporal dimensions.

### 3.2 Graph Attention Convolutional Network for Imputation

We proposed a graph attention convolutional network (GACN) for traffic data imputation in this paper. As shown in Fig. 2, GACN has an encoder-decoder structure, where the encoder extracts the traffic characteristics, and the decoder recovers the input sequence. Both the encoder and the decoder are composed of two spatial-temporal blocks (ST block). In each block, there is a graph attention layer for feature extraction in the spatial dimension and a standard convolutional layer for time-series modeling along the temporal dimension. Specifically, blocks

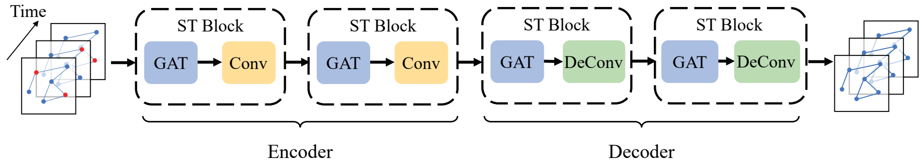



**Fig. 2.** Structure of the proposed graph attention convolution network.

in the encoder apply convolution layers, which perform merging and learn temporal characteristics. Blocks in the decoder apply deconvolution layers, which recover the internal features to their original size.

As illustrated in Fig. 2, the input $\boldsymbol{X}$ of GACN is the observation from a specific region embedded on a graph. Each node has the observation from a sensor in a day, which is denoted as $\boldsymbol{x}_i = \{x_i^1, x_i^2, ..., x_i^T\}$. The red points are missing points which are set to zero in $\boldsymbol{X}$. And the output of GACN is a graph that has the identical size of the input. The missing positions may change on different timestamps (i.e., in Fig. 2, the red points are on different nodes in different timestamps). We define the reconstruction loss $L(\boldsymbol{S}, \boldsymbol{Y})$ of GACN as the mean squared error between input and output feature maps. Let the network parameters be $\theta$, the objective of the training is to minimize the reconstruction loss as follow,

$$\arg\min_{\theta} L(S, Y) = \arg\min_{\theta} \sum_{i,t} \|s_i^j - y_i^j\|^2. \tag{5}$$

Note that this reconstruction loss is different from the imputation error mentioned in Eq. 4. As shown in Fig. 1, reconstruction loss is calculated on the entire output with dark shadow while imputation error is only calculated on the missing points. The reason is that missing positions may change on different days. In the training stage, instead of only focusing on the missing points, we pay more attention to learn the correlation and distribution from the entire observation. The network parameter can be optimized by gradient descent algorithms.

### 3.3 Spatial Graph Attention

The traffic condition of a road section is changing over time. In addition, it is affected by adjacent road sections. Such highly dynamic influence poses challenges in spatial modeling. To address this issue, we apply the graph attention network (GAT) [13] to learn the spatial correlations. GAT captures dynamic spatial correlations by applying a self-attention strategy and calculating dynamic weights between vertexes.

The input of a graph attention layer can be denoted as $\boldsymbol{H} = \{h_1, h_2, ..., h_N\}$, where $N$ is the number of vertexes, $h_i \in \mathbb{R}^F$ represents each vertex that has $F$-dimension features. As mentioned in the above section, each sensor has $T$ observed values in a day, i.e., $T$-dimension features. Thus, the input of the first graph attention layer in GACN is $\boldsymbol{X} \in \mathbb{R}^{N \times T}$, which is with missing points. Since the temporal features are captured by convolutional layers, we set the output dimension the same as the input dimension for all graph attention layers in GACN. With a weight matrix $\boldsymbol{W} \in \mathbb{R}^{F \times F}$, the graph attention layer can transform the input to the output feature space. Consequently, the attention coefficient $e_{ij}$ is calculated by a self-attention mechanism $a : \mathbb{R}^T \times \mathbb{R}^T \to \mathbb{R}$,

$$e_{ij} = a(\boldsymbol{W}h_i, \boldsymbol{W}h_j). \tag{6}$$

The coefficient represents the influence of vertex $j$ on vertex $i$. Generally, every vertex in the graph affects each other. However, calculating coefficients in the entire graph can be expensive. Thus, considering the graph topology, we only compute the coefficients between adjacent vertexes. The hypothesis is that the adjacent vertexes have more significant influence than non-adjacent vertexes. The output $h'_i$ is only affected by the input from adjacent nodes. Then, with LeakyReLU [14] as the activation function, attention coefficients from neighbors are normalized by softmax,

$$\alpha_{ij} = \frac{\exp(\mathrm{LeakyReLU}(e_{ij}))}{\sum_{k \in \mathcal{N}_i} \exp(\mathrm{LeakyReLU}(e_{ik}))}, \tag{7}$$

where $\mathcal{N}_i$ is a set of adjacent vertexes of $i$. To prevent overfitting, we randomly dropout normalized attention coefficients in the training stage. Finally, the output features of vertex $i$ are updated by the attention coefficient and the input, which is calculated by,

$$h'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \boldsymbol{W} h_j\right), \tag{8}$$

where $\sigma(\cdot)$ is a non-linear activation function. We apply exponential linear unit (ELU) here, which is the same as in [13].

### 3.4 Temporal Convolution and Deconvolution

After the graph attention layer capturing the spatial information from adjacent sensors, a canonical convolutional operation is performed on the temporal dimension. The convolutional layers are with vector-like convolutional kernels which aggregate neighboring values in time-series. After the convolutional layers in the encoder part transform the input into the internal feature maps, the decoder restores the feature maps back to the original size. Besides convolution layers, deconvolution layers perform reversed convolutions that restore the input. We use LeakyReLU which is relatively simple in gradient calculation as the activation function [14]. Both the convolution and deconvolution layers are carefully designed to ensure that the input and output time-series share the same size. The specific hyperparameters of convolution and deconvolution layers are listed in Table 1.

**Table 1.** Settings of convolution and deconvolution layers

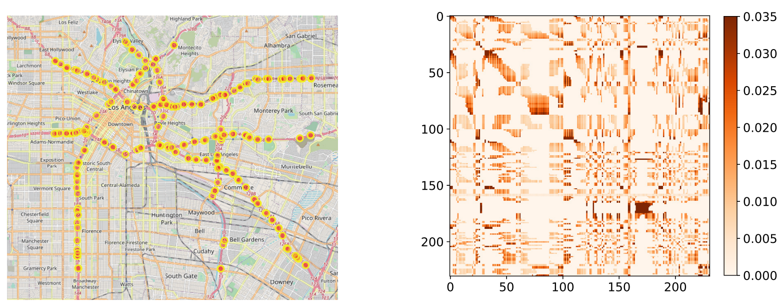
| Layer | Kernel | Stride | Padding | Out Dimension |
|---|---|---|---|---|
| Conv1 | 4 | 2 | 1 | $T/2$ |
| Conv2 | 4 | 2 | 1 | $T/4$ |
| DeConv1 | 4 | 2 | 1 | $T/2$ |
| DeConv2 | 2 | 2 | - | $T$ |

## 4 Case Studites

In this section, we evaluate the performance of the proposed model by case studies. First, we introduce the dataset and detailed experiment settings. Then, we conduct simulations with different missing scenarios and compare the proposed model with other baselines.

### 4.1 Dataset

PeMSD7[1] is collected from District 7 in California Performance Measurement System (PeMS). PeMS provides real-time and historical 5-min average traffic speed data, which is applied in this study. There are in total over one thousand sensors on the arterial roads of District 7. Similar to [18], we selected 231 sensors from the central area, which have a more complicated spatial distribution. The locations of 231 sensors are shown in Fig. 3a, which are in the downtown of Los Angeles. The investigated time ranges from May 1, 2012, to June 30, 2012, excluding weekends. No missing points are found in this period.




(a) Location of sensors (b) Attention coefficients

**Fig. 3.** Sensor distribution of PeMSD7 and results of spatial attention mechanism.

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

### 4.2 Experiment Configurations

We investigate two missing data types as mentioned in Section 3.1 in this study, i.e., random missing (RM) and block missing (BM). To investigate the proposed model in typical missing scenarios as well as in extreme cases, we select a wide missing rate range from 10% to 90% with 10% interval. For each sample, the missing points are randomly erased according to the missing rate and missing type. Then, the missing points together with the observations are fed into the network. For cross-validation, we sequentially group 80% of the data for training, 10% for validation, and 10% for testing. For samples in both the training and testing stage, the missing points are randomly erased ten times, i.e., the number of input samples is enlarged ten times compared to the original dataset. Similarly, the experiment is repeated ten times to reduce the impact of randomness when evaluating the baselines.

The proposed network is trained by Adam optimizer [7]. The learning rate starts at 0.0005 and is subsequently adjusted with a decay rate of 0.5. The negative slope of LeakyReLU is set to 0.1. And the dropout layer in GAT is with a dropout rate of 0.2. During the training stage, the batch size is set to 8. Same as [18], the thresholds $\sigma^2, \varepsilon$ in Eq. 1 are set to 10 and 0.5, respectively. The model is iterated for 150 epochs. All experiments are implemented with PyTorch and conducted on an NVIDIA GeForce RTX 2080Ti GPU.

The imputation accuracy is evaluated by Mean Absolute Percentage Error (MAPE), which is defined as follows:

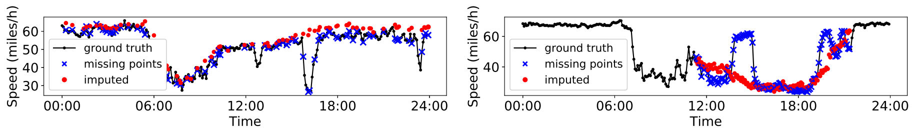
$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\widehat{x}_i - x_i}{x_i}\right| \times 100\%, \tag{9}$$

where $\widehat{x}_i$ is the imputed traffic speed at $i$, and $x_i$ is the ground truth observation.

### 4.3 Baselines

We select the following imputation methods as a comparison of the proposed model:

- **Historical Average (HA)**: In this paper, we average the previous 5 days to estimate the missing values.
- **k-Nearest Neighborhood (k-NN)**: k-NN is a typical example of interpolation-based methods. The imputation is performed by calculating the average value of neighboring points [10]. In this paper, we set $k = 4$.
- **Support Vector Regression (SVR)**: We select SVR as a representative of regression-based imputation methods [2].
- **Denoising Stacked Autoencoder (DSAE)**: We choose DSAE as an example of imputation methods based on deep learning. In this paper, we use the DSAE with the same hyperparameters in [4].
- **Bayesian Gaussian CP decomposition (BGCP)**: Based on probabilistic matrix factorization, BGCP utilized variational Bayes and achieved better performance compared to other tensor-based imputation methods [3].




(a) Random Missing (RM) (b) Block Missing (BM)

**Fig. 4.** Imputation results of the proposed GACN. (30% missing rate)

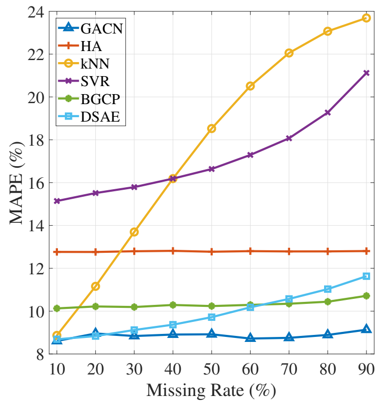
### 4.4 Experimental Results

**Imputation Results** The attention coefficients matrix is shown in Fig. 3b, where the $i$-th row presents the spatial correlation between sensor $i$ and each other. As we constructed an undirected graph, the matrix is approximately symmetric. Since the graph attention is calculated on the adjacent nodes, the attention matrix is sparse. To see the detailed imputation results, we randomly select one sensor and visualize the imputation results using the proposed GACN in Fig. 4. The black dot-dashed line is the ground truth observations, and the red dotted line is the output of GACN. The missing points are marked as blue crosses. In Fig. 4a where a random missing scenario is demonstrated, though some sharp fluctuations are smoothed, the proposed model can accurately recover the missing points as well as other observations. Compared to random missing, block missing in Fig. 4b is more difficult. Although the imputed output has the same overall trend as the ground truth, relatively large fluctuation with continuous missing points is not restored well.

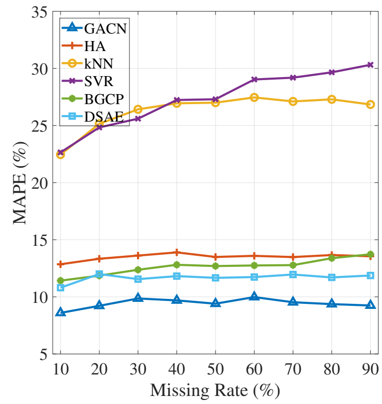
**Performance Comparison** Fig. 5 presents the performance comparison of the proposed GACN and other baselines. In the comparison, we can have the following observations. First of all, GACN achieves the best imputation accuracy in most scenarios and has similar results to other methods at low missing rates (e.g., 10%). Secondly, for the same missing types, while other baselines' performance drops as the missing rate increases, the proposed GACN maintains its outstanding recovery accuracy. The reason is that the proposed GACN learns the overall traffic distribution by reconstruction loss. Methods like KNN and SVR, which only focus on the missing points, can be effective when only a few points are missing. Thirdly, all the methods have better performance in RM than BM, which is in accordance with the results shown in Fig. 4. The proposed GACN has relatively steady MAPE while other methods degenerate rapidly. This observation indicates that spatial correlations of adjacent sensors may provide auxiliary information when missing points are continuously distributed.

## 5 Conclusion

In this paper, we propose a novel GACN for traffic data imputation. An undirected graph is first utilized to represent the traffic network. Then, data impu-




(a) Random Missing (RM)




(b) Block Missing (BM)

**Fig. 5.** Imputation MAPE (%) of different methods for different missing types.

tation is implemented by a graph attention convolutional network that follows an encoder-decoder structure. Both the encoder and the decoder consist of two stacked spatial-temporal blocks. In each block, a graph attention layer extracts spatial correlations, and a convolution/deconvolution layer models temporal relations of the traffic data. Comprehensive case studies are conducted to evaluate the imputation accuracy of the proposed model. Specifically, we consider two missing types and a wide missing rate range from 10% to 90%. Experimental results show that the proposed method outperforms other imputation baselines and maintains steady performance in extreme missing scenarios. In future studies, we plan to further improve the imputation accuracy by employing the missing mask in the training stage.

## References


1. Benkraouda, O., Thodi, B.T., Yeo, H., Menendez, M., Jabari, S.E.: Traffic data imputation using deep convolutional neural networks. IEEE Access **8**, 104740–104752 (2020)
2. Castro-Neto, M., Jeong, Y.S., Jeong, M.K., Han, L.D.: Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Systems with Applications **36**, 6164–6173 (2009)
3. Chen, X., He, Z., Sun, L.: A bayesian tensor decomposition approach for spatiotemporal traffic data imputation. Transportation Research Part C: Emerging Technologies **98**, 73–84 (2019)
4. Chen, Y., Lv, Y., Wang, F.Y.: Traffic flow imputation using parallel data and generative adversarial networks. IEEE Transactions on Intelligent Transportation Systems **21**(4), 1624–1630 (2020)
5. Duan, Y., Lv, Y., Liu, Y.L., Wang, F.Y.: An efficient realization of deep learning for traffic data imputation. Transportation Research Part C: Emerging Technologies **72**, 168–181 (2016)

6. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, AAAI-19. vol. 33, pp. 922–929. Honolulu, Hawaii, USA (2019)
7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations, ICLR. San Diego, CA, USA (2015)
8. Li, L., Zhang, J., Wang, Y., Ran, B.: Missing value imputation for traffic-related time series data based on a multi-view learning method. IEEE Transactions on Intelligent Transportation Systems **20**(8), 2933–2943 (2019)
9. Li, Q., Jianming, H., Li, L., Yi, Z.: PPCA-Based missing data imputation for traffic flow volume: A systematical approach. IEEE Transactions on Intelligent Transportation Systems **10**(3), 512–522 (2009)
10. Li, Y., Li, L., Li, Z.: Missing traffic data: comparison of imputation methods. IET Intelligent Transport Systems **8**(1), 51–57 (2014)
11. Luo, Y., Zhang, Y., Cai, X., Yuan, X.: E2GAN: End-to-end generative adversarial network for multivariate time series imputation. In: Proceedings of 28th International Joint Conference on Artificial Intelligence, IJCAI-19. pp. 3094–3100. Macao, SAR, China (2019)
12. Tan, H., Feng, G., Feng, J., Wang, W., Zhang, Y.J., Li, F.: A tensor-based method for missing traffic data completion. Transportation Research Part C: Emerging Technologies **28**, 15–27 (2013)
13. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
14. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015)
15. Yang, B., Kang, Y., Yuan, Y., Huang, X., Li, H.: ST-LBAGAN: Spatio-temporal learnable bidirectional attention generative adversarial networks for missing traffic data imputation. Knowledge-Based Systems **215**, 106705 (2021)
16. Ye, J., Zhao, J., Ye, K., Xu, C.: How to build a graph-based deep learning architecture in traffic domain: A survey. IEEE Transactions on Intelligent Transportation Systems pp. 1–21 (2020)
17. Yoon, J., Jordon, J., van der Schaar, M.: GAIN: Missing data imputation using generative adversarial nets. In: Proceedings of the 35th International Conference on Machine Learning, ICML-18. pp. 5689–5698. Stockholm, Sweden (2018)
18. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI-18. pp. 3634–3640. Stockholm, Sweden (2018)
19. Yu, J.J.Q., Gu, J.: Real-time traffic speed estimation with graph convolutional generative autoencoder. IEEE Transactions on Intelligent Transportation Systems **20**(10), 3940–3951 (2019)
20. Zhang, C., Yu, J.J.Q., Liu, Y.: Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting. IEEE Access **7**, 166246–166256 (2019)
21. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. AI Open **1**, 57–81 (2020)
22. Zhu, L., Yu, F.R., Wang, Y., Ning, B., Tang, T.: Big data analytics in intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems **20**(1), 383–398 (2019)