Missing Data Repairs for Traffic Flow With Self-Attention Generative Adversarial Imputation Net

Weibin Zhang, Pulin Zhang, Yinghao Yu, Xiying Li, *Member, IEEE*, Salvatore Antonio Biancardo, and Junyi Zhang

***Abstract*— With the rapid development of sensor technologies, time series data collected by multiple and spatially distributed sensors have been widely used in different research fields. Examples of such data include geo-tagged temperature data collected by temperature sensors, air pollutant monitoring data, and traffic data collected by road traffic sensors. Due to sensor failure, communication errors and storage loss, etc., data collected by sensors inevitably includes missing data. However, models commonly used in the analysis of such large-scale data often rely on complete data sets. This paper proposes a model for the imputation of missing data of traffic flow, which combines a self-attention mechanism, an auto-encoder, and a generative adversarial network, into a self-attention generative adversarial imputation net (SA-GAIN). The introduction of the self-attention mechanism can help the proposed model to effectively capture correlations between spatially-distributed sensors at different time points. Adversarial training through two neural networks, called generators and discriminators, allows the proposed model to generate imputed data close to the real data. In comparison with different imputation models, the proposed model shows the best performance in imputing missing data.**

***Index Terms*— Data imputation, spatio-temporal analysis, deep learning, generative adversarial network, self-attention.**

I. Introduction

IN RECENT years, with the deployment of a large number of sensors, many traffic models involve the analysis of time-series data. The data required for these tasks are usually spatially distributed and collected by multiple sensors [1]. For example, such data includes air pollutant monitoring data, traffic data collected by road traffic sensors, GPS satellite positioning data, etc. However, due to sensor failure, communication errors, storage loss, and other reasons, data collected by sensors inevitably have missing data. The complete data collected by sensors facilitate subsequent data analysis tasks, such as data classification, regression prediction, and traffic control optimization [2]. However, due to the diversity of missing data patterns, imputing data is very challenging. It is necessary to design appropriate algorithms to extract patterns from multidimensional data, especially for automatic machine learning models of the internal interdependencies of the data.

Traditional data imputation methods often have some limitations. For example, there are many assumptions made about data such as assumptions about linearity [4] or smoothness [5]. Classical statistical time series models such as ARMA and ARIMA [6] can impute missing values, but these models are essentially linear. Matrix completion has been used to impute missing values (e.g., [7]. However, it usually only applies to statistical data and requires a low rank. Lin *et al.* [8] developed a model based on HMI (a hybrid MI system) to impute missing data in an arbitrary pattern, but the repair accuracy relies on feature extraction methods. Moreover, some scholars have expanded the two-dimensional matrix into high-dimensional tensors; Chen *et al.* [9] further incorporated a Bayesian approach into a tensor decomposition model. Models based on recurrent neural networks (RNN) usually assume that the relationships between datasets are sequential. These cannot be processed in parallel, and it is difficult to directly model the interdependence between input data with different timestamps. With the continuous deepening of research in this field, researchers have begun to use deep generation models for data imputation tasks. Recent research has shown that deep learning models perform well in data repair and prediction problems [10]. Benkraouda *et al.* [11] proposed a Convolutional Auto-Encoder to impute missing traffic data. The traffic data imputation problem can be reformed into an image inpainting problem [12]. Zhuang *et al.* [12] also proposed a Convolutional Auto-Encoder that can effectively repair missing traffic speed data with high missing rates, but this model neglects the relationship between missing data and known data in the distribution. Generative Adversarial Networks (GAN) as typical deep generation models, can learn to capture the distribution and potential structure

Manuscript received 6 June 2020; revised 9 November 2020 and 23 January 2021; accepted 17 April 2021. Date of publication 4 May 2021; date of current version 8 July 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1601100 and in part by the National Natural Science Foundation of China under Grant 71971116. The Associate Editor for this article was S. Siri. *(Corresponding author: Xiying Li.)*

Weibin Zhang, Pulin Zhang, and Yinghao Yu are with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: wbin.zhang@outlook.com).

Xiying Li is with the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mail: stslxy@mail.sysu.edu.cn).

Salvatore Antonio Biancardo is with the School of Polytechnic and Basic Sciences, University of Naples Federico II, 80138 Naples, Italy (e-mail: Salvatoreantonio.biancardo@unina.it).

Junyi Zhang is with the Graduate School of Advanced Science and Engineering, Hiroshima University, Higashihiroshima 739-8511, Japan (e-mail: zjy@hiroshima-u.ac.jp).

Digital Object Identifier 10.1109/TITS.2021.3074564

of incomplete and heterogeneous data through an iterative training process, estimate missing values and detect outliers, and can be used to impute missing data and diagnose traffic jams. In addition, methods based on attention mechanisms have gradually attracted the attention of researchers. Note that the core goal of the mechanism is to choose from the excess of information that is available. The Self-Attention structure in the Transformer [13] model can explicitly capture the correlation between two timestamps, removing the limitations of the RNN model sequential processing, greatly speeding up the training time, and significantly improving the seq-to-seq natural language processing (NLP) task performance. The Self-Attention Generation Adversarial Network (SAGAN) [14] combines the attention mechanism with a convolution calculation to provide attention-driven remote dependency modeling for image generation tasks. In SAGAN, information from all feature positions can be used to generate additional features. Inspired by SAGAN's attention structure, a self-attention mechanism-based missing data imputation model for traffic flow was proposed; namely, the self-attention generative adversarial imputation net (SA-GAIN). In SA-GAIN, the goal of the generator is to accurately impute missing data, while the goal of the discriminator is to distinguish the real data from the imputed data. To repair the lost time series data, SA-GAIN adjusts and improves the standard SAGAN architecture, and repairs lost data through the automatic attention automatic encoder, meanwhile, the discriminator is used to judge the authenticity of the input data to force the generator to improve the imputation process. The main contributions of this article are as follows:

1) A traffic data imputation method based on Self-Attention was developed. By considering that there was variance in the interdependence between different sensors, the proposed model used the self-attention mechanism to learn the spatio-temporal correlation of traffic data in a latent way to improve the repair effect.

2) The GAN network structure was introduced to distinguish the imputed traffic data from the real data, allowing the generator to impute data closer to the real distribution.

This paper is organized as follows: the existing studies on traffic data imputation are reviewed in Section II. The methodology is described in Section III. In Section IV, the experiments are conducted, followed by discussion in Section V. This study is concluded in Section VI.

II. Related Works

In this section, we review the existing studies on traffic data imputation in detail. Imputation of time series (ITS) is a key component of lost traffic flow data imputation, and many statistical and machine learning methods have been used to solve this problem.

Statistics-based imputation methods often use the mean [15], the last observation value [16], or the mode [17] to impute missing data. Many previous works have shown that machine learning-based imputation methods are useful for data imputation or time series imputation. Autoregressive Integrated Moving Average models (ARIMA) [6], Auto-Regressive Fractionally Integrated Moving Average models (ARFIMA) [18], and seasonal ARIMA [19] models are used as representatives of the autoregressive algorithm model, and are also used in the imputation of missing time series data. In addition, Multivariate Imputation by Chained Equations (MICE) [20] uses an iterative regression model to impute missing values, and the imputation of each missing value is solved by an independent model. K nearest neighbor (KNN) [21] uses the average value of the K nearest neighbors to the missing values to repair the data. The matrix factorization algorithm [22], [23] decomposes the missing data set into low-rank matrices and uses the product of these matrices to calculate the missing values. Probabilistic principal component analysis (PPCA) [24] retrieves and utilizes the implicit variation to impute missing data.

Many neural network-based methods are also used to estimate missing data. The approaches based on DAE (Deep Denoising Autoencoders) [25] have been shown to work well in practice, which however require complete data during training. An alternative method for DAE [26] allows use of incomplete data sets, however, it only uses the observed components to learn the representation of the data. Algorithms for repairing missing data based on recurrent neural networks usually build models based on the long-term and short-term dependencies of the time series. Berglund *et al.* [27] proposed two probabilistic interpretations of bidirectional RNNs that can be used to reconstruct missing data efficiently.

Gated Recurrent Unit-D (GRU-D) [28] takes two representations of missing patterns and effectively incorporates them into a deep model architecture. It not only captures the long-term temporal dependencies in the time series but also utilizes the missing patterns to improve the prediction results. Bidirectional Recurrent ITS (BRITS) [29] is a novel neural network method that can directly learn missing values in a two-way recursive dynamic system without any specific assumptions. Considering the spatiotemporal correlation of the collected data, a data imputation method based on convolution recurrent autoencoders is used to capture spatial and temporal patterns and estimate missing values in the data [30]. To estimate the missing values for traffic-related time series data, a multi-view learning method was proposed, which combines data-driven algorithms (long-short term memory and support vector regression) and collaborative filtering techniques [31].

Generating model-based missing data imputation methods usually starts with the data distribution and imputes missing data by fitting the true distribution of the data. For example, Dalca *et al.* [32] proposed a variational approximation learning algorithm based on Convolutional Neural Networks (CNN) and sparse perception. The improved HI-VAE (Heterogeneous-Incomplete Variational Autoencoders) algorithm based on the variational auto-encoder proposed in [33] can accurately impute a variety of missing data. Fortuin *et al.* [34] proposed a new deep latent variable model GP-VAE (Gaussian Process Variational Autoencoders), which uses structured variational inference of the data distribution to improve the scalability of dimensionality reduction and data imputation methods. Generative Adversarial Networks (GAN) as a kind of generative model, through the adversarial training of two networks, generate new samples that follow the distribution of the training data set. In [35], a novel method using parallel data and GAN to enhance the imputation of traffic data was proposed. GAIN is a GAN-based method that uses a hint matrix based on the real observed values for unsupervised data generation. In [36], improved Gated Recurrent Unit (GRU) and GAN models were proposed to model the time irregularities of the missing data. E2-GAN [37] is an end-to-end generative model for calculating missing values in multivariate time series. With the help of the discriminative loss and the reconstruction loss functions, E2-GAN can impute data for incomplete time series by using the closest complete time series generated in a single stage.

Models based on attention mechanisms have also recently attracted the attention of scholars. The self-attention module calculates a weight value at a position with a weighted sum of the features at all positions, where the weights, or attention vectors, only involve a small computational cost. According to Zhang *et al.* [14], compared with a convolution operator, a self-attention module can exhibit a better balance between long-range dependencies with computational and statistical efficiency. In [14], a Self-Attention Generative Adversarial Network (SAGAN) was proposed, which allows attention-driven, long-range dependency modeling for image generation tasks. CDSA (Cross-Dimensional Self-Attention) proposed in [38] is a new method of data imputation based on cross-dimensional self-attention and RNN. Lin *et al.* [39] proposed a 3D Convolutional Ambient Generative Adversarial Network to predict traffic flow by using incomplete datasets. The proposed model can learn the underlying distribution of traffic flow from incomplete traffic data and utilize the captured spatio-temporal features of the traffic data for traffic flow prediction.

In view of the importance of ITS and related downstream analyses for traffic data imputation methods, machine learning-based imputation methods (e.g., ARIMA, ARFIMA) always fail to solve the problem of high missing rates and multiple missing modes. A neural network-based model such as DAE (Deep Denoising Autoencoder) [25] has been shown to work well in practice, but it requires complete data during the training stage, which is often unsatisfactory in reality. The question of how to apply cutting-edge deep learning techniques to improve data imputation is worthy of further investigation.

To summarize past studies, there are two categories of methods to extract spatio-temporal relationships between traffic data: one is to explicitly extract features through mathematical methods such as Tang *et al.* [40], used a Fourier transform to extract the period in traffic data. The other is to learn features through models and implicitly extract spatio-temporal correlations from data sets through data-driven methods such as in Zhuang, *et al.* [12], who extracted spatial correlations through a convolutional neural network. In practice, spatio-temporal features of traffic data are generally complicated; it is difficult to extract the features completely using mathematical methods. However, the effects of the subsequent model largely depend on the previous feature extraction methods. Therefore, a data-driven approach is chosen in this paper to capture the spatio-temporal correlations.

III. Methodology

In this section, a new model of generative adversarial networks based on a self-attention autoencoder is proposed to estimate missing traffic data. First, the definition and premise of the problem are given. Second, the basic principles of self-attention mechanisms and GANs are introduced. Third, the construction of the proposed model is introduced. Finally, this model is applied to the imputation of traffic data.

A. Problem Formulation

A d-dimensional data space is defined as $\chi = \{\chi_1, \chi_2, \chi_3, \ldots, \chi_d\}$. Then, a random multidimensional time series is taken from χ as $\mathrm{X} = \{\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3 \ldots, \mathrm{x}_d\}$, where the joint probability distribution of X is $P(X)$. Suppose the mask variable $M = (m_1, m_2, \ldots, m_d)$ takes values from $\{0, 1\}^d$, whose distribution can be expressed as $P(M)$. For $i \in \{1, \ldots, d\}$, a new space with missing values $\{*\}$ can be defined as $\tilde{\chi}_i = \chi_i \cup \{\cdot\}$. Then randomly sampled data variables with missing values from the new variable space can be expressed as $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_d)$, whose distribution can be expressed as $P\left(\tilde{X}\right)$. The relationship between a pair of mask variables M^i and the data variables $\tilde{X}$ can be expressed as Eq. (1):

$$\tilde{x}_i = \begin{cases} x_i, & \text{if } m_i = 1 \\ *, & \text{otherwise} \end{cases} \qquad (1)$$

During the data imputation process, sampling occurs n times from $P\left(\tilde{X}\right)$ to obtain the real data sample set $\{\tilde{X}^i\}_{i=1}^n$, and sampling occurs n times from $P(M)$ to obtain $\{M^i\}_{i=1}^n$. The problem is solved by the model learning conditional probability distribution $P(X^i \big| (\tilde{X}^i, M^i))$.

B. Self-Attention Mechanism

An attention function describes the ternary function Attention (Q, K, V). These three variables are called "query", "key" and "value". The attention function maps a "query" and a set of "key"-"value" pairs to the output, where "query", "key", "value" and the output are all vectors. The output is a weighted sum of "values", where the weight assigned to each value is calculated by the "query" and the corresponding "key".

The feature $x \in \mathbb{R}^{C \times N}$ from the previous layer is transformed into two feature spaces "query" and "key" by two feature mapping functions $Q = W_q x$ and $K = W_k x$ to calculate the attention weight,

$$\beta_{j,i} = \frac{exp\ \left(s_{i,j}\right)}{\sum_{i=1}^{N} exp\ \left(s_{i,j}\right)}, \quad where\ s_{i,j} = Q^T K \qquad (2)$$

Here, $\beta_{j,i}$ indicates the extent to which the model attends to the i^{th} location when synthesizing the i^{th} region. C is the number of channels and N is the number of features in the previously hidden layer. The convolutional self-attention calculation is shown in Fig. 1. The output of the attention

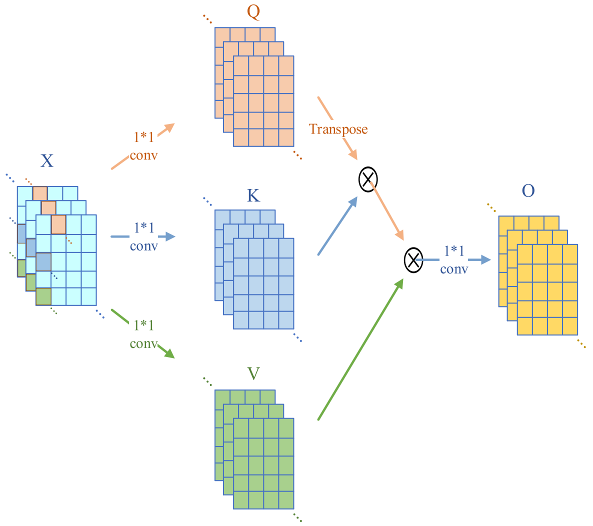


Fig. 1. Convolutional self-attention calculation.

function can be formulated as $o = \left(o_1, \ldots, o_j, \ldots, o_N\right) \in \mathbb{R}^{C\times N}$, where,

$$o_j = t\left(\sum\nolimits_{i=1}^{N}\beta_{j,i}V\right), \quad V = W_v x_i,\ t\left(x_i\right) = W_t x_i \quad (3)$$

In Eq. (3), $W_k \in \mathbb{R}^{\bar{C}\times C}$, $W_q \in \mathbb{R}^{\bar{C}\times C}$, $W_v \in \mathbb{R}^{\bar{C}\times C}$, and $W_t \in \mathbb{R}^{C\times \bar{C}}$ are trainable convolution kernels in 1×1 convolution operations. It was found in [14] that reducing the number of channels after mapping does not result in significant performance loss. In this paper, the scaling factor $k = 8$ was used to scale the number of intermediate channels $\bar{C} = C/k$ for attention calculation.

Note that the output of the layer is multiplied by the trainable scale parameter and then added to the input feature map. Therefore, the final output is

$$y_i = \gamma o_i + x_i \quad (4)$$

where, γ controls the proportion of the result of the attention calculation in the final output, which is initialized to 0. The greater γ, the greater the attention layer's output perception of the global position data.

C. Generative Adversarial Network (GAN)

GANs consists of a generator (G) and a discriminator (D). G learns the mapping G (z), which attempts to map the random noise vector z to the real data. D tries to find a mapping D (.) to discriminate the true probability of the input data. The original GAN generator and discriminator consist of neural networks. The discriminator and generator are alternately trained and competed with each other. The generator is a continuous, differentiable transformation function mapping a prior distribution from the latent space Z into the data space X, and it aims to cheat the discriminator. The discriminator distinguishes its input whether it derives from a real data distribution or not. With this approach, a generator can learn to create solutions that are similar to a real data distribution and are thus difficult to classify by D. This adversarial process gives GAN notable advantages over the other generative models [41].

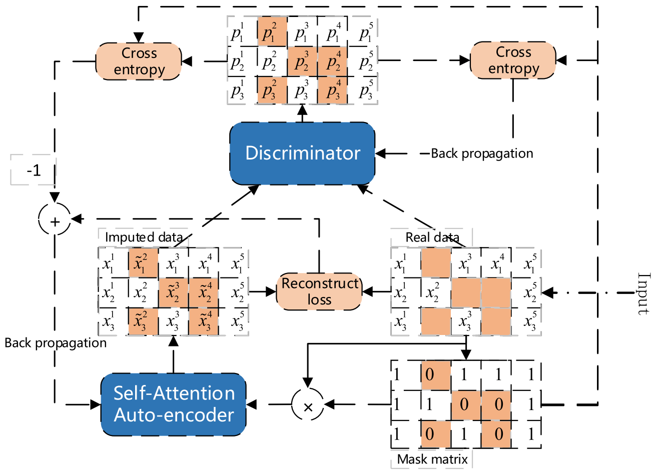


Fig. 2. Self-attention adversarial auto-encoder network structure.

The stopping condition for training is that the discriminator cannot judge whether the data comes from real samples, and the generator cannot generate further fake samples, thereby confusing the discriminator.

The generator G and the discriminator D simultaneously perform minimax training as two competitors with value function $V(D, G)$ [42]. Formally, with $P_z(z)$ as the input prior distribution and $P_{data}(x)$ as the training data distribution, the minimax goal of GAN is defined as

$$\min_G \max_D V(D, G) = E_{x\sim P_{data}(x)} log\left(D\left(x\right)\right) + E_{x\sim P_z(z)} log\left(1 - D\left(G\left(z\right)\right)\right) \quad (5)$$

where x represents the real data, $P_{data}(x)$ represents the distribution of the real data, and z is the input to the generator network, usually taken from the hypothetical prior data distribution $P_z(z)$. $D(x)$ and $D(G(x))$ are the outputs when inputting real data x and generating data $G(x)$ to the discriminator network, respectively.

D. Self-Attention Generative Adversarial Imputation Net (SA-GAIN)

In the original GANs model algorithm, the input of the generator is a random variable. It is worth noting that the input of the missing data imputation problem consists of incomplete samples. With this in mind, in the model building process proposed in this paper, the autoencoder is used as a generator, and the generated samples are constrained to be similar to the corresponding input samples through discriminator loss and reconstruction loss. Here we introduce the proposed SA-GAIN for multi-dimensional missing data imputation. The proposed model is illustrated in Fig. 2.

The difference between GAN and traditional supervised learning is that GAN's discriminator network implicitly learns the diversity similarity measures of the input data in order to distinguish "real data" from "constructed data". Therefore, the proposed model adds the similarity measurement loss learned by the discriminator in the process of training the generator, to help the generator improve the imputation performance. The error learned by the discriminator

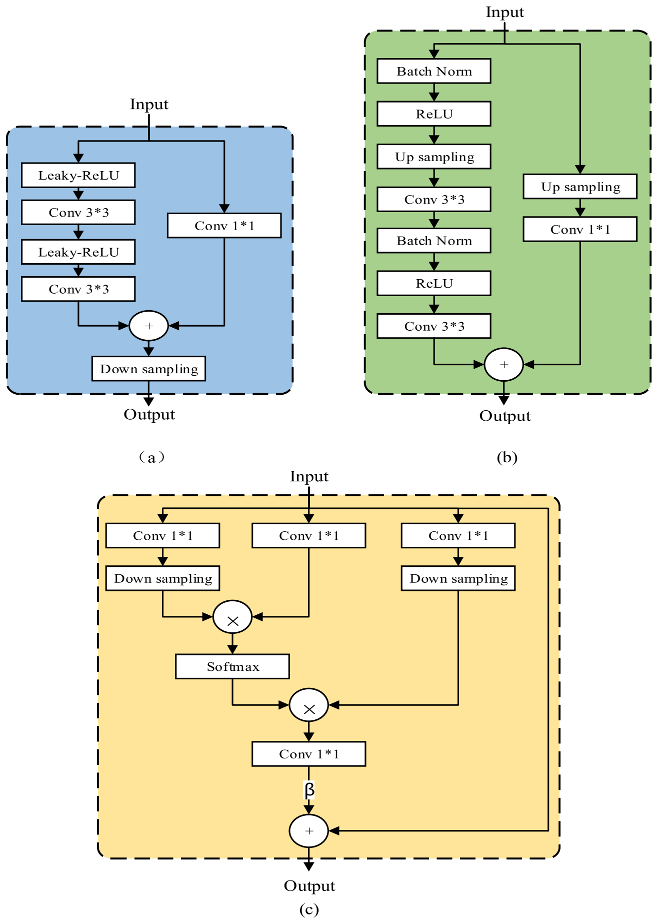


Fig. 3. Neural network block internal computing unit structure. (a) Down residual convolutional block. (b) Up residual convolutional block. (c) Convolutional self-attention block.

is added to the reconstruction loss of the prior hypothesis, and finally, a method combining the advantages of GAN's high-quality generative model and minimizing reconstruction loss is obtained, which can accurately repair missing data, and the distribution of repaired data is more in line with the distribution of real data.

Inspired by SAGAN [14], the position coding and convolutional self-attention mechanisms are added in the model to capture the spatio-temporal correlation in traffic data. With this structure, the model can learn the correlation in data between different loop detectors and timestamps, and this structure improves the imputation accuracy and effect. The convolutional self-attention structure is shown in Fig. 3 (c), and the position coding structure is introduced in Section III.C.

The following is a detailed introduction to each part of the network structure in Fig. 2.

1) Generator: In the proposed framework, the generator is composed of an auto-encoder neural network based on the self-attention mechanism, and its role is to impute the missing data as well as possible. The imputation process of the generator can be formulated in Eq. (6):

$$\hat{x} = m \odot G\left[x, m, (1-m) \odot z\right] + (1-m) \odot x \tag{6}$$

where $\odot$ denotes element-wise multiplication. m indicates which components of x are observed. x denotes incomplete data and $\hat{x}$ represents the imputed data. Eq. (6) describes the process of using the generated data and observation data to obtain the imputed data. In order to improve the generalization ability of the model, we draw on the underlying concept of the DAE (Deep Denoising Autoencoder) [25] model and add noise z to the input. It is worth noting that throughout the generation process, the target conditional distribution, $P\left(\hat{x}|x, m, (1-m) \odot z\right)$, is essentially $\|1-m\|_1$-dimensional. By minimizing the objective function shown in Eq. (7),

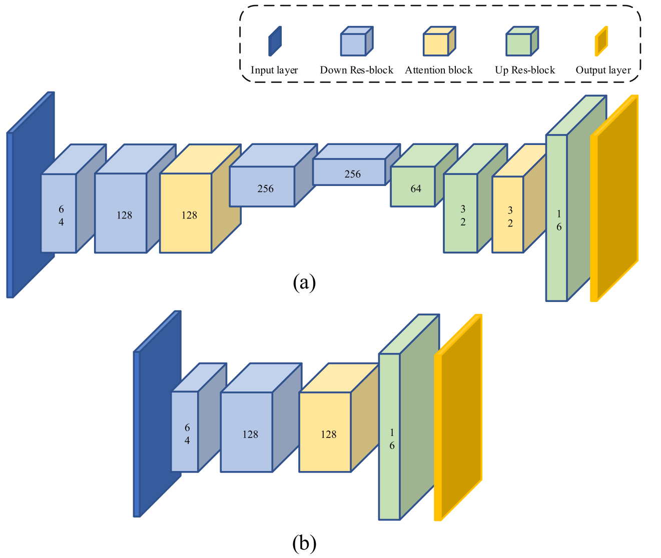


Fig. 4. Generator and discriminator network structure. (a) Self-attention auto-encoder. (b) Self-attention discriminator.

$$V_G(G, \hat{D}) = \mathop{\mathbb{E}}_{x \sim P_r, \hat{x} \sim P_i, m \sim P_m} \left[\left\|\hat{x} - x\right\|^2 - \mathscr{L}_D\left(\hat{x}, m\right)\right] \tag{7}$$

the distribution of the training data is learned by the generator. Here, P_i, P_r, and P_m represent the distribution of the imputed data, the observed distribution of the data, and the distribution of the missing data, respectively. $\left\|\hat{x} - x\right\|^2$ denotes the reconstruction loss, $\mathscr{L}_D\left(\hat{x}, m\right)$ denotes the discriminator loss and can be formulated as

$$\mathscr{L}_D\left(\hat{x}, m\right) = \mathbb{E}\left[m\, log\left(D\left(\hat{x}\right)\right) + (1-m)\, log\left(1 - D\left(\hat{x}\right)\right)\right]. \tag{8}$$

The motivation for applying discriminator loss in the model is to constrain the similarity of the generated data and the real data, and force the generator to learn the observed data distribution. The network structure of the generator is shown in Fig. 4 (a), and the block structure that makes up the network is shown in Fig. 3. The block structure in Fig. 3 is similar to Res-Net's short-cut connection [43] for feature fusion.

2) Discriminator: In the GAN training framework, it is necessary to introduce a discriminator D, which will be used as an opponent to train G. In the standard GAN, the output of the generator is either completely true or completely fake, and in the data imputation problem, the output is the coexisting observed and generated values. The discriminator tries to distinguish which part of the input is real (observed) and which part of the input is generated by the generator (imputed). The network structure of the discriminator is shown in Fig. 4(b), and its output is the same as the input dimension, which characterizes the authenticity of the input data. The greater the generator output value, the greater the probability that the network considers the input at this location to be an observation.

3) Positional Encoding: Since the model does not include the input timestamp information as part of the calculation, and the calculation of the entire process is done in parallel, this may mean that the time-series information in the traffic flow data is not considered. In order for the model to make use of the timestamp and the periodic information contained in the sequence, some information about the relative or absolute position of the markers in the sequence needs to be artificially generated and merged into the input data. A transformer model [13] is a model architecture that eschews recurrence and instead relies entirely on an attention mechanism to draw global dependencies between an input and an output. Inspired by the position coding of the Transformer model [13], when constructing the model input data, the data sequence information is added by adding a position encoding to the input data. The dimensions of the position encoding information are the same as the feature dimensions of the input data at each time point, and the concatenated data and encoded position are used as the input of the model. In the proposed model, the positional encoding method shown in Eq. (9) is used,

$$\begin{aligned} PE_{(t,2i)} &= sin\left(\frac{t}{10000^{\frac{2i}{F}}}\right) \\ PE_{(t,2i+1)} &= cos\left(t/10000^{2i/F}\right) \end{aligned} \tag{9}$$

where t is the timestamp in a day of the input data, i represents the dimensions, and F represents the sampling frequency of the data. In other words, each dimension of the encoded position corresponds to a sinusoidal signal. This function is chosen because it allows the model to easily learn to participate in calculations through relative positions. For any fixed offset k, $PE_{(t+k,i)}$ can be expressed as a function of $PE_{(t,i)}$. In traffic data imputation, different location encodings are applied to different collection timestamps during the day, so the generator can use this information to construct the missing data.

E. Techniques to Stabilize the Training of GANs

The training of GANs is known to be unstable and sensitive to the choices of hyperparameters. Several works that have attempted to stabilize the GAN training process have achieved excellent results. For example, [44]–[46] improved sample diversity by designing a new network architecture. [47]–[50] modified the learning objectives and dynamics to achieve convergence. Some studies have shown that adding regularization methods [51], [52] and introducing heuristic tricks [53], [55] can increase the convergence speed or achieve a better convergence effect. Therefore, the methods of spectral normalization [52] and two time-scale update rules [56] are adopted to improve convergence speed and model stability. The following is an introduction to these two methods.

1) Spectral Normalization: Miyato *et al.* [52] proposed a spectral normalization technique to establish if the trainable weight of each layer in the discriminator network during the entire training process meets the Lipschitz restriction condition. A study [13] found that the use of spectral normalization for both generators and discriminators can stabilize the entire adversarial training process. In view of this, spectral normalization is used in both the generator and discriminator of the proposed model. Eq. (10) and Eq. (11) illustrate how to apply spectral normalization to trainable weights.

$$\bar{W}_{\text{SN}}(W) := W/\sigma(W) \tag{10}$$

$$\sigma(A) := max_{h:h\neq 0}\frac{\|Ah\|_2}{\|h\|_2} = \max_{\|h\|_2\leq 1}\|Ah\|_2 \tag{11}$$

where W represents the trainable weight and $\sigma(A)$ represents the modulus of the weight matrix whose spectral radius is equal to the maximum eigenvalue of the matrix.

2) Two Time-Scale Update Rule: In the model training process, it is often inappropriate for the discriminator and the generator to use the same learning rate for parameter optimization. During the training process, Heusel *et al.* [56] proposed that model training with separate learning rates (TTUR) for the generator and discriminator can improve the training efficiency of the network and obtain better model performance in the same iteration rounds. In the model training part of this paper, the Adam optimizer [57] is used for both the self-attention auto-encoder network and the discriminator network. The parameter setting of the discriminator training process is $\beta_1 = 0$, $\beta_2 = 0.9$, the learning rate is 0.0004, the parameter setting of the generator training process is $\beta_1 = 0$, $\beta_2 = 0.9$, and the learning rate is 0.0001.

Algorithm 3.1 is the pseudo-code for the training process of the algorithm proposed in this paper. In the training phase of the model, the Adam gradient descent algorithm is used for parameter optimization and learning rate adjustment; all weights are uniformly initialized using Xavier; the input data batch size (mini-batch) is set to 32; the input data are normalized to [0, 1]. Spectral Normalization (SN) is imposed on the trainable weights in the generator and discriminator. 30 epochs were used as the stopping condition for model training.

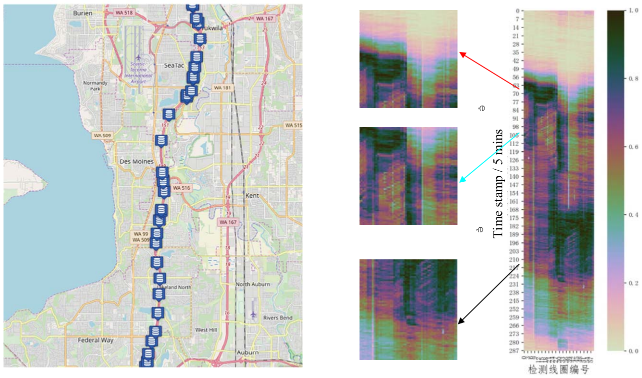
IV. Experiments

In this section, the proposed method was evaluated on real-world datasets. The experimental results are analyzed and compared with other baseline models in detail.

A. Dataset

The data set used in the experimental part of this paper was collected from the induction ring detector deployed by the Washington State Department of Transportation (WSDOT) on the road surface of the I-5 highway in Seattle, USA. Multiple detectors laid on the road are connected to deploy detector stations every half-mile. The data collected at each station are grouped according to the direction and aggregated into traffic data based on the station. This aggregated data and quality control data sets contain traffic speed, volume, and

```
Algorithm 3.1 SA-GAIN Training Process
Input: $P_r$, observation data distribution; $m$, batch size; $G$, generator; $D$, discriminator; $\theta_D$, trainable parameters in $D$: $\theta_G$, trainable parameters in $G$
Output: $G$, self-attention autoencoder. $D$, discriminator.
while training loss has not converged do
    $\{x_r^j\}_{j=1}^m \sim P_r$, Draw $m$ i.i.d samples from $P_r$
    $\{z^j\}_{j=1}^m \sim P_r$, Draw $z$ i.i.d samples from $P_z$
    $\{m^j\}_{j=1}^m \sim \{0,1\}^d$, Draw $m$ i.i.d samples from $\{0,1\}^d$
    for $j = 1, 2, \ldots, m$ do
        $\tilde{x}^j \leftarrow m^j \odot x_r^j$
        $\bar{x}_m^j \leftarrow G(\tilde{x}^j, m^j, (1 - m^j) \odot z^j)$
        $\hat{x}^j \leftarrow m^j \odot \bar{x}_m^j + (1 - m^j) \odot \tilde{x}^j$
        $\mathcal{L}_D(\hat{x}, m) \leftarrow \mathbb{E}[m^j \log(D(\hat{x}^j)) + (1 - m^j)\log(1 - D(\hat{x}^j))]$
        $\mathcal{L}_G \leftarrow \|\hat{x} - x\|^2$
        $\theta_D \overset{+}{\leftarrow} -\nabla_{\theta_D}(\mathcal{L}_D)$, Update $D$
        $\theta_G \overset{+}{\leftarrow} -\nabla_{\theta_G}(\mathcal{L}_G - \lambda\mathcal{L}_{llike}^{Dis_l})$, Update $G$
    end for
end while
```



(a) (b)

Fig. 5. Data source and training sample construction process. (a) The geographical location of the data acquisition equipment. (b) Training sample construction process.

occupancy information. Only traffic volume data are used in this study. The traffic volume data in the experiment were collected from 80 selected detector stations across all days from November 1, 2015 to December 31, 2016, totaling 427 days. The data acquisition time step is 5 minutes, and 288 samples are generated every day. Fig. 5 shows the data collection position and training data structure.

Fig. 5 (b) shows the data construction process, which generates the experimental training data set and test data set by sliding window selection of time bins. In this way, the input size of each sample is set to (80, 80), which represents 80 traffic flow records detected by 80 detector stations at five-minute intervals. 80% of the dataset was randomly selected as training data, with the remaining 20% of the data used as test data.

In the experiment, three traffic flow loss scenarios were designed based on standard characteristics of traffic flow data loss, and it is assumed that the data loss process is random. The three missing data patterns are shown in Fig. 6.

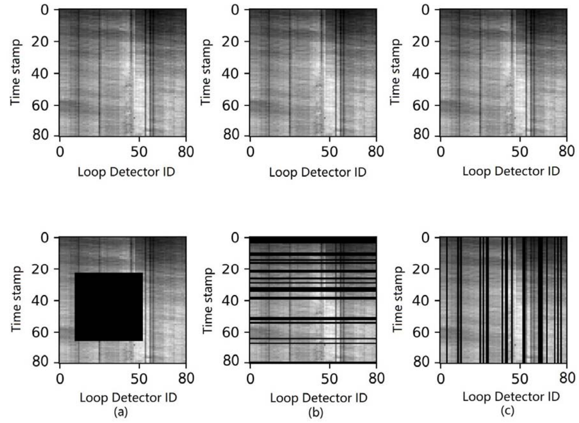


Fig. 6. Three patterns of missing data. (a) Missing data on consecutive road sections in consecutive time periods. (b) Some road sections have missing data over consecutive time periods. (c) Missing data on consecutive sections at some times.

Fig. 6(a) represents the continuous loss of data from the continuous detection coils which may be caused by the loss of data from the roadside control processing unit transmitted to the monitoring sub-center. Fig. 6 (b) shows the scenario where some of the coil data is not collected, probably due to coil damage. Fig. 6 (c) represents the scenario where the data was not collected at some moments, which may be caused by communication or system failure.

B. Baseline Models

This section presents detailed deployment information for the four comparison baseline models. The neural network model in the experimental part was built with the Keras neural network computing library, using the TensorFlow scientific computing package as the backend; the StatsModels scientific computing library was used for data processing operations such as data set division and normalization. The fancyimpute package was used to build a KNN data imputation model. Detailed descriptions of the models are as follows:

1) Historical Average, HA: The sliding historical average model uses the time average of historical traffic flow as the fill value. The model uses the average value of historical traffic flow in the past week to represent the change in traffic flow. Through the model, a traffic pattern table that changes dynamically with time can be established. When data is missing, the data can be retrieved from the corresponding traffic pattern table for imputation.

2) K-Nearest Neighbor, KNN: For each sampling record with missing values, first the missing features are identified, then for each missing feature, the k nearest neighbors of that feature are found, that is, k sample records, and finally the interpolation function is used to impute the missing values. In our experiment, K was set to 3, and the model was implemented with the fancyimpute package in python.

3) Generative Adversarial Imputation Nets, GAIN: GAIN [58] is a GAN-based unsupervised data imputation method that uses a hint vector to impute missing values. GAIN

TABLE I
IMPUTATION PERFORMANCE IN TERMS OF RMSE MAE AND MMD IN MISSING DATA SCENARIO 1

Model	Index	Missing Rate（%）							
		10	20	30	40	50	60	70	80
HA	MAE	7.38	15.16	23.55	33.37	41.73	52.53	59.48	72.56
	MMD	0.03	0.06	0.09	0.10	0.12	0.15	0.18	0.18
	RMSE	32.06	47.08	58.87	71.72	80.20	92.55	97.36	110.08
KNN	MAE	3.55	8.13	13.79	19.95	25.97	32.45	38.61	48.48
	MMD	0.01	0.03	0.04	0.06	0.07	0.08	0.09	0.10
	RMSE	16.43	26.99	36.77	46.09	53.02	60.10	66.52	78.90
GAIN	MAE	3.201	8.739	20.971	32.394	31.868	46.635	55.975	81.081
	MMD	0.029	0.097	0.167	0.261	0.292	0.333	0.342	0.460
	RMSE	15.836	33.704	41.597	55.701	52.679	85.185	105.693	132.033
SAGAN	MAE	2.668	8.797	10.879	19.631	22.026	28.753	36.330	43.121
	MMD	0.033	0.102	0.124	0.226	0.234	0.275	0.352	0.376
	RMSE	12.697	29.176	28.769	45.424	42.213	49.868	56.794	61.508
SA-GAIN	MAE	**2.75**	**5.78**	**8.04**	**12.48**	**17.50**	**22.36**	**32.82**	**35.74**
	MMD	**0.0442**	**0.1093**	**0.2991**	**0.2074**	**0.2592**	**0.2734**	**0.3635**	**0.4162**
	RMSE	**12.52**	**18.41**	**21.39**	**27.32**	**34.44**	**40.93**	**59.04**	**54.24**

TABLE II
IMPUTATION PERFORMANCE IN TERMS OF RMSE MAE AND MMD IN MISSING DATA SCENARIO 2

Model	Index	Missing Rate（%）							
		10	20	30	40	50	60	70	80
HA	MAE	6.40	12.05	16.44	22.12	25.89	30.19	33.91	36.40
	MMD	0.03	0.04	0.06	0.08	0.08	0.10	0.12	0.13
	RMSE	28.01	38.03	43.45	51.77	55.48	60.46	63.93	65.47
KNN	MAE	5.94	7.82	14.61	17.80	18.66	23.26	26.97	28.95
	MMD	0.07	0.10	0.17	0.20	0.21	0.23	0.27	0.30
	RMSE	32.06	28.24	44.96	45.62	42.61	49.52	53.75	54.82
GAIN	MAE	2.513	4.394	5.232	6.310	7.846	9.691	10.610	10.723
	MMD	0.019	0.044	0.058	0.071	0.090	0.113	0.118	0.120
	RMSE	11.389	14.398	14.683	15.786	17.607	20.714	21.144	20.986
SAGAN	MAE	2.915	3.949	11.66	12.26	12.33	12.95	13.45	13.88
	MMD	0.041	0.065	0.099	0.138	0.155	0.165	0.183	0.196
	RMSE	11.220	14.602	15.631	17.609	20.945	23.002	23.535	26.551
SA-GAIN	MAE	**2.05**	**3.12**	**4.67**	**5.86**	**6.74**	**8.72**	**8.81**	**10.55**
	MMD	**0.0126**	**0.0258**	**0.0500**	**0.0668**	**0.0779**	**0.0989**	**0.1069**	**0.1189**
	RMSE	**9.15**	**10.07**	**12.97**	**14.13**	**14.87**	**16.82**	**17.36**	**19.62**

used MSE as a loss function for numerical variables and cross-entropy for binary variables. The network specifications and hyperparameters are reported in [58].

4) Self-Attention Generative Adversarial Network, SAGAN: SAGAN [14] introduces a self-attention mechanism into the convolutional GAN, which can generate high-quality images and impute missing data.

C. Results

Based on the experimental data generated by the above models, this section analyzes the generated data error for each model. Table I, Table II, and Table III show the error results of the imputation for the missing traffic data using the proposed method compared to the baseline methods. These show the imputation performance of the different imputation algorithms for the different missing data scenarios. Loss rate represents the proportion of lost data from the total data. By using three evaluation indicators, MAE (Mean absolute error), MMD (Maximum Mean Discrepancy) and RMSE (Root mean squared error), the imputation pros and cons of different models are compared. The MMD is a nonparametric method for measuring the distance between two distributions [59]–[61], and can evaluate the data distribution characteristics. We use this method to measure the differences in distributions between the repaired data and the real data. It can be seen from the simulated data in all cases, the SA-GAIN method can reach the best imputation accuracy.

In order to conduct further analysis of the experimental results and obtain a more intuitive performance comparison, a visual display of data missing scenario 3 is shown in Fig. 7.

Based on the aforementioned results, some conclusions can be drawn from Fig. 7, Table I, Table II, and Table III. Firstly, when the missing rate increases, the repair effect of all models in the experiment decreases. Compared with the missing

TABLE III

IMPUTATION PERFORMANCE IN TERMS OF RMSE MAE AND MMD IN MISSING DATA SCENARIO 3

Model	Index	Missing Rate（%）							
		10	20	30	40	50	60	70	80
HA	MAE	4.65	8.86	12.68	16.06	19.02	27.42	27.48	27.45
	MMD	0.02	0.03	0.05	0.07	0.08	0.08	0.09	0.11
	RMSE	21.04	29.21	34.78	38.96	42.45	45.61	48.88	50.78
KNN	MAE	4.74	8.29	12.02	15.56	18.62	24.27	26.64	33.22
	MMD	0.05	0.10	0.15	0.18	0.21	0.25	0.27	0.31
	RMSE	24.09	28.65	35.03	40.41	43.40	52.18	52.85	62.96
GAIN	MAE	1.816	5.236	7.304	9.210	11.031	13.473	15.323	18.608
	MMD	0.010	0.058	0.084	0.108	0.125	0.143	0.160	0.184
	RMSE	9.212	26.701	30.593	36.650	34.683	38.162	33.292	53.864
SAGAN	MAE	2.566	4.636	6.076	8.860	11.365	12.876	12.040	15.907
	MMD	0.028	0.066	0.103	0.138	0.150	0.174	0.177	0.192
	RMSE	12.609	15.980	20.363	25.303	26.540	30.106	25.441	30.642
SA-GAIN	MAE	**1.76**	**3.05**	**4.62**	**6.45**	**6.95**	**9.89**	**10.28**	**11.36**
	MMD	**0.0099**	**0.0254**	**0.0496**	**0.0733**	**0.0822**	**0.1178**	**0.1196**	**0.1261**
	RMSE	**8.87**	**11.00**	**13.76**	**17.44**	**17.74**	**28.16**	**24.74**	**26.77**

scenario 2 and missing scenario 3, the result in the missing scenario 1 shows relatively poor repair effects for various models in the same missing rate. This may be the reason why the missing positions are concentrated and why there are few data around the missing parts, which increases the difficulty of data repair. Secondly, when designing an algorithm, the lost data scenario should be considered. For example, in the missing scenario 1 with a high missing rate (70%, 80%), the repair model based on KNN, HA, and GAIN is less effective than the model based on SAGAN and SA-GAIN. In addition, a single evaluation index cannot be used to evaluate the repair effect of a model. For example, HA has low MMD indicators under various missing conditions, but this does not mean that it achieves good repair results because the values of MAE and RMSE are still very high. This may be caused by the defects in the MMD measurement method. From the final results, compared with HA, KNN, and GAIN, the model with the self-attention mechanism (SAGAN, SA-GAIN) has higher imputation quality. Compared with SAGAN, SA-GAIN introduces the structure of the Auto-Encoder and the hint matrix, which can improve the quality of model repair and is suitable for training scenarios with missing data. In all three traffic volume data missing scenarios, SA-GAIN achieves the best interpolation performance.

V. DISCUSSION

In the above chapters, after defining the missing data imputation problem and the data imputation method proposed in this article, we first gave a detailed explanation of the implementation of the proposed model and supplied the corresponding training pseudocode. Subsequently, in the comparison of the imputation performance with the other imputation models, the following findings can be summarized:

1) The proposed method can repair data with different rates of missing data under three different missing data scenarios.

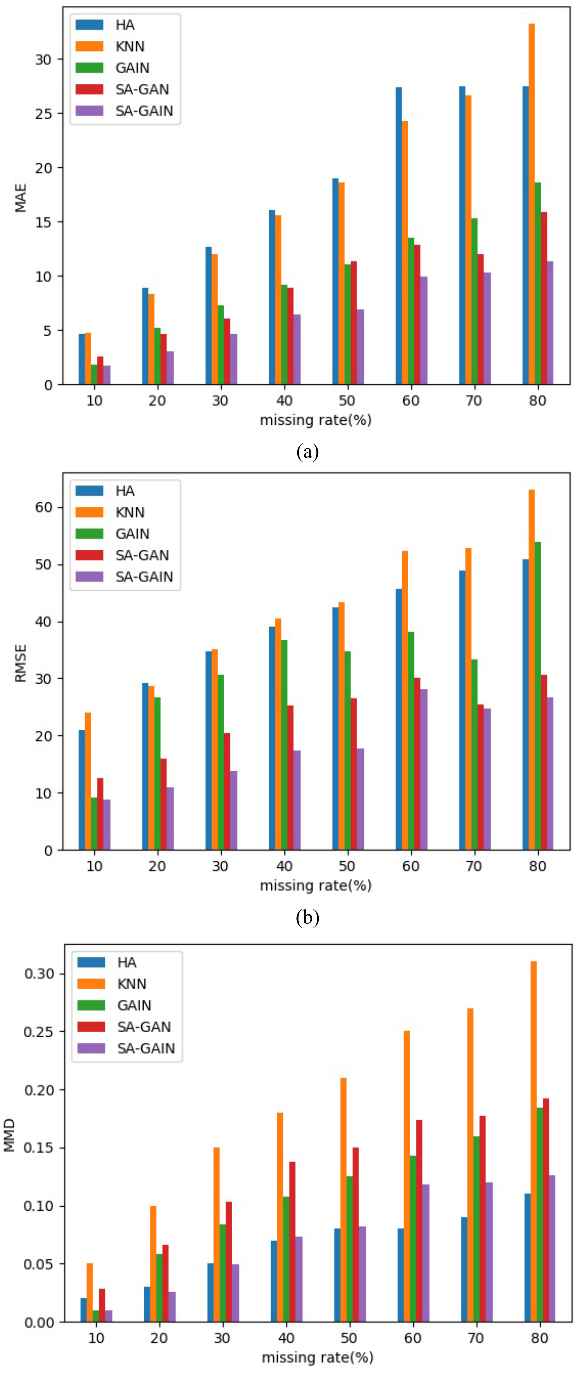
2) Compared with several similar algorithms, the proposed method obtains the lowest imputation error.

3) The training process of the proposed method is stable and the model convergence speed is fast.

We propose the SA-GAIN algorithm, which has higher prediction accuracy than the other algorithms because of the following reasons:

1) The method proposed in this article is a data-driven method for repairing missing data. By constructing a neural network model for self-learning of data features and repairing missing values in the data, the model can adaptively learn and optimize from massive data, and can, in practice, adapt to diverse missing data scenarios. In addition, the introduction of unsupervised learning methods means that the model is able to adapt to changes in the missing data pattern during the learning process.

2) In the past, the imputation of missing traffic data was usually based on statistics and traditional machine learning methods. It was difficult to make full use of the spatio-temporal features of historical traffic data so that the patterns in the data could not be fully mined. This paper proposes a method for modeling the interdependence between data through the convolutional attention mechanism, which can model the spatial geographic distribution of data with different sampling times. The self-attention module cooperates with the convolution operation method and helps to model the dependencies of long-distance, long-term, and cross-regional data. With the help of "self-attention", the generator can improve the effectiveness of data imputation by learning the characteristics of space-time evolution, so that the generated missing data can be coordinated with the observed real data. In addition, during the GAN's adversarial training process, the generator not only guides the data imputation process through fixed loss evaluation indicators, but also applies reconstruction errors to the non-missing parts, and applies discriminator losses to the missing parts to improve imputation quality. From the perspective of data distribution fitting, the distribution of traffic flow data is only a manifold in the high-dimensional data space. It is difficult to find a loss function to describe the



(a)

(b)

(c)

Fig. 7. Model error analysis: (a) MAE of compared models. (b) RMSE of compared models. (c) MMD of compared models.

distance of the generated data distribution to this manifold, thus it is difficult to guide the generator to generate data close to the true distribution. However, our model discriminates through the discriminator, allowing the neural network to learn an error metric, thereby improving the quality of data generation.

3) The GAN training process often has mode collapse. In the method proposed in this paper, we use a variety of methods to improve training efficiency, through spectral normalization (SN), the two-scale gradient descent method, residual connection of features, batch normalization in the generator and other methods to improve training stability. In addition, the introduction of position coding information adds guidance conditions for data repair, which can provide time and relative position information for data generation.

In addition to the above, this article has some limitations. The algorithms proposed in this study are suitable for mesh or sequence data formats; however, the real road geographic topology data often does not form a grid data format, instead, it tends to be more of a graph structure. The design of practical traffic data attention perception algorithms for traffic flow, attention learning for traffic data distribution with spatially irregular graph structures, and considerations of distance and other road-related information for filling in and predicting missing traffic volume data are worth further exploration.

VI. Conclusion

This paper proposes a new method for imputing missing traffic flow data by using a generative adversarial network based on attention mechanisms to estimate missing values in multivariate time series. The generator uses discriminator loss and reconstruction loss to generate a complete sample that is closest to the distribution of the training sample data. The discriminator makes true and false judgments on each component of the repaired sample to improve the data generation quality in the confrontation training. In addition, the introduction of attention mechanisms captures the correlations between spatially distributed sensors at different time points, thereby effectively improving the imputation of traffic flow data. Experiments on real data sets show that the proposed method reduces the error of the imputed data compared to baseline algorithms.

References

[1] M. T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet, "Matrix and tensor based methods for missing data estimation in large traffic networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1816–1825, Jul. 2016.

[2] S. Tak, S. Woo, and H. Yeo, "Data-driven imputation method for traffic data in sectional units of road links," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1762–1771, Jun. 2016.

[3] W. Zhang, Y. Feng, K. Lu, Y. Song, and Y. Wang, "Speed prediction based on a traffic factor state network model," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 16, 2020, doi: 10.1109/TITS.2020.2979924.

[4] C. F. Ansley and R. Kohn, "On the estimation of arima models with missing values," in *Time Series Analysis of Irregularly Observed Data*, E. Parzen, Ed. New York, NY, USA: Springer, 1984, pp. 9–37.

[5] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, vol. 1. New York, NY, USA: Springer, 2001.

[6] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA: Wiley, 2015.

[7] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high dimensional time series prediction," in *Proc. Adv. neural Inf. Process. Syst.*, 2016, pp. 847–855.

[8] J. Lin, N. Li, M. A. Alam, and Y. Ma, "Data-driven missing data imputation in cluster monitoring system based on deep neural network," *Int. J. Speech Technol.*, vol. 50, no. 3, pp. 860–877, Mar. 2020.

[9] X. Chen, Z. He, Y. Chen, Y. Lu, and J. Wang, "Missing traffic data imputation and pattern discovery with a Bayesian augmented tensor factorization model," *Transp. Res. C, Emerg. Technol.*, vol. 104, pp. 66–77, Jul. 2019.

[10] J. Zhao, Y. Nie, S. Ni, and X. Sun, "Traffic data imputation and prediction: An efficient realization of deep learning," *IEEE Access*, vol. 8, pp. 46713–46722, 2020.

[11] O. Benkraouda, B. T. Thodi, H. Yeo, M. Menendez, and S. E. Jabari, "Traffic data imputation using deep convolutional neural networks," *IEEE Access*, vol. 8, pp. 104740–104752, 2020.

[12] Y. Zhuang, R. Ke, and Y. Wang, "Innovative method for traffic data imputation based on convolutional neural network," *IET Intell. Transp. Syst.*, vol. 13, no. 4, pp. 605–613, Apr. 2019.

[13] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[14] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018, *arXiv:1805.08318*. [Online]. Available: http://arxiv.org/abs/1805.08318

[15] M. Kantardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*. Hoboken, NJ, USA: Wiley, 2011.

[16] M. Amiri and R. Jensen, "Missing data imputation using fuzzy-rough methods," *Neurocomputing*, vol. 205, pp. 152–164, Sep. 2016.

[17] A. Purwar and S. K. Singh, "Hybrid prediction model with missing value imputation for medical data," *Expert Syst. Appl.*, vol. 42, no. 13, pp. 5621–5631, Aug. 2015.

[18] J. W. Galbraith and V. Zinde-Walsh, "Autoregression-based estimators for ARFIMA models," CIRANO Work. Papers 2001s-11, CIRANO, 2001.

[19] C. Hamzaçebi, "Improving artificial neural networks' performance in seasonal time series forecasting," *Inf. Sci.*, vol. 178, no. 23, pp. 4550–4559, Dec. 2008.

[20] I. R. White, P. Royston, and A. M. Wood, "Multiple imputation using chained equations: Issues and guidance for practice," *Statist. Med.*, vol. 30, no. 4, pp. 377–399, Feb. 2011.

[21] A. T. Hudak, N. L. Crookston, J. S. Evans, D. E. Hall, and M. J. Falkowski, "Nearest neighbor imputation of species-level, plot-scale forest structure attributes from LiDAR data," *Remote Sens. Environ.*, vol. 112, no. 5, pp. 2232–2245, May 2008.

[22] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometric Intell. Lab. Syst.*, vol. 106, no. 1, pp. 41–56, Mar. 2011.

[23] Y. Wang, Y. Zhang, X. Piao, H. Liu, and K. Zhang, "Traffic data reconstruction via adaptive spatial-temporal correlations," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1531–1543, Apr. 2019.

[24] L. Qu, J. Hu, L. Li, and Y. Zhang, "PPCA-based missing data imputation for traffic flow volume: A systematical approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 512–522, Sep. 2009.

[25] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, Jul. 2008, pp. 1096–1103.

[26] L. Gondara and K. Wang, "Multiple imputation using deep denoising autoencoders," 2017, *arXiv:1705.02737*. [Online]. Available: https://arxiv.org/abs/1705.02737

[27] M. Berglund, T. Raiko, M. Honkala, L. Kärkkäinen, A. Vetek, and J. Karhunen, "Bidirectional recurrent neural networks as generative models," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, vol. 1. Montreal, QC, Canada, Dec. 2015, pp. 856–864.

[28] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Sci. Rep.*, vol. 8, no. 1, pp. 1–12, Apr. 2018.

[29] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "BRITS: Bidirectional recurrent imputation for time series," in *Proc. Adv. Neural Inf. Process. Syst.*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett, and Eds. Red Hook, NY, USA: Curran Associates, 2018, pp. 6775–6785.

[30] R. Asadi and A. Regan, "A convolution recurrent autoencoder for spatio-temporal missing data imputation," 2019, *arXiv:1904.12413*. [Online]. Available: http://arxiv.org/abs/1904.12413

[31] L. Li, J. Zhang, Y. Wang, and B. Ran, "Missing value imputation for traffic-related time series data based on a multi-view learning method," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 2933–2943, Aug. 2019.

[32] A. V. Dalca, J. Guttag, and M. R. Sabuncu, "Unsupervised data imputation via variational inference of deep subspaces," 2019, *arXiv:1903.03503*. [Online]. Available: http://arxiv.org/abs/1903.03503

[33] A. Nazabal, P. M. Olmos, Z. Ghahramani, and I. Valera, "Handling incomplete heterogeneous data using VAEs," 2018, *arXiv:1807.03653*. [Online]. Available: http://arxiv.org/abs/1807.03653

[34] V. Fortuin, G. Rätsch, and S. Mandt. (2019). *Multivariate Time Series Imputation With Variational Autoencoders*. Accessed: Mar. 24, 2020. [Online]. Available: /paper/Multivariate-Time-Series-Imputation-with-Fortuin-R%C3%A4tsch/bf0cdea090ff25a3f8bafe5a46622de99c55cbc3

[35] Y. Chen, Y. Lv, and F.-Y. Wang, "Traffic flow imputation using parallel data and generative adversarial networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1624–1630, Apr. 2020.

[36] Y. Luo, X. Cai, Y. Zhang, J. Xu, and Y. Xiaojie, "Multivariate time series imputation with generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2018, pp. 1596–1607.

[37] Y. Luo, Y. Zhang, X. Cai, and X. Yuan, "E^{2}GAN: End-to-end generative adversarial network for multivariate time series imputation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3094–3100.

[38] J. Ma, Z. Shou, A. Zareian, H. Mansour, A. Vetro, and S. Chang, "CDSA: Cross-dimensional self-attention for multivariate, geo-tagged time series imputation," *CoRR*, vol. abs/1905.09904, 2019. [Online]. Available: http://arxiv.org/abs/1905.09904

[39] F. Lin, H. Zheng, and X. Feng, "An attention-based ambient network with 3D convolutional network for incomplete traffic flow prediction," in *Proc. ACM Turing Celebration Conf. China*, May 2019, pp. 1–5.

[40] J. Tang, F. Liu, Y. Zou, W. Zhang, and Y. Wang, "An improved fuzzy neural network for traffic speed prediction considering periodic characteristic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2340–2350, Sep. 2017, doi: 10.1109/TITS.2016.2643005.

[41] Y. Hong, U. Hwang, J. Yoo, and S. Yoon, "How generative adversarial networks and their variants work: An overview," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–43, Feb. 2019.

[42] I. J. Goodfellow *et al.*, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, vol. 3, pp. 2672–2680.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[44] J. Li, J. Jia, and D. Xu, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. 4th Int. Conf. Learn. Represent., (ICLR)*, 2016, pp. 97–108.

[45] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4396–4405.

[46] H. Zhang *et al.*, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5908–5916.

[47] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn., (ICML)*, vol. 1, 2017, pp. 298–321.

[48] T. Salimans, H. Zhang, A. Radford, and D. Metaxas, "Improving GANs using optimal transport," in *Proc. Int. Conf. Learn. Represent.*, 2018. [Online]. Available: https://openreview.net/forum?id=rkQkBnJAb

[49] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," Nov. 2016, *arXiv:1611.02163*. [Online]. Available: https://arxiv.org/abs/1611.02163

[50] A. Jolicoeur-Martineau, "The relativistic discriminator: A key element missing from standard GaN," in *Proc. 7th Int. Conf. Learn. Represent., (ICLR)*, Montreal, QC, Canada: Université de Montréal, 2019, pp. 1–25.

[51] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 5768–5778.

[52] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," presented at the Int. Conf. Learn. Represent., Feb. 2018, Accessed: Mar. 24, 2020.

[53] S. Azadi, C. Olsson, T. Darrell, I. Goodfellow, and A. Odena, "Discriminator rejection sampling," 2018, *arXiv:1810.06758*. [Online]. Available: http://arxiv.org/abs/1810.06758

[54] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. NIPS*, 2016, pp. 1–10.

[55] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. ICML*, 2017, pp. 2642–2651.

[56] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, I. Guyon *et al.*, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 6626–6637.

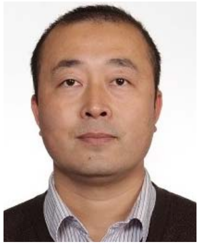
[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980 and https://dare.uva.nl/search?identifier=a20791d3-1aff-464a-8544-268383c33a75

[58] J. Yoon, J. Jordon, and M. Schaar, "GAIN: Missing data imputation using generative adversarial nets," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2018, pp. 5689–5698, Accessed: Mar. 24, 2020.

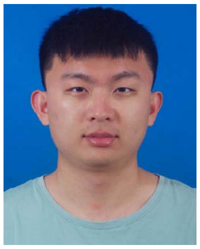
[59] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 513–520.

[60] Y. Li, K. Swersky, and R. S. Zemel, "Generative moment matching networks," in *Proc. ICML*, 2015, pp. 1718–1727.

[61] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.



Weibin Zhang received the Ph.D. degree in automation from Xi'an Jiaotong University, China, in 2008. From 2014 to 2017, he worked as a Research Associate with the Department of Civil and Environmental Engineering, University of Washington, Seattle, WA, USA. He is currently a Professor with the Nanjing University of Science and Technology. His research fields include intelligent transportation systems, connected vehicle, big data in transportation modeling, machine learning techniques, and marine safety.



Pulin Zhang received the B.E. degree in 2019. He is currently pursuing the master's degree with the Nanjing University of Science and Technology. His research interests cover traffic data imputation and intelligent transportation systems.

Yinghao Yu received the master's degree from the Nanjing University of Science and Technology in 2020. His research interests cover traffic data imputation, traffic prediction, and intelligent transportation systems.

Xiying Li (Member, IEEE) received the Ph.D. degree in optical engineering from the Beijing Institute of Technology in 2002. She is currently an Associate professor with the School of Intelligent Systems Engineering, Sun Yat-sen University. Her research interests include intelligent transportation systems, traffic information collection, traffic video, and image big data processing and application.

Salvatore Antonio Biancardo received the Ph.D. degree in civil engineering from the University of Naples Federico II, Italy, in 2016, where he obtained the title Doctor Europaeus in civil engineering. He is currently with the University of Naples Federico II, as an Assistant Professor. His research interests are BIM for infrastructures, road pavement materials, and transportation safety.

Junyi Zhang received the Ph.D. degree in engineering from Hiroshima University, Japan, in 1996. His research fields include urban and regional planning, transportation planning, traffic engineering, environment and energy policies, tourism policy, health policy, human behavior analysis, and systematic approaches.