

Article

Traffic-Data Recovery Using Geometric-Algebra-Based Generative Adversarial Network

Di Zang ^{1,*}, Yongjie Ding ¹, Xiaoke Qu ¹, Chenglin Miao ¹ , Xihao Chen ¹, Junqi Zhang ¹ and Keshuang Tang ^{2,*}

¹ Department of Computer Science and Technology, Tongji University, Shanghai 200092, China; dinyongjie2000@163.com (Y.D.); quxiaoke@tongji.edu.cn (X.Q.); 1851804@tongji.edu.cn (C.M.); 2030803@tongji.edu.cn (X.C.); zhangjunqi@tongji.edu.cn (J.Z.)

² Department of Transportation Information and Control Engineering, Tongji University, Shanghai 200092, China

* Correspondence: zangdi@tongji.edu.cn (D.Z.); tang@tongji.edu.cn (K.T.)

Abstract: Traffic-data recovery plays an important role in traffic prediction, congestion judgment, road network planning and other fields. Complete and accurate traffic data help to find the laws contained in the data more efficiently and effectively. However, existing methods still have problems to cope with the case when large amounts of traffic data are missed. As a generalization of vector algebra, geometric algebra has more powerful representation and processing capability for high-dimensional data. In this article, we are thus inspired to propose the geometric-algebra-based generative adversarial network to repair the missing traffic data by learning the correlation of multidimensional traffic parameters. The generator of the proposed model consists of a geometric algebra convolution module, an attention module and a deconvolution module. Global and local data mean squared errors are simultaneously applied to form the loss function of the generator. The discriminator is composed of a multichannel convolutional neural network which can continuously optimize the adversarial training process. Real traffic data from two elevated highways are used for experimental verification. Experimental results demonstrate that our method can effectively repair missing traffic data in a robust way and has better performance when compared with the state-of-the-art methods.

Keywords: traffic data recovery; geometric algebra; deep learning; intelligent transportation system



Citation: Zang, D.; Ding, Y.; Qu, X.; Miao, C.; Chen, X.; Zhang, J.; Tang, K. Traffic-Data Recovery Using Geometric-Algebra-Based Generative Adversarial Network. *Sensors* **2022**, *22*, 2744. <https://doi.org/10.3390/s22072744>

Academic Editor: Francisco J. Martinez

Received: 7 March 2022

Accepted: 31 March 2022

Published: 2 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Traffic data are of great significance to intelligent transportation systems (ITS), which provide useful information for traffic flow prediction, congestion judgment and urban transportation network planning. Accurate traffic data can make the analysis results more reliable. In a large-scale traffic-flow-monitoring system, sensors deployed in different locations can collect a large amount of useful time series data. However, due to the influence of the hardware device itself, the sensors often fail to work, resulting in incomplete data collection [1]. At the same time, accidents that occur during the storage of a large amount of traffic data will also cause the lack of these traffic data. In order to repair the missing traffic data, researchers have tried a variety of methods including regression-model-based methods, probability-model-based methods and deep-learning-based methods.

2. Related Work

Regression-model-based methods evaluate the mathematical expectations of missing data through known data points. Local binary pattern (LBP)-based support vector machines (SVMs) [2] have shown better recovery results when a small amount of traffic data are missing. Least squares support vector machines (LS-SVMs) introduced by Zhang and Liu [3], and the K-value proximity algorithm based on spatial and temporal correlation [4], also illustrate better imputation performance when missing types and data are mixed.

Online support vector machines (OL-SVR) proposed by Manoel [5] have more timely responses in repetitive traffic data. However, most regression models cannot recover data with high signal-to-noise ratio (SNR) or long sequences of missing data, which often occurs in ITS systems [6].

Probability models include the principal component analysis (PCA) [7] method, based on historical data mining, and the fully Bayesian generative model [8], based on tensor decomposition for estimating missing data. Bayesian principal component analysis (BPCA) [9] combines these two algorithms mentioned above to achieve a balance between the periodicity of the flow, local predictability, and statistical properties of traffic. The Bayesian Gaussian CANDECOMP/PARAFAC tensor decomposition (BGCP) [10] algorithm extends the tensor decomposition to higher dimension and applies it to the spatio-temporal traffic data interpolation task, solving the problem of missing data attribution in a spatio-temporal multidimensional environment. The variational Bayesian (VB) [11] algorithm exploits the spatio-temporal properties of network traffic to improve the quality of lost data recovery, fully capturing the multidimensional and spatio-temporal characteristics of traffic data.

Deep learning has demonstrated its great potential in many fields, including transportation [12]. Deep-learning-based data recovery models rely on high scale traffic data and incorporate the influence of nonlinear factors in a better way. Convolutional neural networks (CNNs) [13] are commonly used for image data recovery and improving image resolution, and super-resolution convolutional neural networks (SRCNNs) proposed by Dong et al. [14] can learn the recovery process from low-resolution images to high-resolution images in an end-to-end manner.

Generative adversarial networks (GAN) [15] are generative models which can create new data instances that resemble the training samples, they have been widely applied in many domains such as image restoration [16], video prediction [17] and security [18]. GANs are also used for traffic information recovery by using historical traffic data to improve recovery accuracy. He and Luo et al. [19] propose the research of GAN in traffic-data recovery. Arora, S. [20] conducts a study on the generalization ability of GAN in different situations. The encoded multiagent generative adversarial network (E-MGAN) proposed by Zhao [21] proves to be very effective in overcoming GAN pattern collapse. Deep Convolutional Generative Adversarial Network (DCGAN) [22] and Generalized Adversarial Interpolation Network (GAIN) [23] can solve the model instability problem to a certain extent. M. Arif [24] establishes a deep learning model with nonparametric regression to improve the prediction of lost data under nonlinear spatio-temporal effects. D. Tran et al. [25] finds that 3D convolution is more suitable for spatio-temporal feature learning than 2D convolution, easier to train and use. K. Xie [26] proposes a sequential tensor completing method to reduce the computing cost of high-dimensional neural network algorithms. All the above studies have promoted the application of 3D convolutional generative adversarial networks [27,28] that can effectively recover traffic data in large-scale traffic networks.

In summary, existing research has made some progress in the field of traffic-data restoration. But the accuracy of repairing large-scale missing data still needs to be improved. Traffic data are composed of multiple parameters, such as flow, speed and occupancy. These parameters are interrelated and contain complex high-dimensional traffic laws. Geometric algebra has strong expressive ability for multidimensional signals, and can better realize the learning of high-dimensional correlation. In this paper, considering the advantages of deep learning, we propose a geometric-algebra-based generative adversarial network (GAGAN) to recover missing traffic data by learning the correlation of multidimensional traffic parameters. The performance of traffic-data repair can be improved by coupling geometric algebra and generative adversarial network into a single model. We first preprocess the original traffic data, which include speed, flow and occupancy, to generate scalar-valued spatio-temporal matrices. By embedding the traffic data in the framework of geometric algebra, multivector-valued spatio-temporal matrices, which contain elements of high-dimensional entities, are created and used as the inputs of the proposed GAGAN model. The generator of GAGAN consists of a geometric algebra convolutional module,

an attention module and a deconvolutional module. The discriminator of GAGAN is composed of a multichannel convolutional neural network.

The main contributions of this paper are summarized as follows:

- We present a geometric algebra based generative adversarial network (GAGAN) to handle the problem of traffic data recovery. To represent and process multidimensional signals more efficiently, original traffic data are embedded in the framework of geometric algebra to form multivector-valued spatial-temporal matrices.
- The generator of the proposed GAGAN contains a geometric algebra convolutional module, an attention module and a deconvolutional module. The geometric algebra convolutional module is capable of learning the correlations of multidimensional inputs more efficiently. The loss function of the generator considers both the global and local traffic data mean squared errors.
- We conduct various experiments based on traffic data from two urban expressways of Shanghai, China. Experimental results prove that our method can effectively repair missing traffic data in a robust way. Compared with the state-of-the-art work, our approach shows the best performance.

3. Geometric Algebra of Euclidean 3D Space

Geometric algebra [29,30] is a generalization of vector algebra and it has been successfully applied in the domain of physics and engineering [31]. Compared with the classical vector algebra, modeling capability based on geometric algebra is tremendously extended. As a coordinate-free system, it captures the geometric characteristics of the problem in a better way and enables a more powerful representation and processing framework for multidimensional signals. Since the traffic data recovery problem is handled in the 3D Euclidean space (\mathbf{R}^3), in this section, we briefly introduce the geometric algebra of Euclidean 3D space (\mathbf{R}_3).

As shown in Equation (1), there are 8 basis elements of the geometric algebra of 3D Euclidean space (\mathbf{R}_3).

$$\mathbf{R}_3 = \text{span}\{1, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_{12}, \mathbf{e}_{23}, \mathbf{e}_{31}, \mathbf{e}_{123}\} \quad (1)$$

where 1 indicates the scalar basis, \mathbf{e}_1 , \mathbf{e}_2 and \mathbf{e}_3 refer to orthonormal basis vectors; \mathbf{e}_{12} , \mathbf{e}_{23} and \mathbf{e}_{31} indicate unit bivectors; \mathbf{e}_{123} means the unit trivector.

For a unit cube, \mathbf{e}_1 , \mathbf{e}_2 and \mathbf{e}_3 represent three axes, \mathbf{e}_{12} , \mathbf{e}_{23} and \mathbf{e}_{31} correspond to three surfaces and \mathbf{e}_{123} indicates the cube. By combining these basis elements, a multivector can be formed to represent multidimensional entities in an efficient way, e.g., $\mathbf{M} = 3 + 5\mathbf{e}_1 + 7\mathbf{e}_2 + 9\mathbf{e}_3 + 11\mathbf{e}_{12} + 13\mathbf{e}_{23} + 15\mathbf{e}_{31} + 17\mathbf{e}_{123}$. Geometric product is the basic product of geometric algebra, it is noncommutative and can be decomposed as the combination of inner product and outer product, Table 1 shows the results of geometric products of basis elements.

Table 1. The geometric product of basis elements of \mathbf{R}_3 .

	1	\mathbf{e}_1	\mathbf{e}_2	\mathbf{e}_3	\mathbf{e}_{12}	\mathbf{e}_{23}	\mathbf{e}_{31}	\mathbf{e}_{123}
1	1	\mathbf{e}_1	\mathbf{e}_2	\mathbf{e}_3	\mathbf{e}_{12}	\mathbf{e}_{23}	\mathbf{e}_{31}	\mathbf{e}_{123}
\mathbf{e}_1	\mathbf{e}_1	1	\mathbf{e}_{12}	\mathbf{e}_{13}	\mathbf{e}_2	\mathbf{e}_{123}	$-\mathbf{e}_3$	\mathbf{e}_{23}
\mathbf{e}_2	\mathbf{e}_2	$-\mathbf{e}_{12}$	1	\mathbf{e}_{23}	$-\mathbf{e}_1$	\mathbf{e}_3	$-\mathbf{e}_{123}$	$-\mathbf{e}_{13}$
\mathbf{e}_3	\mathbf{e}_3	$-\mathbf{e}_{13}$	$-\mathbf{e}_{23}$	1	$-\mathbf{e}_1$	\mathbf{e}_{123}	$-\mathbf{e}_1$	\mathbf{e}_{12}
\mathbf{e}_{12}	\mathbf{e}_{12}	$-\mathbf{e}_2$	\mathbf{e}_1	\mathbf{e}_{123}	-1	\mathbf{e}_{13}	\mathbf{e}_{23}	$-\mathbf{e}_3$
\mathbf{e}_{23}	\mathbf{e}_{23}	\mathbf{e}_{123}	$-\mathbf{e}_3$	\mathbf{e}_2	$-\mathbf{e}_{13}$	-1	$-\mathbf{e}_{12}$	$-\mathbf{e}_1$
\mathbf{e}_{31}	\mathbf{e}_{31}	\mathbf{e}_3	\mathbf{e}_{123}	\mathbf{e}_1	$-\mathbf{e}_{23}$	\mathbf{e}_{12}	-1	$-\mathbf{e}_2$
\mathbf{e}_{123}	\mathbf{e}_{123}	\mathbf{e}_{23}	$-\mathbf{e}_{13}$	\mathbf{e}_{12}	$-\mathbf{e}_3$	$-\mathbf{e}_1$	$-\mathbf{e}_2$	-1

Given two multivectors $\mathbf{M}_1 = 3\mathbf{e}_1 + 5\mathbf{e}_{23}$ and $\mathbf{M}_2 = 3\mathbf{e}_2 + 7\mathbf{e}_{12}$, their geometric product is given by

$$\mathbf{M}_1 \otimes \mathbf{M}_2 = \mathbf{M}_1 \cdot \mathbf{M}_2 + \mathbf{M}_1 \wedge \mathbf{M}_2 = 21\mathbf{e}_2 - 15\mathbf{e}_3 + 9\mathbf{e}_{12} + 35\mathbf{e}_{31} \quad (2)$$

where \otimes , \cdot and \wedge represent geometric product, inner product and outer product, respectively.

4. Proposed Methodology

4.1. Overview

This paper aims to realize the repair of damaged traffic data. Figure 1 shows our system architecture. Raw traffic data which include speed, flow and occupancy are collected by the detectors deployed on the elevated highway at specific time intervals, and there is a position interval between these detectors. First, raw traffic data are preprocessed, and then converted into spatio-temporal matrices, each of which integrates certain traffic information of a day in both spatial and time domains. The matrix containing speed information is used to generate damaged speed matrix using point-by-point multiplication with a randomly generated mask of the same size. Next, the damaged speed matrix, the complete flow matrix and the complete occupancy matrix become a sample whose label is the complete speed matrix. Samples of all days constitute a data set. We randomly divide the samples into training samples to train proposed GAGAN model and test samples to repair and test the performance of our model. Recovered speed matrix is obtained by multiplying predicted speed matrix generated by GAGAN with the mask-inverted matrix.

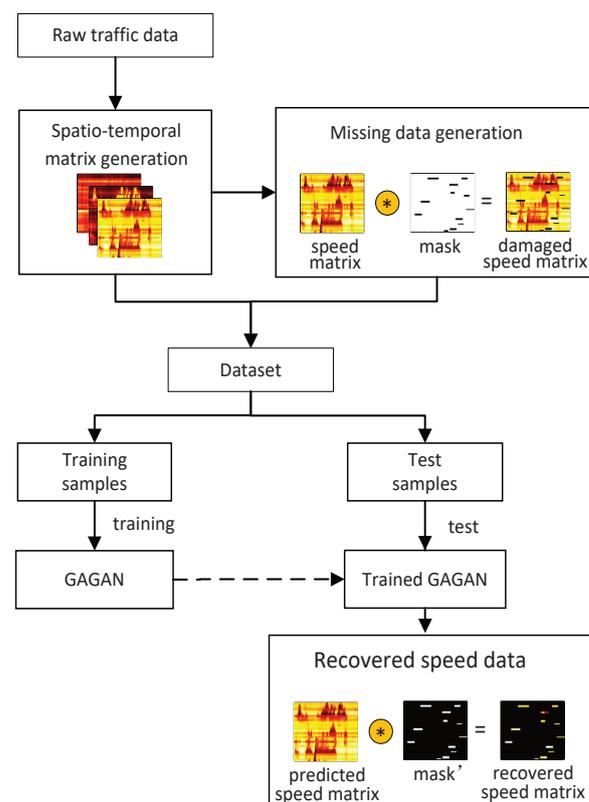


Figure 1. The system architecture of traffic speed imputation using GAGAN.

4.2. Damaged Data Set Generation

Traffic data are collected by detectors deployed on the road. Different roads have different value ranges for the same traffic parameter. Therefore, it is necessary to normalize

the traffic data including flow, speed and occupancy. For example, the regularization of speed can be described as:

$$s_{norm} = \frac{s - s_{min}}{s_{max} - s_{min}} \quad (3)$$

s_{norm} represents normalized data while s is original speed data. s_{max} means the maximum value of the original speed data, and s_{min} is the minimum value. Flow and occupancy are also processed in the same way.

Because the detectors are deployed at different locations on the road and collect traffic data at regular intervals, the traffic data itself has time and space properties. In order to make full use of the correlation between time and space, we construct the traffic spatial-temporal matrix. A row of the matrix indicates the location of a detector, and different columns represent different times of a day. The matrix elements refer to values of traffic speed. Mathematically, traffic speed spatial-temporal matrix can be represented as:

$$S = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1n} \\ S_{21} & S_{22} & \dots & S_{2n} \\ \dots & S_{ij} & \dots & \dots \\ S_{m1} & S_{m2} & \dots & S_{mn} \end{bmatrix} \quad (4)$$

The matrix S represents the traffic speed information for each day. Where m and n are the number of loop detectors and the number of time intervals respectively, S_{ij} is normalized speed of the i_{th} loop detector at the j_{th} time period. Similarly, we can get the flow spatial-temporal matrix and occupancy spatial-temporal matrix, which are represented as F and O , respectively.

Next, we simulate the damage to the traffic speed data. Traffic data corruption usually occurs in various locations. Moreover, the shape of the damaged part is also different. Therefore, we use two different shapes of masks to randomly destroy the data. One is the strip damage, in this case damaged data is continuous with time, which in the space-time matrix is displayed as a rectangle. The other is the discrete damage, that means the damaged data is discontinuous, which in the space-time matrix is displayed as dots. Mathematically, the mask can be defined as:

$$Mask = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1n} \\ k_{21} & k_{22} & \dots & k_{2n} \\ \dots & k_{ij} & \dots & \dots \\ k_{m1} & k_{m2} & \dots & k_{mn} \end{bmatrix} \quad (5)$$

where the value of k_{ij} is 0 or 1. If it is 0, the data of this point is damaged. If it is 1, the data of this point is retained.

Finally, we multiply the speed spatial-temporal matrix and the mask point-by-point to obtain a corrupted data set.

4.3. The GAGAN Model for Traffic Speed Recovery

The GAN model has been proved to perform very well in the application of image generation, and geometric algebra has the advantage of representing and processing multidimensional signals in an efficient way; therefore, we are inspired to propose the GAGAN model for traffic speed recovery by coupling GAN and geometric algebra into a single framework.

As shown in Figure 2, three scalar-valued matrices, i.e., damaged speed, complete flow and occupancy, are employed as the input of GAGAN model. By embedding these scalar-valued matrices in the geometric algebra, a multivector-valued matrix which represents multidimensional signals can be obtained and considered as the input of the generator. The generator of GAGAN is a geometric algebra convolutional neural network (GACNN) with multivector-valued neurons; it aims to learn the correlation of multidimensional traffic data and generate a recovered speed matrix. The discriminator of GAGAN contains a scalar-

valued multichannel CNN, which is applied to determine whether the result generated by the generator is true or false, and to continuously feed back information to the generator, thereby improving the model's repair accuracy.

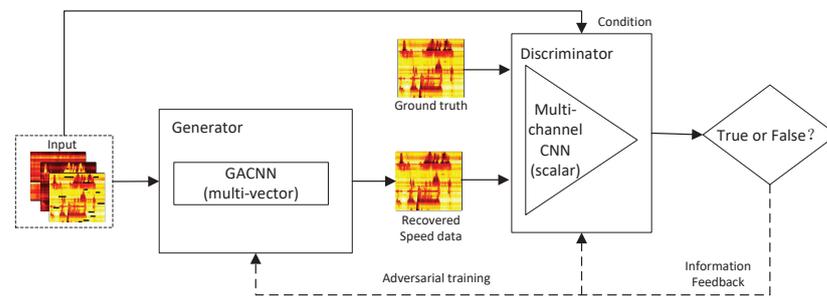


Figure 2. The geometric algebra based generative adversarial network (GAGAN).

Even though the presented GAGAN model in this paper is used to recover missing traffic speed data, it also can be generalized to recover different types of data based on multidimensional inputs.

4.3.1. The Generator of GAGAN

The structure of the generator is a GACNN, as illustrated in the Figure 3. It consists of two parts: encoding and decoding. The encoding part includes 3 geometric algebra convolutional layers, 3 pooling layers and 1 convolutional block attention module (CBAM). The function of the encoding part is to produce advanced feature maps which can efficiently describe the correlation characteristics of the input. The decoding part of the generator consists of 3 deconvolutional layers, aiming to decode the comprehensive spatio-temporal features extracted from the traffic parameters, and output the repaired speed matrix with the same size as the input speed matrix. Compared with scalar-valued CNN, GACNN has better capability to learn the potential dependencies between multidimensional inputs.

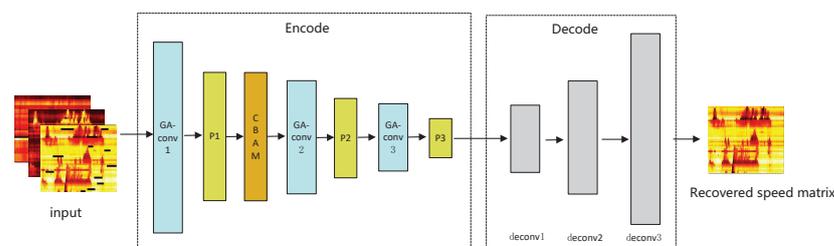


Figure 3. The generator of GAGAN.

The original inputs of the GAGAN model are damaged speed, complete flow and occupancy matrices of a day, they are first embedded in the geometric algebra with bivector basis to yield a multivector valued matrix as the input of the generator, which can be represented as Equation (6).

$$Z = \begin{bmatrix} Z_{11} & Z_{12} & \dots & Z_{1n} \\ Z_{21} & Z_{22} & \dots & Z_{2n} \\ \dots & Z_{ij} & \dots & \dots \\ Z_{m1} & Z_{m2} & \dots & Z_{mn} \end{bmatrix} \quad (6)$$

where the matrix Z indicates the multivector valued spatio-temporal matrix which encodes the traffic information for a day, m and n are the number of loop detectors and the number

of time intervals respectively, Z_{ij} is the multivector valued traffic parameter of the i_{th} loop detector at the j_{th} time period, Z_{ij} can be further expressed in the following form:

$$Z_{ij} = F_{ij}\mathbf{e}_{12} + S_{ij}\mathbf{e}_{23} + O_{ij}\mathbf{e}_{31} \quad (7)$$

where F_{ij} , S_{ij} and O_{ij} refer to the flow, speed and occupancy, respectively.

The geometric algebra convolutional layers of the generator are able to extract correlated spatio-temporal features by convolving the input with learnable kernels. Different with the conventional scalar-valued convolution, in this case, both the input and kernel are multivector-valued. For the L_{th} geometric algebra convolutional layer, the input of the multivector-valued neuron is the output of the previous layer, which can be denoted as

$$X_{ij}^{L-1} = X_{ij}^{L-1,r} + X_{ij}^{L-1,1}\mathbf{e}_{12} + X_{ij}^{L-1,2}\mathbf{e}_{23} + X_{ij}^{L-1,3}\mathbf{e}_{31} \quad (8)$$

where X_{ij}^{L-1} means the output of the previous layer, r indicates the scalar part of the multivector X_{ij}^{L-1} , 1, 2, 3 represent the corresponding bivector parts.

For the first layer, since there are only 3 traffic parameters, the scalar part of X_{ij}^{L-1} is zero, i.e., $X_{ij}^{L-1} = Z_{ij}$. However, according to the results of geometric product, X_{ij}^{L-1} at other layers will contain scalar parts. To perform the geometric algebra convolution, the weights W_{ij}^L of the kernel in the L_{th} layer also take multivector values as shown in Equation (9)

$$W_{ij}^L = W_{ij}^{L,r} + W_{ij}^{L,1}\mathbf{e}_{12} + W_{ij}^{L,2}\mathbf{e}_{23} + W_{ij}^{L,3}\mathbf{e}_{31} \quad (9)$$

Hence, the convolved output of a neuron in the L_{th} geometric algebra convolution layer reads

$$\begin{aligned} X_{ij}^L &= f\left(\sum_{i=1}^p \sum_{j=1}^q X_{ij}^{L-1} \otimes W_{ij}^L + B_{ij}^L\right) \\ &= f\left(\sum_{i=1}^p \sum_{j=1}^q (X_{ij}^{L-1} \cdot W_{ij}^L + X_{ij}^{L-1} \wedge W_{ij}^L) + B_{ij}^L\right) \end{aligned} \quad (10)$$

where f is the ReLU activation function, the kernel has a size of $p \times q$, \otimes , \cdot , \wedge respectively represent geometric product, inner product and outer product, B_{ij}^L means the bias parameter of this layer. According to the relationship shown in Table 1, the geometric product of two multivectors X_{ij}^{L-1} and W_{ij}^L is defined as:

$$\begin{aligned} X_{ij}^{L-1} \otimes W_{ij}^L &= X_{ij}^{L-1} \cdot W_{ij}^L + X_{ij}^{L-1} \wedge W_{ij}^L \\ &= D_r + D_1\mathbf{e}_{12} + D_2\mathbf{e}_{23} + D_3\mathbf{e}_{31} \end{aligned} \quad (11)$$

where D_r , D_1 , D_2 and D_3 are scalar coefficients which can be further expressed as:

$$\begin{aligned} D_r &= X_{ij}^{L-1,r}W_{ij}^{L,r} - X_{ij}^{L-1,1}W_{ij}^{L,1} \\ &\quad - X_{ij}^{L-1,2}W_{ij}^{L,2} - X_{ij}^{L-1,3}W_{ij}^{L,3} \end{aligned} \quad (12)$$

$$\begin{aligned} D_1 &= X_{ij}^{L-1,r}W_{ij}^{L,1} + X_{ij}^{L-1,1}W_{ij}^{L,r} \\ &\quad + X_{ij}^{L-1,2}W_{ij}^{L,3} - X_{ij}^{L-1,3}W_{ij}^{L,2} \end{aligned} \quad (13)$$

$$\begin{aligned} D_2 &= X_{ij}^{L-1,r}W_{ij}^{L,2} - X_{ij}^{L-1,1}W_{ij}^{L,3} \\ &\quad + X_{ij}^{L-1,2}W_{ij}^{L,r} + X_{ij}^{L-1,3}W_{ij}^{L,1} \end{aligned} \quad (14)$$

$$\begin{aligned} D_3 &= X_{ij}^{L-1,r}W_{ij}^{L,3} + X_{ij}^{L-1,1}W_{ij}^{L,2} \\ &\quad - X_{ij}^{L-1,2}W_{ij}^{L,1} + X_{ij}^{L-1,3}W_{ij}^{L,r} \end{aligned} \quad (15)$$

The geometric algebra convolution layer is mainly based on the geometric product operation to realize the information transfer between multivector neurons. The neurons are connected locally and the weights are shared. From Equations (10)–(15), it is demonstrated that the traditional scalar-valued convolution, indicated by the inner product, is just a part of the geometric algebra convolution. In addition, the geometric algebra convolution includes the computation of outer products, it provides the potential to learn the correlations of multidimensional inputs. Compared with 3D convolution, which ignores the relationship between channels and causes information loss, the geometric algebra convolution is capable of learning correlation features of multidimensional signals in a more efficient way.

Geometric algebra is the basic mathematical framework to model our problem, however, in the real implementation, we follow the way illustrated in Figure 4 to perform the computation of a geometric algebra convolutional layer L . For the geometric product, we map multivector-valued neurons to multiple scalar neurons according to the number of dimensions. In this case, one multivector-valued neuron corresponds to four scalar-valued neurons, their outputs can be obtained according to equations from (11) to (15) by adding and subtracting the results of 4 ordinary convolutions. The geometric algebra convolution is similar to learning the compound characteristics by aggregating several separate standard convolution results. The four convolved results are then combined by basis to form the multivector-valued input for the next layer.

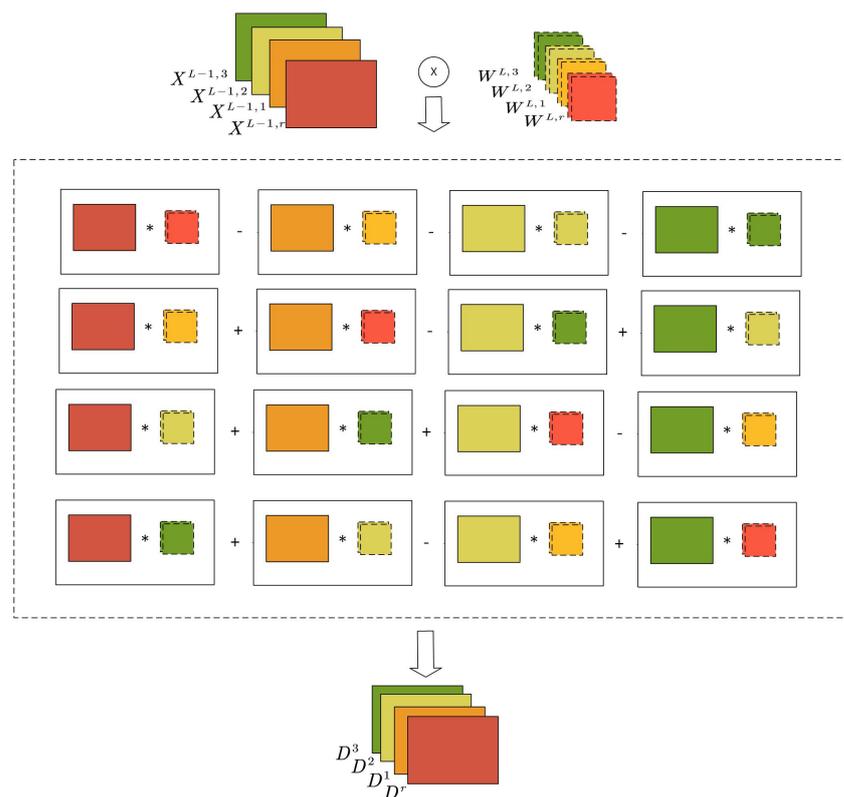


Figure 4. The implementation of geometric algebra convolutional layer.

It is worth noting that for the area to be repaired, the traffic area far away from it does not provide much information and may even interfere with the repair result. We introduce the convolutional block attention module (CBAM) to extract useful information and filter useless information, thereby improving feature extraction capabilities. Mathematically, the process of CBAM can be defined as:

$$F_c = \sigma(FC(AvgPool(F_{ga})) + FC(MaxPool(F_{ga}))) \odot F_{ga} \quad (16)$$

$$F_s(F_c) = \sigma(f^{7 \times 7}(AvgPool(F_c) \oplus MaxPool(F_c))) \odot F_c \quad (17)$$

F_{ga} , F_c and F_s represent feature maps obtained from geometric algebra convolution layer, channel attention module and spatial attention module, respectively, *AvgPool* and *MaxPool* represent average pooling and maximum pooling, respectively, *FC* and $f^{7 \times 7}$ refer to the fully connected layer and convolutional layer using the convolution kernel with a size of 7×7 , σ denotes the sigmoid function, \odot represents point-by-point multiplication between matrices, \oplus denotes concatenating channels.

The CBAM layer is composed of two parts, first the channel attention module, and then the spatial attention module. The channel attention module first adopts global average pooling and global maximum pooling. Then the feature maps are delivered to the fully connected layers to model the correlation between the channels. The weight of the feature channel is defined in Formula (16) as the part between symbols of = and \odot , they are multiplied channel-by-channel to complete the recalibration of the original feature in the channel dimension. The spatial attention module takes the output of the channel attention module as the input. The global average pooling and global maximum pooling are also used. The difference is that the pooling operation compresses the multichannel feature map into a single channel, so that the subsequent convolution only focuses on the spatial dimension. Finally, it is the same recalibration operation that the newly obtained weight in the spatial dimension is multiplied by the feature map to yield the result adjusted by double attention models.

After completing the extraction of the high-dimensional features of the traffic parameters, the extracted high-dimensional features need to be decoded. As mentioned above, we perform feature encoding based on three bivectors and one scalar. For decoding, deconvolution is performed for the four dimensions of three bivectors and one scalar. The feature maps obtained by deconvolution decode the traffic speed information layer by layer. Finally, we fuse the feature maps generated from these four dimensions, stitch them together according to the channels, and then pass them to the last deconvolution layer to produce the recovered traffic speed matrix.

4.3.2. The Discriminator Structure

The discriminator of GAGAN can be regarded as a binary classifier, it aims to distinguish as accurately as possible whether the input is the ground truth or the recovered value yielded by the generator. The discriminator fights against the generator, which further encourages the generator to produce more realistic recovered values. It has been proved that the performance of GAN will be improved if it is conditioned. The proposed GAGAN model is a conditional GAN, as illustrated in Figure 5. The discriminator is composed of multiple CNNs, each CNN consists of 2 convolutional layers, 2 pooling layers and 2 fully connected layers. Multidimensional data including flow, occupancy and damaged speed are taken as conditions and fed to three CNNs for learning patterns and distributions, features of the predicated speed matrix is also learned by another CNN. Concatenating results from these four CNNs produces P_1 , a value which indicates the probability of the output of generator coming from training samples. In addition to this, the ground truth of speed matrix is also delivered to the fifth CNN, and P_2 , the probability of real values coming from training samples can be obtained. In this paper, the multiple conditions applied to our model enable the discriminator to make a more reliable decision of the probability that the predicted value is the true value under the constraints of the current FSO matrices.

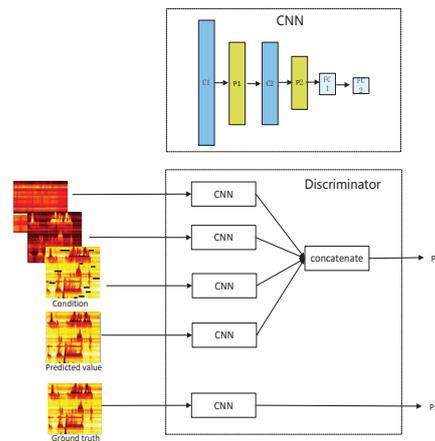


Figure 5. The structure of discriminator of GAGAN.

4.3.3. Model Optimization

Model training is a process of continuously adjusting the weight parameters. The model is composed of the generator network (G) and the discriminator network (D), they compete with each other and are trained alternately.

The goal of the discriminator is to distinguish as accurately as possible whether the input is the data generated by the generator or the real data, by minimizing the probability P_1 and maximizing the probability P_2 , as shown in Figure 5. Thus, the loss function of D is the cross entropy, which can be defined as:

$$L_D = -\log(1 - D(G(\hat{C}))) - \log(D(x|\hat{C})) \quad (18)$$

where \hat{C} means the FSO condition matrices, $G(\hat{C})$ indicates the output generated by the generator, $D(G(\hat{C}))$ denotes the probability P_1 , x refers to the training data coming from real distribution and $D(x|\hat{C})$ denotes the probability P_2 .

The goal of the optimization is to make the repaired value, that is the output of the generator, as close to the real value as possible. Based on the traditional GAN [32], we optimize the loss function of the generator as

$$L_G = \alpha L_g + \beta L_{totalMSE} + \gamma L_{localMSE}, \quad (19)$$

with

$$L_g = -\log(D(G(\hat{C}))) \quad (20)$$

$$L_{totalMSE} = \frac{1}{N} \sum_{t=1}^N \left(\frac{1}{mn} \left(\sum_{i=1}^m \sum_{j=1}^n (S_{ijt} - \hat{S}_{ijt})^2 \right) \right) \quad (21)$$

$$L_{localMSE} = \frac{1}{N} \sum_{t=1}^N \left(\frac{1}{C_t} \left(\sum_{i=1}^m \sum_{j=1}^n ((S_{ijt} - \hat{S}_{ijt}) * (1 - Mask_{ijt}))^2 \right) \right) \quad (22)$$

where α , β , and γ , whose sum equals 1, are weights associated with 3 parts of the loss function of GAGAN. L_g is used to measure the authenticity of the generated results and make the generated value from G more approximate to the real value. $L_{totalMSE}$ represents the global mean squared error (MSE), it measures the overall loss between the speed matrix generated by the generator and the real matrix. N refers to the number of test samples, m and n are the number of rows and columns of a speed matrix, respectively. S_{ijt} denotes the true speed value of the i_{th} loop detector at the j_{th} time period of the t_{th} test sample, \hat{S}_{ijt} means the corresponding recovered value. $L_{localMSE}$ is used to measure the loss between the recovered value and true value in the damaged area, so as to learn the characteristics

of the damaged area in a targeted manner. C_t indicates the number of damaged points in the t_{th} speed matrix. Similar to S_{ijt} , $Mask_{ijt}$ means the mask value of the the i_{th} loop detector at the j_{th} time period of the t_{th} test sample. The multiplier ' $1 - Mask_{ijt}$ ' aims to keep damaged points and remove other irrelevant points.

5. Experiment

5.1. Datasets and Settings

In this study, the experimental data are collected from two urban expressways named Yan'an and Neihuan of Shanghai, China in 2011. Figure 6 shows the map of these two elevated highways, which are important parts of Shanghai's urban transportation network and effectively increase the traffic capacity.



Figure 6. Marking of two elevated highways. Red and green bold lines mark Yan'an elevated highway and Neihuan elevated highway, respectively.

On each elevated highway, there is a loop detector every 400 m. The detector collects and stores traffic data at its location every 5 min, including flow, speed, and occupancy. There are 35 and 72 detectors on the Yan'an and Neihuan elevated highways, respectively. These detectors collect 288 time points in a day.

To use the correlation between time and space, we first convert the raw data collected from loop detectors into daily spatio-temporal matrices. However, there may exist errors in the spatio-temporal matrix, because of the inevitable damage of the detector and storage. Therefore, these data needs further process. Firstly, we use neighbour average filtering to handle the invalid value '0' in the matrix. Secondly, we choose to use data collected from 7 a.m. to 10 p.m. for experiments, because some loop detectors may be maintained at night and fail to collect traffic data. Lastly, in terms of the Yan'an elevated highway, we only have data from 361 days to make the data set, due to the lack of data from 20 March to 23 March. After making these processes, to simulate the traffic data damage, we use the two masks mentioned in Section 4.2. A value of 0 in the mask indicates that data is damaged. Multiplying the mask and the original speed space-time matrix point-by-point yields the damaged speed matrix. Figure 7 shows the strip damage. It may appear when a detector fails and lasts for a period of time. Figure 8 illustrates the discrete damage. This situation may occur when the transient detector fails or data is lost during storage. The damaged speed space-time matrix, the flow matrix, and the occupancy matrix together are taken as the input, and the label is the complete speed space-time matrix. All the matrices for the whole of 2011 constitute the basic data set of each elevated highway. To evaluate the performance of our proposed model, we randomly select 36 samples as the test set for each data set, and the remaining samples are regarded as the training set. For Yan'an and Neihuan elevated highways, their respective training sets include 325 and 329 samples.

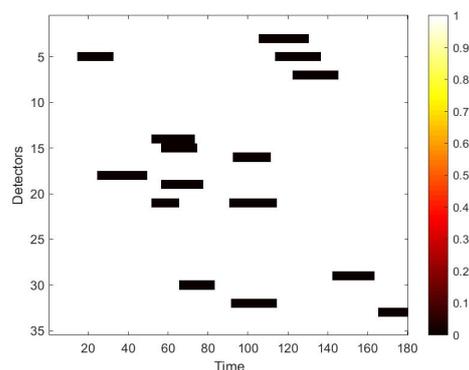


Figure 7. Mask used to simulate strip damage.

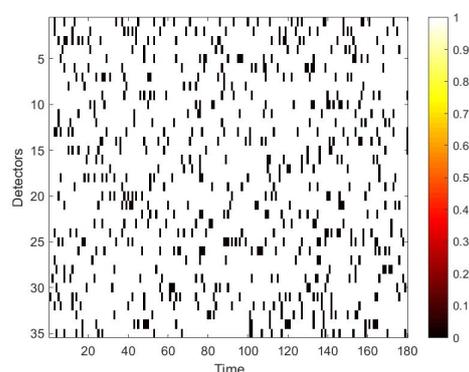


Figure 8. Mask used to simulate discrete damage.

The experiments are conducted on a server with i7-5820K CPU, 48 GB memory and NVIDIA GeForce GTX1080 GPU. The proposed model is implemented on the TensorFlow framework of deep learning, whose parameter configuration is shown in Tables 2 and 3. Note that, the parameters and network structure of the two elevated highways are the same. The step size of all convolution kernels is set as 1×1 . The learning rate of both the generator and the discriminator is 0.0001, and the total number of iterations of our network is 10,000.

Table 2. Parameter configuration of the generator of GAGAN.

Layers	Name	Description
1	GA-conv Layer1	32 kernels of size $3 \times 3 \times 1$
2	Pooling1	kernels of size 2×2
3	CBAM	attention module
4	GA-conv Layer2	64 kernels of size $3 \times 3 \times 32$
5	Pooling2	kernels of size 2×2
6	GA-conv Layer3	64 kernels of size $3 \times 3 \times 64$
7	Pooling3	kernels of size 2×2
8	deconv Layer1	64 kernels of size $3 \times 3 \times 64$
9	deconv Layer2	64 kernels of size $3 \times 3 \times 32$
10	deconv Layer3	32 kernels of size $3 \times 3 \times 1$

The numbers of training samples of Yan'an and Neihuan expressways are 325 and 329, respectively. These samples are considered as inputs to train the proposed GAGAN model by minimizing the loss function. Once the training process is terminated, the learned weight matrices will be immediately saved. There are 36 randomly selected test samples for every elevated highway, each sample is delivered to the saved generator of GAGAN to yield a predicted speed matrix by forward calculation. Combining the predicted speed matrix and its associated mask, missing speed values can be recovered.

Table 3. Parameter configuration of the discriminator of GAGAN.

Layers	Name	Description
1	convolution1	32 kernels of size $5 \times 5 \times 31$
2	pooling1	kernels of size 2×2
3	convolution2	64 kernels of size $3 \times 3 \times 32$
4	pooling2	kernels of size 2×2
5	FC1	128 neuron nodes
6	FC2	1 neuron nodes

5.2. Results and Analysis

We first visualize speed matrices as heat maps, which reveal the traffic speed values in a whole day, to demonstrate the repaired results of our model. In each heat map, the x-axis represents the time series of one day, and the y-axis indicates the position of these detectors. In addition, the different values of speed are represented with different colors. The darker the color is, the smaller the speed value is. Figure 9 includes the heat maps of Yan'an elevated highway with strip damage. From left to right are the mask, the damaged speed matrix, the ground truth and its corresponding repaired speed matrix on the 1st day of test set. Similarly, Figure 10 shows the results of 8th day of the test set with strip damage. Figure 11 contains the results of 31st day of the test set with discrete damage. Obviously, the repaired speed data of our model are very close to the ground truth for the Yan'an elevated highway with both strip damage and discrete damage. Then we conduct the same experiments on the Neihuan elevated highway, and the results are depicted in Figures 12–14. It is also illustrated that our proposed method achieves a close result to the ground truth for the Neihuan elevated highway with both strip damage and discrete damage.

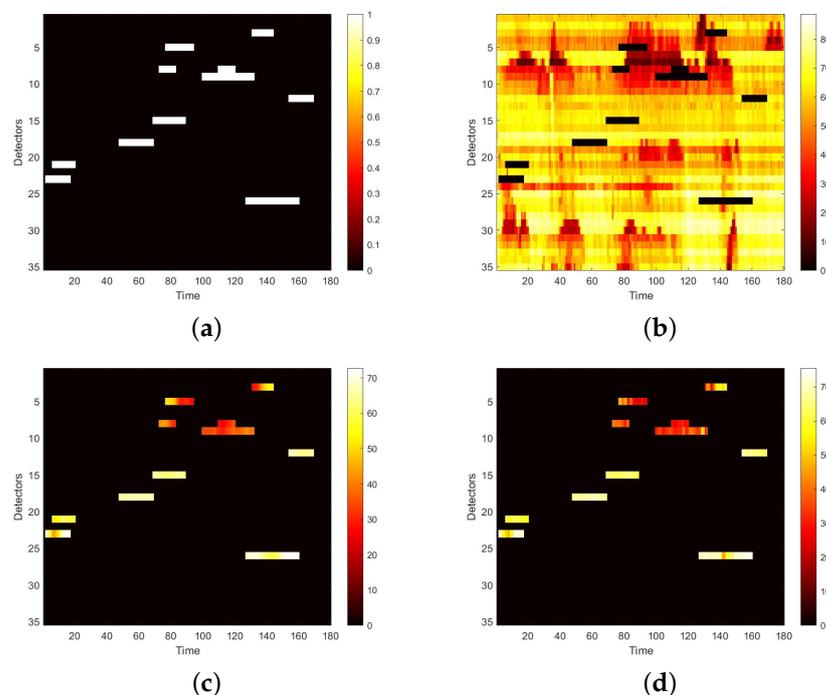


Figure 9. Speed matrices of Yan'an elevated highway (the 1st day of testset) with strip damage visualized as heat maps. (a) Mask matrix visualized as a heat map. (b) Damaged speed matrix visualized as a heat map. (c) Repaired speed matrix visualized as a heat map. (d) Real speed matrix visualized as a heat map.

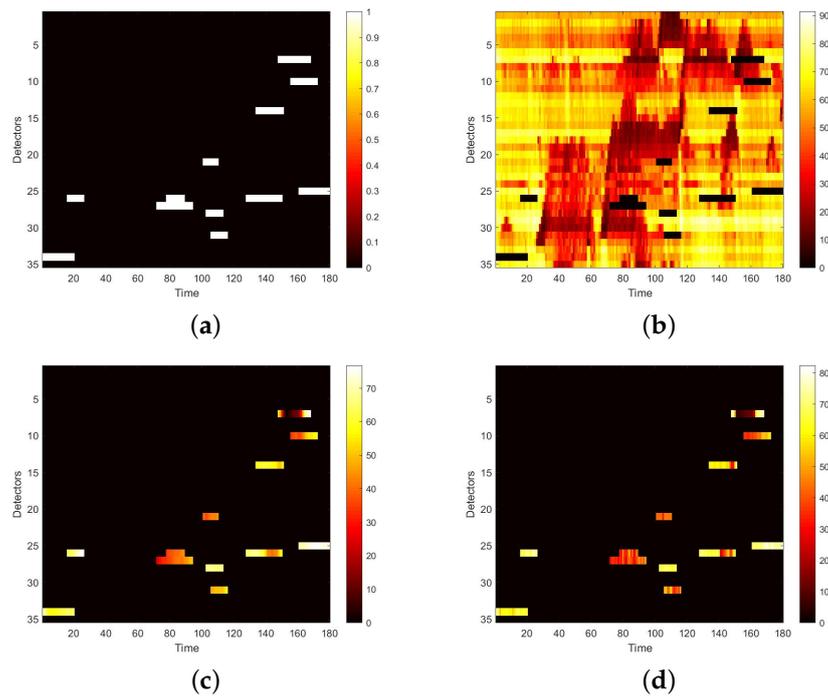


Figure 10. Speed matrices of Yan'an elevated highway (the 8th day of testset) with strip damage visualized as heat maps. (a) Mask matrix visualized as a heat map. (b) Damaged speed matrix visualized as a heat map. (c) Repaired speed matrix visualized as a heat map. (d) Real speed matrix visualized as a heat map.

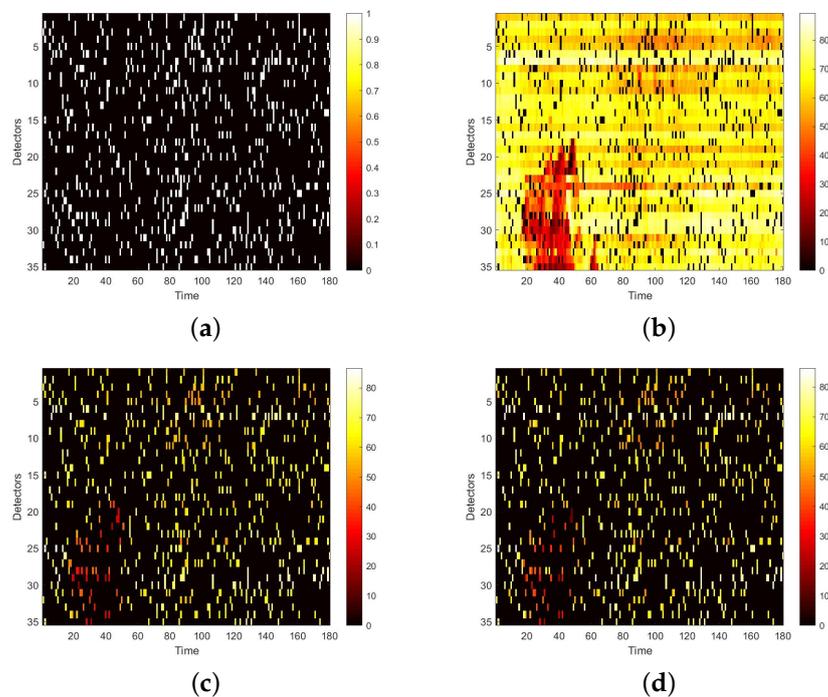


Figure 11. Speed matrices of Yan'an elevated highway (the 31st day of testset) with discrete damage visualized as heat maps. The ratio of damaged area to total area is 10%. (a) Mask matrix visualized as a heat map. (b) Damaged speed matrix visualized as a heat map. (c) Repaired speed matrix visualized as a heat map. (d) Real speed matrix visualized as a heat map.

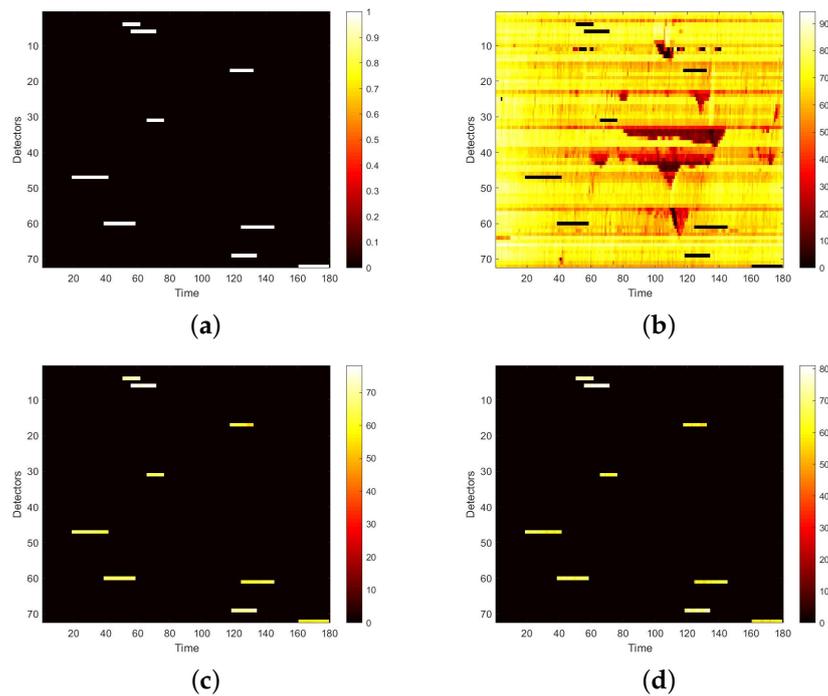


Figure 12. Speed matrices of Neihuan elevated highway (the 6th day of testset) with strip damage visualized as heat maps. (a) Mask matrix visualized as a heat map. (b) Damaged speed matrix visualized as a heat map. (c) Repaired speed matrix visualized as a heat map. (d) Real speed matrix visualized as a heat map.

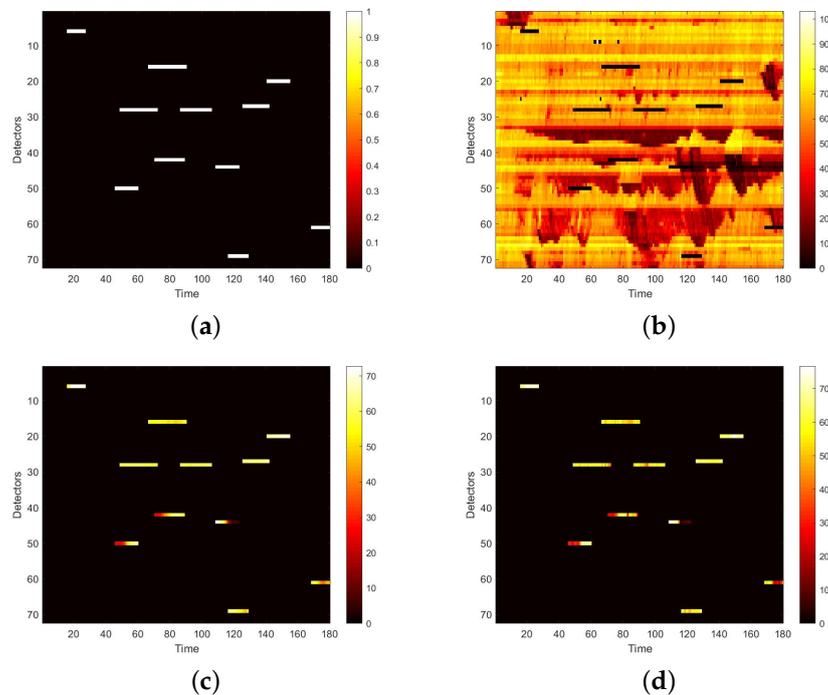


Figure 13. Speed matrices of Neihuan elevated highway (the 23rd day of testset) with strip damage visualized as heat maps. (a) Mask matrix visualized as a heat map. (b) Damaged speed matrix visualized as a heat map. (c) Repaired speed matrix visualized as a heat map. (d) Real speed matrix visualized as a heat map.

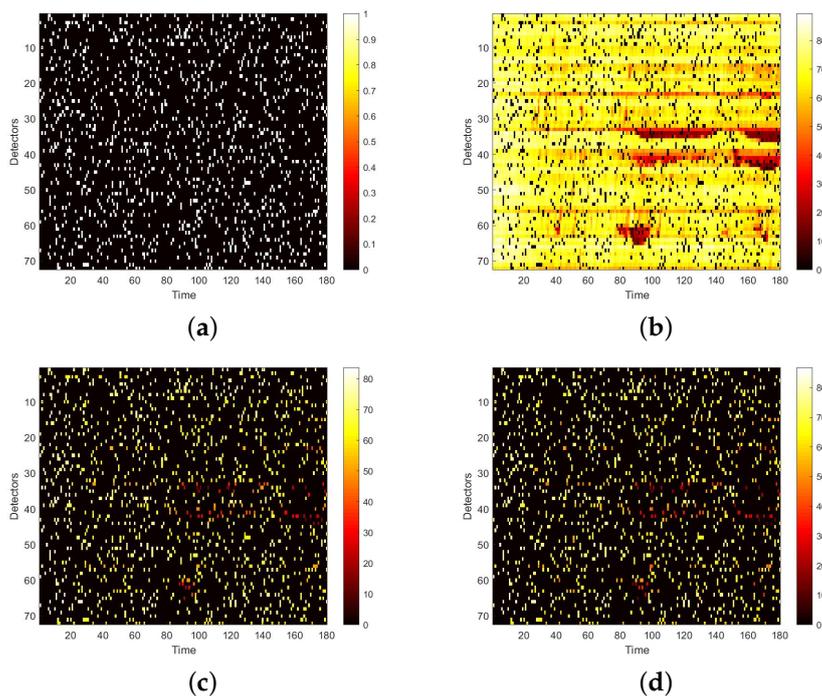


Figure 14. Speed matrices of Neihuan elevated highway (the 36th day of testset) with discrete damage visualized as heat maps. The ratio of damaged area to total area is 10%. (a) Mask matrix visualized as a heat map. (b) Damaged speed matrix visualized as a heat map. (c) Repaired speed matrix visualized as a heat map. (d) Real speed matrix visualized as a heat map.

In this paper, we use L1 loss and L2 loss to evaluate the repair performance. The L1 loss indicates the average absolute error (MAE) of the damaged location, and the L2 loss is used to measure the mean squared error (MSE). The formulas of L1 loss and L2 loss are defined as

$$L_1 = \frac{1}{v} \sum_{j=1}^v \left(\frac{1}{u} \sum_{i=1}^u |y_{ij} - \hat{y}_{ij}| \right) \quad (23)$$

$$L_2 = \frac{1}{v} \sum_{j=1}^v \left(\frac{1}{u} \sum_{i=1}^u (y_{ij} - \hat{y}_{ij})^2 \right) \quad (24)$$

where y_{ij} means the true value of the i_{th} damaged point in the speed matrix of j_{th} test sample, \hat{y}_{ij} is the corresponding repaired value, u indicates the number of damaged points in a recovered speed matrix and v denotes the number of samples in the test set.

To improve the efficiency of the proposed model, we compare our method with CNNBranch3 [27], CNN3 [28], CNN1 [22] and CNNBranch3_fc, they all have the GAN architecture. In order to prove the advantages of geometric algebra convolution, CNNBranch3 is used as its comparative experiment. The difference between CNNBranch3 and our model is only the convolutional layer of the generator. CNNBranch3 uses traditional scalar convolution, while our model has geometric algebra convolution. Compared with CNNBranch3, which takes a multibranch structure to process inputs with parameters of F , S and O , CNN3 simply uses 3D convolution to process inputs with these three parameters. In order to prove the influence of parameter correlation on the repair effect, CNN1 was employed for comparison. It only takes the damaged speed parameter as the input; flow and occupancy are not included. CNN1 also uses scalar convolution. For the last CNNBranch3_fc model, compared with the CNNBranch3 model, the deconvolution layer of the generator is replaced with a fully connected layer, and the other modules remain unchanged to prove the importance of decoding high-dimensional features.

In the experiment, the generator of CNNBranch3 contains three branches, which are used to process traffic, speed, and occupancy data. These three branches encode and decode the corresponding parameters. The results generated by the branches are merged, the repaired data can be obtained from the last deconvolution layer. The encoding part has three convolutional layers, and the decoding also includes three deconvolutional layers. The generator of CNN3 employs 3D convolution to process the three traffic parameters, without merging feature maps such as branches, and other module structures are the same as that of CNNBranch3. The input of the CNN1 generator is the impaired speed matrix, which is also composed of three convolutional layers and three deconvolutional layers. The structure of CNNBranch3_fc and CNNBranch3 is basically the same, except that the three deconvolution layers in the generator is replaced with three fully connected layers.

Curves of Figures 15 and 16 demonstrate the recovered values and their corresponding ground truth of the Yan'an and Neihuan elevated highways for some detectors. More specifically, we randomly selected six diagrams of each highway, which represent the speed values of different detectors on different days. In these subgraphs, the blue solid line denotes the ground truth, the yellow solid line represents the repaired value of our model, and other curves with different colors indicate the results of baseline methods. It can be seen from these figures that the repaired results generated by our model are the closest to the ground truth.

Finally, we use strip mask and show the results compared with baseline methods in Tables 4 and 5. It can be found that the proposed model achieves the lowest error among all of the methods. More specifically, CNN1 and CNNBranch3_fc perform the worst, as CNN1 does not consider the correlation of traffic parameters, and CNNBranch3_fc cannot decode the extracted features effectively. The performance of CNN3 is better than the previous two, because it makes full use of the excellent decoding ability of parameter correlation and deconvolution. Compared with CNN3, CNNBranch3 performs feature extraction on each parameter separately, so that all parameters can be fused after they have been fully learned, and the performance is better. Our model shows the best performance due to the use of geometric algebra convolution. For the Yan'an elevated highway, L1 and L2 indicators produced by our method are 3.264% and 0.259%, which are the lowest of all listed methods. Compared with the presented models, L1 and L2 measures of our approach for the Neihuan elevated highway have the values of 2.616% and 0.180%, which are still the smallest.

In order to further verify the generalization ability, we conduct a comparative experiment with different degrees of damage in the case of discrete damage. The ratios of damaged area to total area are 10%, 20%, 30%, 40% and 50%. Figure 17 demonstrates that as the degree of damage increases, the performance of the models also declines within a reasonable range. However, the proposed method still performs the best, which proves the robustness of our model.

In this section, various experiments are conducted to evaluate the robustness of our model and to compare with other state-of-the-art work. It is illustrated that our method outperforms CNNBranch3 [24], CNN3 [25], CNN1 [19] and CNNBranch3_fc. In addition, the proposed approach performs well with both strip damage and discrete damage on two highways. Specifically, in the case of discrete damage, the generalization ability of our model with different degrees of damage is also proved. The performance of the proposed GAGAN model greatly contributes to the joint learning of correlation between high-dimensional traffic parameters.

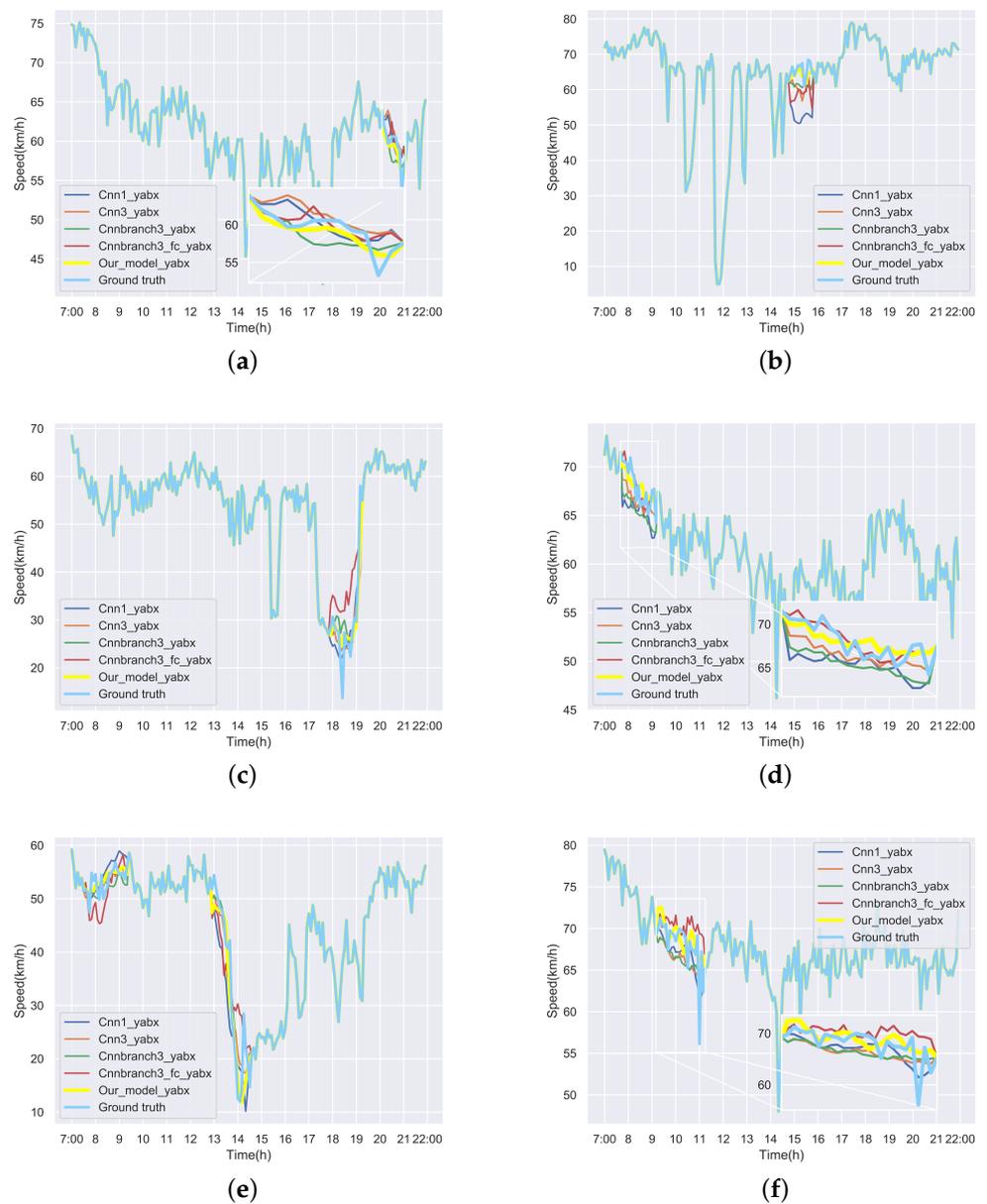


Figure 15. Repaired speed curves and corresponding ground truth of six loop detectors of Yan'an elevated highway. (a) The repaired values and the ground truth of the 10th loop detector on the Yan'an elevated highway in the 2nd test day. (b) The repaired values and the ground truth of the 35th loop detector on the Yan'an elevated highway in the 3rd test day. (c) The repaired values and the ground truth of the 8th loop detector on the Yan'an elevated highway in the 7th test day. (d) The repaired values and the ground truth of the 11th loop detector on the Yan'an elevated highway in the 10th test day. (e) The repaired values and the ground truth of the 19th loop detector on the Yan'an elevated highway in the 32nd test day. (f) The repaired values and the ground truth of the 15th loop detector on the Yan'an elevated highway in the 35th test day.

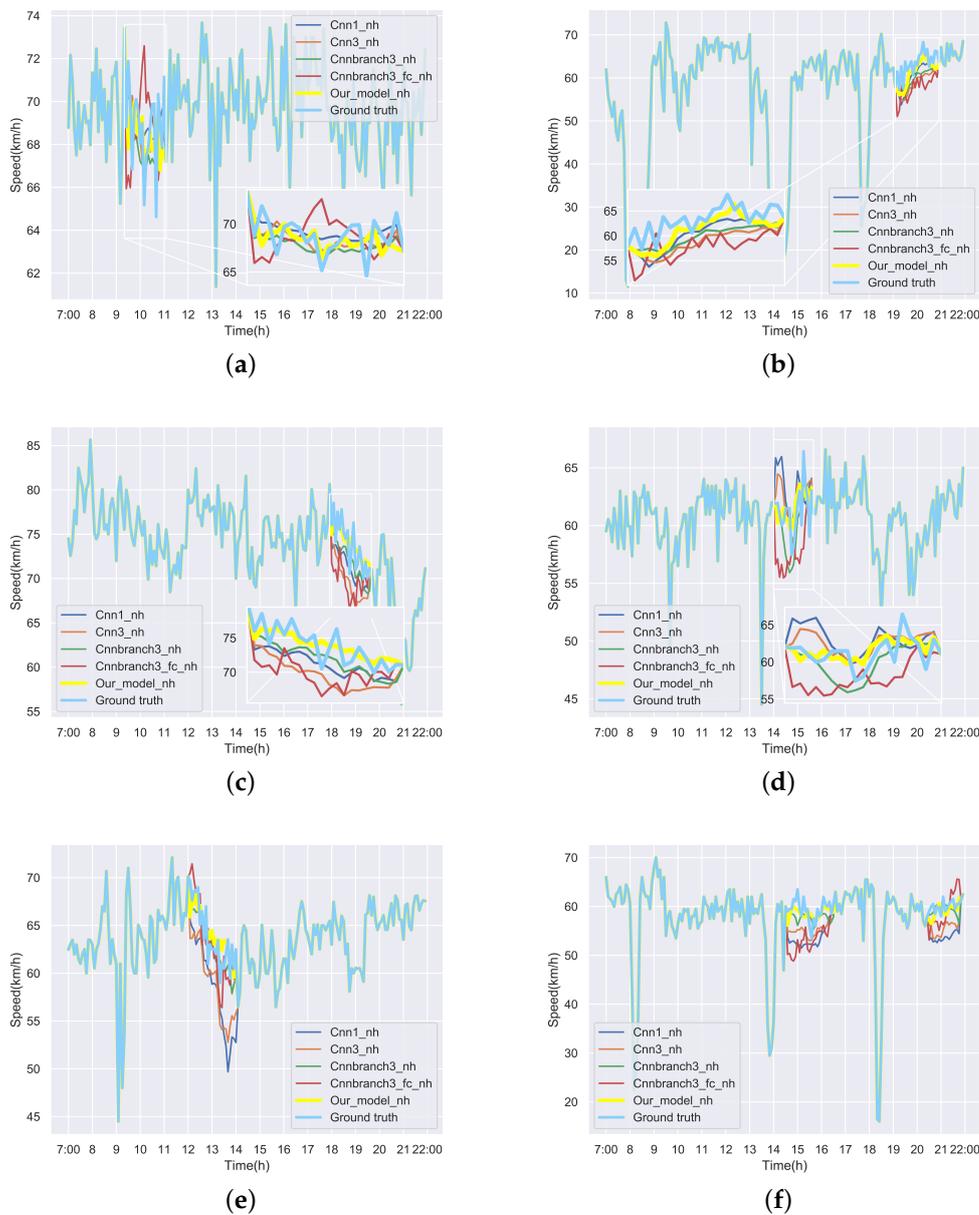


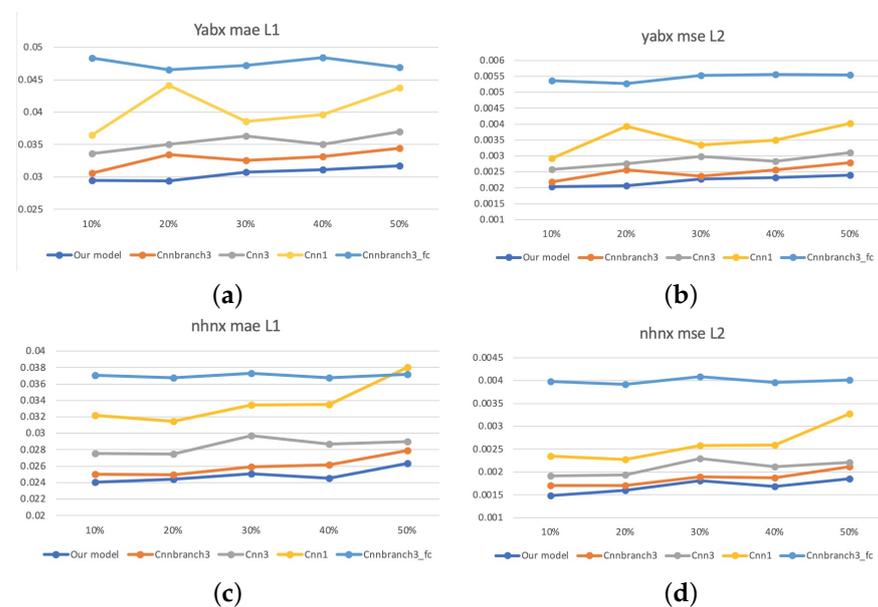
Figure 16. Repaired speed curves and corresponding ground truth of six loop detectors of Neihuan elevated highway. (a) The repaired values and the ground truth of the 7th loop detector on the Neihuan elevated highway in the 2nd test day. (b) The repaired values and the ground truth of the 24th loop detector on the Neihuan elevated highway in the 10th test day. (c) The repaired values and the ground truth of the 14th loop detector on the Neihuan elevated highway in the 12th test day. (d) The repaired values and the ground truth of the 10th loop detector on the Neihuan elevated highway in the 15th test day. (e) The repaired values and the ground truth of the 32nd loop detector on the Neihuan elevated highway in the 21st test day. (f) The repaired values and the ground truth of the 31st loop detector on the Neihuan elevated highway in the 27th test day.

Table 4. Comparison results for Yan'an elevated highway.

Models	L1	L2
Our model	0.03264	0.00259
Cnnbranch3	0.03495	0.00271
Cnn3	0.03960	0.00357
Cnn1	0.05014	0.00573
Cnnbranch3fc	0.05198	0.00636

Table 5. Comparison results for Neihuan elevated highway.

Models	L1	L2
Our model	0.02616	0.00180
Cnnbranch3	0.02995	0.00254
Cnn3	0.03741	0.00312
Cnn1	0.04830	0.00490
Cnnbranch3fc	0.04034	0.00399

**Figure 17.** Results of different levels of damage on two elevated highways. (a) L1 of different levels of damage on Yan'an elevated highways. (b) L2 of different levels of damage on Yan'an elevated highways. (c) L1 of different levels of damage on Neihuan elevated highways. (d) L2 of different levels of damage on Neihuan elevated highways.

6. Conclusions

In this paper, we propose a geometric-algebra-based generative adversarial network to deal with the important task of repairing missing traffic speed data. The original traffic data which include speed, flow and occupancy are first processed as spatial-temporal matrices. To make full use of the correlation between different traffic parameters, the speed, flow and occupancy data are embedded in the geometric algebraic framework to form multivectors and used as the input of the proposed model. The geometric algebra convolution module in the generator encodes high-dimensional data and enables efficient joint learning of multidimensional traffic parameters. The deconvolution module in the generator decodes the extracted features and generates recovered traffic speed matrix. In the proposed model, the generator loss function takes into account the feedback information from the discriminator, the global and local traffic speed data characteristics at the same time. The discriminator based on the multichannel convolutional network makes the repair value more realistic. Traffic data obtained from the elevated highway loop detectors are

used to evaluate the performance of the proposed method. Experimental results show that our approach outperforms the state-of-the-art work and can effectively recover missing traffic speed data in a robust way.

Author Contributions: Conceptualization, D.Z., X.Q. and K.T.; methodology, D.Z., K.T., X.Q. and Y.D.; software, Y.D., X.Q. and C.M.; validation, Y.D. and X.C.; formal analysis, C.M. and X.C.; investigation, C.M. and X.C.; resources, D.Z. and K.T.; data curation, C.M. and X.C.; writing—original draft preparation, D.Z., Y.D., X.Q. and C.M.; writing—review and editing, D.Z., K.T., Y.D., X.C. and J.Z.; visualization, Y.D., X.Q. and C.M.; supervision, D.Z., K.T. and J.Z.; project administration, D.Z., K.T. and J.Z.; funding acquisition, D.Z., J.Z. and K.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by National Natural Science Foundation of China (No. 61876218), Innovation Program of Shanghai Municipal Education Commission (202101070007E00098), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0100) and the Fundamental Research Funds for the Central Universities.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Du, J.; Chen, H.; Zhang, W. A Deep Learning Method for Data Recovery in Sensor Networks Using Effective Spatio-Temporal Correlation Data. *Sens. Rev.* **2019**, *39*, 208–217. [[CrossRef](#)]
- Prasad, D.; Kapadni, K.; Gadpal, A.; Visave, M.; Sultanpure, K. HOG, LBP and SVM Based Traffic Density Estimation at Intersection. In Proceedings of the 2019 IEEE Pune Section International Conference (PuneCon), Pune, India, 18–20 December 2020.
- Yang, Z.; Liu, Y. Missing Traffic Flow Data Prediction Using Least Squares Support Vector Machines in Urban Arterial Streets. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Data Mining, Nashville, TN, USA, 30 March–2 April 2009; pp. 76–83.
- Tak, S.; Woo, S.; Yeo, H. Data-Driven Imputation Method for Traffic Data in Sectional Units of Road Links. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1762–1771. [[CrossRef](#)]
- 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 Syst. Appl.* **2009**, *36*, 6164–6173. [[CrossRef](#)]
- Zhang, H.S.; Zhang, Y.; Li, Z.H.; Hu, D.C. Spatial-Temporal Traffic Data Analysis Based on Global Data Management Using Mas. *IEEE Trans. Intell. Transp. Syst.* **2004**, *5*, 267–275. [[CrossRef](#)]
- Qu, L.; Li, L.; Zhang, Y.; Hu, J. Ppca-Based Missing Data Imputation for Traffic Flow Volume: A Systematical Approach. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 512–522.
- Zhao, Q.; Zhou, G.; Zhang, L.; Cichocki, A.; Amari, S.I. Bayesian Robust Tensor Factorization for Incomplete Multiway Data. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *27*, 736–748. [[CrossRef](#)] [[PubMed](#)]
- Li, Q.; Yi, Z.; Hu, J.; Jia, L.; Li, L. A Bpca Based Missing Value Imputing Method for Traffic Flow Volume Data. In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands, 4–6 June 2008.
- Chen, X.; He, Z.; Sun, L. A Bayesian Tensor Decomposition Approach for Spatiotemporal Traffic Data Imputation. *Transp. Res. Part C Emerg. Technol.* **2019**, *98*, 73–84. [[CrossRef](#)]
- Zhou, H.; Zhang, D.; Xie, K.; Chen, Y. Robust Spatio-Temporal Tensor Recovery for Internet Traffic Data. In Proceedings of the 2016 IEEE Trustcom/BigDataSE/I SPA, Tianjin, China, 23–26 August 2016.
- Liang, X. *Applied Deep Learning in Intelligent Transportation Systems and Embedding Exploration*; New Jersey Institute of Technology: Newark, NJ, USA, 2019.
- Ma, X.; Dai, Z.; He, Z.; Ma, J.; Wang, Y.; Wang, Y. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors* **2017**, *17*, 818. [[CrossRef](#)] [[PubMed](#)]
- Dong, C.; Loy, C.C.; He, K.; Tang, X. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307. [[CrossRef](#)] [[PubMed](#)]
- Mudavathu, K.D.B.; Rao, M.V.P.C.S.; Ramana, K.V. Auxiliary Conditional Generative Adversarial Networks for Image Data Set Augmentation. In Proceedings of the 2018 3rd International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 15–16 November 2018.
- Hussein, S.A.; Tiner, T.; Giryes, R. Image-Adaptive Gan Based Reconstruction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
- Oprea, S.; Martinez-Gonzalez, P.; Garcia-Garcia, A.; Castro-Vargas, J.A.; Orts-Escolano, S.; Garcia-Rodriguez, J.; Argyros, A. A Review on Deep Learning Techniques for Video Prediction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *1*. [[CrossRef](#)] [[PubMed](#)]
- Kwon, H.; Ko, K.; Kim, S. Optimized Adversarial Example with Classification Score Pattern Vulnerability Removed. *IEEE Access* **2021**, *1*. [[CrossRef](#)]
- He, M.; Luo, X.; Wang, Z.; Yang, F.; Qian, H.; Hua, C. Global Traffic State Recovery Via Local Observations with Generative Adversarial Networks. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020.
- Arora, S.; Ge, R.; Liang, Y.; Ma, T.; Zhang, Y. Generalization and Equilibrium in Generative Adversarial Nets (Gans). In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.

21. Zhao, H.; Li, T.; Xiao, Y.; Wang, Y. Improving Multi-Agent Generative Adversarial Nets with Variational Latent Representation. *Entropy* **2020**, *22*, 1055. [[CrossRef](#)] [[PubMed](#)]
22. Yeh, R.A.; Chen, C.; Lim, T.Y.; Schwing, A.G.; Hasegawa-Johnson, M.; Do, M.N. Semantic Image Inpainting with Deep Generative Models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
23. Yoon, J.; Jordon, J.; Schaar, M. Gain: Missing Data Imputation Using Generative Adversarial Nets. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 14–15 July 2018.
24. Arif, M.; Wang, G.; Chen, S. Deep Learning with Non-Parametric Regression Model for Traffic Flow Prediction. In Proceedings of the 2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Athens, Greece, 12–15 August 2018.
25. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning Spatiotemporal Features with 3d Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
26. Xie, K.; Wang, L.; Wang, X.; Xie, G.; Wen, J.; Zhang, G.; Cao, J.; Zhang, D. Accurate Recovery of Internet Traffic Data: A Sequential Tensor Completion Approach. *IEEE/ACM Trans. Netw.* **2018**, *26*, 793–806. [[CrossRef](#)]
27. Chen, Y.; Lv, Y.; Wang, F. Traffic Flow Imputation Using Parallel Data and Generative Adversarial Networks. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1624–1630. [[CrossRef](#)]
28. Yang, B.; Kang, Y.; Yuan, Y.; Li, H.; Wang, F. St-Fvgan: Filling Series Traffic Missing Values with Generative Adversarial Network. *Transp. Lett.* **2021**, *1–9*. [[CrossRef](#)]
29. Ablamowicz, R.; Parra, J.; Lounesto, P. *Clifford Algebras with Numeric and Symbolic Computations*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1996.
30. Hestenes, D.; Li, H.; Rockwood, A. New Algebraic Tools for Classical Geometry. In *Geometric Computing with Clifford Algebras*; Springer: Berlin/Heidelberg, Germany, 2001.
31. Doran, C.J.L. Geometric Algebra and Its Application to Mathematical Physics. Ph.D. Thesis, University of Cambridge, Cambridge, UK, 1994.
32. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*.