**SDSCNN: A Hybrid Model Integrating Static and Dynamic Spatial Correlation Neural Network for Traffic Prediction**

JUNMING DAI¹, JIASHUANG HUANG¹, QINQIN SHEN², QUAN SHI¹,², SIYUN FENG³, AND ZHENQUAN SHI¹,²

¹School of Information Science and Technology, Nantong University, Nantong 226000, China
²School of Transportation and Civil Engineering, Nantong University, Nantong 226000, China
³Information Department, The Third Affiliated Hospital of Nantong University, Nantong 226000, China

Corresponding author: Zhenquan Shi (szq@ntu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62176131 and Grant 61771265, and in part by the Key Science and Technology Foundation of Nantong under Grant MS22021034.

**ABSTRACT** Traffic flow prediction is of great significance for traffic control, and it has been challenging for capturing the complex spatial-temporal correlation. However, most existing prediction methods only consider the spatial adjacency of the nodes (i.e., static spatial correlation), lacking sufficient analysis of non-stationary traffic conditions (i.e., dynamic spatial correlation). The combination of static spatial correlation and dynamic spatial correlation enables the model to comprehensively analyze the feature of traffic flow at each moment and improve the mining capability. To address this problem, we use the multi-head self-attention mechanism to establish a hybrid model integrating static and dynamic spatial correlation neural network (SDSCNN) for traffic flow prediction. Specifically, we first construct static adjacency matrix and dynamic adjacency matrix according to different methods. These two matrices are simultaneously input into Graph Attention Network for analysis. The two outputs are integrated by the sum operation. Then the fused static and dynamic spatial features are fed into the multi-head self-attention layer to analyze the temporal correlation. Also, multi-layer SDSCNNs are stacked to further analyze the dynamic correlations between road sections, as well as to improve the model’s multi-step prediction capability. Finally, Multi-layer Perceptron is used to output the prediction results. Extensive experiments are conducted using the datasets PEMS04, PEMS08, and METR-LA. And the results demonstrate that our model shows a good prediction performance.

**INDEX TERMS** Traffic flow prediction, static and dynamic spatial correlation, graph attention network, multi-head self-attention mechanism.

**I. INTRODUCTION**

With the development of intelligent transportation systems, more and more scholars pay attention to traffic flow prediction. Accurate and effective predictions can help traffic management departments guide vehicles more reasonably, alleviate traffic congestion and improve road traffic efficiency [1]. Current research on traffic flow prediction mainly focuses on the mining of temporal and spatial correlations [2]. Temporal correlation refers to the impact of the historical traffic flow state on the future state [3]. It focuses on periodicity. For example, the morning peak hours of weekdays are generally fixed, and the traffic flow state during these periods is similar. Also, the traffic state of the previous week is similar to the general trend of the current week, which means that there is a temporal correlation between the current traffic status and the historical traffic status. Currently, the existing models for analyzing temporal correlation include Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) [4], [5]. However, these models can only analyze...
FIGURE 1. Illustration of the proposed SDSCNN model for traffic prediction. There are five main elements: (a) static spatial correlation analysis; (b) dynamic spatial correlation analysis; (c) self-attention for temporal correlation analysis; (d) stacked SDSCNN layers for deep analysis; (e) MLP for prediction.

In the case of smooth traffic flow, the spatial correlation of traffic flow depends on the static topology of the road. The spatial correlation between the roads is usually consistent with the topology and remains unchanged. Feeding the original adjacency matrix into GCN can better extract static spatial correlation. However, for some unstable traffic situations (e.g., traffic congestion during peak hours, sudden traffic accidents, and one-way road restrictions during peak hours), the spatial correlation of traffic flow not only changes with time but also shows dynamic characteristics. In these cases, the simple topology cannot accurately represent the spatial correlation between roads at that moment. Adding the dynamic information based on static spatial correlation can fully consider the dynamic characteristics of spatial correlation of traffic flow which changes with time, and make up for the shortcomings of static spatial correlation [12], [13]. By integrating static spatial correlation and dynamic spatial correlation, we can conduct a comprehensive analysis of traffic flow and improve the generalization ability and prediction accuracy of the model.

To this end, we propose a hybrid model integrating static and dynamic spatial correlation neural network for traffic prediction. Figure 1 shows the model structure proposed in this paper. The model divides the spatial correlation into two parts, including the static spatial correlation analysis and dynamic spatial correlation analysis. Specifically, static spatial correlation analysis relies on the original topology of the road. It transforms the road topology into an adjacency matrix and uses Graph Convolutional Network (GCN) to extract spatial correlation between road sections [7]. Only static spatial correlation between neighboring traffic flow is extracted via the adjacency matrix. Dynamic spatial correlation analysis is part of dynamic graph learning [8], [10]. In dynamic graph learning, in addition to considering the weights of edges between nodes, the dynamic changes of nodes in the graph are also considered. In traffic flow prediction, most traffic flow prediction studies focus on mining the weights of edges between nodes, and rarely consider the changes in sensor nodes. These methods dynamically obtain optimal weight parameters between nodes through an end-to-end training process [11]. In this way, the spatial weight obtained from training depends on the input traffic value rather than the original road network topology. However, the drawback of this method is that for a large road network, the number of parameters is large and the computational complexity is high.

The spatial correlation of traffic flow cannot be analyzed comprehensively using either of these two methods alone.
(1) We propose a combination mining method, which jointly uses static and dynamic spatial correlations for traffic flow prediction. Compared to using only static or dynamic information, our method can better analyze sudden traffic conditions and adapt to more complex traffic conditions.

(2) We construct a sequence-to-sequence (Seq2Seq) model structure, which stacks several spatial-temporal layers. Compared with the single spatial-temporal layer, our method can mine deeper spatial-temporal correlations. It can effectively avoid the shortcoming that the correlation mining will become weak due to the long intervals.

(3) We conduct experiments on three real datasets, PEMSO4, PEMSO8, and METR-LA. The results show that our model has better prediction performance compared to other benchmark models.

II. RELATED WORK

The problem of traffic flow prediction is to model the traffic status of the urban road network or highway to predict the future traffic situation [14]. Existing traffic flow prediction methods are mainly divided into two categories: model-driven methods and data-driven methods [15].

A. MODEL-DRIVEN METHODS

The model-driven method is also known as the parameter method. The main representative models include the differential autoregressive moving average model (ARIMA) [16], the vector support autoregression model (SVR) [17], the gray prediction model (GM(1,m)) [18], etc. These models usually have strict assumptions and fixed conditions. Therefore, parametric methods cannot mine the nonlinear characteristics of traffic flow, resulting in generally low prediction accuracy. Data-driven methods are further divided into traditional machine-learning methods and deep-learning methods. Classical representatives of the former are support vector machines [19], [20], k-nearest neighbors [21], Bayesian networks [22], MLP [23], and so on. The latter are mainly artificial neural networks and their variants [24]. These models have strong nonlinear mapping ability, and the data requirements are not as strict as model-driven methods. Therefore, they can better adapt to the uncertainty of traffic flow and effectively improve the prediction effect.

B. DATA-DRIVEN METHODS

In recent years, to fully extract spatial-temporal characteristics of traffic flow, deep neural network models with high-dimensional data processing capabilities and nonlinear data feature mining efficiency have been favored by more and more researchers. Since traffic flow forecasting is a typical time series forecast with significant time correlation, LSTM and GRU have been widely used to deal with the temporal characteristics of traffic flow. However, the model based on time series cannot completely cover all the factors of traffic flow prediction. The traffic flow in each area of the urban road network is not only affected by the traffic flow in the previous period, but also by the congestion of adjacent roads. The time series model can only extract the traffic flow information of a single area, and it is far from enough to rely on the information of a single area to predict the traffic flow. Therefore, the Convolutional Neural Network (CNN) [25], which can handle the spatial structure of roads, is also considered. For example, Ma et al. [26] convert traffic flow into images and utilize CNN to extract traffic features in adjacent regions. Meanwhile, to comprehensively consider the spatial-temporal dependencies of traffic data, many researchers combine CNN and LSTM models for traffic flow prediction or traffic speed prediction, such as ConvLSTM [27], DMVST-Net [28], CLTFP [29], and so on. However, these CNN-based methods [30] are not suitable for non-Euclidean space, so they cannot extract features in the traffic network with complex structures.

Since the spatial structure of the road network is in non-Euclidean space, the classic convolutional neural network cannot be used to extract the spatial features between adjacent observation points. Therefore, Graph Convolutional Networks (GCN) [31], [33] have received extensive attention, and many studies have proposed GCN-based traffic prediction models [34], [39]. For example, Li et al. [40] proposed Diffusion Convolutional Gated Recurrent Unit (DCGRU) to improve the gating of GRU to capture the spatial-temporal dependence of traffic data, and combined encoder-decoder to propose a DCRNN model of Seq2Seq. Guo et al. [41] used three different spatial-temporal components to extract the information from historical data, integrated the graph structure of the traffic network and the dynamic spatial-temporal pattern of the traffic data to characterize the spatial-temporal correlation between neighbor nodes and predicted nodes, and proposed an attention mechanism-based method named Spatial-temporal Graph Convolutional Neural Network (ASTGCN). Song et al. [42] used three consecutive time slices to construct a local spatial-temporal graph, while using a sliding window to segment different periods, stacking multiple graph convolutional layers to form a spatial-temporal synchronized graph volume. Wu et al. [43] used an adaptive adjacency matrix to capture hidden spatial dependencies and combined dilated causal convolution for modeling. Zhang et al. [44] proposed the Structure Learning Convolution Neural Networks (SLCNN) which consider the local and global spatial correlation and used the 3D convolutions to capture temporal dependencies. Overall, GCN shows good prediction performance on the non-Euclidean datasets.

C. DIFFERENCES BETWEEN SDSCNN AND EXISTING WORKS

Although the above methods take into account the spatial and temporal dependencies relatively well, there are certain drawbacks in dealing with complex spatiotemporal correlations.
Yan et al. [45] improved the transformer structure by introducing a global decoder and a global-local decoder respectively. The multi-head attention is used to extract non-local features, and the masked multi-head attention is focused on extracting local features. In addition, temporal correlation analysis is performed using LSTM. However, the model only considered the dynamic characteristics, and the static correlation of traffic flow is ignored. Also, the computational complexity of the LSTM is high. Wu et al. [46] also proposed a multi-attention mechanism model to predict traffic flow. But the paper focused on emphasizing data integrity and verified that initial interpolation affects the stability of model performance, especially for complex missing scenarios with large-scale data.

For the temporal correlation, both convolutional and RNN-based time series models have shortcomings such as the weak ability to capture long-range connections and the inability to capture the periodicity of traffic information. For spatial correlation, GCN-based deep learning models cannot assign different weights according to node importance and cannot be applied to directed graphs. Furthermore, GCN uses a static Laplacian matrix as a filter and does not take into account dynamic spatial dependencies, i.e., situations where spatial weights change over time, such as traffic jams, traffic accidents, and one-way lanes that only appear during peak hours, and so on. To capture complex spatial-temporal associations, we propose Hybrid Model Integrating Static and Dynamic Spatial Correlation Neural Networks (SDSCNN).

The SDSCNN model is based on the idea of SLCNN, which considers global and local spatial graphs, including static and dynamic spatial dependencies. Different from SLCNN, SDSCNN extensively uses the attention mechanism to obtain different features in space and time. Specifically, we construct spatial-temporal graphs based on static road networks and dynamic correlations, use attention to extract global and local spatial dependencies, and assign different weights to them. Meanwhile, to capture the periodicity of traffic information, we design a multi-head self-attention mechanism based on Seq2Seq temporal module, which can perform computations at different periods to extract temporal dependencies. Finally, we stack the two components to aggregate spatial-temporal correlations for prediction.

### III. METHODOLOGY

#### A. PROBLEM DEFINITION

Given a graph $G = (V, E)$ with $N$ nodes, where $V$ and $E$ are the set of nodes and edges respectively, the historical traffic data embedded on graph $G$ with $M$ input channels and $T$ time intervals can be denoted as $X \in \mathbb{R}^{N \times M \times T}$. The task of traffic forecasting is to learn a mapping function $f(\cdot)$, which takes historical traffic data $X$ and graph $G$ as inputs to forecast future $K$ time intervals of traffic data:

$$\hat{X} = f(G, X)$$  \hspace{1cm} (1)

where $\hat{X} \in \mathbb{R}^{N \times M \times K}$ is the prediction value.

### B. SDSCNN

Many works use ChebNet or GCN to analyze the spatial correlation. However, whether it is ChebNet or GCN, there are three shortcomings:

1. From the history of GCN-related research, GCN and its original model ChebNet are both models of the Spectral domain, which uses graph theory to perform convolution operations on topological graphs. Convolution operations on topological graphs (graph Fourier transform and graph convolution) require the use of Laplacian matrices. The Laplace matrix of an undirected graph is a positive semi-definite matrix. The spectral decomposition of the positive semi-definite matrix can be decomposed into $N$ linearly independent eigenvectors, to carry out the subsequent Graph Fourier transformation and convolution, and finally obtain the first-order derivation formula of GCN. However, the Laplacian matrix of a directed graph is not a positive semi-definite matrix, so spectral decomposition is not possible. Theoretically, the derivation formula of GCN cannot be obtained.

2. GCN cannot handle dynamic graphs. GCN relies on the specific graph structure during training, and it is also performed on the same graph during testing. It is a full graph calculation method. GCN can only handle transductive learning tasks and cannot be applied to inductive tasks.

3. GCN cannot assign different weights to each neighbor. GCN treats all neighbor nodes uniformly during convolution, and cannot assign different weights according to the importance of nodes.

Based on the above issues, GAT [47], [48] is proposed. There are at least three benefits of using GAT rather than using GCNs.

1. GAT is a spatial domain model that does not require the Laplacian matrix to join operations. Therefore, it is more suitable for directed graphs.

2. The operation mode of GAT is node-wise operation, instead of GCN using the Laplacian matrix for convolution. Therefore, it can be used for dynamic graphs.

3. The spatial weights of GAT are computed dynamically, instead of GCN utilizing the Laplacian matrix. Therefore, GAT allows for assigning different weights to nodes of the same neighborhood.

We introduce GAT to analyze the temporal and spatial correlation respectively, which is used to construct the SDSCNN model. The model includes four parts, static spatial module, dynamic spatial module, spatial fusion module, and temporal module. The specific description is as below.

1) STATIC SPATIAL MODULE

GAT introduces a self-attention mechanism in the propagation process, and the hidden state of each node is calculated by paying attention to its neighbor nodes. The GAT network is implemented by stacking a simple graph attention layer. For each attention layer to nodes, the attention coefficient
calculation method is as shown in equation (2):

$$\alpha_{ij} = \frac{\exp \left( \text{LeakyReLU} \left( a^T \left[ Wh_i || Wh_j \right] \right) \right)}{\sum_{k \in N_i} \exp \left( \text{LeakyReLU} \left( a^T \left[ Wh_i || Wh_k \right] \right) \right)}$$  (2)

Among them, \(\alpha_{ij}\) is the attention coefficient from node \(i\) to \(j\), and \(N_i\) represents the neighbor node of node \(i\). The input feature of the node is \(h = \{h_1, h_2, \ldots, h_N\}, h_i \in \mathbb{R}^d\), where \(N, F\) represents the number of nodes and the feature dimension respectively, \(h\) is the feature value of the nodes. \(W \in \mathbb{R}^{d\times F}\) is the linear transformation weight matrix applied on each node, and \(a \in \mathbb{R}^{2d\times F}\) is the weight vector, which can map the input to \(R\). Finally, softmax is used for normalization, and LeakyReLU is added for activation.

The feature output of the final node is obtained by equation (3):

$$h'_i = \sigma \left( \sum_{j \in N_i} \alpha_{ij} Wh_j \right)$$  (3)

Among them, \(\alpha_{ij}\) is the attention coefficient of nodes \(i\) to \(j\) calculated in equation (2), \(W\) is the weight matrix to be trained, \(h_j\) is the feature of the neighbor node \(j\) of node \(i\), and \(h'_i\) is the node \(i\) calculated by the attention mechanism output.

In addition, GAT utilizes multi-head self-attention [49], [50] to stabilize the learning process. It applies an independent attention mechanism to calculate the hidden state. For the results of the self-attention layers, this method concatenates the output and feeds into the fully connected layer to get the final result. The process is shown in Figure 2, equation (4) is the specific calculation process of multi-head attention:

$$h'_i = ||_{k=1}^{K} \sigma \left( \sum_{j \in N_i} \alpha_{ij}^k W^k h_j \right)$$  (4)

where \(\alpha_{ij}^k\) is the normalized attention coefficient of the first attention head and || represents the concatenation operation.

![Figure 2. Illustration of multi-head self-attention. K is the number of the head. The output of the three self-attention layers is concatenated, and the spatial correlation analysis is completed by the Fully Connected layer.](image)

Our model introduces the multi-head self-attention method to mine the correlation. Static spatial correlation analysis depends on the original road topology. Generally existing work uses a 0-1 matrix to represent the road structure. However, it cannot accurately reflect the static spatial structure of the road network. Therefore, this paper uses the adjacency matrix based on the distance from the road network, which not only better represents the actual spatial structure of the road network, but also reduces the computational complexity of the model. And the product of adjacency matrix and parameter matrix is used as the input of GAT. The static adjacency matrix calculation formulas are as follows:

$$\tilde{A}_{ij} = \begin{cases} \exp \left( \frac{\text{dist}(v_i, v_j)^2}{\sigma^2} \right), & \text{dist}(v_i, v_j) < d^s \\ 0, & \text{dist}(v_i, v_j) \geq d^s \end{cases}$$  (5)

where the adjacency matrix \(\tilde{A} \in \mathbb{R}^{N \times N}\) is calculated using a threshold Gaussian kernel [51] and the road distance between each node of the road network. \(\text{dist}(v_i, v_j)\) represents the distance between sensor \(v_i\) and sensor \(v_j\), \(\sigma^2\) is the variance of the road distance, and \(d^s\) is the threshold. Since the actual road network is generally a directed graph, \(\text{dist}(v_i, v_j)\) and \(\text{dist}(v_j, v_i)\) are inconsistent in some cases, so \(\tilde{A}\) is an asymmetric matrix. \(d^s\) is the maximum threshold of road network distance, and different datasets have different values.

The feature value \(X\) and the matrix \(W^s\) are input into equation (4) to obtain the static spatial correlation analysis result. The calculation rule is as follows.

$$X^s = ||_{k=1}^{K} \sigma \left( \sum_{j \in N_i} \alpha_{ij}^k W^k X \right)$$  (6)

$$W^s = \tilde{A} \theta^s$$  (7)

where \(X^s\) is the output of static spatial correlation analysis. \(\theta^s \in \mathbb{R}^{N \times N}\) is a learnable parameter. Sometimes, the adjacency matrix \(\tilde{A}\) of the road distance cannot be directly used in the traffic flow prediction task. We consider the impact of road construction, natural disasters, etc., and use the parameter \(\theta^s\) to get the matrix \(W^s\). \(\alpha_{ij}^k\) is calculated by equation (2), and keeps fixed during the calculation. It represents the weights of the edges trained by the GAT. The product \(\alpha_{ij}^k\) and \(W^s\) determines the relationship of each node in the static spatial correlation module. The edges and the associated weights of the nodes in this model are static.

2) DYNAMIC SPATIAL MODULE

Dynamic spatial correlation often exists in the historical data. Similarly, we also use the multi-headed attention mechanism for dynamic spatial correlation mining. The difference is that the attention coefficients of each layer changes according to the input feature values.

Following the same idea as the static weight calculation, the dynamic part uses the fusion of two weight matrices to construct the dynamic weights. One is the matrix \(W^d\), and the other is the matrix \(\alpha_{ij}^d\) which is calculated by GAT.

The weight parameters depend on the current feature values and the correlation between nodes at different moments shows dynamic characteristics, so we introduce the dynamic weight training method to construct the \(W^d\) [44]. The optimal dynamic feature at the current moment is determined by combining the latest feature of the nodes and the weight matrix.
of the model which is self-trained in real-time. So, we first parameterize the matrix $W_\phi$. And then the matrix $W_\phi$ and the latest feature value $X$ are input into the heuristic function, the specific formula is as follows.

$$W^d = \phi (X, W_\phi) = X^T W_\phi X$$  \hspace{1cm} (8)$$

where $\phi (\cdot)$ is the heuristic function. $W^d$ is the output of dynamic spatial correlation analysis.

For the $\alpha^s_{ij}$, the latest values of the nodes are input into equation (2) to get the output.

Also, the dynamic spatial correlation is analyzed using the GAT which is calculated as follows.

$$X^d = \|\sum_{k=1}^{K} \sigma \left( \sum_{j \in N_i} \alpha^d_{ij} W^d X \right)\|$$  \hspace{1cm} (9)$$

where $\alpha^d_{ij}$ is calculated by equation (2), and the input is the latest features of the nodes. It represents the training parameters of the dynamic spatial module. $X^d$ is the output of dynamic spatial correlation analysis.

3) SPATIAL FUSION MODULE

Spatial correlation analysis includes static spatial correlation analysis and dynamic spatial correlation analysis. We add both of them to achieve the fusion of spatial correlation. Therefore, based on the above analysis, the spatial correlation result is as follows.

$$\tilde{X} = X^s + X^d$$  \hspace{1cm} (10)$$

where $\tilde{X}$ is the result of the spatial correlation.

4) TEMPORAL MODULE

To improve the model’s ability of mining temporal correlation, especially the periodicity, the multi-head self-attention mechanism is also used in this module. Since the self-attention mechanism is to calculate the attention at every time point and all time points, it is possible to directly calculate the dependency relationship regardless of the interval between time points. The model structure is shown in Figure 3 and the input of the model is $\tilde{X}$. The calculation rule is as below.

$$\tilde{X} = \|\sum_{k=1}^{K} \sigma \left( \sum_{j \in N_i} \alpha^s_{ij} W^s \tilde{X} \right)\|$$  \hspace{1cm} (11)$$

where $\alpha^s_{ij}$ is calculated by equation (2) and the input is the current node’s value. $\tilde{X}$ is the result of the temporal correlation. Based on multi-head self-attention, this paper constructs a temporal correlation analysis module of Seq2Seq, which can be extracted in different periods to predict information of different time intervals.

C. DETAILED IMPLEMENTATION

Based on the above analysis, SDSCNN is proposed. To mine the deeper spatial-temporal correlation, we stack several SDSCNN layers. Since the receptive field size varies from layer to layer, the physical meaning of each layer’s weight matrix varies from layer to layer. At the first level, a matrix element may correspond to the relationship between two roads. At a high level, matrix elements can represent road information within an area. Therefore, the calculation of the weight matrix varies from layer to layer, and SDSCNN learns different parameters at different layers to characterize the graph structure, which is more in line with the actual situation.

The training process of the SDSCNN is shown in Algorithm 1.

---

Algorithm 1 The SDSCNN Training Process

**Input:** Historical traffic dataset $X \in \mathbb{R}^{N \times M \times T}$, $Y \in \mathbb{R}^{N \times 1}$; Static adjacency matrix $A$; Number of training iterations $P$; Small batch sample size $B$

**Output:** Hybrid network $SDSCNN (X, W^s, \theta^s, W^d, \theta^d, b^s)$

1. Randomly initialize training parameters $W^s, \theta^s, W^d, \theta^d, b^s$
2. for $p \leftarrow 1$ to $P$
   3. for $b \leftarrow 1$ to $B$
      4. $W^s = A \theta^s$
      5. $\tilde{X}^s = \|\sum_{k=1}^{K} \sigma \left( \sum_{j \in N_i} \alpha^s_{ij} W^s X \right)\|$
      6. $W^d = X^T W_\phi X$
      7. $\tilde{X}^d = \|\sum_{k=1}^{K} \sigma \left( \sum_{j \in N_i} \alpha^d_{ij} W^d X \right)\|$
      8. $\tilde{X} = \tilde{X}^s + \tilde{X}^d$
      9. $\tilde{Y} = W^d \tilde{X} + b^d$
   10. end for
11. $\Delta W^s, \Delta \theta^s, \Delta W^d, \Delta \theta^d, \Delta b^s$ = BackPropagation($Y, \tilde{Y}$)
12. $W^s \leftarrow W^s + \Delta W^s, \theta^s \leftarrow \theta^s + \Delta \theta^s, \ldots$
13. end for
14. return $SDSCNN (X, W^s, \theta^s, W^d, \theta^d, b^s)$
For the static spatial module, the static attention coefficient $\alpha_{ij}^{s}$ is first calculated by GAT with equation (3). Then, we use equation (4) to obtain the static adjacency matrix and parameterize it the matrix $W^s$. We feed the input $X$, $W^s$, and $\alpha_{ij}^{s}$ into equation (7) to get the output of the static spatial module $X^s$.

For the dynamic spatial module, the dynamic attention coefficient $\alpha_{ij}^{d}$ is also calculated by equation (3). In this module, we directly parameterize the dynamic adjacency matrix $W_d$. The $W_d$ and the input $X$ are input into equation (8) to get matrix $W^d$. And then, the input $X$, $W^d$, and $\alpha_{ij}^{d}$ are input into the equation (9) to get the output of the dynamic spatial module $X^d$.

After the above step is complete, the $X^s$ and $X^d$ are summed to get the spatial module result $\tilde{X}$. Then, the is input to the multi-headed self-attention to analyze the temporal correlation with equation (11). And the output of the temporal module is used as the output of the first SDSCNN layer.

Next, several layers of SDSCNN are stacked according to the above steps, and the parameters of each layer are initialized independently. The output of each layer is used as input for the next layer. We choose the optimal numbers of the layer and the head number of the multi-head self-attention through extensive experiments, and the detailed values are shown in the following experiments section. Finally, the result of the last SDSCNN layer is input to the MLP layer to get the prediction value.

IV. EXPERIMENTS
To evaluate the effectiveness of the proposed model, a series of experiments have been conducted. They are organized into the following steps.

A. EXPERIMENTS SETTINGS
1) DATA COLLECTION
The SDSCNN in this paper is validated using three traffic datasets. The highway datasets PEMS04, PEMS08, and METR-LA in the Caltrans Performance Evaluation System (PeMS, https://pems.dot.ca.gov) are used for analysis.

PEMS04 is the traffic data collected from San Francisco Bay, which contains 3848 sensors on 29 roads. The PEMS dataset contains traffic information such as traffic flow, vehicle speed, and density. In addition to this, the road distance between the sensors is also included. Among them, 307 sensors are selected for prediction. The period of this dataset is from January 1, 2018, to February 28, 2018, with a total of 59 days of data.

PEMS08 is the traffic data of San Bernardino for 62 days from July 1, 2016, to August 31, 2016. It contains 1979 sensors on 8 roads, and 170 of them are selected for prediction.

METR-LA is traffic data for the Los Angeles Freeway from March 1, 2012, to June 27, 2012, with 207 sensors. METR-LA dataset mainly records vehicle speed information but also includes the GPS location of the sensors and the road distance between the sensors.

2) EVALUATION METRIC
This article uses three measurement functions to evaluate the prediction effect of the model, namely Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE).

\[
MAE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} |y_j^i - \hat{y}_j^i| 
\]  

(12)

\[
RMSE = \sqrt{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (y_j^i - \hat{y}_j^i)^2}
\]  

(13)

\[
MAPE = \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| \frac{y_j^i - \hat{y}_j^i}{y_j^i} \right| 
\]  

(14)

Among them, represent the actual and predicted traffic flow of the j-th road in the i-th sample, $M$ represents the number of samples, $N$ represents the number of roads, and $\bar{y}$ is the average value of the sample.

3) MODEL PARAMETER SETTINGS
This paper builds a new SDSCNN model based on the deep learning framework PyTorch and uses 2 NVIDIA TITAN 12GB GPUs for experiments. In the experiment, the mean absolute error (MAE) was used as the Loss function, and used Adam optimizer to optimize the model parameters. The optimizer training step size is set to 0.0001, the number of batches is set to 40, the maximum number of iterations is set to 500, and the early_stop is set to 10. Also, initialize the training parameters with a random function. In this paper, grid search is used to optimize some hyper-parameters. The recommended hyper-parameters are shown in Table 1. The $k$-fold cross-validation was selected as the dataset division method, and the hyper-parameter cross-validation in Table 1 was used to repeat 10 times, and the mean and standard deviation were used as the results.

| Type                     | Setting |
|--------------------------|---------|
| Time steps $(T)$         | 8       |
| Layers                   | 3       |
| Head of multi-head attention | 8,6,5  |

In addition, the adjacency matrix $\tilde{A} \in \mathbb{R}^{N \times N}$ in equation 6 uses $d^s$ as the maximum threshold of the road network distance, and different datasets have different values. Figure 4 shows the histograms of the road network distances for each dataset. According to the histograms, this paper sets the $d^s$ of PEMS04, PEMS08, and METR-LA to 1500, 1000, and 12000, respectively.

B. METHODS FOR COMPARISON
We first compare the proposed SDSCNN method with two model-driven methods, including ARIMA [16]and SVR [17].
TABLE 2. Performance comparison of traffic flow prediction of different models.

| Methods     | Datasets | MAE     | RMSE    | MAPE(%) |
|-------------|----------|---------|---------|---------|
|             | PEMS04   |         |         |         |
| ARIMA [16]  | MAE      | 32.11 ± 0.09 | 28.27 ± 0.01 | 24.38 ± 0.35 |
|             | RMSE     | 68.13 ± 0.16 | 44.56 ± 0.01 | 40.51 ± 0.37 |
|             | MAPE(%)  | 17.96 ± 0.03 | 19.20 ± 0.01 | 16.29 ± 0.38 |
|             | PEMS08   |         |         |         |
| ARIMA [16]  | MAE      | 24.04 ± 0.08 | 23.25 ± 0.01 | 19.07 ± 0.31 |
|             | RMSE     | 43.3 ± 0.12 | 36.16 ± 0.11 | 35.83 ± 0.39 |
|             | MAPE(%)  | 15.42 ± 0.07 | 14.64 ± 0.02 | 13.09 ± 0.32 |
|             | METR-LA  |         |         |         |
| ARIMA [16]  | MAE      | 4.01 ± 0.03 | 3.85 ± 0.05 | 3.77 ± 0.23 |
|             | RMSE     | 6.96 ± 0.06 | 6.57 ± 0.11 | 6.32 ± 0.29 |
|             | MAPE(%)  | 10.18 ± 0.03 | 9.54 ± 0.06 | 9.07 ± 0.20 |

We further compare the SDSCNN with five data-driven methods, i.e., 1) LSTM [7], 2) DCRNN [40], 3) Spatial-Temporal Graph Convolutional Network (STGCN) [41], 4) Graph WaveNet (GWN) [43] and 5) SLCNN [44].

ARIMA [16]: It is a classic time series data prediction method. In the experiments, we first check the data stability and deal with non-stationary data. We input the data into the ARIMA model and check the model parameters. After the model parameters are checked for reasonableness, the prediction is performed and the results will be output.

SVR [17]: This method is an application of the support vector machine (SVM) in regression tasks. In the experiments, we choose the linear function as the kernel function. And the regularization coefficient is set to 1.

LSTM [7]: This method is a variant of the recurrent neural network. It only considers the temporal correlation. The LSTM uses the gating mechanism, including input gates, output gates, and forgetting gates. We input the traffic flow data into the LSTM to obtain the prediction results.

DCRNN [40]: The DCRNN model represents the road network as a weighted directed graph. The model first performs a diffusion convolution operation of the graph signal and the filter to map PD features to QD features which mines the spatial correlation of traffic flow. Meanwhile, the output is used as the input of GRU to further analyze the temporal correlation. To perform better multi-step prediction, a Seq2Seq model structure is built by diffusion convolution and GRU network. We predict by inputting the traffic flow data into the encoder and initializing the decoder with its final state.

STGCN [41]: STGCN consists of several spatial-temporal convolution modules. Each module contains two gated sequence convolution layers and one spatial graph convolution module. Gated sequence convolution is used to capture temporal correlation and consists of a 1D convolution and a GLU. Compared with the RNN series models, this method reduces the computational complexity and simplifies the model training. The spatial correlation is analyzed by Chebyshev graph convolution.

GWN [43]: The GWN model stack multiple spatial-temporal layers. A spatial-temporal layer consists of a GCN and a Gated TCN. Each layer contains residual connections. The adaptive adjacency matrix is employed in GCN. Meanwhile, the diffusion convolution is extended to graph convolution to represent diffusion graph convolution, which complements the spatial structure of road network to a certain degree. GWN uses causal convolution and dilated convolution to convert all inputs into one-dimensional vectors according to the time dimension, and performs 1D convolution operations.

SLCNN [44]: SLCNN is mainly used to address the problem of analyzing dynamic spatial correlation. SLCNN consists of multiple SLCNN layers. Each SLCNN layer includes a Global SLC module and a Local SLC module. The Global SLC module combines Chebyshev graph convolution with static adjacency matrix and dynamic adjacency matrix, respectively, to achieve global spatial correlation analysis. The Local SLC module only considers the spatial correlation between the k-nearest neighbor nodes. Both modules use P3D-SLC networks for the analysis of temporal correlation.

C. RESULTS OF TRAFFIC PREDICTION

The experimental results are shown in Table 2. In the PEMS08 dataset, the SDSCNN model is always better than other benchmark models in accuracy. In the PEMS04 dataset, SDSCNN has the smallest MAE and MAPE, and the RMSE is slightly larger than SLCNN. Taking MAPE as the evaluation index, SDSCNN is 6.52% higher than DCRNN, 1.78% higher.
than STGCN, 17.11% higher than GWN, and 4.02% higher than SLCNN.

Meanwhile, we visualize and compare the prediction results of the SLCNN model with that of the SDSCNN model, as shown in Figure 5. The fit of the SDSCNN model is better than SLCNN. Although there is a certain regularity in the prediction results of SLCNN, there is still a certain distance from the accuracy of the SDSCNN model. This proves that the SDSCNN model adopts the multi-head attention mechanism to replace the 1D convolution of SLCNN and obtains a certain performance improvement. In addition, SDSCNN shows stronger adaptability when there are sudden and large changes in the traffic flow state. This shows that compared with the global and local graph structures of SLCNN, SDSCNN can better capture the dynamic transformation of the traffic flow by combining static and dynamic graph structures.

V. DISCUSSION
A. SPATIAL CORRELATION ANALYSIS
SDSCNN converts the distances between sensors into static spatial correlations, as shown in Figure 6 (a). From the perspective of road network structure, the structure can distinguish adjacent road segments and non-adjacent road segments, to provide a reference for prediction. Figures 6 (b), (c), and (d) represent the dynamic spatial correlation coefficients trained from the first layer of the SCSCNN Layer to the last layer.

We define HCN as the number of correlation coefficients greater than 0.5 in the correlation matrix. From Figure 6, we can be found that the correlation in Figure 6 (a) is not similar to that in Figures 6 (b), (c), (d), which indicates that the real road network structure cannot fully express the spatial correlation, we can calculate the previous Traffic flow further improves the model’s ability to capture spatial correlations.

The density of strong correlation coefficients is not as large as shown in Figure 6 (a). When the information aggregated by the previous hidden layer of SDSCNN is used as the input of the next hidden layer, the next hidden layer can obtain more information about non-adjacent nodes. Therefore, the density of strong correlation coefficients in Figures 6 (c), (d) is higher than that in Figure 6 (b).

B. MULTI STEP PREDICTION ANALYSIS
Table. 3 and Figure 7 show the change in the prediction performance of various methods as the prediction output step increases. In general, as the prediction output step gets longer, the corresponding difficulty of the prediction gets greater, so the prediction error also increases. As can be seen from the images, methods that only consider temporal correlation can achieve good results in short-term prediction, such as ARIMA, SVR, and LSTM. However, their forecasting performance dropped sharply as the forecast output step increased. In contrast, graph-based methods degrade slowly. This is mainly because they take into account the traffic flow effects of adjacent areas and the previous period. The prediction errors of graph-based methods increase slowly with the output step, and they all perform well overall.

SDSCNN model achieves the best prediction performance in almost every scenario, except the PEMS08 dataset. We carefully analyzed the PEMS08 dataset and found that the dataset is noisy with some extreme values. The RMSE is more sensitive to noise data, so the error rises slightly when the prediction output step is 5. Moreover, compared with the METR-LA dataset, the PEMS series data are more volatile and have many sudden flow changes, which makes the prediction task more arduous, and as a result, their RMSE values will be larger.

Especially, in multi-step prediction tasks, the difference between ASTGCN and other baselines is more significant, indicating that the strategy combining graph and attention mechanism can better mine the dynamic spatial-temporal patterns of traffic data.
C. HYPER-PARAMETER OPTIMIZATION

We also compare the performance of SDSCNN with different hyper-parameters, including the influence of the time step, the number of layers of SDSCNN, and the number of heads of the multi-head self-attention mechanism on MAPE. The experimental results are shown in Table 4 and Table 5.

From Table 4, we can find that with the increase of the time step and the increase of the number of SDSCNN layers, the MAPE decreases first and then increases. For different datasets, the optimal hyper-parameters of SDSCNN are different. As can be seen in Table 4, the PEMS04 and PEMS08 datasets perform well when the time step is 8 and the number of SDSCNN layers is 3. However, in the METR-LA dataset, our method achieves the best results when the time step is 7 and the number of SDSCNN layers is 2. This may be caused by the inconsistency between the sensor types of METR-LA and the PEMS04 and PEMS08 datasets. In Table 5, as the number of heads of the multi-head self-attention mechanism increases, the MAPE tends to decrease slightly. When the number of heads is greater than 5, the accuracy of the model no longer improves. This result is reasonable because the multi-head self-attention mechanism will increase the parameter and increase the risk of overfitting problems.
D. ABLATION EXPERIMENT

To further analyze the performance of our model, we added the other two variants under different combinations of modules. These two variants are compared with the SLCNN model and the SDSCNN model in the PEM504 dataset, and the four models differ in:

1. SLCNN (GL+1D): the basic model, which uses global and local modules for spatial convolution, and uses 1D convolution to extract temporal dependencies.

2. SD+1D: The spatial correlation analysis method of this model is the same as the method of our model. And the temporal module uses the 1D CNN.

3. GL+ATT: The temporal module of this variant is designed based on a multi-head attention mechanism rather than 1D CNN.

4. SDSCNN (SD+ATT): The SDSCNN model deploys a multi-layer spatial module and a temporal module based on a multi-head attention mechanism for prediction.

As shown in Figure 8, the model with static and dynamic spatial modules (SD) achieves a relatively better prediction performance than the model with global and local modules (GL). Compared with the 1D convolutional temporal module, In the case where the spatial component is (GL), the multi-head attention mechanism module (ATT)-based model achieves 3.15%, 1.1%, and 3.78% improvement in MAE, RMSE, and MAPE, respectively. With the spatial component being (SD), the multi-head attention mechanism module (ATT)-based model achieves 2.86%, 0.91%, and 2.59% improvement in MAE, RMSE, and MAPE, respectively.
For the task of road network prediction, previous research either started with adjacency matrix and established a static spatial matrix to represent the node relationship, or used training parameters to try to mine the node relationship in the data. In this paper, static spatial matrix and dynamic spatial matrix are constructed respectively. The static spatial matrix is constructed by the static adjacency matrix calculated from the distance of the road network; the dynamic spatial matrix is determined by combining the historical data characteristics of the nodes and the weight matrix of the real-time self-training model. Use the static spatial matrix and the dynamic spatial matrix to perform node aggregation operations respectively, and finally perform fusion weighting and input them into the temporal module. Ablation experiments show that the spatial component (SD) proposed in this paper can improve the accuracy of the model.

Additionally, traffic information (vehicle flow, speed, etc.) has obvious periodicity, and the 1D convolution components used in the original model cannot extract the periodicity. Moreover, the multi-head self-attention mechanism can mine periods, and there is no disadvantage that the RNN-like model has a weak ability to capture long-term dependencies. Therefore, the multi-head attention mechanism can significantly improve the performance of the SDSCNN model.

E. LIMITATIONS AND FUTURE WORK

Although our model has achieved good results in traffic prediction, there are still some limitations to consider in this study. Firstly, we ignored the multi-modality feature of the input data, such as heterogeneous datasets of weather, car accidents, and unexpected situations. The model prediction performance may be affected by the relatively homogeneous characteristics of the dataset. Secondly, we directly employ the summation operation for the fusion of static and dynamic spatial correlation analysis results. This is a relatively simple fusion method. To further improve the model prediction performance, we can study the automatic learning of these two component weights by the model in the future. Furthermore, we can introduce the traffic sensing method to improve the data quality to enhance the prediction accuracy [52], [53]. The biased estimated traffic speed data can be corrected by Wi-Fi and Bluetooth passive sensing technology. For some missing values, the data can be calculated by detecting the time differences. Thirdly, this paper does not consider the application of real-world scenarios such as the compressed transmission of model datasets, and abnormal data removal in the Internet of Vehicles (IOV) [54], [55]. Using the model for the Internet of Vehicles will also be a direction of our future work.

VI. CONCLUSION

In this paper, we propose a hybrid model that integrates static and dynamic spatial correlation neural network for traffic flow prediction. Specifically, in our model, the graph attention network is used to construct static and dynamic modules based on the correlation of distance and traffic information of the road network. Then, the periodicity and temporal correlation of traffic flow are captured by using the multi-head self-attention mechanism. Finally, multiple spatial-temporal layers are stacked for prediction. The implementation with three public traffic flow datasets shows that our SDSCNN outperforms several state-of-the-art methods.

REFERENCES

[1] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Goliás, “Short-term traffic forecasting: Where we are and where we’re going,” Transp. Res. C, Emerg. Technol., vol. 43, pp. 3–19, Jun. 2014, doi: 10.1016/j.trc.2014.01.005.
[2] H. Yuan and G. Li, “A survey of traffic prediction: From spatio-temporal data to intelligent transportation,” Data Sci. Eng., vol. 6, no. 1, pp. 63–85, Mar. 2021, doi: 10.1007/s41019-020-00151-z.
[3] D. Pavlyuk, “Feature selection and extraction in spatiotemporal traffic forecasting: A systematic literature review,” Eur. Transp. Res. Rev., vol. 11, no. 1, Jan. 2019, doi: 10.1186/s12544-019-0345-9.
[4] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, “Long short-term memory neural network for traffic speed prediction using remote microwave sensor data,” Transp. Res. C, Emerg. Technol., vol. 54, pp. 187–197, May 2015, doi: 10.1016/j.trc.2015.03.014.
[5] Y. Gu, W. Lu, L. Qin, M. Li, and Z. Shao, “Short-term prediction of lane-level traffic speeds: A fusion deep learning model,” Transp. Res. C, Emerg. Technol., vol. 106, pp. 1–16, Sep. 2019, doi: 10.1016/j.trc.2019.07.003.
[6] Y. Zhang, “Short-term traffic flow prediction methods: A survey,” in Proc. J. Phys., Conf., Apr. 2020, vol. 1486, no. 5, Art. no. 052018, doi: 10.1088/1742-6596/1486/5/052018.
[7] L. Zhao, Y. Song, C. Zhang, and Y. Liu, “T-GCN: A temporal graph convolutional network for traffic prediction,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 9, pp. 3848–3858, Sep. 2020, doi: 10.1109/TITS.2019.2935152.
[8] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, “Temporal graph networks for deep learning on dynamic graphs,” 2020, arXiv:2006.10637.
[9] B. Chamberlain, J. Rowbottom, M. I. Gorinova, M. Bronstein, S. Webb, and E. Rossi, “GRAND: Graph neural diffusion,” in Proc. 36th Int. Conf. Mach. Learn., 2021, pp. 1407–1418.
[10] Y. Zhang, S. Gao, J. Pei, and H. Huang, “Improving social network embedding via new second-order continuous graph neural networks,” in Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining, New York, NY, USA, Aug. 2022, pp. 2515–2523, doi: 10.1145/3534468.3537945.
[11] B. Yu, Y. Lee, and K. Sohn, “Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN),” Transp. Res. C, Emerg. Technol., vol. 114, pp. 189–204, May 2020, doi: 10.1016/j.trc.2020.02.013.
X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, “Learning traffic as graphs on graphs with fast localized spectral filtering,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9, Accessed: Jul. 19, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2016/hash/0a44343d481c3b307d32e6b6fdec65-Abstract.html

J. Zhu, Q. Wang, C. Tao, H. Deng, L. Zhu, and H. Li, “AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting,” *IEEE Access*, vol. 9, pp. 35973–35983, 2021, doi: 10.1109/ACCESS.2021.3062114.

S. Guo, Y. Guo, P. Zhao, C. Zheng, and X. Chen, “A graph-based temporal attention framework for multi-sensor traffic flow forecasting,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7743–7758, Jul. 2022, doi: 10.1109/TITS.2021.3072118.

S. Feng, J. Ke, H. Yang, and J. Ye, “Multi-task model for traffic prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5704–5716, Jun. 2022, doi: 10.1109/TITS.2021.3056415.

J. Zhu, Q. Wang, C. Tao, H. Deng, L. Zhu, and H. Li, “AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting,” *IEEE Access*, vol. 9, pp. 35973–35983, 2021, doi: 10.1109/ACCESS.2021.3062114.

S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, “Attention based spatial–temporal graph convolutional networks for traffic flow forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 1–8, doi: 10.1609/aaai.v33i01.3301922.

C. Song, Y. Lin, S. Guo, and H. Wan, “Spatial–temporal synchronous graph convolutional networks: A new framework for spatial–temporal network data forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 1, pp. 1–8, doi: 10.1609/aaai.v34i01.5438.

Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph WaveNet for deep spatial–temporal graph modeling,” 2019, arXiv:1906.00121.

Q. Zhang, J. Chang, G. Meng, S. Xiang, and C. Pan, “Spatio-temporal graph structure learning for traffic forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, Apr. 2020, no. 1, pp. 1–9, doi: 10.1609/aaai.v34i01.54015470.

H. Yan, X. Ma, and Z. Pu, “Learning dynamic and hierarchical traffic spatiotemporal features with transformer,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22386–22399, Nov. 2022, doi: 10.1109/TITS.2021.3102983.

M. Fang, M. Xu, J. Fang, and X. Wu, “A multi-attention tensor completion network for spatiotemporal traffic data imputation,” *Internet Things J.*, vol. 9, no. 20, pp. 20203–20213, Oct. 2022, doi: 10.1109/JIOT.2022.3171780.

D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013, doi: 10.1109/MSP.2012.2335192.

G. Li, V. L. Knoop, and H. van Lint, “Multistep traffic forecasting by dynamic graph convolution: Interpretations of real-time spatial correlations,” *Transp. Res. Part C: Emerg. Technol.*, vol. 128, Jul. 2021, Art. no. 103185, doi: 10.1016/j.trc.2021.103185.

A. Vyasani, N. Shazeer, and N. Parmar, “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11, Accessed: Jul. 19, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee23435d4ee1f8b0d53c1c4a854a-Abstract.html.

S. Reza, M. C. Ferreira, J. J. Machado, and J. M. R. S. Tavares, “A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks,” *Expert Syst. Appl.*, vol. 202, Sep. 2022, Art no. 117275, doi: 10.1016/j.eswa.2022.117275.

Q. Li, Z. Han, and X. M. Wu, (2018). *Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning*. Accessed: Jul. 19, 2022. [Online]. Available: http://www.xuefushan.com/publication/2784814091
Z. Pu, Z. Cui, J. Tang, S. Wang, and Y. Wang, “Multimodal traffic speed monitoring: A real-time system based on passive Wi-Fi and Bluetooth sensing technology,” *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12413–12424, Jul. 2022, doi: 10.1109/JIOT.2021.3136031.

A. Haghani, M. Hamedi, K. F. Sadabadi, S. Young, and P. Tarnoff, “Data collection of freeway travel time ground truth with Bluetooth sensors,” *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2160, no. 1, pp. 60–68, Jan. 2010, doi: 10.3141/2160-07.

C. Chen, Z. Liu, S. Wan, J. Luan, and Q. Pei, “Traffic flow prediction based on deep learning in internet of vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3776–3789, Oct. 2021, doi: 10.1109/TITS.2020.3025856.

C. Chen, L. Liu, S. Wan, X. Hui, and Q. Pei, “Data dissemination for industry 4.0 applications in internet of vehicles based on short-term traffic prediction,” *ACM Trans. Internet Technol.*, vol. 22, no. 1, pp. 1–18, Feb. 2022.

**JUNMING DAI** was born in Jiangsu, China. He is currently pursuing the master’s degree with Nantong University. His research interests include intelligent transportation, graph neural networks, and time-series networks.

**JIASHUANG HUANG** received the Ph.D. degree from the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, in 2020. From 2018 to 2019, he was a Visiting Scholar at the University of Wollongong (UOW), Wollongong, NSW, Australia. He is currently a Lecturer with the School of Information Science and Technology, Nantong University. His recent research is to analyze the brain network by using machine learning methods. He has published more than 20 research peer-reviewed journals and conference papers, including *IEEE Transactions on Medical Imaging, IEEE Journal of Biomedical and Health Informatics, Medical Image Analysis, Bioinformatics, AAAI*, and *MICCAI*.

**QINQIN SHEN** received the Ph.D. degree from the School of Rail Transportation, Soochow University, in 2021. She is currently a Senior Experimentalist of the School of Transportation and Civil Engineering, Nantong University. Her research interests include intelligence transportation and numerical computation. She has published over ten papers in high level journals, including *Computational and Applied Mathematics, Computers and Mathematics with Applications*, and *Numerical Algorithms*.

**QUAN SHI** (Member, IEEE) received the M.S. and Ph.D. degrees in management information systems from the University of Shanghai for Science and Technology, Shanghai, China, in 2005 and 2011, respectively. He is currently a Professor with the School of Transportation, Nantong University. His research interests include the development of signal and image processing, intelligent information processing, and big data techniques for computer.

**SIYUN FENG** was born in 1995. She received the master’s degree. Her research interests include intelligent transportation technology, graph convolutional networks, deep learning, and medical systems.

**ZHENQUAN SHI** received the master’s degree from the School of Computer Science and Technology, University of Shanghai for Science and Technology, in 2009, and the Ph.D. degree in management information systems from the School of Management, University of Shanghai for Science and Technology, in 2021. Currently, he is working with the School of Transportation, Nantong University. He has published eight relevant papers in high level journals. His research interests include intelligent transportation and deep learning.

* * *