A correlation information-based spatiotemporal network for traffic flow forecasting

Weiguo Zhu¹,² · Yongqi Sun¹,² · Xintong Yi¹,² · Yan Wang¹,² · Zhen Liu¹,²

Received: 21 June 2022 / Accepted: 28 June 2023 / Published online: 3 August 2023
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2023

Abstract
Traffic flow forecasting technology plays an important role in intelligent transportation systems. Based on graph neural networks and attention mechanisms, most previous works utilize the transformer architecture to discover spatiotemporal dependencies and dynamic relationships. However, they have not thoroughly considered correlation information among spatiotemporal sequences. In this paper, based on the maximal information coefficient, we present two elaborate spatiotemporal representations, spatial correlation information (SCorr) and temporal correlation information (TCorr). Using SCorr, we propose a correlation information-based spatiotemporal network (CorrSTN) that includes a dynamic graph neural network component for integrating correlation information into spatial structure effectively and a multi-head attention component for modeling dynamic temporal dependencies accurately. Utilizing TCorr, we explore the correlation pattern among different periodic data to identify the most relevant data, and then design an efficient data selection scheme to further enhance model performance. The experimental results on the highway traffic flow (PEMS03, PEMS04, PEMS07 and PEMS08) and metro crowd flow (HZME inflow and outflow) datasets demonstrate that CorrSTN outperforms the state-of-the-art methods in terms of predictive performance. In particular, on the HZME (outflow) dataset, our model makes significant improvements compared with the ASTGNN model by 13.2%, 15.3% and 29.3% in the metrics of MAE, RMSE and MAPE, respectively.

Keywords Correlation information · Feature extraction · Attention mechanism · Graph neural network · Traffic forecasting

1 Introduction
Currently, many smart cities are making dramatic efforts to improve the performance of intelligent transportation systems (ITSs) [1, 2]. As one of the most fundamental and crucial techniques in smart city construction, traffic flow forecasting has become a hot research topic.

Traffic flow forecasting utilizes historical traffic data to predict future flow timestamps. Early works focusing on time series prediction have produced excellent results. Traditional methods, such as support vector regression (SVR) [3], support vector machines (SVMs) [4, 5] and k-nearest neighbors (KNN) [6, 7], have been extensively applied to traffic forecasting. However, these methods need to identify the data characteristics and ignore the spatial features.

With the rapid development of deep learning, deep neural networks have been used to extract spatiotemporal features for traffic forecasting. The convolutional neural...
network (CNN) is introduced into traffic forecasting tasks as an effective method to extract spatial features [8–11]. However, these methods are coarse-graining processes utilizing CNN to capture neighbor block attributes and extract nonlinear spatial dependencies by a grid representation.

Since grid representation cannot adequately represent the flow between sensors, graph representation is proposed to encode the elaborate relationships among sensors [12, 13]. In recent years, the graph neural network (GNN), as an efficient and effective method, has gradually become the essential traffic prediction module for graph representation. GNN-based methods concentrate on the relationships among adjacent sensors with the propagate-aggregate mode [12–17]. Furthermore, attention-based methods, such as [14–20], form the feature extraction processing as a query within queries (Q) and keys (K) and calculate attention weights as interrelationships for values (V).

Nevertheless, three key issues need to be given more attention. First, GNN-based methods cannot construct correct features with a sparse similarity matrix by the propagate-aggregate mode. Since the sparse similarity matrix is generated based on the spatial sensors and road, the neighboring sensors typically do not have a similar pattern. Researchers have made efforts to take mixed-hop propagation to explore deep neighborhoods [21], take the DTW algorithm to construct a graph matrix [16, 22], and take dynamic graph convolution to capture dynamic relationships [23, 24]. Even though these methods improve traffic forecasting accuracy, their performance could be further improved if they take into account more elaborate and density correlation information.

Second, the attention mechanism makes unwanted distractions during the attention weight calculation, and it cannot rectify the false activations induced by distractions of keys (K). As shown in Fig. 1, the most relevant sequence part distracts attention weights to the other parts with vanilla attention.

Third, for periodic data, it is challenging to select appropriate data as input for neural network-based methods. The traditional methods iterate over all possible schemes and choose the best scheme. However, these methods are not practicable for neural network-based methods due to the expensive computation cost.

In this paper, to solve the above issues, we first introduce a spatial correlation representation (SCorr) to elaborate reflect the relevant features among temporal sequences. Based on SCorr, a dynamic correlation information GNN (CIGNN) is designed to integrate the spatial graph network with SCorr and capture more comparable patterns from each sensor. In addition, to handle the unwanted distractions in the attention mechanism, we propose a correlation information multi-head attention mechanism (CIATT), which can stabilize contextual features by aggregating the top-k comparable patterns and concentrate attention weights on the most relevant sequence as matching templates (see Fig. 1). Then, for spatiotemporal features, we propose an effective correlation information-based spatiotemporal network for traffic flow forecasting (CorrSTN), which employs CIGNN to discover the dynamic spatial dependencies and CIATT to match the relevant temporal traffic patterns. Finally, to avoid the traditional exhaustive search for the data selection scheme, we build a temporal correlation information representation (TCorr) to mine the relevant sequence among different periodic data. By TCorr, we design an appropriate data selection scheme to further enhance model performance.

For traffic flow forecasting, we summarize three key contributions of our work as follows,

- We propose an elaborate and dense spatial correlation information representation to fully express the similarity among each sensor.
- In our model, the proposed CIGNN can construct correct features by associating the density similarity representation and predefined graph structure with the propagate-aggregate mode. Meanwhile, the CIATT is developed to stabilize contextual features and concentrate attention weights on the most relevant sequence.
- A temporal correlation information representation (TCorr) is proposed to mine the relevant sequence and seek the best data scheme. To the best of our knowledge, the efficient data selection scheme is the first used for neural network-based methods in traffic forecasting tasks.

The experimental results on four real-world datasets show that the predictive performance of CorrSTN outperforms the state-of-the-art model. The remainder of this paper is organized as follows. Section 2 gives a brief overview of the related work. Section 3 formulates the traffic flow
forecasting problem. Section 4 presents a detailed description of correlation information representations and the proposed model. Section 5 presents and analyzes the evaluation results. Section 6 provides detailed experiments about model settings. Section 7 gives the case studies. Section 8 summarizes our work.

2 Related work

2.1 Traditional traffic forecasting methods

The early works for traffic forecasting are based on machine learning methods. Drucker et al. [3] propose a linear support vector machine (SVR) to predict traffic flow data. Eric et al. [25] propose an advanced time series model based on vector autoregression (VAR) to capture the pairwise relationships among spatial sequences on traffic data. Hochreiter and Schmidhuber [26] design the long short-term memory (LSTM) network, a special recurrent neural network (RNN), to predict time series data and solve the vanishing gradient problem. With only temporal sequences considered and spatial relationships ignored, their prediction accuracy cannot satisfy the practical requirements.

2.2 Deep neural network traffic forecasting methods

The deep neural network has dramatically improved traffic prediction accuracy by combining CNN and RNN variants. Li et al. [13] propose a diffusion convolutional recurrent neural network (DCRNN) to employ a diffusion graph convolutional network and a gated recurrent unit (GRU) in seq2seq to predict traffic data. Yu et al. [12] design a spatial-temporal graph convolutional network (STGCN) architecture for spatiotemporal datasets in the traffic forecasting task. Wu et al. [27] propose Graph WaveNet (GWN), which combines a graph convolution network with a temporal convolution network to capture spatial-temporal dependencies.

Since the attention mechanism can effectively model the dependencies among sequences, many works utilize it in traffic flow forecasting tasks. Guo et al. [14] present an ASTGCN model that uses spatial and temporal attention mechanisms to improve prediction accuracy. Song et al. [15] propose a spatial-temporal synchronous graph convolution network (STSGCN) to extract temporal adjacency features by considering the local spatiotemporal relation. Recently, Guo et al. [17] propose a GNN-based model called ASTGNN formed as an encoder-decoder architecture [28] with residual connection [29] and layer normalization [30] to learn the dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting, which is an extension of their previous ASTGCN model. Its prediction accuracy is significantly improved by effectively capturing the local data trend and dynamically aggregating the spatial features.

Although these methods have yielded outstanding results, they do not take the crucial correlation information among spatiotemporal sequences into account.

2.3 Correlation information traffic forecasting methods

The GWN adopts an adaptive adjacency matrix as a supplement to the structural matrix by calculating the vector correlations during the training stage [27]. However, it cannot accurately reflect the correlation information by short-term sequences. Zheng et al. [23] design a graph multi-attention network (GMAN) integrating spatial and temporal attention to capture sensor correlations. Bai et al. [31] propose adaptive graph generation to dynamically generate the graph during training in AGCRN. Wu et al. [21] make use of the graph learning layer in the MTGNN to construct an adaptive graph by multivariate node features. Li and Zhu [16] introduce a spatial-temporal fusion graph matrix combining the correlation information with the structural network (STFGNN). However, since the matrix is very sparse, the correlation among the sensors is not elaborated for traffic forecasting. Fang et al. [22] combine the spatial and semantic neighbors to consider spatial correlations in STGDE. Han et al. [24] propose a dynamic graph constructor and graph convolution in the DMSTGCN to learn the dynamic spatial dependencies as an extensive predefined adjacency matrix.

However, their methods cannot extract correct features in the GNN-based component and avoid unwanted distractions in the attention-based component. Based on the correlation information, we aim to obtain more correct features and more focused attention weights in this paper.

3 Preliminaries

Definition 1 Traffic Network. We define a traffic network as a directed or an undirected graph \( G = (V, E) \), where \( V \) is a set of \( |V| = N \) nodes, each node represents a traffic sensor, and \( E \) is a set of edges.

Definition 2 Traffic spatiotemporal sequence. We define a traffic spatiotemporal sequence as \( X = (X^1, X^2, \ldots, X^T) \in \mathbb{R}^{T \times N \times C} \), where \( \mathbf{X}^t = (\mathbf{x}_{j}^{t}, \mathbf{x}_{k}^{t}, \ldots, \mathbf{x}_{N}^{t}) \in \mathbb{R}^{N \times C} \) denotes the vector of the \( N \) sensors with \( C \) attributes at timestamp \( t \).
Definition 3  Periodic Data. We define the hourly, daily and weekly data intervals as $T_h$, $T_d$ and $T_w$, respectively. Given time window $\tau$, the historical periodic data can be defined as

$$X = (X^{t-T_h+1}, X^{t-T_h+2}, \ldots, X^{t-T_h+\tau}),$$

$$X^{t-T_h+1}, X^{t-T_h+2}, \ldots, X^{t-T_h+\tau},$$

$$X^{t-T_h+1}, X^{t-T_h+2}, \ldots, X^{t-T_h+\tau},$$

where the time interval of each timestamp is 5 min on the datasets, and $\tau = 12$ in this paper.

Proposition 1  Given the historical periodic data $X \in \mathbb{R}^{T_{hde} \times N \times C}$ defined as Eq. (1), where $T_{hde} \in [\tau, 2\tau, 3\tau]$ will change according to our data selection scheme for different datasets. Then our focus is to predict traffic flow for all sensors over the next $L$ timestamps,

$$f(X) \rightarrow (X^{t+1}, X^{t+2}, \ldots, X^{t+L}) \in \mathbb{R}^{L \times N \times 1},$$

where $f(\cdot)$ is the mapping function aimed at learning and $L = 12$ in our model.

4 Methodology

In this section, we will introduce our spatiotemporal correlation information representations (SCorr and TCorr) and the correlation information-based components (CIGNN and CIATT). The overall framework of CorrSTN is based on an encoder-decoder architecture, as shown in Fig. 2. The encoder (decoder) network consists of temporal position embedding, spatial position embedding and encoder (decoder) layer components. The CIGNN and CIATT components are connected by the residual connection and layer normalization in each encoder and decoder layer. The second CIATT component of each decoder layer is designed to receive the encoder output as historical memory.

4.1 Spatiotemporal correlation information representation

In this subsection, we propose two elaborate spatiotemporal representations based on the maximal information coefficient [32], spatial correlation information (SCorr) and temporal correlation information (TCorr). To measure the degree of correlation information among sensor sequences, the maximal information coefficient (MIC) method is employed in the two spatiotemporal representations. Compared with DTW and cosine similarity methods, MIC can not only capture diverse associations but also execute fast calculations.

To explain the calculation process of the MIC in detail, we first define two sequences, $\var{var_1} \in \mathbb{R}^M$ and $\var{var_2} \in \mathbb{R}^M$. By partitioning the $x$-axis into $A$ parts and the $y$-axis into $B$ parts, the degree of correlation information between $\var{var_1}$ and $\var{var_2}$ can be calculated as follows:

$$\text{MIC} (\var{var_1}, \var{var_2}) = \frac{\max_{A, B < M} \{\var{I}_{A,B} (\var{var_1}, \var{var_2})\}}{\log_2 \min\{A, B\}},$$

where $\eta$ is a parameter to control the number of partitions and $\var{I}_{A,B}$ denotes the mutual information. $\var{I}_{A,B}$ is calculated as follows:

$$\var{I}_{A,B} = \sum_{a < A, b < B} q(a,b) \log_2 \left( \frac{q(a,b)}{q(a)q(b)} \right),$$

where $q(a,b)$ is the joint probability density and $q(a)$ and $q(b)$ are the edge probability densities when choosing the $(a, b)$ grids.

4.1.1 Spatial correlation information

In traffic forecasting tasks, the input data sequence shows interrelated spatial characteristics. In this subsection, we propose SCorr to represent the dependence among spatiotemporal sensor sequences.

Let $X_i^c$ denote the sensor $i$ vector with attribute $c$ of $T$ timestamps. Then, SCorr is defined as follows:

$$\text{SCorr} (X_i^c)_{i,j} = \text{MIC} (X_i^c, X_j^c),$$

where $\text{SCorr} (X) \in \mathbb{R}^{N \times N \times C}$ is the degree of correlation information, and $\text{SCorr} (X)_{i,j} \in [0, 1]$ denotes the degree of correlation information between sensor $i$ and sensor $j$ in attribute $c$.

Concretely, as shown in Fig. 3a, data sequences with $T$ timestamps are displayed. We select the upper and lower sensor sequences to calculate the degree of correlation information as one example. According to Eq. (4), we calculate the degree for each sensor pair, and then the SCorr matrices are shown in Fig. 3b.

Generally, each sensor has varying correlative relationships with other sensors in the traffic road or station network. The sparse adjacency matrix cannot represent it just by 1 or 0. In contrast, the SCorr matrices are elaborate and density spatiotemporal representations with values ranging from 0 to 1. High values signify strong correlative relationships and similar patterns between sensors, while low values express significant differences and low reference characteristics.

4.1.2 Temporal correlation information

It has been found that traffic flow data have different temporal associations among different temporal sequences.
In this subsection, we propose an effective representation, called TCorr, to explore and capture similar patterns among different periodic data.

We define $X_{\text{hourly}} \in \mathbb{R}^{T \times N \times C}$ (resp. $X_{\text{daily}}, X_{\text{weekly}}$) as the vector of the last hour (resp. day, week) before the predicted data $X \in \mathbb{R}^{T \times N \times C}$. Then, we define TCorr as follows,

$$T\text{Corr}(X) = \frac{1}{T} \sum_{t=1}^{T} \text{MIC}(X_{t}^{c,t+\tau}, X_{t}^{c,t+\tau}),$$

where $T\text{Corr}(X) \in \mathbb{R}^{N \times C}$ is the average degree of all sensors temporal correlation information, $T\text{Corr}(X) \in [0, 1]$ denotes the degree of temporal correlation information of sensor $i$ in attribute $c$, and $X_{t}^{c,t+\tau}$ denotes the vector of sensor $i$ in attribute $c$ between timestamp $t$ and

timestamp $t + \tau$ of $X_{\text{hourly}}, X_{\text{daily}}$ or $X_{\text{weekly}}$. We set a lower weight for timestamps shifted with a greater number of steps. Here, we set different weights for different periodic types of data as follows:

$$T\text{Corr}_h = \alpha T\text{Corr}(X_{\text{hourly}})$$

$$T\text{Corr}_d = \beta T\text{Corr}(X_{\text{daily}})$$

$$T\text{Corr}_w = \gamma T\text{Corr}(X_{\text{weekly}}),$$

(6)

where the weights $\alpha, \beta$ and $\gamma$ are set at 0.95, 0.95 and 0.85 according to the shifted steps of hour, day and week in this paper, respectively. We illustrate the TCorr calculation between the last hour (day, week) data and the predicted data, as shown in Fig. 4a.

The previous neural network-based methods take the data selection scheme as hyperparameters and exhaustively search for the appropriate scheme to improve the model performance. In the following, with the help of TCorr, we will design an efficient data selection scheme as follows.

**Data selection scheme.** First, we evaluate the contributions of different periodic data for the predicted data. With the help of TCorr, we define $\Delta_{\text{hourly}}, \Delta_{\text{daily}}$ and $\Delta_{\text{weekly}}$ to represent the contributions as follows:
Fig. 4 An illustration of the TCorr calculation

\[
\Delta_{hd}^{c} = \frac{1}{N} \sum_{i=1}^{N} T\text{Corr}_{d,i}^{c} - \frac{1}{N} \sum_{i=1}^{N} T\text{Corr}_{h,i}^{c},
\]

\[
\Delta_{hw}^{c} = \frac{1}{N} \sum_{i=1}^{N} T\text{Corr}_{w,i}^{c} - \frac{1}{N} \sum_{i=1}^{N} T\text{Corr}_{h,i}^{c},
\]

\[
\Delta_{dw}^{c} = \frac{1}{N} \sum_{i=1}^{N} T\text{Corr}_{w,i}^{c} - \frac{1}{N} \sum_{i=1}^{N} T\text{Corr}_{d,i}^{c},
\]

where \( \Delta_{hd}^{c} \) (resp. \( \Delta_{hw}^{c}, \Delta_{dw}^{c} \)) denotes the gap of contribution between hourly and daily (resp. hourly and weekly, daily and weekly) data with attribute \( c \). To clarify, we illustrate the TCorr of all sensors, as the scatter chart shows in Fig. 4b, while the \( \Delta_{hd}^{c}, \Delta_{hw}^{c} \) and \( \Delta_{dw}^{c} \) are notated on the average values, as the histogram shows in Fig. 4b.

Then, according to the data contribution, we are able to make the appropriate data selection scheme by the following three rules: i) to capture the short-term tendency, we set the hourly data as the basis of the input data; ii) the daily (weekly) data are combined into input data if \( \Delta_{hd}^{c} > 0 \) (\( \Delta_{hw}^{c} > 0 \)) to capture the long-term season; iii) according to Occam’s Razor principle, when \( \Delta_{hd}^{c} > 0 \) and \( \Delta_{hw}^{c} > 0 \), we select the periodic data to combine with input data as follows,

\[
data = \begin{cases} 
\text{daily and weekly data} & , \text{for } \Delta_{dw}^{c} > 0, \\
\text{daily or weekly data} & , \text{for } \Delta_{w}^{c} = 0, \\
\text{daily data} & , \text{for } \Delta_{dw}^{c} < 0.
\end{cases}
\]

where the Eq. \( \Delta_{w}^{c} > 0 \) denotes that data have weekly periodic properties. As shown in Fig. 4b, we can see that \( \Delta_{hd}^{c} > 0, \Delta_{hw}^{c} > 0 \) and \( \Delta_{dw}^{c} > 0 \). Thus, the appropriate scheme selects the hourly, daily and weekly data as input data for the example.

Based on our three rules and Eq. (8), we can search for an appropriate data selection scheme for datasets of various types. Thus, we can make use of the relevant sequence among different periodic data to further enhance model performance. Additionally, we will give more TCorr representations of different datasets and discuss different schemes in Sect. 6.1.

4.2 Correlation information graph neural network

In this subsection, we propose a correlation information graph neural network (CIGNN) incorporating the spatial correlation information into the structure network, as shown in Fig. 2b. The vanilla graph neural network is defined as follows:

\[
Z^{(l)} = \sigma \left( AZ^{(l-1)}W^{(l)} \right),
\]

where \( Z^{(l)} \) and \( Z^{(l-1)} \in \mathbb{R}^{N \times d_{model}} \), \( W^{(l)} \in \mathbb{R}^{d_{model} \times d_{model}} \), and \( \sigma \) is the sensor feature representation output and the input of \( d_{model} \) dimensions, linear projection weight matrix, and nonlinear activation function, respectively. \( l \) and \( l - 1 \) denote the layer number. \( A \in \mathbb{R}^{N \times N} \) denotes the normalized structural adjacency matrix by the Laplacian regularization

\[
A = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2},
\]

where \( \tilde{A} \in \mathbb{R}^{N \times N} \) is the graph adjacency matrix and \( \tilde{D} \) is the diagonal matrix with the \( i \)-th element \( \tilde{D}_{ii} = \sum_{j} \tilde{A}_{ij} \).

According to Eq. (9), the network only considers the neighbor sensor relationships, even if the neighbor sensors do not have similar features and patterns. Aiming to widely capture similar patterns from other sensor nodes, we propose the CIGNN module as follows,

\[
\tilde{Z}^{(l)} = \text{AGG} \left( \psi_{c} \sigma \left( \text{SCorr}, S_{w}Z^{(l-1)}W^{(l)} \right) \right),
\]

where \( \text{SCorr}_{c} \in \mathbb{R}^{N \times N} \) denotes our proposed spatial correlation information in dimension \( c \), and \( \psi_{c} \) is the trainable parameter to control the aggregation level of each attribute.

To cover the dynamic change among sensors over time, we add a spatial dynamic weight matrix \( S_{w} \) to adaptively adjust the degree of correlation information as in [17]. The spatial dynamic weight matrix is calculated as follows:

\[
S_{w} = \text{softmax} \left( \frac{Z^{(l-1)}Z^{(l-1)^{T}}}{\sqrt{d_{model}}} \right) \in \mathbb{R}^{N \times N}.
\]

Although Scorr is static based on the dataset distribution, we can make use of \( S_{w} \) to dynamically control the feature construction. To discuss the effort of dynamic and static Scorr, we present detailed experiments in Sect. 6.2.
Finally, we integrate the predefined graph structure to avoid losing local information by a trainable parameter $\Omega$ as follows:

$$Z^{(l)} = \text{AGG}(Z^{(l)}, \Omega Z^{(l)}).$$

(12)

Different from previous works, the CIGNN module integrates correlation information with structural information and provides accurate and dense interconnections for the propagate-aggregate mode. As a result, it can improve the feature aggregation efficiency among similar sensors and produce more correct features for other components.

4.3 Correlation information multi-head attention

In this subsection, we propose a correlation information multi-head attention (CIATT) component utilizing spatial correlation information to construct more stable contextual features and focus on the most relevant sequence, as shown in Fig. 2c.

Currently, many works use the attention mechanism for traffic flow forecasting. The attention mechanism, as the major component of the transformer architecture, utilizes the queries ($Q$), keys ($K$), and values ($V$) as the inputs to model the dependencies of time points among sequences and construct a new embedding representation output as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax} \left( \frac{Q K^T}{\sqrt{d_{\text{head}}}} \right) V,$$

(13)

where $Q \in \mathbb{R}^{N \times L \times d_{\text{head}}}$, $K \in \mathbb{R}^{N \times L \times d_{\text{head}}}$, $V \in \mathbb{R}^{N \times L \times d_{\text{head}}}$ and $d_{\text{head}}$ is the input dimension of one head.

However, unwanted distractions will appear caused by distractions of keys ($K$) in Eq. (13). To address this problem, we define the CIATT as follows:

$$\text{CIATT}(Q, K, V) = \text{Linear} (\text{Cat}(\text{CIATT}_1, \ldots, \text{CIATT}_H)),$$

(14)

where there are $H$ heads for the CIATT, and the outputs of all heads are concatenated and reweighted by the Cat and Linear operators. The $h_{th}$ head is defined as follows:

$$\text{CIATT}_h = \text{Softmax} \left( \frac{Q^{(h)} K^{(h)}^T}{\sqrt{d_{\text{head}}}} \right)_V,$$

(15)

Considering that different sensors with strong correlation information have similar patterns in the traffic road network, we construct $\tilde{K}^{(h)}$ by using SCorr to combine similar patterns of correlative sensors as follows:

$$\tilde{K} = F(\text{SCorr}, K) = \frac{1}{C} \sum_{c=1}^{C} \text{SCorr}_c^T \tilde{K},$$

(16)

where $F$ is the reconstruction function. The inputs are the spatial correlation information and the original keys $K$ and the outputs $\tilde{K} \in \mathbb{R}^{N \times L_{K} \times d_{\text{head}}}$ are more stable and reliable representations for the attention mechanism, where $L_{K}$ is the length of $K$. According to the degree of correlation information, the top $U$ items are selected from SCorr and normalized via the softmax function as,

$$\text{SCorr}_{i,m} = \frac{\exp(\text{SCorr}_{i,m}^{(c)})}{\sum_{u=1}^{U} \exp(\text{SCorr}_{i,u}^{(c)})},$$

and $\tilde{K} \in \mathbb{R}^{N \times U \times L_{K} \times d_{\text{head}}}$ are the items selected from $K$ in the same order. In detail, $\tilde{K}_i$ of sensor $i$ in $\tilde{K}$ is calculated as follows:

$$\tilde{K}_i = \frac{1}{C} \sum_{c=1}^{C} \sum_{u=1}^{U} \text{SCorr}_{i,u}^{(c)} \bar{k}_{i,u},$$

where $\bar{k}_{i,u} \in \mathbb{R}^{L_{K} \times d_{\text{head}}}$ is the element of sensor $i$ in $\tilde{K}$. Finally, the representation $\tilde{K} \in \mathbb{R}^{N \times L_{K} \times d_{\text{head}}}$ can replace the original keys ($K \in \mathbb{R}^{N \times L_{K} \times d_{\text{head}}}$) in the attention mechanism in Eq. (13).

Our method aggregates similar patterns of different sensors with SCorr weights to construct more stable contextual features for the attention mechanism. Hence, the attention mechanism can earn more focused attention weights, as shown in Fig. 1. With the help of CIATT, a time series sequence can accurately find its most relevant sequence to increase the accuracy of traffic flow forecasting tasks.

4.4 Encoder–decoder

The encoder-decoder is the basic architecture of the transformer network. In this paper, the input data of the encoder are the periodic data defined as Eq. (1). The encoder input data are $X = (X_{\text{weekly}}, X_{\text{daily}}, X_{\text{hourly}})$, which are combined with the spatial position embedding and temporal position embedding. Then, CIATT constructs more stable contextual features and CIGNN integrates correlation information with structural information to learn a similar pattern for each sensor node, which are connected by the residual connection and layer normalization in each layer. The decoder input data are $X = (X_t, X_{t+1}, \ldots, X_{t+L-1})$, which are processed as the encoder input. Furthermore, the decoder layer receives the encoder output as historical memory for prediction. Finally, the output data are $X = (\hat{X}_{t+1}, \hat{X}_{t+2}, \ldots, \hat{X}_{t+L})$. 

© Springer
5 Experiments

In this section, we conduct experiments to evaluate the performance of CorrSTN on four real-world spatiotemporal traffic network datasets and discuss the effect of each component. We partition each dataset into training, validation and test sets in a ratio of 6:2:2 by timestamps, and all data are normalized into \([-1, 1]\) by the min-max method. In SCorr and TCorr, \(\eta\) is set at 0.6 to control the number of partitions. We evaluate the performance of our model by the metrics of mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE). They are defined as

\[
\text{MAE}(\hat{X}, X) = \frac{1}{L} \sum_{i=1}^{L} \|X^i - \hat{X}^i\|
\]

\[
\text{RMSE}(\hat{X}, X) = \sqrt{\frac{1}{L} \sum_{i=1}^{L} (X^i - \hat{X}^i)^2}
\]

\[
\text{MAPE}(\hat{X}, X) = \frac{1}{L} \sum_{i=1}^{L} \left| \frac{X^i - \hat{X}^i}{X^i} \right|
\]

(17)

The means and standard deviations are collected in Table 5, and the standard deviation (s) is defined as,

\[
s = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (E_i - \bar{E})^2},
\]

(18)

where \(E\) denotes the MAE, RMSE or MAPE, and \(N\) denotes the number of metric values.

5.1 Parameter settings

Our model chooses the MAE loss function and Adam optimizer for training. We feed the historical data into the encoder and decoder networks during training. Once the decoder network generates prediction results, the optimizer adjusts the model parameters according to the training loss. We set the learning rate at 0.001, and other hyperparameters are described in Table 1. To compare all the baseline methods, we adopt three kinds of periodic traffic data (hourly, daily and weekly) to predict the traffic flow in the next hour. With the help of TCorr, we search data schemes for each dataset listed in Table 2. To demonstrate the correctness of TCorr, we illustrate TCorr and conduct experiments to evaluate the scheme on the HZME (outflow) dataset in Sect. 6.1.

5.2 Datasets

The details of the four traffic datasets are given in Table 3. The datasets of the highway traffic flow are collected every 30 s by the Caltrans Performance Measurement System (PeMS) in California [33] with various types of vehicle detector stations, including inductive loops, side-fire radar, and magnetometers. Then, the data are aggregated into 5-minute intervals, named the PEMS03, PEMS04, PEMS07 and PEMS08 datasets. There are 358, 307, 883 and 170 sensors in these datasets, respectively, which indicate vehicle detector stations in the real world. The Traffic Flow Prediction dataset is collected every 15 min at 36 sensor locations along two major highways in the Northern Virginia/Washington, D.C., capital region. The datasets of the metro crowd flow type are collected from the Hangzhou metro system [17], namely HZME, including inflow and outflow datasets. The HZME datasets contain 80 nodes and 168 edges (undirected network) with a sparse spatial structural relationship.

| Table 1 | Hyperparameters of our CorrSTN model |
|---------|-------------------------------------|
|          | Encoder layers | Decoder layers | Kernel size | Heads | Batchsize | U |
| PEMS03   | 3              | 3              | 3           | 8     | 4         | 3 |
| PEMS03(p) | 3              | 3              | 3           | 8     | 4         | 5 |
| PEMS04   | 4              | 4              | 3           | 8     | 4         | 2 |
| PEMS04(p) | 4              | 4              | 3           | 8     | 4         | 3 |
| PEMS07   | 3              | 3              | 3           | 8     | 4         | 5 |
| PEMS07(p) | 3              | 3              | 3           | 8     | 2         | 5 |
| PEMS08   | 4              | 4              | 3           | 8     | 16        | 8 |
| PEMS08(p) | 4              | 4              | 3           | 8     | 8         | 5 |
| Traffic Flow Prediction | 4              | 4              | 3           | 8     | 16        | 3 |
| HZME(inflow) | 4            | 4              | 3           | 8     | 4         | 4 |
| HZME(inflow)(p) | 4          | 4              | 3           | 8     | 4         | 4 |
| HZME(outflow) | 3             | 3              | 5           | 4     | 8         | 5 |
| HZME(outflow)(p) | 4            | 4              | 3           | 4     | 16        | 5 |
5.3 Performance

We compare our model with fifteen baseline methods, including VAR [25], SVR [3], LSTM [26], DCRNN [13], STGCN [12], ASTGCN [14], GWN [27], GMAN [23], AGCRN [31], STSGCN [15], MTGNN [21], STFGNN [16], STGODE [22], DMSTGCN [24] and ASTGNN [17]. We repeat the experiments 5 times on each dataset to evaluate the performance, and the means and standard deviations are collected in Tables 4 and 5, where the bold font highlights the best values. Particularly, due to the normalized data in the Traffic Flow Prediction dataset, the standard deviations are almost zero. Thus, we collect the best results of each model on the Traffic Flow Prediction dataset. In Tables 4 and 5, the results with (p) are achieved with hourly, daily and weekly data, while the results without (p) are achieved with only hourly data.

Accuracy. With only hourly data, CorrSTN achieves superior performance in the metrics of MAE, RMSE and MAPE on both the highway traffic flow and metro crowd flow datasets. We further enhance predictive performance by using periodic data (hourly, daily or weekly) as input to our model. Thus, our model has the ability to outperform the state-of-the-art method ASTGNN(p). In the following, we will give detailed comparisons and analyses of these baselines and our model.

The traditional methods, VAR, SVR and LSTM, only consider the temporal features but ignore the spatial relationships among sensors. Thus, their capabilities of capturing spatiotemporal dependencies are quite limited. For example, LSTM achieves the worst score in the metrics of MAE, RMSE and MAPE on the HZME (outflow) dataset, as shown in Table 5.

The neural network-based methods (DCRNN, STGCN, ASTGCN, STSGCN, and ASTGNN) all take efforts to make use of spatiotemporal relationships in feature extraction. The DCRNN combines diffusion convolution and RNN to predict traffic flow. However, the ability for long-term forecasting is limited by the RNN capability. The STGCN, ASTGCN, STSGCN, and ASTGNN are four methods mixed of CNN-based and GNN-based components, where the temporal features are captured by the CNN, and the spatial features are captured by the GNN. Nevertheless, the critical correlation information is not taken into consideration in these methods. Due to the combination of correlation information into the GNN component, GWN, GMAN, AGCRN, MTGNN, STFGNN, STGODE, and DMSTGCN can obtain a more satisfactory performance for each dataset, as shown in Table 5. The main reasons are that the adaptive graph structure is built with training parameters (e.g., GWN, GMAN, and AGCRN), a sparse graph matrix is constructed with the DTW algorithm (e.g., STFGNN and STGODE), or a dynamic graph convolution is employed to capture neighbor sensor features (e.g., MTGNN and DMSTGCN).

However, the graph structures of the above methods cannot adequately represent similar patterns among various sensors. To address this problem, we use the MIC algorithm to detect more widespread relationships for combining similar features in the CIGNN component. As a result, CorrSTN (p) outperforms AGCRN on the HZME (outflow) dataset by 21.6%, 23.0% and 43.8% in the metrics of MAE, RMSE and MAPE, respectively, as shown in Table 5. On the other hand, unlike dynamic graph convolution (e.g., MTGNN and DMSTGCN), we employ a spatial dynamic weight matrix to fit the varying changes throughout the training and test processes. Compared with DMSTGCN on the HZME (outflow) dataset, our model improves the performance by 18.2%, 19.2% and 35.3% in the metrics of MAE, RMSE and MAPE, respectively, as shown in Table 5.

| Data type               | Dataset          | Sensors | Time span                |
|-------------------------|------------------|---------|--------------------------|
| Highway traffic flow    | PEMS03           | 358     | 09/01/2018–11/30/2018    |
|                         | PEMS04           | 307     | 01/01/2018–02/28/2018    |
|                         | PEMS07           | 883     | 05/01/2017–08/31/2017    |
|                         | PEMS08           | 170     | 07/01/2016–08/31/2016    |
| Traffic Flow Prediction | 36               |         |                          |
| Metro crowd flow        | HZME(inflow)     | 80      | 01/01/2019–01/26/2019    |
|                         | HZME(outflow)    | 80      | 01/01/2019–01/26/2019    |
Table 4 Performance comparison on the highway traffic flow datasets (PEMS03, PEMS04 and traffic flow prediction)

| Highway traffic flow | PEMS03 | PEMS04 | Traffic flow prediction |
|----------------------|--------|--------|------------------------|
|                      | MAE    | RMSE   | MAPE (%)               | MAE    | RMSE   | MAPE (%)               | MAE    | RMSE   | MAPE (%)               |
| VAR (2006) [25]      | 21.08  | 34.75  | 22.35                  | 23.75  | 36.66  | 18.09                  | 0.0341 | 0.0494 | 16.72                  |
| SVR (1997) [3]       | 22.01 ±| 35.28 ±| 22.93 ±                | 28.66 ±| 44.59 ±| 19.15 ±                | 0.0491 | 0.0663 | 17.70                  |
| LSTM (1997) [26]     | 20.62 ±| 33.54 ±| 28.94 ±                | 26.81 ±| 40.74 ±| 22.33 ±                | 0.0421 | 0.0613 | 20.64                  |
| DCRNN (2018) [13]    | 18.39 ±| 30.56 ±| 20.22 ±                | 23.65 ±| 37.12 ±| 16.05 ±                | 0.0387 | 0.0495 | 14.83                  |
| STGCN (2018) [12]    | 18.28 ±| 30.73 ±| 17.52 ±                | 22.27 ±| 35.02 ±| 14.36 ±                | 0.0323 | 0.0461 | 13.28                  |
| ASTGCN (2019) [14]   | 17.85 ±| 29.88 ±| 17.65 ±                | 22.42 ±| 34.75 ±| 15.87 ±                | 0.0311 | 0.0524 | 14.67                  |
| GWN (2019) [27]      | 14.79 ±| 25.51 ±| 14.32 ±                | 19.36 ±| 31.72 ±| 13.31 ±                | 0.0306 | 0.0458 | 12.31                  |
| GMAN (2020) [23]     | 17.12 ±| 28.32 ±| 20.01 ±                | 19.84 ±| 32.89 ±| 16.22 ±                | 0.0296 | 0.0512 | 16.21                  |
| AGCRN (2020) [31]    | 16.21 ±| 28.67 ±| 16.98 ±                | 19.76 ±| 32.30 ±| 12.93 ±                | 0.0361 | 0.0474 | 11.99                  |
| STSGCN (2020) [15]   | 17.51 ±| 29.05 ±| 16.92 ±                | 21.08 ±| 33.83 ±| 13.88 ±                | 0.0363 | 0.0479 | 12.83                  |
| MTGNN (2020) [21]    | 16.74 ±| 28.43 ±| 16.59 ±                | 20.92 ±| 33.18 ±| 13.88 ±                | 0.0302 | 0.0471 | 12.71                  |
| STFGNN (2021) [16]   | 16.77 ±| 28.34 ±| 16.30 ±                | 19.83 ±| 31.88 ±| 13.02 ±                | 0.0359 | 0.0501 | 12.03                  |
| STGODE (2021) [22]   | 16.61 ±| 27.98 ±| 16.81 ±                | 21.24 ±| 32.97 ±| 13.83 ±                | 0.0314 | 0.0471 | 12.84                  |
| DMSTGCN (2021) [24]  | 15.12 ±| 25.72 ±| 15.85 ±                | 19.83 ±| 31.35 ±| 14.06 ±                | 0.0333 | 0.0467 | 12.85                  |
| ASTGNN (2021) [17]   | 14.78 ±| 25.00 ±| 14.79 ±                | 18.60 ±| 30.91 ±| 12.36 ±                | 0.0308 | 0.0468 | 11.84                  |
| ASTGNN (p) (2021) [17]| 14.55 ±| 24.96 ±| 13.66 ±                | 18.44 ±| 31.02 ±| 12.37 ±                | /      | /      | /                     |
| CorrSTN              | 14.75 ±| 25.28 ±| 13.40 ±                | 18.76 ±| 30.93 ±| 12.31 ±                | 0.0298 | 0.0450 | 11.50                  |
| CorrSTN (p)          | 14.48 ±| 24.61 ±| 13.58 ±                | 18.46 ±| 30.91 ±| 12.21 ±                | /      | /      | /                     |

For the attention mechanism, several previous methods suffer from unwanted distractions due to vanilla attention, such as spatiotemporal attention (GMAN) and trend-aware attention (ASTGNN). In our model, with the help of SCorr, CIATT can aggregate similar patterns from different sensors and apply more focused attention weights to match the most relevant sequence pattern. Thus, compared with GMAN on the HZME (outflow) dataset, the performance of CorrSTN (p) can be enhanced by 13.4%, 15.0% and 32.3% in the metrics of MAE, RMSE and MAPE, respectively, as shown in Table 5. On the PEMS04 dataset, ASTGNN (p) achieves the best performance in the MAE metric, and CorrSTN (p) obtains the best performance in the RMSE and MAPE metrics.

**Selection scheme.** The data selection scheme for periodic data is a crucial part to enhance model performance. Instead of the traditional exhaustive search, we design an appropriate data selection scheme by using the efficient representation TCorr. In contrast to ASTGNN (p) using hourly and weekly data, our CorrSTN (p) adopts hourly and daily data as input for model training on the HZME
Table 5 Performance comparison on the highway traffic flow and metro crowd flow datasets (PEMS07, PEMS08, HZME (inflow) and HZME (outflow))

| Datasets (highway traffic flow) | PEMS07 |         |         |
|-------------------------------|--------|---------|---------|
| Metrics                       | MAE    | RMSE    | MAPE (%)|
| VAR (2006) [25]               | 101.20 | 155.14  | 39.69   |
| SVR (1997) [3]                | 32.97 ± 0.98 | 50.15 ± 0.15 | 15.43 ± 1.22 |
| LSTM (1997) [26]              | 29.71 ± 0.09 | 45.32 ± 0.27 | 14.14 ± 1.00 |
| DCRNN (2018) [13]             | 23.60 ± 0.05 | 36.51 ± 0.05 | 10.28 ± 0.02 |
| STGCN (2018) [12]             | 27.41 ± 0.45 | 41.02 ± 0.58 | 12.23 ± 0.38 |
| ASTGCN (2019) [14]            | 25.98 ± 0.78 | 39.65 ± 0.89 | 11.84 ± 0.69 |
| GWN (2019) [27]               | 21.22 ± 0.24 | 34.12 ± 0.18 | 9.07 ± 0.20 |
| GMAN (2020) [23]              | 21.56 ± 0.26 | 34.97 ± 0.44 | 9.51 ± 0.16 |
| AGCRN (2020) [31]             | 22.56 ± 0.33 | 36.18 ± 0.46 | 9.67 ± 0.14 |
| STSGCN (2020) [15]            | 23.99 ± 0.14 | 39.32 ± 0.31 | 10.10 ± 0.08 |
| MTGN (2020) [21]              | 20.57 ± 0.61 | 33.54 ± 0.73 | 9.12 ± 0.13 |
| STFGNN (2021) [16]            | 20.72 ± 0.11 | 35.80 ± 0.18 | 9.21 ± 0.07 |
| STGODE (2021) [22]            | 22.89 ± 0.15 | 37.47 ± 0.07 | 10.10 ± 0.06 |
| DMSTGCN (2021) [24]           | 20.77 ± 0.57 | 33.67 ± 0.54 | 8.94 ± 0.42 |
| ASTGNN (2021) [17]            | 20.62 ± 0.12 | 34.00 ± 0.21 | 8.86 ± 0.10 |
| ASTGNN(p) (2021) [17]         | 19.26 ± 0.17 | 32.75 ± 0.25 | 8.54 ± 0.19 |
| CorrSTN                       | 19.62 ± 0.05 | 33.11 ± 0.23 | 8.22 ± 0.06 |
| CorrSTN(p)                    | 18.10 ± 0.10 | 31.61 ± 0.12 | 7.58 ± 0.12 |
|                                |         |         |         |
| Datasets (metro crowd flow)    | PEMS08 |         |         |
| Metrics                       | MAE    | RMSE    | MAPE (%)|
| VAR (2006) [25]               | 22.32  | 33.83   | 14.47   |
| SVR (1997) [3]                | 23.25 ± 0.01 | 36.15 ± 0.02 | 14.71 ± 0.16 |
| LSTM (1997) [26]              | 22.19 ± 0.13 | 33.59 ± 0.05 | 18.74 ± 2.79 |
| DCRNN (2018) [13]             | 18.22 ± 0.06 | 28.29 ± 0.09 | 11.56 ± 0.04 |
| STGCN (2018) [12]             | 18.04 ± 0.19 | 27.94 ± 0.18 | 11.16 ± 0.10 |
| ASTGCN (2019) [14]            | 18.86 ± 0.41 | 28.55 ± 0.49 | 12.50 ± 0.66 |
| GWN (2019) [27]               | 21.22 ± 0.24 | 34.12 ± 0.18 | 9.07 ± 0.20 |
| GMAN (2020) [23]              | 21.56 ± 0.26 | 34.97 ± 0.44 | 9.51 ± 0.16 |
| AGCRN (2020) [31]             | 22.56 ± 0.33 | 36.18 ± 0.46 | 9.67 ± 0.14 |
| STSGCN (2020) [15]            | 23.99 ± 0.14 | 39.32 ± 0.31 | 10.10 ± 0.08 |
| MTGN (2020) [21]              | 20.57 ± 0.61 | 33.54 ± 0.73 | 9.12 ± 0.13 |
| STFGNN (2021) [16]            | 20.72 ± 0.11 | 35.80 ± 0.18 | 9.21 ± 0.07 |
| STGODE (2021) [22]            | 22.89 ± 0.15 | 37.47 ± 0.07 | 10.10 ± 0.06 |
| DMSTGCN (2021) [24]           | 20.77 ± 0.57 | 33.67 ± 0.54 | 8.94 ± 0.42 |
| ASTGNN (2021) [17]            | 20.62 ± 0.12 | 34.00 ± 0.21 | 8.86 ± 0.10 |
| ASTGNN(p) (2021) [17]         | 19.26 ± 0.17 | 32.75 ± 0.25 | 8.54 ± 0.19 |
| CorrSTN                       | 19.62 ± 0.05 | 33.11 ± 0.23 | 8.22 ± 0.06 |
| CorrSTN(p)                    | 18.10 ± 0.10 | 31.61 ± 0.12 | 7.58 ± 0.12 |

Datasets (metro crowd flow) (inflow/outflow) dataset. In Table 5, compared with other models, our model CorrSTN (p) makes notable improvements over all datasets, which indicates that our data selection scheme can generally improve model performance.
Fig. 5 Performance comparison of CorrSTN and ASTGNN as the prediction interval increases (PEMS03, PEM04 and Traffic Flow Prediction)

Fig. 6 Performance comparison of the CorrSTN and ASTGNN as the prediction interval increases (PEMS07, PEM08, HZME (inflow) and HZME (outflow))
Stability. In order to show the stability of our model, we present an illustration at each prediction point of the CorrSTN and ASTGNN, as shown in Figs. 5 and 6. For the PEMS03 and PEMS04 datasets, the stability of our model is similar to that of the ASTGNN model, as shown in Fig. 5. For the PEMS07, HZME (inflow), HZME (outflow) and Traffic Flow Prediction datasets, our model achieves a more stable predictive performance, as shown in Fig. 6; meanwhile, our model makes notable advancements compared with ASTGNN.

Overall, our work offers more accurate results for traffic flow forecasting. The improvement is attributed to the accurate aggregation in CIGNN and the focused attention weights in CIATT with SCorr. Moreover, based on TCorr, a novel method searching for an effective scheme further improves our model performance.

5.4 Ablation experiments

In this subsection, we verify the effectiveness of our CIGNN and CIATT components on the PEMS08 dataset. For comparison, we assign the state-of-the-art ASTGNN (without CIGNN and CIATT) as the baseline. And, with-CIGNN (with-CIATT) denotes the model using only the CIGNN (CIATT) component, and CIGNN+CIATT denotes the model using the CIGNN and CIATT components. Our model takes the hourly data as the input for training and shows the prediction results for all prediction points, as shown in Fig. 7.

From the experimental results, we find that with-CIGNN (resp. with-CIATT) outperforms the baseline by 1.04%, 0.72% and 1.50% (resp. 2.74%, 2.42% and 2.13%) in the metrics of MAE, RMSE and MAPE, respectively. Moreover, CIGNN+CIATT outperforms the baseline by 5.72%, 4.16% and 3.61% in the metrics of MAE, RMSE and MAPE, respectively. Thus, the CIGNN and CIATT modules can make significant increases in prediction accuracy.

We also find that the combined contribution (CIGNN+CIATT) is larger than the sum of the contributions of CIGNN and CIATT (with-CIGNN and with-CIATT). This shows that our CIGNN and CIATT are able to interact and positively influence each other by using the correlation information between spatial and temporal features.

Furthermore, comparing the results, we find that the CIATT component shows higher effectiveness than the CIGNN component in the traffic flow forecasting task. In our model, the CIATT component focuses the attention weights on the most relevant sequence and constructs more correct features for the other components, as shown in Fig. 1. Thus, a powerful improvement can be achieved by our CIATT component.

5.5 Time cost study

We implement the proposed model in Python 3.8 and PyTorch 1.7.0. The model has been successfully executed and tested on the Linux platform with an Intel (R) Xeon (R) Gold 6240R CPU@2.40 GHz and NVIDIA TESLA V100 (PCI-E) GPU 32 GB card. To compare the time cost among our model and baselines, we collect the training and test times in Table 6. In addition, we list the time costs of SCorr and TCorr (CPU: AMD 3970x 32C64T) in Table 7. Although our model spends more time during training and test, the forecasting performance is significantly improved on these datasets. Meanwhile, in Table 7, the time costs of SCorr and TCorr for each dataset spend only once before the model training. Hence, we can search for an appropriate

![Fig. 7 Ablation results on the PEMS08 dataset](image-url)
data scheme for each dataset in a short time to save training and test time costs.

6 Comparison and analysis beyond performance

In this section, we perform experiments to discuss and analyze the data selection schemes, dynamic SCorr and the influence of different top-$U$.

6.1 Data selection schemes

In this subsection, we consider the four real-world datasets. Different datasets have different temporal correlation information representations, as shown in Fig. 8. For the sake of analysis, we sum and average the scores of all sensors, as shown in Table 8. It is obvious that the weekly data have the highest correlation degree with the prediction data on the highway traffic flow datasets (PEMS07 and PEMS08). In contrast, the daily data have the highest correlation degree with the prediction data on the metro traffic flow datasets (HZME (inflow) and HZME (outflow)).

It is reasonable that these four datasets have different temporal correlation information representations. As stated in Sect. 5.2, the highway traffic flow datasets (PEMS07 and PEMS08) and the metro crowd flow datasets (HZME (inflow) and HZME (outflow)) are collected from different locations and environments. Concretely, highway traffic flow datasets are collected from highway roads in California, which reflect the characteristics of long-distance travel. In contrast, the metro crowd flow datasets are collected from the metro in Hangzhou, which reflects the characteristics of short-distance travel.

Hence, for the different representations, we need to design different schemes, as stated in Sect. 4.1.2. In the following, we will verify the model performance with different data selection schemes on the HZME (outflow) dataset. We conduct experiments to compare the performance of our scheme and traditional exhaustive search schemes. The type of scheme is designed as scheme $h_1d_1w_1$, as shown in Table 9. The $h_1$ ($h_0$), $d_1$ ($d_0$), and $w_1$ ($w_0$) denote that the input data (do not) contain hourly, daily and weekly data, respectively.

The average performance and each prediction point performance in the metrics of MAE, RMSE and MAPE are shown in Fig. 9. First, we analyze the effect of hourly data. Compared with the $h_0d_1w_0$ (or $h_0d_0w_1$) scheme, the $h_1d_0w_0$ scheme gives rise to predictive performance at the short-term prediction points. However, the performance sharply decreases at the long-term prediction points with the $h_1d_0w_0$ scheme. It shows that hourly data only reflect the
short-term trend in the HZME (outflow) dataset. Moreover, by comparing the \(h_0d_1w_1\) and \(h_1d_1w_1\) schemes, we can also observe the improvement by hourly data at the short-term prediction points. Then, we consider the daily and weekly data by comparing the \(h_1d_1w_0\), \(h_1d_0w_1\) and \(h_1d_1w_1\) schemes. The experimental results demonstrate that the daily data can improve the model performance at short-term and long-term prediction points. In contrast, the weekly data decrease the model performance. Thus, we can conclude that the daily data make more improvements in predictive performance than the weekly data. Moreover, as shown in Table 8, the result of \(\Delta_{\text{dv}} > 0\) demonstrates that there is no regular weekly periodic property in the HZME (outflow) dataset.

Overall, with the \(h_1d_1w_0\) scheme, our model can achieve outstanding performance at all prediction points. Compared with the traditional exhaustive search method, our method with TCorr can effectively search for an appropriate data selection scheme without trying all possible schemes. With the appropriate scheme, we can further improve the model performance for traffic flow forecasting tasks.

### 6.2 Static and dynamic SCorr

In this subsection, we compare the effect of static and dynamic SCorr on the HZME (inflow) dataset. The dynamic SCorr is calculated by cyclic forwarding along the timeline as Eq. (4) with \(T = 12\) timestamps (one hour).

As shown in Fig. 10, compared with CorrSTN with dynamic SCorr, CorrSTN with static SCorr achieves markedly better performance in the metrics of MAE, RMSE and MAPE. Although the dynamic SCorr contains rich information at each time point, the high-frequency change makes it difficult for the model to fit all training data. In fact, the correlative relationships do not frequently change. Thus, the dynamic SCorr leads to a performance decrease. Moreover, we use the spatial dynamic weight matrix to adaptively adjust the relationships in the CIGNN component. Therefore, we can also dynamically control the feature construction during the prediction process.

### 6.3 CIATT with different top \(U\)

In this subsection, we adopt six different \(U\) hyperparameters to compare the performance with different levels of spatial correlation information on the HEME (inflow) dataset. We set the values of \(U\) at 2, 3, 4, 5 and 8, and the other hyperparameters are set as shown in Table 1.

As shown in Fig. 12, we find that the CorrSTN model achieves the best performance when \(U\) is set at 4 on the HEME (inflow) dataset. Moreover, the performance can be improved as \(U\) increases from \(U = 2\) until \(U = 4\), and the peak performance is achieved at \(U = 4\). We also find that the CIATT component is affected by the hyperparameter \(U\). Concretely, as \(U\) increases, the increasing number of relevant sequences can improve predictive performance at the beginning. However, after achieving the best performance, the performance decreases due to the increasing number of irrelevant features (see \(U = 5\) and \(U = 8\) in Fig. 12). Therefore, with an appropriate hyperparameter \(U\), CIATT can further improve predictive performance.

### 7 Case study

In this section, we conduct case studies to show that SCorr and TCorr can accurately represent the data correlations. Moreover, we also provide visual comparisons to clearly demonstrate our improvement in predictive performance.
7.1 The effect of SCorr

In the traffic flow data, correlation information exists among sensors in different traffic datasets. Taking the PEMS08 dataset as an example, we calculate the correlation of sensors by using SCorr, as shown in Fig. 11. The size of each sensor denotes its correlation degree with sensor 48. Note that both sensor 74 and sensor 83 have strong correlation representations with sensor 48, although there are no structural paths among them directly. The counterintuitive phenomenon shows that correlation information is more significant than structural information in discovering spatiotemporal dependencies and dynamic relationships in traffic forecasting tasks. Furthermore, according to the sensor 48 sequence and four other sequences, as shown in Fig. 11, we find that SCorr can capture widespread associations accurately and provide reliable representations for evaluating the correlation strength.

7.2 The effect of TCorr

We take a part of the data as an example to show the effect of TCorr on four real-world datasets, as shown in Fig. 13. We can see that the PEMS07 and PEMS08 datasets have a stronger correlation with the weekly data, as shown on May 13–14 (weekend) in the PEMS07 dataset and July 9–10 (weekend) in the PEMS08 dataset. Compared with other days, the two weekends are special days in each week, which do not have morning peaks or evening peaks. It is demonstrated that the regular weekly periodic property exists in the two datasets. Meanwhile, the other days also show regular daily periodic properties. In contrast, the HZME inflow and outflow datasets strongly correlate with

![Fig. 11 The spatial correlation information between sensor 48 and other sensors on the PEMS08 dataset](image1)

![Fig. 12 Results on the HZME (inflow) dataset with six different U schemes](image2)

![Fig. 13 The original data on the PEMS07, PEMS08, HZME (inflow) and HZME (outflow) datasets](image3)
daily data, as shown on January 12-13 (weekend) on the two datasets. Compared with workdays, weekends also show similar sequences in the morning peaks and evening peaks. It is shown that all days have regular daily periodic properties.

Based on the case study, we find that the effect of TCorr accurately corresponds with the data characteristics. Additionally, TCorr is effective to detect the periodic properties.

7.3 Visual comparisons

We visualize the forecasting results of the CorrSTN and ASTGNN models, as shown in Fig. 14. To clearly show the improvement, we smooth all sequences by the locally estimated scatterplot smoothing (LOESS) method [34], which is a nonparametric regression method. And the metrics of MAE, RMSE and MAPE over the time periods are illustrated to compare the model forecasting performances.

The visualization results show that our model can fit the target sequences better than the ASTGNN model. Although the target sequences are hard to fit, we can also obtain satisfactory predicted results, for example, the magnified sequences on August 18, 2017, of the PEMS07 dataset and August 26, 2016, of the PEMS08 dataset. Moreover, CorrSTN can achieve better performance at the sequences of peaks and troughs, for example, the magnified sequences on August 20, 2017, of the PEMS07 dataset and January 25, 2019, of the HZME (outflow) dataset.

8 Conclusion

In this paper, we propose an effective neural network-based network CorrSTN to predict traffic flow data in intelligent transportation systems. Considering the correlation information deeply, we first propose two elaborate spatiotemporal representations named SCorr and TCorr for spatiotemporal sensor sequences. Then, by using SCorr, we
design two crucial components named CIGNN and CIATT in our CorrSTN model. The CIGNN component can improve the feature aggregation efficiency and produce more correct features. The CIATT component can construct more focused attention weights to extract features from relevant sequences. We use TCorr to mine the correlations among different periodic data and design an effective data scheme for periodic datasets. Finally, we conduct experiments to compare the CorrSTN with fifteen baseline methods on the highway traffic flow and metro crowd flow datasets. The experimental results demonstrate that our CorrSTN outperforms the state-of-the-art methods in terms of predictive performance. In particular, on the HZME (outflow) dataset, our model makes significant improvements compared with the ASTGNN model by 13.2%, 15.3% and 29.3% in the metrics of MAE, RMSE and MAPE, respectively. Moreover, the case studies show that SCorr can elaborately present the spatiotemporal features, while TCorr can help to select the appropriate data schemes.

Acknowledgements This research is supported by the National Key R &D Program of China (No. 2021ZD0113002), National Natural Science Foundation of China (No. 62072292, 61572005, 61771058) and Fundamental Research Funds for the Central Universities of China (No. 2020YJ0532). The support and resources from the Center for High Performance Computing at Beijing Jiaotong University are also gratefully acknowledged.

Data Availability The datasets, code and pre-trained models generated and analyzed during the current study are available in the CorrSTN repository, https://github.com/bjtu-ccdlab/CorrSTN.

Declarations

Conflicts of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Muhammad AN, Aseere AM, Chiroma H, Shah H, Gital AY, Hashem IAT (2021) Deep learning application in smart cities: recent development, taxonomy, challenges and research prospects. Neural Comput Appl 33(7):2973–3009. https://doi.org/10.1007/s00521-020-05151-8
2. Qureshi KN, Ahmad A, Piccialli F, Casella G, Jeon G (2021) Nature-inspired algorithm-based secure data dissemination framework for smart city networks. Neural Comput Appl 33(17):10637–10656. https://doi.org/10.1007/s00521-020-04900-z
3. Drucker H, Burges CJ, Kaufman L, Smola A, Vapnik V (1997) Support vector regression machines. In: Mozer MC, Jordan M, Petsche T (eds) Advances in neural information processing systems, vol 9. MIT Press, Cambridge, pp 155–161
4. Jeong Y-S, Byron Y-J, Castro-Neto MM, Easa SM (2013) Supervised weighting-online learning algorithm for short-term traffic flow prediction. IEEE Trans Intell Transp Syst 14(4):1700–1707. https://doi.org/10.1109/TITS.2013.2267735
5. Sun Y, Leng B, Guan W (2015) A novel wavelet-svm short-time passenger flow prediction in Beijing subway system. Neurocomputing 166:109–121. https://doi.org/10.1016/j.neucom.2015.03.085
6. Van Lint J, Van Hinsbergen C (2012) Short-term and travel time prediction models. Artif Intell Appl Critic Transp Issues 22(1):22–41
7. Luo X, Li D, Yang Y, Zhang S (2019) Spatiotemporal traffic flow prediction with knn and lstm. J Adv Transp 2019:10. https://doi.org/10.1155/2019/414535
8. Cui Z, Ke R, Wang Y (2018) Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. CoRR abs/180102143, 1–12
9. Yao H, Wu F, Ke J, Tang X, Jia Y, Lu S, Gong P, Ye J, Li Z (2018) Deep multi-view spatial-temporal network for taxi demand prediction. In: proceedings of the AAAI conference on artificial intelligence, pp 2588–2595
10. Zhang J, Zheng Y, Qi D, Li R, Yi X, Li T (2018) Predicting citywide crowd flows using deep spatio-temporal residual networks. Artif Intell 259:147–166
11. Wang J, Zhu W, Sun Y, Tian C (2021) An effective dynamic spatiotemporal framework with external features information for traffic prediction. Appl Intell 51(6):3159–3173. https://doi.org/10.1007/s10489-020-02043-1
12. Yu B, Yin H, Zhu Z (2018) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: proceedings of the international joint conference on artificial intelligence
13. Li Y, Yu R, Shahabi C, Liu Y (2018) Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In: international conference on learning representations
14. Guo S, Lin Y, Feng N, Song C, Wan H (2019) Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: proceedings of the AAAI conference on artificial intelligence, vol 33, pp 922–929 https://doi.org/10.1609/aaaiv33i01.3301922
15. Song C, Lin Y, Guo S, Wan H (2020) Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In: proceedings of the AAAI conference on artificial intelligence, vol 34, pp 914–921 https://doi.org/10.1609/aaai.v34i01.5438
16. Li M, Zhu Z (2021) Spatial-temporal fusion graph neural networks for traffic flow forecasting. In: proceedings of the AAAI conference on artificial intelligence vol 35, pp 4189–4196
17. Guo S, Lin Y, Wan H, Li X (2021) Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. IEEE Trans Knowledge Data Eng 99:1–1. https://doi.org/10.1109/TKDE.2021.3056502
18. Liu CH, Piao C, Ma X, Yuan Y, Tang J, Wang G, Leung K K (2021) Modeling citywide crowd flows using attentive convolutional lstm. In: IEEE international conference on data engineering (ICDE), pp 217–228 https://doi.org/10.1109/ICDE51399.2021.00026
19. Wang Y, Yin H, Chen T, Liu C, Wang B, Wo T, Xu J (2021) Gallat: A spatiotemporal graph attention network for passenger demand prediction. In: IEEE international conference on data engineering (ICDE), pp 2129–2134 https://doi.org/10.1109/ICDE51399.2021.00212
20. Zhou X, Shao Y, Huang L, Zang T, Zhu Y (2021) Multi-level attention networks for multi-step citywide passenger demands prediction. IEEE Trans Knowl Data Eng 33(5):2096–2108. https://doi.org/10.1109/TKDE.2019.2948005
21. Wu Z, Pan S, Long G, Jiang J, Chang X, Zhang C (2020) Connecting the dots: multivariate time series forecasting with graph
neural networks. In: proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 753–763 https://doi.org/10.1145/3394486.3403118

22. Fang Z, Long Q, Song G, Xie K (2021) Spatial-temporal graph node networks for traffic flow forecasting. In: proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, pp 364–373. Association for computing machinery, New York, USA https://doi.org/10.1145/3447548.3467430

23. Zheng C, Fan X, Wang C, Qi J (2020) Gman: a graph multi-attention network for traffic prediction. In: proceedings of the AAAI conference on artificial intelligence vol 34, pp 1234–1241 (2020). https://doi.org/10.1609/aaai.v34i01.5477

24. Han L, Du B, Sun L, Fu Y, Ly Y, Xiong H (2021) Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting. In: proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, pp 547–555. Association for computing machinery, New York, USA https://doi.org/10.1145/3447548.3467275

25. Zivot, E., Wang, J.: Vector autoregressive models for multivariate time series. Modeling financial time series with S-PLUS®. 385–429 (2006)

26. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

27. Wu Z, Pan S, Long G, Jiang J, Zhang C (2019) Graph wavenet for deep spatial-temporal graph modeling. In: Kraus, S. (ed.) proceedings of the international joint conference on artificial intelligence, pp 1907–1913 https://doi.org/10.24963/ijcai.2019/264

28. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L, Polosukhin I (2017) Attention is all you need. In: proceedings of the 31st international conference on neural information processing systems, pp 6000–6010. Curran Associates Inc

29. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition, pp 770–778 https://doi.org/10.1109/CVPR.2016.90

30. Ba J L, Kiros J R, Hinton G E (2016) Layer normalization. arXiv preprint arXiv:1607.06450

31. Bai L, Yao L, Li C, Wang X, Wang C (2020) Adaptive graph convolutional recurrent network for traffic forecasting. In: advances in neural information processing systems. advances in neural information processing systems, vol 33, pp 17804–17815

32. Reshef DN, Reshef YA, Finucane HK, Grossman SR, McVean G, Turnbaugh PJ, Lander ES, Mitzenmacher M, Sabeti PC (2011) Detecting novel associations in large data sets. Science 334(6062):1518–1524. https://doi.org/10.1126/science.1205438

33. Chen C, Petty K, Skabardonis A, Varaiya P, Jia Z (2001) Freeway performance measurement system: mining loop detector data. Transp Res Rec 1748(1):96–102. https://doi.org/10.3141/1748-12

34. Cleveland WS (1979) Robust locally weighted regression and smoothing scatterplots. J Am Stat Assoc 74(368):829–836. https://doi.org/10.1080/01621459.1979.10481038

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.