Heliyon 9 (2023) e19927

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

STHSGCN: Spatial-temporal heterogeneous and synchronous graph convolution network for traffic flow prediction

Xian Yu [a], Yin-Xin Bao [a], Quan Shi [a,b,*]

[a] School of Information Science and Technology, Nantong University, Nantong 226019, China
[b] School of Transportation and Civil Engineering, Nantong University, Nantong 226019, China

ARTICLE INFO

Keywords:
Traffic flow prediction
Graph convolution
Heterogeneity
Causality

ABSTRACT

Nowadays, as a crucial component of intelligent transportation systems, traffic flow prediction has received extensive concern. However, most of the existing studies extracted spatial-temporal features with modules that do not differentiate with time and space, and failed to consider spatial-temporal heterogeneities. Furthermore, although previous works have achieved synchronous modeling of spatial-temporal dependencies, the consideration of temporal causality is still lacking in their graph structures. To address these shortcomings, a spatial-temporal heterogeneous and synchronous graph convolution network (STHSGCN) is proposed for traffic flow prediction. To be specific, separate dilated causal spatial-temporal synchronous graph convolutional networks (DCSTSGCNs) for various node clusters are designed to reflect spatial heterogeneity, different dilated causal spatial-temporal synchronous graph convolutional modules (DCSTSGCMs) for diverse time steps are deployed to take account of temporal heterogeneity. In addition, causal spatial-temporal synchronous graph (CSTSG) is proposed to capture temporal causality in spatial-temporal synchronous learning. We further conducted extensive experiments on four real-world datasets, and the results verified the consistent superiority of our proposed approach compared with various existing baselines.

1. Introduction

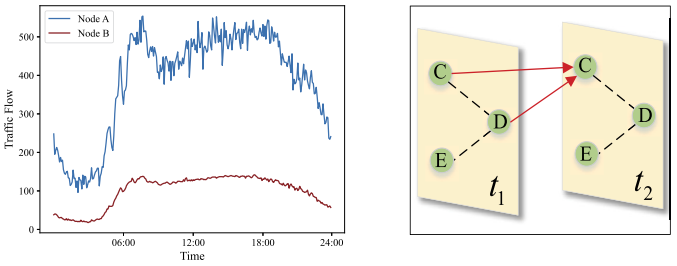
Applications of intelligent transportation systems (ITS) has been proved to be powerful in addressing urban transportation problems, and traffic flow prediction plays a key role in ITS. Modeling spatial-temporal dependencies to accurately predict traffic flow enables reasonable planning of travel routes and rational avoidance of traffic congestion [1]. Recently, many studies have been conducted to capture spatial correlation by defining graph structures and then employing graph convolutional network (GCN) [2–11]. Besides, recurrent neural network variants, such as long-short term memory (LSTM) and gated recurrent unit (GRU), as well as convolutional neural network (CNN) are introduced to learn temporal dependency [12–23]. Although existing works have achieved promising results, modeling spatial-temporal dependencies of traffic flow is still a challenging task due to the spatial-temporal dynamics.

First, traffic flow varies with both spatial and temporal changes. As indicated in Fig. 1(a), on the one hand, the traffic flow of the same node exhibits different rules at different times. For example, the traffic peak in office areas occurs during commuting hours, while the flow is relatively flat at other times. On the other hand, the traffic flow of nodes in different regions shows different

* Corresponding author at: School of Information Science and Technology, Nantong University, Nantong 226019, China.
E-mail addresses: yuxuancos@ntu.edu.cn (X. Yu), 1930320029@stmail.ntu.edu.cn (Y.-X. Bao), sq@ntu.edu.cn (Q. Shi).

https://doi.org/10.1016/j.heliyon.2023.e19927
Received 29 March 2023; Received in revised form 5 September 2023; Accepted 6 September 2023

Available online 11 September 2023



(a) Spatial-temporal variations (b) Temporal correlation

Fig. 1. Example of spatial-temporal variations and temporal correlation of traffic flow. (a) shows that the flow of node A varies with time, and the flow of node B from another area exhibits different characteristics compared with A. (b) illustrates unidirectional correlations between different time steps by the red arrows, and the black dotted lines indicate the spatial adjacencies.

patterns. For instance, the flow in residential areas has distinct peaks in the morning and evening of weekdays, while the flow in tourist areas has more peaks during holidays. Second, traffic flow is correlated in the temporal dimension. There are unidirectional influences between different time steps. To be specific, the traffic flow of a node and its spatial neighbors at the previous time step affects the features of the node at future time steps. As shown in Fig. 1(b), the traffic flow of node A at time step t_2 is influenced by the features of node A and its spatial neighbor B at time step t_1. If node A or node B has traffic congestion at t_1, the traffic flow of node A at t_2 will change significantly. Two shortcomings remain in modeling the dynamic spatial-temporal correlations of traffic flow caused by the above reasons.

(1) Modeling both spatial and temporal heterogeneities. The variations of traffic flow in temporal and spatial dimensions lead to the necessity of taking both spatial and temporal heterogeneities into account when modeling spatial-temporal dependencies. The study by Zhang *et al.* [24] proposed a two-step K-Means clustering model to divide subway stations into different categories for passenger flow forecasting. Ryu *et al.* [25] designed a clustering-based traffic flow prediction method by introducing spatial-temporal correlation matrices. These works took account of spatial heterogeneity but not temporal heterogeneity by clustering traffic nodes. STSGCN [26] and STFGNN [27] reflected temporal heterogeneity by deploying various modules at different time steps. However, they treated all the nodes as one cluster in each time step, so both ignored spatial heterogeneity. Failure to consider both spatial and temporal heterogeneities limits the abilities of existing models to capture spatial-temporal correlations.

(2) Considering temporal causality of spatial-temporal synchronous graph. DCRNN [28], Graph WaveNet [29], STGCN [30], AST-GCN [31], as well as T-GCN [32] utilized graph convolutions to capture spatial correlation, and applied GRU or CNN to modeling temporal dependency, but they all employed separate components to learn spatial and temporal features. STSGCN [26] and STFGNN [27] proposed localized spatial-temporal graph and spatial-temporal fusion graph, respectively, to model temporal and spatial dependencies simultaneously, but each node at time step t_m not only captured correlations from the historical time step t_{m-1} but also from the future time step t_{m+1} in their graph structures, which is contrary to the reality that the traffic flow of the previous time step determines the features of the next time step, not the other way around, and for time step t_m, the flow at t_{m+1} is unknown, so they failed to take into consideration temporal causality.

To overcome the mentioned shortcomings, a novel framework named Spatial-Temporal Heterogeneous and Synchronous Graph Convolution Network (STHSGCN) is proposed in this paper. Specifically, we designed separate dilated causal spatial-temporal synchronous graph convolutional networks (DCSTSGCNs) for various node clusters to reflect spatial heterogeneity and deployed different dilated causal spatial-temporal synchronous graph convolutional modules (DCSTSGCMs) for diverse time steps to take account of temporal heterogeneity. Moreover, causal spatial-temporal synchronous graph (CSTSG) is proposed to capture temporal causality in spatial-temporal synchronous learning. The main contributions of this paper can be summarized as follows:

- We proposed a novel STHSGCN to learn spatial-temporal correlations, which utilizes separate DCSTSGCNs for various node clusters to reflect spatial heterogeneity and deploys different DCSTSGCMs for diverse time steps to take account of temporal heterogeneity.
- We designed CSTSG to capture temporal causality in spatial-temporal synchronous modeling. In contrast to existing studies, CSTSG considers temporal causality by aggregating traffic features from the previous time step and the current time step instead of the next one.
- In order to evaluate the performance of our model, extensive experiments are conducted on four real-world datasets. The results demonstrate that STHSGCN obtains the best prediction compared with the baselines, indicating the effectiveness of our model.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 presents the preliminaries. Section 4 describes the details of the proposed methodology. In Section 5, extensive experiments are conducted on four real-world datasets, and the results are discussed. In the end, Section 6 concludes this paper.

2. Related work

2.1. Graph convolution network

Graph convolution network (GCN) has been playing a significant role in many graph-based studies, such as classifying diverse nodes and learning hidden representations. GCN can be defined in both spectral and spatial domains. In terms of spectral domain, GCN was formulated by finding the corresponding Fourier basis [27]. In [33], Bruna *et al.* proposed a learnable diagonal matrix to replace the convolution kernel, which enabled the graph convolution operation. Defferrard *et al.* addressed the problem of excessive computation by employing Chebyshev polynomials instead of convolution kernels [34]. The graph convolution operation was further simplified by considering only first-order Chebyshev polynomials [35], which greatly reduced the number of parameters. As for spatial domain, GCN was defined by generalizing spatial neighbors. Graph sample and aggregate (GraphSAGE) performed feature learning by sampling a fixed number of neighbors and then aggregating the information from neighbors [36]. Graph attention network (GAT) achieved adaptive learning of the weights for different neighbors by utilizing attention mechanisms to perform aggregation operations on neighboring nodes [37].

2.2. Traffic prediction

In recent years, neural networks have been widely applied in traffic flow prediction because of their outstanding ability to model spatial-temporal correlations [32]. Most of the studies integrated graph structures and temporal components to learn spatial-temporal dependencies [38]. DCRNN [28] proposed an encoder-decoder model, in which bidirectional random walks on the graph were utilized to model the spatial correlation, and GRU was employed to learn the temporal dependence. On the basis of this, Graph WaveNet [29] incorporated self-adaptive adjacency matrix with dilated causal convolution to strengthen the learning ability of the model. STGCN [30] first introduced GCN in capturing spatial features, and made use of CNN to learn temporal correlation, which had faster-training speed with fewer parameters compared to GRU. Based on this, ASTGCN [31] applied both spatial and temporal attention mechanisms to enhancing the model's performance in dependence learning. GCN followed by GRU was designed in T-GCN [32] to capture spatial-temporal correlations. Besides, multi-head attention was applied in GMAN [39] to spatial-temporal modeling due to its powerful learning capability. STSGCN [26] proposed localized spatial-temporal graph to learn spatial and temporal dependencies synchronously. Based on this, STFGNN [27] designed a temporal graph and deployed gated convolution modules to improve the accuracy of predictions, Auto-DSTSG [38] further presented dilated convolution and graph structure search approach to enhance the performance, AC-STSGCN [40] proposed a fused feature attention module to consider the influence of other traffic factors on traffic flow, and STGSA [41] designed a tailored graph aggregation method to model spatial-temporal dependencies more lightly.

However, the existing studies not only failed to take into account both spatial and temporal heterogeneities, but also ignored the temporal causality in spatial-temporal synchronous graph. Different from them, in this research we proposed a novel approach that treats traffic nodes as various clusters and deployed parallel modules for different time steps to take account of both spatial and temporal heterogeneities, we also designed causal spatial-temporal synchronous graph which is able to consider temporal causality.

3. Preliminaries

3.1. Problem definition

A road network with N traffic sensors can be represented by a graph structure as $\mathcal{G} = (V, E, A)$, where $V\,(|V| = N)$ denotes the set of traffic nodes, E denotes the connectivity between the nodes, and A denotes the spatial adjacency matrix between the nodes, which does not change with time. The graph signal matrix at time t can be expressed as $X_{\mathcal{G}}^{(t)} \in \mathbb{R}^{N \times C}$, where C denotes the dimensions of traffic features. The problem of traffic flow prediction can be defined as follows: Forecasting the future traffic signals $\left[X_{\mathcal{G}}^{(t+1)}, X_{\mathcal{G}}^{(t+2)}, \ldots, X_{\mathcal{G}}^{(t+T_p)}\right]$ of the road network $\mathcal{G}$ by learning a nonlinear transformation function $f(\cdot)$ when the historical traffic signals $\left[X_{\mathcal{G}}^{(t-T_h+1)}, X_{\mathcal{G}}^{(t-T_h+2)}, \ldots, X_{\mathcal{G}}^{(t)}\right]$ of $\mathcal{G}$ is given, where T_h denotes the time span of the historical traffic signals and T_p denotes the time span of the future traffic signals. The mathematical expression of the problem can be defined as:

$$\left[X_{\mathcal{G}}^{(t-T_h+1)}, \ldots, X_{\mathcal{G}}^{(t)}\right] \xrightarrow{f(\cdot)} \left[X_{\mathcal{G}}^{(t+1)}, \ldots, X_{\mathcal{G}}^{(t+T_p)}\right].$$

3.2. Fast-DTW

Dynamic Time Warping (DTW) is a classical algorithm used to compare the similarity between two time series. Suppose there are two time series $O = (o_1, o_2, \ldots, o_n)$ and $Q = (q_1, q_2, \ldots, q_m)$, DTW algorithm first calculates the distance matrix $M_{n \times m}$ between the two series by $\left|o_i - q_j\right|$, and then finds an optimal path from the upper left corner of the matrix $M_{n \times m}$ to the lower right corner, so that the sum of elements on the path is minimized, and its calculation process can be defined as:

$$M(i,j) = \left|o_i - q_j\right| + min\left(M(i-1,j), M(i,j-1), M(i-1,j-1)\right).$$

The $M(i,j)$, after the computation is done denotes the distance between time series O and Q. The smaller $M(i,j)$, the more similar O and Q are. By limiting the search range in the warping path to T, STFGNN [27] proposes fast-DTW algorithm. As in [27], fast-DTW algorithm is used in this paper to calculate the similarity between the time series of different nodes, which in turn generates the temporal graph, and the temporal adjacency matrix A_{TG} is formed by setting the threshold $a = 0.01$ that denotes the sparsity rate.

3.3. BiKmeans

BiKmeans algorithm is a typical Partition-based Method, which is improved on the basis of Kmeans algorithm, solving the shortcoming that Kmeans algorithm will fall into local optima [42]. BiKmeans algorithm uses the within-cluster sum of squared errors (SSE) as a measure of clustering effectiveness. For a sample $V = \left(V_{G_1}, V_{G_2}, \ldots, V_{G_K}\right)$, SSE is defined as follows:

$$SSE = \sum_{k=1}^{K} \sum_{v_i \in G_k} \|v_i - \mu_k\|_2^2 \tag{1}$$

where K denotes the number of clusters, G_k denotes the k-th cluster, and μ_k denotes the centroid of G_k. BiKmeans algorithm is based on the principle of SSE minimization, where nodes are first considered as one cluster, and then the existing clusters are continuously dichotomized with the aim of minimizing SSE until a predetermined number of clusters is obtained. Compared with Kmeans algorithm, SSE computed by BiKmeans algorithm is not only more stable but also smaller. In order to cluster traffic nodes, we combined graph auto-encoder (GAE) with BiKmeans algorithm to design BiKmeans algorithm based on GAE, which can learn the high-dimensional embedding of traffic features, and obtain a better clustering effect compared to the original BiKmeans algorithm.

4. Methodology

The overall architecture of STHSGCN is shown in Fig. 2(a). We trained GAE using traffic sequence data and spatial adjacency matrix to obtain a higher-order representation of traffic features, and then employed BiKmeans algorithm to cluster the traffic nodes. To better model the spatial-temporal correlations of various nodes group, we designed different Dilated Causal Spatial-Temporal Synchronous Graph Convolutional Networks (DCSTSGCNs) for various clusters, which reflects the spatial heterogeneity. In each DCSTSGCN, we deployed independent modules at different time steps, which takes into account temporal heterogeneity, and addresses the first shortcoming mentioned in Section 1. As for each cluster, we constructed Causal Spatial-Temporal Synchronous Graph (CSTSG) based on the spatial-temporal relationships of the traffic nodes within the cluster to capture spatial and temporal dependencies synchronously. Specifically, in CSTSG, traffic nodes at each time step only aggregate the spatial-temporal features from nodes at the previous time step, but not from those at the future time step, thus taking into consideration the temporal causality, which deals with the second shortcoming mentioned in Section 1.

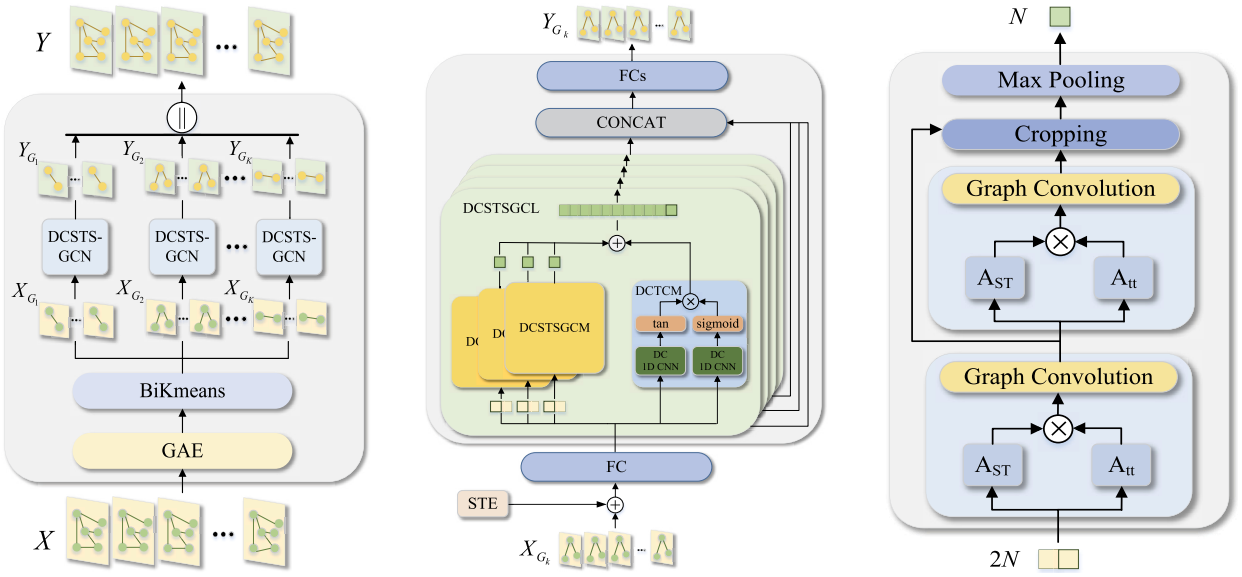
Fig. 2(b) presents the architecture of DCSTSGCN. In each DCSTSGCN, a fully connected layer is utilized to map the traffic sequence data from low dimensions to high dimensions, which enhance the expressiveness of the model, and then multiple Dilated Causal Spatial-Temporal Synchronous Graph Convolution Layers (DCSTSGCLs) are stacked to capture the deep spatial-temporal dependencies. In each DCSTSGCL, we designed parallel Dilated Causal Spatial-Temporal Synchronous Graph Convolution Modules (DCSTSGCMs) for different time steps, which take into account temporal heterogeneity. Specifically, each DCSTSGCM consists of several graph convolutions that incorporate a spatial-temporal attention mechanism to model the complex spatial-temporal correlations, which is illustrated in Fig. 2(c). Besides, Jumping Knowledge Network (JK-Net) between different graph convolutions are applied to mitigate vanishing gradient. In addition, cropping and max pooling are also included in a DCSTSGCM. We also deployed a Dilated Causal Time Convolution Module (DCTCM) in each DCSTSSGCL to extract the long-term dependencies in the temporal dimension. The output of each DCSGCL is obtained by summing the outputs of all DCSGCMs and the output of the DCTCM, and the output of the previous DCSGCL forms the input of the next DCSGCL. On the other hand, the outputs of all DCSGSGCLs are also concatenated by introducing JK-Net, and then pass through two fully connected layers to gain the output of DCSGSGCN. And all the outputs of DCSGSGCNs are put together to finally obtain the prediction results of STHSGCN.

4.1. BiKmeans based on GAE

BiKmeans algorithm is an improvement of Kmeans algorithm, which overcomes the problem caused by the possible extreme positions of incipient centroids when initialization [43], that makes it suitable for clustering traffic nodes. Traffic features are effective expressions of node attributes and can be utilized for node clustering, but limited by the expressiveness of low dimension, there are shortcomings in applying original traffic features as inputs to BiKmeans algorithm. GAE is a typical model for Representation Learning based on Deep Learning, which can learn the embedding of inputs in a high-dimensional feature space [44]. Therefore, we first employed GAE for training the traffic sequence data to obtain the high-dimensional embedding of traffic features, then combined it with BiKmeans algorithm to cluster traffic nodes, which is shown in Alg. 1.

GAE is composed of an encoder and a decoder. In the first place, the encoder is designed as a two-layer attention mechanism, which is mathematically defined as:

$$z_i^{(2)} = \sigma\left(\sum_{v_j \in N e_i} \alpha_{i,j}^{(1)} W^{(1)} \left(\sigma\left(\sum_{v_j \in N e_i} \alpha_{i,j}^{(0)} W^{(0)} x_j\right)\right)\right),$$



(a) STHSGCN architecture (b) DCSTSGCN architecture (c) DCSTSGCM architecture

Fig. 2. Detailed framework of STHSGCN. (a) indicates the overall architecture of STHSGCN. BiKmeans based on GAE is employed to cluster traffic nodes, and parallel DCSTSGCNs are deployed for various clusters. (b) presents the architecture of DCSTSGCN. Each DCSTSGCN contains four DCSTSGCLs with dilated step sizes. Independent DCSTSGCMs for different time steps as well as a DCTCM are designed in each DCSTSGCL. (c) illustrates the architecture of DCSTSGCM. Several graph convolutions that incorporate a spatial-temporal attention mechanism are followed by cropping operation as well as max pooling.

where $z_i^{(2)}$ denotes the high-dimensional embedding of node v_i, $\sigma(\cdot)$ denotes the sigmoid activation function, Ne_i denotes the set of neighboring nodes of v_i, $W^{(0)}$ and $W^{(1)}$ denote the weight matrices in the first layer and the second layer respectively, $\alpha_{i,j}^{(0)}$ and $\alpha_{i,j}^{(1)}$ individually denotes the attention coefficients between v_i and v_j in the two different layers. The attention coefficient $\alpha_{v_i,v_j}^{(l)}$ is obtained in three steps. Firstly, the nonlinear transformation function is defined as:

$$g(x) = LeakyReLU(xW + b),$$

where W, b are learnable parameters, and $LeakyReLU(\cdot)$ is the activation function. Secondly, the correlation coefficient is calculated by employing the scaled dot product:

$$e_{v_i,v_j}^{(l)} = \frac{\langle g_1\left(z_{v_i}^{(l)}\right), g_2\left(z_{v_j}^{(l)}\right)\rangle}{\sqrt{C^{(l)}}}, \quad l = 0, 1 \quad ,$$

where $C^{(l)}$ denotes the number of features in the l-th layer, and then the correlation coefficient is normalized by the softmax function to obtain the attention coefficient:

$$\alpha_{v_i,v_j}^{(l)} = \frac{\exp\left(e_{v_i,v_j}^{(l)}\right)}{\sum_{v_r \in Ne_i} \exp\left(e_{v_i,v_r}^{(l)}\right)}, \quad l = 0, 1 \quad .$$

In the next place, the decoder is formulated as the inner product of node pairs as:

$$\hat{A}_{ij} = \sigma\left(z_i^T z_j\right).$$

After that, GAE is trained in combination with the reconstruction loss function:

$$L_{GAE} = -\sum_{i,j=1}^{N} A_{ij} \log \hat{A}_{ij}.$$

Finally, the embedding of the nodes are obtained: $Z \in \mathbb{R}^{T \times N \times C'}$, where T denotes the length of time series, and C' denotes the number of high-dimensional features.

The embedding Z is introduced to BiKmeans algorithm, in which Euclidean Distance is adopted as the distance between nodes, which can be formulated as:

$$d_{v_i,v_j} = \sqrt{(Z_i - Z_j)^2},$$

where Z_i, $Z_j \in \mathbb{R}^{T\times C'}$ are the high-dimensional features of v_i, v_j respectively. According to Eq. (1), SSE can be written as:

$$SSE = \sum_{k=1}^{K} \sum_{v_i \in G_k} d_{v_i,\mu_k}^2,$$

where G_k denotes the k-th cluster, μ_k denotes the centroid of G_k. Then, as shown in Alg. 1, all nodes are first considered as one cluster, and then the existing clusters are continuously dichotomized on the principle of minimizing SSE until the number of clusters reaches K, which is predetermined. Different K and the corresponding SSE can be obtained by Alg. 1. The curve of K and SSE within a certain range is in the shape of an elbow, and the value of K corresponding to the turning point of the elbow is the optimal number of clusters.

Algorithm 1 BiKmeans base on GAE.

Input: The traffic series data: $X = (X_1, X_2, \ldots, X_N)$
Output: The cluster list: $CL = (V_{G_1}, V_{G_2}, \ldots, V_{G_K})$, the minimum SSE.

```
Build GAE with a two-layer attention mechanism;
Train GAE to obtain the high-dimensional features: Z;
Define distance d_{v_i,v_j} and SSE according to Z;
Initialization: n = 1, j = 1, CL = V_{G_1} = V;
while n < K do
    SSE_{max_re} = 0;
    for i in n do
        Dichotomize V_{G_k} into V_{G_k}[0], V_{G_k}[1];
        if SSE_{before} − SSE_{after} > SSE_{max_re} then
            SSE_{max_re} = SSE_{before} − SSE_{after};
            j = i; SSE = SSE_{after}
        end if
    end for
    Remove V_{G_j} from CL, Add V_{G_j}[0], V_{G_j}[1] into CL;
    n = n + 1, j = 1
end while
Return CL, SSE
```

After the clustering is completed, the traffic nodes are divided into K clusters:

$$V = \left(V_{G_1}, V_{G_2}, \ldots, V_{G_K}\right),$$

where $V_{G_k}, 1 \le k \le K$ denotes the k-th cluster, the number of traffic nodes is:

$$N = \left(N_{G_1} + N_{G_2} + \ldots + N_{G_K}\right),$$

where $N_{G_k}, 1 \le k \le K$ denotes the number of traffic nodes in the k-th cluster, and the input sequence can be formulated as:

$$X = \left(X_{G_1}, X_{G_2}, \ldots, X_{G_K}\right),$$

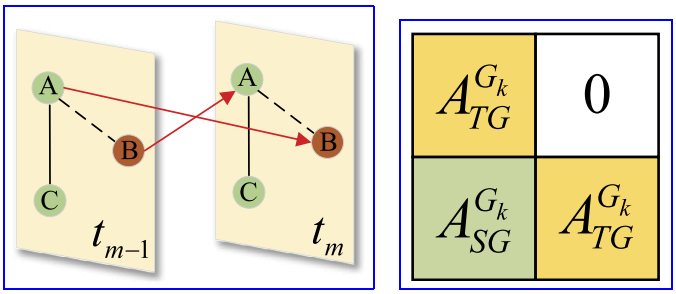
where $X_{G_k}, 1 \le k \le K$ denotes the embedding of the k-th cluster.

4.2. Causal spatial-temporal synchronous graph construction

In order to capture both temporal and spatial correlations, STSGCN [26] designed Localized Spatial-Temporal Graph, which is comprised of three time steps, and its adjacency matrix has the shape of $3N\times 3N$, containing the spatial adjacency matrix A_{SG} and the temporal connectivity matrix A_{TC}. In Localized Spatial-Temporal Graph, each node is connected to both its spatial neighbors at the same time step and itself at the previous and subsequent time steps. By introducing fast-DTW, STFGNN [27] generated the temporal graph and temporal adjacency matrix A_{TG}, then the Spatial-Temporal Fusion Graph is proposed, where each node has connections not only with its spatial and temporal neighbors but also with nodes in the same temporal pattern, its adjacency matrix is extended to $4N \times 4N$. In these works, each node captures correlations from the previous time step and the next time step simultaneously, but for the nodes at time t, the features at time t_{m+1} are unknown, so it is unreasonable to extract spatial-temporal features from time t_{m+1} for obtaining the hidden state at time t, which ignores the temporal causality.

To deal with the shortcoming, Causal Spatial-Temporal Synchronous Graph (CSTSG) is proposed. As shown in Fig. 3, two adjacent time steps t_{m-1} and t_m are included in each CSTSG. For each node cluster V_{G_k}, the causal spatial-temporal adjacency matrix $A_{ST}^{G_k}$ is designed as $2N \times 2N$ with $[A_{TG}^{G_k}, A_{TG}^{G_k}]$ on the main diagonal, indicating that each node in V_{G_k} is connected to the nodes in the same time pattern at the same time step, where $A_{TG}^{G_k}$ is the temporal adjacency matrix of V_{G_k}, which is calculated by fast-DTW and defined as:

$$A_{TG}^{G_k} = \begin{cases} 1, & \text{if } v_i, v_j \text{ in the same pattern}, v_i, v_j \in V_{G_k} \\ 0, & \text{else} \end{cases}.$$



(a) Description of CSTSG **(b)** Structure of the spatial-temporal adjacency matrix

Fig. 3. Description of CSTSG and the spatial-temporal adjacency matrix. (a) indicates that each CSTSG contains two adjacent time steps t_{m-1} and t_m. (b) is the structure of the spatial-temporal adjacency matrix.

The counter-diagonal is $[A_{SG}^{G_k}, 0]$, denoting the one-way connection between nodes in V_{G_k} at time step t and their spatial neighbors at time step $t-1$, which ensures that each node only obtains spatial-temporal features from the previous time step. $A_{SG}^{G_k}$ is the spatial adjacency matrix of V_{G_k}, which is generated by the spatial graph structure and can be formulated as:

$$A_{SG}^{G_k} = \begin{cases} 1, & \text{if } v_i \text{ connects to } v_j, \quad v_i, v_j \in V_{G_k} \\ 0, & \text{else} \end{cases}$$

By proposing CSTSG, we could not only model temporal and spatial dependencies simultaneously, but also take account of the temporal causality.

4.3. Dilated causal spatial-temporal synchronous graph convolutional network

In addition to temporal heterogeneity, spatial heterogeneity should also be taken into consideration in modeling spatial-temporal correlations. To achieve this goal, we deployed a DCSTSGCN for each cluster, Fig. 2(b) illustrates the architecture of DCSTSGCN. Four DCSTSGCLs with different step sizes are stacked in each DCSGCN to extract long-term spatial-temporal dependencies. A DCSTSGCL consists of parallel DCSTSGCMs and a DCTCM. A DCSTSGCN also contains Spatial-Temporal Embedding (STE), input layer, Jumping Knowledge Network (JK-Net), and output layer. It can be described as follows:

$$Y_{G_k} = FCs\left(CONCAT\left(DCSTSGCLs\left(FC\left(X_{G_k}\right)\right)\right)\right),$$

where $X_{G_k} \in \mathbb{R}^{T_h \times N_{G_k} \times C}, Y_{G_k} \in \mathbb{R}^{T_p \times N_{G_k} \times C}$ are the input and output of the k-th DCSTSGCN, respectively. $FC(\cdot), FCs(\cdot)$ means fully connected layers, $CONCAT(\cdot)$ denotes the concatenate operation. The outputs of K DCSTSGCNs form the output of the STHSGCN:

$$Y = \left(Y_{G_1}, Y_{G_2}, \ldots, Y_{G_K}\right) \in \mathbb{R}^{T_p \times N \times C}. \tag{2}$$

4.3.1. Dilated causal spatial-temporal synchronous graph convolutional layer

In order to expand the receptive field of DCSTSGCN, we employed dilated rather than fixed step sizes to construct CSTSGs in different DCSTSGCLs. Specifically, the step size of the l-th layer is defined as:

$$d^l = 2^{min(l-1,2)}, \quad 1 \le l \le 4. \tag{3}$$

The input sequence of the l-th DCSGCL can be described as:

$$X^l = \left(x_1^l, x_2^l, \ldots, x_{T_l}^l\right),$$

where T_l denotes the length of the input sequence, we selected time step pairs with distance d^l to construct CSTSGs: $\left[\left(x_1^l, x_{1+d^l}^l\right), \left(x_2^l, x_{2+d^l}^l\right), \ldots, \left(x_{T_l-d^l}^l, x_{T_l}^l\right)\right]$, which is shown in Fig. 4. By stacking four DCSGCLs with step sizes $[1,2,4,4]$ which is calculated by Eq. (3) respectively, the receptive field of each DCSGCN can cover up to 12 input time steps, which is exactly the length of the input traffic sequence in this paper.

Traffic sequence data at different times exhibit various patterns, which is temporal heterogeneity, so it is more reasonable to utilize multiple modules instead of one to capture temporal correlation for different time steps. In each DCSGCL, we deployed diverse DCSGCMs for different CSTSGs, so the number of DCSGCMs deployed in the l-th DCSGCL can be calculated as:

$$M^l = T_l - d^l.$$

Each time step pair as well as the spatial-temporal adjacency matrix $A_{ST}^{G_k}$ are fed into a DCTSGCM to model the spatial and temporal dependencies synchronously. Moreover, the input sequence of the l-th DCSGCL is also treated as the input of DCTCM.

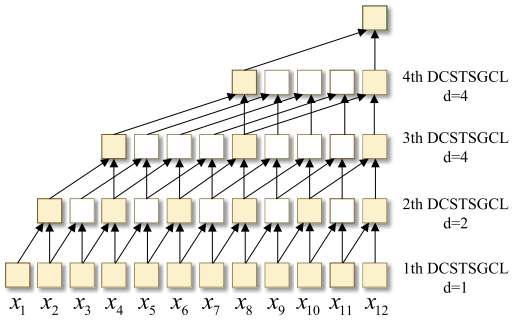


Fig. 4. Dilated step sizes in four DCSTSGCLs.

4.3.2. Dilated causal spatial-temporal synchronous graph convolutional module

The architecture of DCSTSGCM is illustrated in Fig. 2(c). Multiple graph convolutions with spatial-temporal attention mechanisms are stacked in DCSTSGCM to extract deep spatial-temporal features, cropping operation as well as max pooling are also included.

First, we enhanced the module's ability to characterize the relationships between nodes and their spatial-temporal neighbors by designing a spatial-temporal attention matrix A_{tt} with the shape of $2N_{G_k} \times 2N_{G_k}$. On the main diagonal, there are two temporal attention matrices $\left[E^{1'}, E^{2'}\right]$, denoting the attention of nodes and their temporal neighbors at the same time step, whose computational formula is defined as:

$$E^n = \sigma\left(\left(X^{(l,m)}V_1\right)\left(X^{(l,m)}V_2\right)^T + b_t\right), \quad n = 1,2$$

$$E^{n'}_{i,j} = \frac{\exp\left(E^n_{i,j}\right)}{\sum_{r=1}^{N_{G_k}} \exp\left(E^n_{i,r}\right)}, \quad n = 1,2$$

where $X^{(l,m)} \in \mathbb{R}^{2N_{G_k} \times D}$ denotes the input of m-th DCSTSGCM in l-th DCSTSGCL, $V_1, V_2 \in \mathbb{R}^{D \times N_{G_k}}$ and $b_t \in \mathbb{R}^{N_{G_k} \times N_{G_k}}$ are learnable parameters. On the counter-diagonal, there are a spatial attention matrix and an all-zero matrix $\left[S', 0\right]$, denoting the attention of nodes in the current time step with their spatial neighbors at the previous time step. The mathematical equation of S' is formulated as:

$$S = \sigma\left(\left(X^{(l,m)}U_1\right)\left(X^{(l,m)}U_2\right)^T + b_s\right),$$

$$S'_{i,j} = \frac{\exp\left(S_{i,j}\right)}{\sum_{r=1}^{N_{G_k}} \exp\left(S_{i,r}\right)},$$

where $U_1, U_2 \in \mathbb{R}^{D \times N_{G_k}}$ and $b_s \in \mathbb{R}^{N_{G_k} \times N_{G_k}}$ are learnable parameters.

Then the spatial-temporal adjacency matrix used in graph convolution can be obtained by:

$$A'_{ST} = A^{G_k}_{ST} \otimes A_{tt}$$

where $\otimes$ denotes element-wise product. Considering that adding spatial-temporal attention matrices in all the DCSTSGCLs will over-inflate the number of learnable parameters and increase the training difficulty, we only arranged attention mechanisms in DCSTSGCMs of the first DCSTSGCL, which is able to reduce the complexity of the model while ensuring that each input time step has a corresponding spatial-temporal attention matrix.

Employing GLU as the activation function enables a richer expression of the convolution operation than utilizing the sigmoid function alone. Thus, we introduced Graph Convolution with GLU as the mapping function to extract spatial-temporal features, the mathematical expression can be formulated as:

$$h^r = \left(A'_{ST}h^{r-1}W_1 + b_1\right) \otimes \sigma\left(A'_{ST}h^{r-1}W_2 + b_2\right),$$

where h^{r-1}, $h^r \in \mathbb{R}^{2N_{G_k} \times D}$ are the input and output of the r-th graph operation respectively, $W_1, W_2 \in \mathbb{R}^{D \times D}$ as well as $b_1, b_2 \in \mathbb{R}^{2N_{G_k} \times D}$ are learnable parameters.

After the convolution operation, the spatial-temporal features of time step t_{q-1} have already been aggregated in time step t_q, and discarding time step t_{q-1} can reduce the complexity without affecting the module's ability to capture spatial-temporal dependencies. Therefore, we removed the previous time step from each CSTSG by cropping operation, which changes the shape of the output from $h^r \in \mathbb{R}^{2N_{G_k} \times D}$ to $h^r_c \in \mathbb{R}^{N_{G_k} \times D}$.

Finally, the outputs of multiple graph convolutions are aggregated by max pooling to obtain the output of DCSTSGCM:

$$h_{MP} = MaxPooling\left(h^1_c, h^2_c, \ldots, h^R_c\right) \in \mathbb{R}^{N_{G_k} \times D},$$

where R denotes the number of graph convolutions. The output of all DCSTSGCMs in l-th DCSTSGCL can be described as:

$$Y_{M^l} = \left(h^1_{MP}, h^2_{MP}, \ldots, h^{M^l}_{MP}\right) \in \mathbb{R}^{M^l \times N_{G_k} \times D}. \tag{4}$$

4.3.3. Dilated causal temporal convolutional module

In the time dimension, both local and global dependencies are important [27,38]. Various DCSTSGCMs have diverse learnable parameters at different time steps, which makes them more suitable for capturing temporal dependency in the local temporal range, but also leaves them with limitations in extracting global dependency of the input sequence. Therefore, 1D dilated causal convolutions sharing weights in long term is employed to extract the global temporal correlation, which also integrates the gating mechanism. The mathematical expression of 1D dilated causal convolutions can be defined as:

$$Y_T = tanh(\Theta_1 * X^l + b_1) \otimes \sigma(\Theta_2 * X^l + b_2) \in \mathbb{R}^{M^l \times N_{G_k} \times D}, \tag{5}$$

where X^l is the input of l-th DCSTSGCL, $tanh(\cdot)$ means tanh mapping function, Θ_1, Θ_2 denote two 1D dilated causal convolutions respectively, b_1, b_2 are learnable parameters. The kernel sizes of Θ_1 and Θ_2 are always 2, and the step sizes are the same as the step sizes of current DCSTSGCL, which are $[1, 2, 4, 4]$.

The output of each DCSTSGCL consists of the outputs of all DCSTSGCMs and the output of DCTCN, which can be described as:

$$Y^l = Y_{M^l} + Y_T \in \mathbb{R}^{M^l \times N_{G_k} \times D}, \tag{6}$$

where Y_{M^l}, Y_T come from Eq. (4) and Eq. (5), respectively.

4.3.4. Other components

Spatial-temporal embedding Integrating nodes at different time steps into the same graph will blur the temporal and spatial attributes to some extent [26]. To address this issue, temporal embedding $T_{emb} \in \mathbb{R}^{T \times C}$ and spatial embedding $S_{emb} \in \mathbb{R}^{N \times C}$ are added in the input sequence to improve the model's ability of capturing spatial-temporal dependencies:

$$X_{\mathcal{G}+t_{emb}+s_{emb}} = X_{\mathcal{G}} + T_{emb} + S_{emb} \in \mathbb{R}^{T \times N \times C}.$$

Input layer and output layer In DCSTSGCN, we first transformed the input sequence by a fully connected layer to obtain the high-dimensional embedding of traffic features, which can strengthen the expressiveness. As for output layer, by introducing JK-Net, we performed a concatenate operation on the outputs of DCSTSGCLs, and then went through two fully connected layers to obtain the output of DCSTSGCN, which can be described as:

$$Y_{G_k} = ReLU\left(\left(\left(Y^1, Y^2, \ldots, Y^L\right)\Theta_1 + b_1\right)\Theta_2 + b_2\right).$$

Where $Y_{G_k} \in \mathbb{R}^{T_p \times N_{G_k} \times C}$, L is the number of DCSTSGCLs in the k-th DCSTSGCN, $Y_l\,(1 \le l \le L)$ comes from Eq. (6), $\Theta_1, \Theta_2, b_1, b_2$ are learnable parameters. All the outputs of DCSTSGCN form the output of STHSGCN, which can be obtained according to Eq. (2).

4.4. Loss function

We selected smooth L1 loss as the loss function. Compared to L1 loss, the smooth L1 loss modified the unsmooth problem at the zero-point [45].

$$L(Y, \hat{Y}) = \begin{cases} \frac{1}{2}\left(Y - \hat{Y}\right)^2/\delta, & if \left|Y - \hat{Y}\right| \le \delta, \\ \left|Y - \hat{Y}\right| - \frac{1}{2}\delta, & otherwise \end{cases},$$

where $Y, \hat{Y}$ denote ground truth and predicted values respectively, δ is a threshold parameter.

5. Experiments and discussion

5.1. Datasets

We evaluated the performance of STHSGCN on four public datasets: PEMS03, PEMS04, PEMS07, PEMS08 [26], which are generated by California Transportation Agency Performance Measurement System (PeMS). These datasets are collected by sensors deployed on the freeways in the State of California every five minutes, which means there are 288 points in the flow data for each day. The adjacencies between sensors are constructed based on the actual road network. We standardized the traffic features of each dataset by Z-score normalization. Neighborhood values are employed to fill in the missing data. For these four traffic flow datasets, we assumed that their distributions vary periodically over time. The detailed information of four datasets is illustrated in Table 1.

5.2. Experiment settings and evaluation metrics

All datasets are split into training sets, validation sets and test sets in the ratio of 6: 2: 2. To evaluate the fitness of the model, the validation sets during training are continuously used to test the validity of the model for unknown data. We made use of the

Table 1
Dataset description.

Datasets	Sensors	Time Steps	Traffic Features
PEMS03	358	26208	flow
PEMS04	307	16992	flow,speed,occupancy
PEMS07	883	28224	flow,speed,occupancy
PEMS08	170	17856	flow

historical data in the past hour to forecast the traffic flow in the next hour. Experiments are implemented by Pytorch 1.13 under the environment with AMD Ryzen 5 5600G CPU and NVIDIA RTX 3060 GPU. The hyperparameters are determined by the performance of the model. Each DCSTSGCM comprises 3 graph convolutions, and the hidden representations of convolutional operations are set to [64,64,64]. The output dimension of FCs in the input layer and output layer are set to [64], [128,1], respectively. We utilized Adam as the optimizer. The batch sizes for PEMS03, PEMS04, PEMS07, PEMS08 are set to [64,64,16,64]. The learning rate is initialized to 0.003, and then it is multiplied by 0.3 at the 20th and the 40th epoch, which can speed up training. Early stopping with tolerance 20 for 200 epochs is applied to training process.

We employed mean absolute error (MAE), mean absolute percentage error (MAPE) and root mean square error (RMSE) as the evaluation metrics, which can be defined as follows:

$$MAE\left(y_i, \hat{y}_i\right) = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|,$$

$$MAPE\left(y_i, \hat{y}_i\right) = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|y_i - \hat{y}_i\right|}{y_i},$$

$$RMSE\left(y_i, \hat{y}_i\right) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2},$$

where y_i, $\hat{y}_i$ denote the ground truth and the prediction values, respectively. The smaller the three evaluation metrics, the better the performance of the model.

5.3. Baseline methods

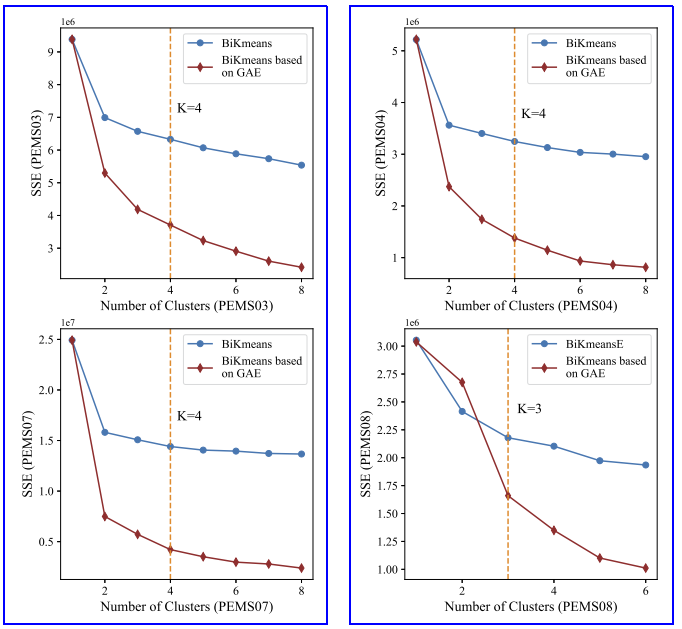
We compared STHSGCN with the following six models:

- GRU: Gate Recurrent Unit, it is a variant of LSTM, which consists of reset gates and update gates. The hidden dimensions of the two gates are set as 64.
- T-GCN: Temporal Graph Convolutional Network, which integrates GCNs into GRUs. The number of GRUs is set as 64 in this model [32].
- DCRNN: Diffusion Convolution Recurrent Neural Network, which combines graph convolutions with encoder-decoder gated recurrent units. The hidden dimensions are set as 64 in this model [28].
- STGCN: Spatio-Temporal Graph Convolution Network, which utilizes GCNs and 1D convolution units to model the spatio-temporal dependencies. It is comprised of two ST-Conv Blocks and an output layer. The hidden dimensions are set as 64 [30].
- STSGCN: Spatial-Temporal Graph Convolutional Network, which introduces localized spatial-temporal graphs and spatial-temporal synchronous graph convolutional modules to capture the dependencies between traffic nodes and their spatial-temporal neighbors. The hidden dimensions in each module are set as 64 [26].
- STFGNN: Spatial-Temporal Fusion Graph Neural Network, which designs temporal graphs and applies spatial-temporal fusion graph neural modules as well as gated convolution modules to modeling the hidden spatial-temporal correlations. The hidden dimensions are set as 64 in this model [27].

5.4. Experimental results

The relationships between the number of clusters and SSE on PEMS03, PEMS04, PEMS07, PEMS08 can be found in Fig. 5. It is observed that SSE decreases gradually with the increasing number of clusters, and BiKmeans based on GAE consistently outperforms original BiKmeans. The turning points on the four datasets are chosen as [4, 4, 4, 3] according to the shape of elbows, respectively, which also represent the optimal numbers of clusters. In addition, we conducted parameters experiments to validate the optimum numbers of clusters, which will be shown in section 5.6.

Table 2 presents the performance of STHSGCN and six baselines for traffic flow prediction on PEMS03, PEMS04, PEMS07 and PEMS08. By means of numerical results, it is verified that STHSGCN overwhelmingly outperforms the baselines on all four datasets. In order to evaluate the ability of models to capture spatial-temporal dependencies at different time steps, a comparison of the prediction performance at 12 time steps on PEMS03 is made, as shown in Fig. 6, STHSGCN consistently outperforms all baselines. Moreover, to make the capacities for prediction intuitive visually, we compared the prediction results of best baselines and the



(a) Number of clusters and SSE on PEMS03 and PEMS04

(b) Number of clusters and SSE on PEMS07 and PEMS08

Fig. 5. Relationships between the number of clusters and SSE on the four datasets. (a) illustrates the relationships on PEMS03 and PEMS04. (b) presents the relationships on PEMS07 and PEMS08.

Table 2
Performance comparison of STHSGCN and baselines for traffic flow prediction.

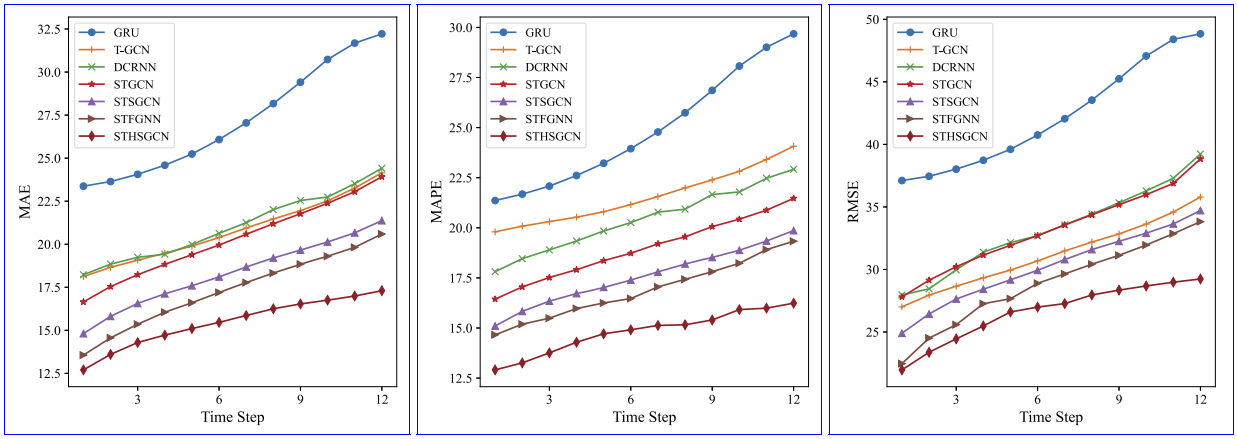
Datasets	Metric	GRU	T-GCN	DCRNN	STGCN	STSGCN	STFGNN	STHSGCN
PEMS03	MAE	27.19	20.83	21.07	20.29	18.30	17.33	**15.46**
	MAPE(%)	24.92	21.58	20.43	18.98	17.58	16.90	**14.81**
	RMSE	42.24	31.18	33.23	33.08	30.20	29.04	**26.61**
PEMS04	MAE	34.13	26.02	27.57	25.37	22.38	20.14	**19.50**
	MAPE(%)	21.53	17.08	18.29	15.50	15.03	13.76	**12.89**
	RMSE	49.98	38.19	42.07	39.11	35.33	32.62	**31.39**
PEMS07	MAE	37.79	30.37	31.29	30.91	25.15	23.76	**21.39**
	MAPE(%)	16.83	13.83	15.09	14.49	10.74	9.96	**9.21**
	RMSE	56.72	43.39	47.21	47.12	40.79	38.48	**34.98**
PEMS08	MAE	28.12	21.37	21.21	21.39	17.72	16.99	**15.50**
	MAPE(%)	16.92	13.66	13.08	13.13	11.61	10.96	**9.95**
	RMSE	41.85	30.69	31.43	31.42	27.21	26.80	**24.51**

proposed model for predicting traffic flow on a random day of PEMS08, as illustrated in Fig. 7, STHSGCN fits the ground-truth better. The better-fitting predictions and higher accuracy indicate that STHSGCN is more suitable for predicting road traffic flows.

5.5. Ablation experiments

To further verify the effectiveness of different modules in STHSGCN, various ablation experiments are conducted on PEMS03 and PEMS08. As illustrated in Table 3, we designed seven variants of STHSGCN and compare STHSGCN with these variants.

- N-Cluster, which removes BiKmeans based on GAE, and treats all the nodes as one cluster.
- N-Causality, which replaces causal spatial-temporal synchronous graph with noncausal spatial-temporal synchronous graph, that means the shape of spatial-temporal adjacency matrix remains $2N \times 2N$, while the counter-diagonal becomes $[A_{SG}^{G_k}, A_{SG}^{G_k}]$.
- N-Dilation, which employs fixed instead of dilated step sizes to construct CSTSGs in all DCSTSGCLs. The fixed size is set as 1.
- N-DCTCM, which removes DCTCMs from each DCSTSGCL.
- N-STAtt, which removes spatial-temporal attention mechanism from DCSTSGCMs.
- N-JK-Net, which removes the concatenate operation on the outputs of four DCSTSGCLs.
- N-STE, which removes spatial embedding and temporal embedding from our model.



(a) MAE (b) MAPE (c) RMSE

Fig. 6. Prediction performance at 12 time steps on PEMS03. (a) illustrates MAE of the seven models. (b) shows MAPE of STHSGCN and the baselines. (c) presents RMSE of STHSGCN and six other models.

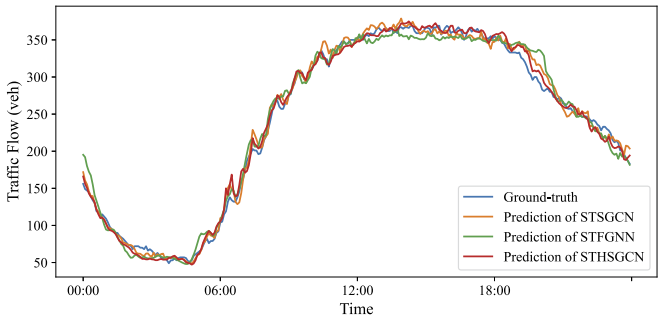
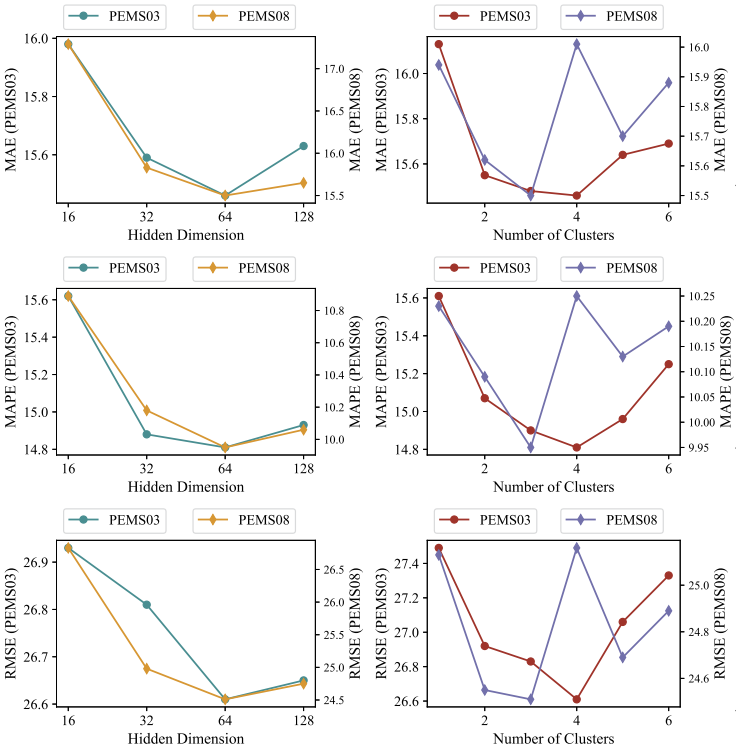


Fig. 7. Prediction results of STSGCN, STFGNN and STHSGCN.

Table 3
Ablation experiments.

Dataset	Model&Variants	MAE	MAPE(%)	RMSE
PEMS03	STHSGCN	**15.46**	**14.81**	**26.61**
	N-Cluster	16.13	15.61	27.49
	N-Causality	15.82	15.13	26.91
	N-Dilation	15.62	15.04	26.88
	N-DCTCM	16.49	15.63	28.24
	N-STAtt	16.36	15.48	27.72
	N-JK-Net	16.72	15.96	29.03
	N-STE	15.66	15.11	26.98
PEMS08	STHSGCN	**15.50**	**9.95**	**24.51**
	N-Cluster	15.94	10.23	25.13
	N-Causality	15.84	10.16	25.02
	N-Dilation	15.88	10.23	24.95
	N-DCTCM	16.77	10.78	26.01
	N-STAtt	16.23	10.66	25.60
	N-JK-Net	16.44	10.73	25.75
	N-STE	16.04	10.41	25.18

Table 3 shows that STHSGCN obtains minimal metrics compared with all the baselines on both PEMS03 and PEMS08. In contrast with the metrics of N-Cluster, STHSGCN improves 4.15%, 5.12% and 3.20% in terms of MAE, MAPE and RMSE on PEMS03. In the meantime, it also achieves 2.76%, 2.74% and 2.47% improvements for MAE, MAPE and RMSE on PEMS08, which clarifies the effectiveness of our BiKmeans algorithm on learning spatial heterogeneity. In contrast with the metrics of N-Causality, STHSGCN improves 2.28%, 2.12% and 1.11% in terms of MAE, MAPE and RMSE on PEMS03. Meanwhile, it also outperforms N-Causality with 2.15%, 2.07% and 2.04% improvements for MAE, MAPE and RMSE on PEMS08, which verifies the effectiveness of causal spatial-temporal synchronous graph on model temporal correlations. In contrast with the metrics of N-STAtt, STHSGCN improves



(a) Results of different hidden dimensions **(b)** Results of different numbers in clustering

Fig. 8. Experimental results of parameters on PEMS03 and PEMS08. (a) shows the results of different hidden dimensions. (b) illustrates the results of different numbers in clustering.

5.50%, 4.33% and 4.00% in terms of MAE, MAPE and RMSE on PEMS03. In the meanwhile, it also achieves 4.50%, 6.66% and 4.26% improvements for MAE, MAPE and RMSE on PEMS08, which demonstrates the effectiveness of spatial-temporal attention mechanism on learning spatial-temporal dependencies.

5.6. Parameters experiments

In order to validate the effectiveness of STHSGCN, parameters experiments are conducted on PEMS03 as well as PEMS08. The parameters consist of the hidden dimension D in graph convolutions and the number of clusters K in BiKmeans based on GAE, Fig. 8 illustrates the experimental results. It can be observed from Fig. 8(a) that STHSGCN obtains minimal MAE, MAPE, RMSE on both PEMS03 and PEMS08 when D is set as 64. As for the number of clusters K, Fig. 8(b) indicates that STHSGCN achieves optimal results on PEMS03 and PEMS08 when K is set as [4, 3] respectively, which is exactly the turning points chosen in Fig. 5, revealing that appropriate number of clusters contributes to learning spatial-temporal correlations.

5.7. Discussion

The experimental results in Section 5.4 illustrate the superiority of STHSGCN over the baselines. GRU is only applicable to feature extraction of time series, and cannot take into account the spatial relationships between nodes, so the performance is not ideal. T-GCN, DCRNN and STGCN extract spatial features by graph convolution, and learn temporal features using GRU or 1D CNN, leading to better performances than the model that only considers time series, but they fail to capture spatial-temporal dependencies synchronously. STSGCN and STFGNN model spatial-temporal correlations synchronously by Localized Spatial-Temporal Graph and Spatial-Temporal Fusion Graph, respectively, deploying parallel modules in the temporal dimension to reflect temporal heterogeneity, but they employ shared networks for all nodes to capture spatial-temporal dependencies and thus fail in considering spatial heterogeneity. In addition, they are also unable to capture temporal causality in spatial-temporal synchronous modeling, resulting in limiting the ability to extract spatial-temporal features. Our model first employs BiKmeans based on GAE to cluster traffic nodes at different spatial locations, and then takes advantage of separate DCSTSGCNs for various node clusters and different DCSTSGCMs for diverse time steps, leading to a consideration of both spatial and temporal heterogeneities. Moreover, our CSTSG not only learns spatial-temporal correlations synchronously, but also reflects temporal causality in the graph structure. These designs enhance the

performance of STHSGCN for extracting spatial-temporal features. The advantages also allow our model to predict traffic flows in the road network more accurately, further improving the availability of ITS.

Ablation Experiments in Section 5.5 indicate that the parallel networks for diverse clusters in our model are able to characterize spatial heterogeneity. Causal Spatial-Temporal Synchronous Graph of STHSGCN can extracts spatial-temporal features more accurately by taking temporal causality into account. Spatial-temporal attention mechanism also contributes to the enhancement of learning spatial-temporal dependencies. Besides, the comparisons with other variants also verify the effectiveness of corresponding parts on capturing spatial-temporal features. These are the reasons why STHSGCN has the ability to outperform the variants.

In addition, parameters experiments in Section 5.6 show that appropriate hidden dimension D and number of clusters K strengthen the capacity for modeling spatial-temporal dependencies. When D is set as 16 or 32, inefficient representation of hidden dimension limits the ability of our model to capture spatial-temporal features, while setting D as 128 leads to over-fitting. When K is too small, the spatial heterogeneity of nodes cannot be fully considered, while an excessive number of clusters results in a fragmented graph structure that losses part of the adjacency relationships between nodes.

5.8. Limitations

Despite various experiments demonstrating the advantages of the proposed method, some limitations still exist. First, there is a lack of considering external factors, such as temperature, visibility and holiday. Second, the structure of the graph in the model is fixed and does not change with time. In the future, we intend to take external factors into account, we also plan to explore the influence of dynamic graph structure on traffic flow prediction.

5.9. Policy recommendations

Research on traffic flow forecasting is clearly influenced by transportation policies, especially those regarding the construction of intelligent transportation systems. Traffic feature acquisition equipment is the main means to obtain traffic flow from the road network, and extensive placement of acquisition equipment can strengthen the basis of traffic flow prediction. Therefore, we suggest that the competent authorities should introduce more policies to support the construction of intelligent traffic systems. The lag in the construction of traffic feature acquisition equipment in some road networks is a possible reason why future research cannot be carried out smoothly.

6. Conclusion

In this paper, we proposed a spatial-temporal heterogeneous and synchronous graph convolution networks to model the dynamic spatial-temporal dependencies in traffic flow prediction. STHSGCN not only takes into consideration both spatial and temporal heterogeneities, but also takes account of temporal causality in spatial-temporal synchronous graph. We further conducted extensive experiments on four real-world datasets, and the results verified the consistent superiority of our proposed approach compared with various existing baselines. This study also has some possible limitations, such as the lack of consideration of external factors and the neglect of the flexible graph structure. In the future work, we will attempt to investigate further improvements of our model and extend it to more general fields of spatial-temporal forecasting.

CRediT authorship contribution statement

Xian Yu: Conceived and designed the experiments; Performed the experiments; Wrote the paper. Yin-Xin Bao: Analyzed and interpreted the data; Wrote the paper. Quan Shi: Contributed reagents, materials, analysis tools or data.

Declaration of competing 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.

Data availability

Data will be made available on request.

Funding statement

This work is supported by the National Natural Science Foundation of China [61771265]; the Science and Technology Project of Nantong City [MS22021034]; the Postgraduate Research and Practice Innovation Program of Jiangsu Province [KYCX23_3396], [KYCX22_3341].

References

[1] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, B. Yin, Deep learning on traffic prediction: methods, analysis, and future directions, IEEE Trans. Intell. Transp. Syst. 23 (2022) 4927–4943.

[2] A. Ali, Y. Zhu, M. Zakarya, Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction, Neural Netw. 145 (2022) 233–247.

[3] R. Dai, S. Xu, Q. Gu, C. Ji, K. Liu, Hybrid spatio-temporal graph convolutional network: improving traffic prediction with navigation data, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, CA, USA, 2020, pp. 3074–3082.

[4] K. Guo, Y. Hu, Z. Qian, H. Liu, K. Zhang, Y. Sun, J. Gao, B. Yin, Optimized graph convolution recurrent neural network for traffic prediction, IEEE Trans. Intell. Transp. Syst. 22 (2021) 1138–1149.

[5] F. Huang, P. Yi, J. Wang, M. Li, J. Peng, X. Xiong, A dynamical spatial-temporal graph neural network for traffic demand prediction, Inf. Sci. 594 (2022) 286–304.

[6] Z. Li, G. Xiong, Y. Tian, Y. Lv, Y. Chen, P. Hui, X. Su, A multi-stream feature fusion approach for traffic prediction, IEEE Trans. Intell. Transp. Syst. 23 (2022) 1456–1466.

[7] W. Li, X. Wang, Y. Zhang, Q. Wu, Traffic flow prediction over muti-sensor data correlation with graph convolution network, Neurocomputing 427 (2021) 50–63.

[8] H. Peng, B. Du, M. Liu, M. Liu, S. Ji, S. Wang, X. Zhang, L. He, Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning, Inf. Sci. 578 (2021) 401–416.

[9] H. Qiu, Q. Zheng, M. Msahli, G. Memmi, M. Qiu, J. Lu, Topological graph convolutional network-based urban traffic flow and density prediction, IEEE Trans. Intell. Transp. Syst. 22 (2021) 4560–4569.

[10] H. Yang, X. Zhang, Z. Li, J. Cui, Region-level traffic prediction based on temporal multi-spatial dependence graph convolutional network from GPS data, Remote Sens. 14 (2022) 303.

[11] B. Yu, Y. Lee, K. Sohn, Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN), Transp. Res., Part C, Emerg. Technol. 114 (2020) 189–204.

[12] M. Abdoos, A.L.C. Bazzan, Hierarchical traffic signal optimization using reinforcement learning and traffic prediction with long-short term memory, Expert Syst. Appl. 171 (2021) 1–9.

[13] R.L. Abduljabbar, H. Dia, P.-W. Tsai, Unidirectional and bidirectional lstm models for short-term traffic prediction, J. Adv. Transp. 2021 (2021) 1–16.

[14] T. Afrin, N. Yodo, A long short-term memory-based correlated traffic data prediction framework, Knowl.-Based Syst. 237 (2022) 1–11.

[15] Y.-X. Bao, Y. Cao, Q.-Q. Shen, Q. Shi, Global-local spatial-temporal residual correlation network for urban traffic status prediction, Comput. Intell. Neurosci. 2022 (2022) 1–15.

[16] A. Essien, I. Petrounias, P. Sampaio, S. Sampaio, A deep-learning model for urban traffic flow prediction with traffic events mined from twitter, World Wide Web 24 (2021) 1345–1368.

[17] J. Li, F. Guo, A. Sivakumar, Y. Dong, R. Krishnan, Transferability improvement in short-term traffic prediction using stacked LSTM network, Transp. Res., Part C, Emerg. Technol. 124 (2021) 1–18.

[18] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, S. Ji, Temporal multi-graph convolutional network for traffic flow prediction, IEEE Trans. Intell. Transp. Syst. 22 (2021) 3337–3348.

[19] C. Ma, G. Dai, J. Zhou, Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method, IEEE Trans. Intell. Transp. Syst. 23 (2022) 5615–5624.

[20] Z. Pan, W. Zhang, Y. Liang, W. Zhang, Y. Yu, J. Zhang, Y. Zheng, Spatio-temporal meta learning for urban traffic prediction, IEEE Trans. Knowl. Data Eng. 34 (2022) 1462–1476.

[21] W. Shu, K. Cai, N.N. Xiong, A short-term traffic flow prediction model based on an improved gate recurrent unit neural network, IEEE Trans. Intell. Transp. Syst. 23 (2022) 16654–16665.

[22] K. Wang, C. Ma, Y. Qiao, X. Lu, W. Hao, S. Dong, A hybrid deep learning model with 1dcnn-lstm-attention networks for short-term traffic flow prediction, Phys. A, Stat. Mech. Appl. 583 (2021) 1–13.

[23] Z. Wang, X. Su, Z. Ding, Long-term traffic prediction based on LSTM encoder-decoder architecture, IEEE Trans. Intell. Transp. Syst. 22 (2021) 6561–6571.

[24] J. Zhang, F. Chen, Q. Shen, Cluster-based LSTM network for short-term passenger flow forecasting in urban rail transit, IEEE Access 7 (2019) 147653–147671.

[25] U. Ryu, J. Wang, U. Pak, S. Kwak, K. Ri, J. Jang, K. Sok, A clustering based traffic flow prediction method with dynamic spatiotemporal correlation analysis, Transportation 49 (2022) 951–988.

[26] C. Song, Y. Lin, S. Guo, H. Wan, 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, New York, USA, 2020, pp. 914–921.

[27] M. Li, Z. Zhu, Spatial-temporal fusion graph neural networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 4189–4196, Online.

[28] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: data-driven traffic forecasting, in: International Conference on Learning Representations, Vancouver, Canada, 2018, pp. 1–16.

[29] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial-temporal graph modeling, in: Proceedings of the International Joint Conference on Artificial Intelligence, Macau, China, 2019, pp. 1907–1913.

[30] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting, in: Proceedings of the International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 2018, pp. 3634–3640.

[31] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, Hawaii, USA, 2019, pp. 922–929.

[32] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, H. Li T-gcn, A temporal graph convolutional network for traffic prediction, IEEE Trans. Intell. Transp. Syst. 21 (2020) 3848–3858.

[33] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs, arXiv:1312.6203, 2014.

[34] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in Neural Information Processing Systems, vol. 29, Barcelona, Spain, 2016, pp. 1–9.

[35] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, Toulon, France, 2017, pp. 1–14.

[36] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, vol. 30, Long Beach, USA, 2017, pp. 1024–1034.

[37] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations, Vancouver, Canada, 2018, pp. 1–12.

[38] G. Jin, F. Li, J. Zhang, M. Wang, J. Huang, Automated dilated spatio-temporal synchronous graph modeling for traffic prediction, IEEE Trans. Intell. Transp. Syst. (2022) 1–11.

[39] C. Zheng, X. Fan, C. Wang, J. Qi Gman, A graph multi-attention network for traffic prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, New York, USA, 2020, pp. 1234–1241.

[40] X. Cheng, Y. He, P. Zhang, Y. Kang, Traffic flow prediction based on information aggregation and comprehensive temporal-spatial synchronous graph neural network, IEEE Access (2023) 1–11.
[41] Z. Wei, H. Zhao, Z. Li, X. Bu, Y. Chen, X. Zhang, Y. Lv, F.-Y. Wang, Stgsa: a novel spatial-temporal graph synchronous aggregation model for traffic prediction, IEEE/CAA J. Autom. Sin. 10 (2023) 226–238.
[42] J. Sun, J. Wang, J. Chen, G. Ding, F. Lin, Clustering analysis for Internet of spectrum devices: real-world data analytics and applications, IEEE Int. Things J. 7 (2020) 4485–4496.
[43] S. Peignier, P. Schmitt, F. Calevro, Data-driven gene regulatory networks inference based on classification algorithms, Int. J. Artif. Intell. Tools 30 (2021) 1–14.
[44] X. Du, J. Yu, Z. Chu, L. Jin, J. Chen, Graph autoencoder-based unsupervised outlier detection, Inf. Sci. 608 (2022) 532–550.
[45] R. Girshick, Fast R-CNN, in: IEEE International Conference on Computer Vision, Santiago, Chile, 2015, pp. 1440–1448.