Information Sciences 588 (2022) 315–330

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Adaptive Dual-View WaveNet for urban spatial–temporal event prediction

Guangyin Jin [a], Chenxi Liu [b], Zhexu Xi [c], Hengyu Sha [a], Yanyun Liu [d], Jincai Huang [a,*]

[a] College of System Engineering, National University of Defense Technology, Changsha, China
[b] College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
[c] Bristol Centre for Functional Nano-materials, University of Bristol, Bristol, UK
[d] College of Economics and Management, Harbin Institute of Technology, Harbin, China

ARTICLE INFO

Article history:
Received 21 February 2021
Received in revised form 17 December 2021
Accepted 25 December 2021
Available online 30 December 2021

Keywords:
Spatial–temporal prediction
Representation learning
WaveNet
Graph convolutional neural network

ABSTRACT

Spatial–temporal event prediction is a particular task for multivariate time series forecasting. Therefore, the complex entangled dynamics of space and time need to be considered. This task is an essential but crucial loop in future smart cities construction, which can be widely applied in urban traffic management, disaster monitoring and mobility analysis. In recent years, video-like spatial–temporal modelling has been the most common approach in many deep learning models. However, the video-like modelling approach cannot consider some latent region-wise correlations other than geographic spatial distance information. To overcome the limitation, we propose a novel neural network framework, Adaptive Dual-View WaveNet (ADVW-Net), for the urban spatial–temporal event prediction. By integrating the spatial representations from Convolutional Neural Network (CNN) and that from adaptive Graph convolutional neural network (GCN), our proposed model can capture not only the geographic correlations but also some latent region-wise dependencies from the input data. In addition, the effective architecture, WaveNet, can be transferred to region-wise spatial–temporal prediction scenarios for long-range temporal dependencies learning. Experimental results on three urban datasets demonstrate the superior performance of our proposed model.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

The urban population has proliferated with urbanization, bringing safety and sustainability challenges. The essential but significant loop of future smart cities is the predictable dynamics in the complex and substantial urban area [1]. For urban transportation, if the intelligent systems can predict the citywide traffic flow or taxi orders, some scheduling strategies can be arranged to reduce traffic congestion or improve ride-hailing demand response. For urban security, if the criminal or disaster incidents can be predicted in advance, the loss of life and property can be reduced, and the emergency security decision-making can also provided for the government. However, the most challenging issue is modelling the complex spatial–temporal dynamics in urban systems.

In the early stages, some classic time series modelling methods and traditional machine learning models were adopted in spatial–temporal prediction tasks. For instance, the self-exciting point, ARIMA and Random Forest were widely used in

* Corresponding author.
E-mail address: jinguangyin18@nudt.edu.cn (J. Huang).

https://doi.org/10.1016/j.ins.2021.12.085
0020-0255/© 2021 Elsevier Inc. All rights reserved.

spatial–temporal prediction [2,3]. Nevertheless, in these traditional methods, the spatial–temporal correlations are usually separated, so they are challenging to achieve satisfactory results. In recent years, deep learning models with convolutional neural network (CNN) and recurrent neural network (RNN) as the mainstream have achieved impressive performance in vision systems and sequence modeling [4,5]. Some previous works combine the CNN-based models with RNN-based models in spatial–temporal prediction [6–8]. In this manner, the spatial–temporal data needs to be processed into video-like inputs [9]. Specifically, the spatial areas should be divided into small grids equally on the longitude and latitude axes respectively. Time dimensions should be divided into different snapshots at a specific time granularity. Then the spatial–temporal events need to be mapped into the spatial grids and time snapshots of their occurrence. The value in the spatial grid of one time snapshot, similarly to the pixel value, represents the frequency of such spatial–temporal events. The video-like input format is compatible with CNN and the spatial–temporal joint features can be captured by the deep neural network stacked with CNNs and RNNs. Many previous works have demonstrated the effectiveness of these hybrid deep learning models [10–15]. However, the most significant limitation of the video-like modeling approach is that only the geographic proximity feature is considered. Specifically, according to the first law of geography, the nearby items are related to each other. However, the urban system is more complex, and there are more latent spatial correlations that only geographic proximity cannot be represented. For instance, the similar Points of Interest (POIs) in different urban regions could lead to the similar patterns of certain spatial–temporal events, even if these regions are far apart. If the POIs data is taken consideration into deep models, there are still two main disadvantages. The one is that the complete POIs data is usually difficult to access. Another one is that the POIs data is not transferable among different cities. To address these bottlenecks, we propose an adaptive graph method to capture the latent correlations among different regions. As far as we know, the graph structure is a fruitful approach to reveal the relations between different nodes, and the graph deep learning methods are widely applied in transportation and other fields [16–25]. In our model, the graph structure can be learned adaptively from the spatial–temporal sequences in selected spatial regions and temporal snapshots, and the graph representation can be updated adaptively by graph convolutional network (GCN). Some high-level latent spatial correlations can be learned automatically without any external data in this way.

Although GCN can effectively capture the spatial correlations from non-Euclidean structures, its global weight sharing mechanism could limit the capability to learn fine-grained local features in spatial proximity. While the main advantage of CNN is that its local weight sharing mechanism has a stronger capability to aggregate the spatial proximity features. Therefore, GCN and CNN should be a mutually complementary relationship in spatial representation learning. As shown in Fig. 1, the difference between the two paradigms is revealed. To improve the capability of spatial representation learning from the above directions, we propose Adaptive Dual-View WaveNet (ADVW-Net) in this paper. In spatial perspective, dual-view representations are integrated in this model: pixel-view representation and graph-view representation. The frontier is obtained by CNN model from the video-like inputs while the latter is obtained by GCN model from the adaptive graph structure. Then we compose the dual-view representation as a hybrid representation to enhance the complete spatial representation. In temporal perspective, we combine our proposed dual-view module with the architecture of WaveNet, which is a more efficient long-short term deep sequence modeling framework compared with RNN-based models. In summary, our contributions are summarized as follow:

- To the best of our knowledge, it is the first exploration to combine the pixel-view features with graph-view features to enhance the spatial representation in region-wise spatial–temporal prediction tasks.
- We have improved the internal architecture of traditional WaveNet, so that the model is transferred from 1D temporal signals scenario to 2D spatial–temporal signals scenario.
- We improve the generation approach of the adaptive graph structures. The adaptive GCN model can learn the graph structure from the input data, enhancing the association with the original input data.
- We evaluate our model with three different domain urban datasets. The experimental results have demonstrated the effectiveness and universality of our model in different urban spatial–temporal prediction tasks.

In the remainder of this paper, we begin with the related work on region-wise spatial–temporal prediction methods in Section 2. Then we briefly review the background of CNN, GCN, WaveNet and modeling approach for region-wise spatial–temporal prediction in Section 3. The dataset processing and proposed methodologies in this research are presented in Section 4, followed by the experimental results and analysis in Section 5. The final section concludes the achievements of this research and proposes some future directions.

2. Related Work

The earliest research on spatial–temporal prediction task is based on the classical time series modeling approach. One mainstream approach is Autoregressive Integrated Moving Average (ARIMA) and its variants. Based on classical ARIMA model, many variants were also proposed in many different domains, for instance, traffic flow prediction, wind forecasting and epidemic control [26–28]. Another one is self-exciting point method, which simulates the changing patterns of spatial–temporal events by stochastic process modeling. And the variants of this model has been applied in urban crime and earthquake prediction [29,2]. Although these methods have satisfactory mathematical interpretability and elegant formulas, they

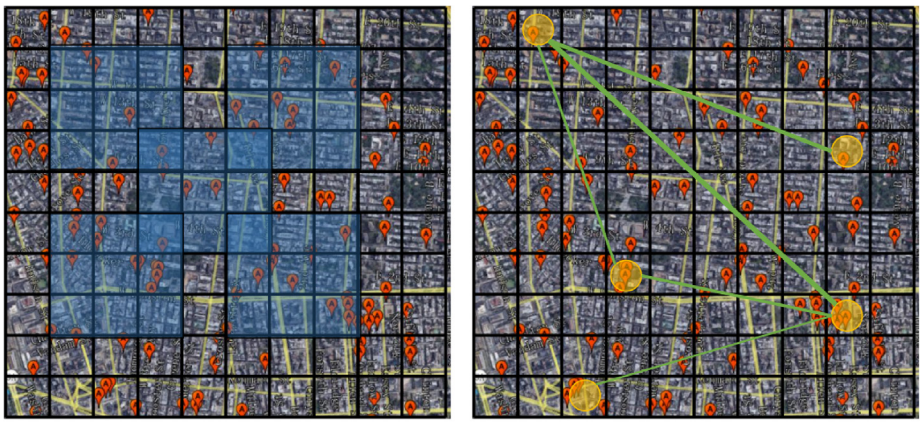

Fig. 1. The two different methods for spatial representation learning. Fig. 1(a) represents the learning paradigm in CNN, whose different filters can capture the different spatial proximity features efficiently. Fig. 2(b) represents the learning paradigm in GCN, which can capture the spatial correlations effectively from non-Euclidean structures even if the spatial distances of different grids are far away from each other..

cannot take full advantage of big data and automatically learn complex patterns from data. To enable the latent patterns of data to be further automatically discovered, some data-driven methods based on statistical learning have been introduced into this field. Especially some ensemble statistical learning models such as Random Forest and Xgboost has been widely used in urban traffic and air pollution prediction [30,31]. However, there are still two main limitations of these statistical learning methods: The one is that the superior performance usually relies on experienced feature engineering. Another one is that the spatial and temporal patterns are captured separately in these methods. This means that some spatial–temporal entanglement features are difficult to extract automatically.

In the recent five years, the deep neural networks (DNN) have become a fruitful approach to address the problems of complex feature engineering and entanglement feature learning, for their powerful feature-extraction automatically. In some previous work, some DNN models have been tentatively applied in spatial–temporal prediction. In [32,33],the DNN-based model was first applied in trajectory prediction and urban air quality prediction respectively. To further explore the methods of spatial–temporal joint representation learning, some hybrid deep learning models are presented. Among them, ConvLSTM [8] is the milestone which is the first work to capture spatial–temporal coupling features through CNN and LSTM simultaneously. Based on ConvLSTM, many improved variant models were proposed. In [34], Wang Yunbo et al. proposed PredRNN, a improved version of ConvLSTM, which address the problem of high-level information loss by spatial–temporal memory unit. In [35], Zhang Junbo et al. proposed an efficient CNN-based model to predict region-wise urban flow. In [36], a deep multi-view hybrid model was proposed to forecast ride-hailing demand. But this series of deep models are based on the video-like inputs. This modeling approach reflects the geographical proximity feature but ignores some other inner correlations among different grids. The graph structure is the appropriate approach to describe the region-wise correlations. Especially in traffic prediction, many graph-based deep learning models were developed. The most common framework is that the structural spatial correlations are captured by graph neural network and the temporal correlations are captured by some RNN-based, CNN-based or attention-based models. For instance, DCRNN [19] and T-GCN [37] are the combined models of GCN and RNN, ST-GCN [38] and Graph WaveNet [39] are the combined models of GCN and CNN, GMAN [40] is the combined model of GCN and attention network. Although these graph-based deep learning models have achieved satisfactory performance in learning structural spatial–temporal data, the main bottleneck is the non-predefined graph structure. In order to refine the high-level correlations between regions, some researchers began to construct regional correlation graphs by some empirical methods such as traffic flow interaction and POI similarity [17,41,42]. However, these graph modeling methods require certain domain knowledge and experience. In addition to traffic prediction, there is usually no definable graph structure in other region-wise urban spatial–temporal prediction tasks. In recent works, both Graph WaveNet [39] and STAG-GCN [43] combine the adaptive graphs with defined graphs for traffic flow prediction, which can adaptively learn some useful latent correlations without external data and domain knowledge. Motivated by these two work, we discovered the potential of adaptive graph learning in spatial–temporal prediction tasks. We assume that if the geographical proximity feature can be combined with adaptive graph feature, the region-wise urban spatial–temporal prediction tasks can be greatly improved. However, these two works both construct adaptive graph by randomly initialized embedding. This operation leads to fewer connections with the original data and easily causes oscillations during the training phase. Hence, we attempt to design a novel adaptive graph generator that can employ original data feature to tackle this limitation in this paper.

Different from all the previous literature, we first integrate the pixel-view representation and adaptive graph-view representation. More importantly, even without external inputs such as POIs, our proposed model can still learn some intrinsically correlated features beyond geographic proximity, and can be generalized to multiple spatial–temporal prediction scenarios.

3. Preliminaries

3.1. Convolutional Neural Network

Convolutional Neural Network (CNN) was proposed by Lecun first in 1995[4]. Up to now, the variations of CNN has been developed rapidly and achieved satisfactory performance in various image recognition-related tasks. The key to employing CNN is to slide the different filters on the image to aggregate the neighbor pixel-level value by learnable parameters via the multi-channel features. We define the convolution operation in the mathematical form as:

$$z(u, v) = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{i,j} \cdot k_{u-i,v-j} \tag{1}$$

where $x_{i,j}$ represents a pixel value in an image and $k_{u-i,v-j}$ represents a parameter in one filter. In CNN model, setting more filters can obtain more different latent features and the parameters in different filters can be optimized automatically during back propagation process. In addition, another two important hyper-parameters, the size of filters and the stride of filters should be preset in convolution operation. The size of the filter is used to determine the granularity of spatial feature extracting while the stride of filters is used to control the sampling frequency of feature learning. These hyper parameters need to be adjusted to equilibrium state according to different application scenarios.

Also, with the development of CNN, its application scenarios have gradually diversified. For instance, 1D CNN is also an efficient method for modeling time series, which achieves the goal by aggregating temporal instead of spatial neighbor features. In this paper, we apply the CNN models to capture spatial and temporal features simultaneously.

3.2. Graph Convolutional Neural Network

Graph Convolutional Neural Network (GCN) is a special case of CNN whose convolutional operation is on graph structured data. The size of traditional CNNs' filters are fixed and based on regular pixel-level grids. The irregular node neighbors in the graph structure make the convolution operation difficult. To overcome the application difficulties in graph, spectral-based and spatial-based approach have been developed [44,45]. The limitation of spectral-based GCN is that the Laplacian matrix needs to be known, in other words, the structure of the graph must be static. But spatial-based GCN does not have such limitation. In this paper, our model is based on the spatial approach proposed by Kipf et al. [46], which is defined as:

$$H^{l+1} = \sigma\left(\widehat{D}^{-\frac{1}{2}}\widehat{A}\widehat{D}^{-\frac{1}{2}}H^{l}W^{l}\right), \widehat{A} = A + I \tag{2}$$

Where H^{l+1} is the hidden state in the $(l+1)_{th}$ layer, H^{l} is the hidden state in the l_{th} layer, W^{l} is the learnable parameter of the l_{th} GCN layer. When $l = 0, H^{0}$ is the initial feature matrix. A and I denotes the original adjacency matrix and identity matrix respectively. $\widehat{A}$ denotes the self-accessible graph adjacency matrix. $\widehat{D}$ represents the degree matrix of $\widehat{A}$. The operation of $\widehat{D}^{-1/2}A\widehat{D}^{-1/2}$ is to normalize $\widehat{A}$ for training stability. σ denotes the activation function, which increase non-linearity to the output.

3.3. WaveNet

WaveNet is a fruitful method in deep sequence learning, which has been widely used in acoustic modeling. The traditional WaveNet model can predict the result of the t_{th} point based on the first t-1 points of a sequence, so it can be used to predict the value of sampling points in the speech. The basic formula is as follows:

$$p(x) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1}) \tag{3}$$

In terms of model structure, WaveNet is an efficient time series prediction model based on CNN, and also combines residual connections and skip connections to better capture long-term dependencies. We analyze its specific model structure in Section 4.

3.4. Region-wise Spatial–temporal Prediction

There are two mainstream spatial–temporal prediction scenarios: station-wise and region-wise respectively. In station-wise scenario, defined stations are provided. But in region-wise scenario, the regions usually need to be divided by geographic location. In most previous works, an entire area is usually divided into H*I small areas evenly based on latitude and longitude. To ensure the continuity in time dimension, the common approach is to generate continuous and discrete time snapshots by certain time slot unit. An event that occurs in the timestamp t, area (i, j) can be regularized into the corresponding time snapshot according to the division of geographic regions and time slots. Events that occur in the same grid

in the same time slot are aggregated and the number of events in a grid is similar to the pixel value. Consequently, the video-like sequences are established. In this paper, our aim is to predict the next snapshot by historical ones, which is defined as:

$$[X_t, X_{t+1}, \ldots, X_{t+T-1}] \longrightarrow X_{t+T} \tag{4}$$

4. Methodologies

The proposed deep learning framework ADVW-Net is displayed in Fig. 2. As reveal in Fig. 2(a), The framework consists of embedding layer, stacked Spatial–Temporal WaveNet (STWN) layers and output layer. The input of ADVW-Net is a historical continuous spatio-temporal event map and the output is the predicted event map in the next time step. As illustrated in Fig. 2(b), aSTWN layer is constructed by an adaptive dual-view module (ADVM) and a gated temporal convolution layer (Gated TCN). The Gated TCN layer consists of two parallel temporal convolution layers (TCN-a and TCN-b) while the ADVM is composed by the adaptive GCN model and CNN model. From the spatial perspective, our model can capture some latent structural spatial dynamics by involving adaptive GCN model. From the temporal perspective, our model can capture spatial dependencies at different temporal levels by stacking multiple STWN layers. For instance, at the bottom STWN layer, ADVM receives short-range temporal information while at the top STWN layer, ADVM tackles long-range temporal information. In addition, to accelerate the convergence speed in training phase and prevent the loss of shallow information, we adopt the residual connection and skip connection in each STWN layer.Fig. 3.

4.1. Embedding layer

The original input X_0 is a four-dimensional tensor with the size $[L, H, I, 1]$ where L is the length of the spatial–temporal sequences, H and I are discrete size of urban area. The dimension of the original feature is 1, The aim of embedding layer is to enhance the feature representation of the original inputs, which is defined as:

$$X_e = tanh(W_e \cdot X_0 + b_e) \tag{5}$$

Where $W_e \in R^{1 * C}$ and $b_e \in R^C$ are respectively the weight and bias of the embedding layer. We use tanh function in this case. We get the output of embedding layer as $X_e \in R^{L* H* I* C}$.

4.2. Temporal Convolution Layer

We adopt the dilated causal convolution as our temporal convolution layer (TCN) to capture the temporal dynamics of a region. Dilated causal convolution networks can capture long-range temporal information by stacking the same layers. Dif-

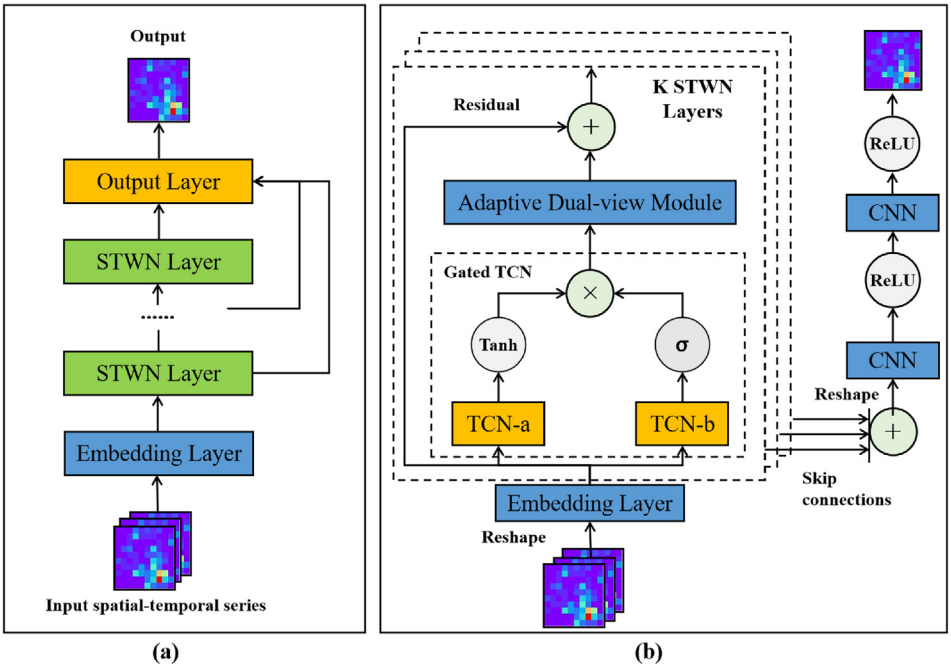

Fig. 2. the overview of ADVW-Net is presented in left sub-Fig. (a) and the details of STWN layer is presented in right sub-Fig. (b)..

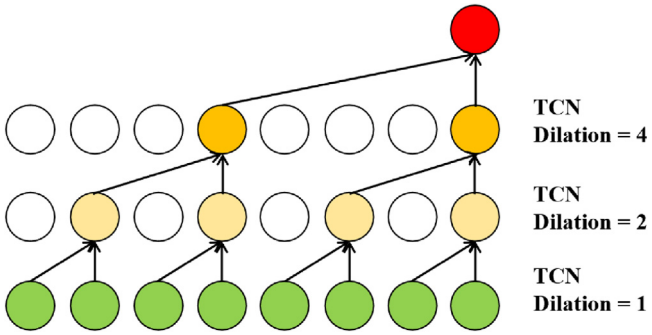

Fig. 3. The overview of dilated causal convolution network with kernel size 2. When the dilation factor increases, the time distance that can be captured increases accordingly.

ferent from RNN-based models, dilated casual convolution networks can process long-range sequences efficiently without recursive manner, which accelerates the calculation speed and alleviates the gradient explosion problem through parallel computation manner. The dilated causal convolution extracts long-range dynamics from the sequences through increasing the dilation factor layer by layer. Also, in order to maintain the consistency of sequential convolution operations, the zero padding mechanism is involved in this case. The computation process of dilated causal convolution operation is illustrated in Fig. 2. Mathematically, the dilated causal convolution operation of x with Γ at step t is represented as:

$$x \text{☆} \Gamma(t) = \sum_{s=0}^{k-1} \Gamma(s) x(t - d * s) \tag{6}$$

where d is the dilation factor which controls the skipping distance, x is the 1D sequence input with length l and dimension d, k is the kernel size of filters. By stacking dilated causal convolution layers with dilation factors in an increasing order, the receptive field of TCN also increases. This stacked structure enables TCN to capture long-term dependencies from complex sequences with less computation burden, which avoids the recursive learning process similarly to RNN-based models.

4.3. Gated TCN

Gating mechanism is a effective approach to control the information flow in sequence learning, which is widely used in some variants of RNN model. Also, this efficient mechanism is involved in 1D CNN-based models to enhance deep sequence learning capabilities. As shown in Fig. 2(b), Gated TCN is composed of two TCN modules, TCN-a and TCN-b respectively. In this case, TCN-a is used as temporal learner while TCN-b is treated as gating controller. The inputs of Gated TCN should be reshaped as the three-dimensional tensors with size $[N_f, L, C]$. $N_f = H * I$ means the total number of grid regions in a city. The region-wise temporal dynamics need to be learned individually and the formulation of Gated TCN for one region is defined as:

$$\begin{aligned} x_g &= \sigma(x \text{☆} \Gamma_1(\theta_1) + b_1) \odot tanh(x \text{☆} \Gamma_2(\theta_2) + b_2) \\ X_g &= \|_{i=0}^{N}(x_g) \end{aligned} \tag{7}$$

where θ_1, θ_2 b_1 and b_2 are the model parameters, $\odot$ is element-wise product operation. In this case, Sigmoid function is usually selected to control the ratio of information passed. The notation $\|$ represents the concentration operation. After concentrating the individual region-wise representation x_g, we can obtain the global region-wise representation $X_g \in R^{N_f * L * C}$

4.4. Adaptive Dual-View Module

As illustrated in Fig. 4, Adaptive Dual-View Module aims to learn regular video-like spatial–temporal features and irregular structural correlation features simultaneously. The regular video-like spatial–temporal features are captured by CNN model. In spatial–temporal prediction tasks, the grid-based video-like series are basic but important elements, whose spatial correlations can be captured efficiently by CNN models. In our framework, the outputs of Gated TCN are pushed into CNN model. Note that, the outputs of Gated TCN X_g should be reshaped as $[L, H, I, C]$. We apply the CNN layer to each of $X_g[i, :, :, :] \in R^{H * I * C}$, which is defined as:

$$h_{cnn} = \|_{i=0}^{L} ReLU(CNN(X_g[i, :, :, :])) \tag{8}$$

The notation $\|$ represents the concentration operation. After the convolution and concentration processes, the output h_{cnn} of CNN should be reshaped in the form $[L, N_f, C]$.

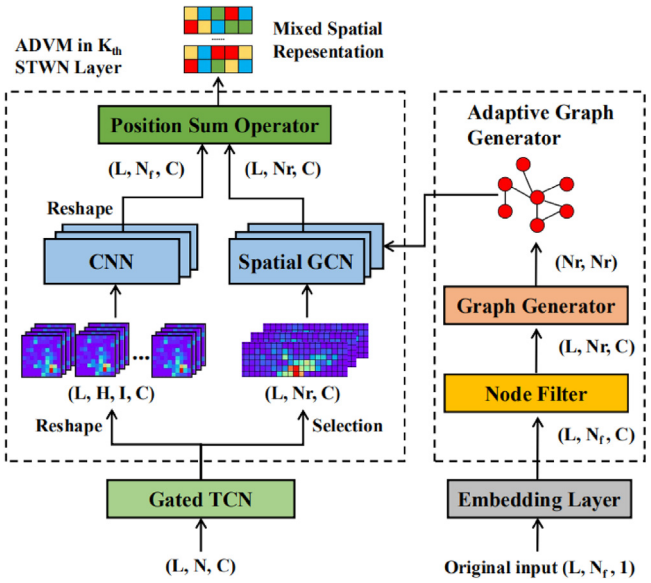

Fig. 4. The detailed architecture of Adaptive Dual-View Module in K_{th} STWN Layer and the framework of the adaptive graph generation..

Only geographic distance-based inner correlation can be reflected from the grid-based spatial representation. However, the distance-based manner cannot represent some high-level patterns in urban systems such as the regional function correlation and region-wise OD flow interaction. Graph is an appropriate structure to represent these relationship in real-word but the lack of prior knowledge and the entanglement of multiple correlations hinder the pre-definition of the whole graph. Hence, we propose an adaptive approach to generate graph structure from original data and improve the generated graph iteratively during training process.

In the work[39], the adaptive graph method is first presented in spatial–temporal prediction task but the graph embedding is initialized randomly. This initialization approach not only separates the association with the original data but also cause the instability in training process. In addition, not all urban regions have strong correlation with the patterns of other regions such as some event sparse region. With regard to the correlation between the event sparse regions and other non-sparse regions may involve more noise. First, we filter out sparse regions based on the missing rate δ of regional events on the time scale, and reserve the non-sparse regions as graph vertices. Given N_r reserved regions and the reshaped input tensor $X_a \in R^{N_r * L \cdot C}$, we define the adaptive graph as:

$$A_{adp} = ReLU\left(SoftMax\left(X_a W_a \cdot U_a X_a^T\right)\right) \tag{9}$$

where $W_a \in R^{L \cdot C * d}$ and $U_a \in R^{d * L \cdot C}$ are learnable parameters in adaptive graph generator. The adaptive mechanism allows the graph generator to perform parameter learning during the training process based on the input original data and learn the latent graph structure from the data automatically.

Meanwhile, the representation of adaptive graph is captured by the spatial GCN model. The corresponding graph feature matrix is actually the selected output of Gate TCN model. Given the output of Gated TCN as $X_g \in R^{N_f * L * C}$, we can obtain the selected tensor $X_s \in R^{N_r * L * C}$ without sparse regions. In this case, we apply the spatial GCN model to each of $X_s[:, i, :] \in R^{N_r * C}$. The calculation of spatial GCN is simple, which is defined as:

$$\begin{aligned} h_{gcn} &= \|_{i=0}^{L} ReLU\left(\widehat{D}^{-\frac{1}{2}} \widehat{A}_{adp} \widehat{D}^{-\frac{1}{2}} \cdot X_s[:, i, :] \cdot W_g\right) \\ \widehat{A}_{adp} &= A_{adp} + I \end{aligned} \tag{10}$$

where W_g is the learnable parameter in spatial GCN, h_{gcn} is the output of spatial GCN.

The last step is to combine the outputs of CNN and GCN. The specific method is to sum the representations of the corresponding regions, which is defined as:

$$h_M = h_C \oplus h_G \tag{11}$$

where h_C, h_G and are respectively the latent representation from CNN and GCN. h_M denotes the mixed representation. The notation $\oplus$ represents corresponding position sum operator. Note that, the alignment technique is based on the index. We can obtain the index by the non-sparse region and then add the graph latent representation into the corresponding

spatial position. Accordingly, we obtain the hybrid representation that contains both geographical distance and high-level latent correlation information.

4.5. Residual Connection

The residual connection is involved to prevent the loss of low-level information as the number of network layers deepens. In the work [47], the residual network was first proposed and achieved the state-of-art performance in image recognition. In our model, we expect to reserve the initial information before Gated TCN and Adaptive Dual-View Module in each Spatial–Temporal WaveNet (STWN) layer. The formula is defined as:

$$h_S = W_r \cdot h_I + h_M \tag{12}$$

where h_S is the output of each STWN layer, h_I is the initial input of each STWN layer, W_r is the learnable parameters of residual connection.

4.6. Output Layer

The stacked STWN layer is to capture multi-level spatial–temporal dynamics from the original data. Hence, we need to reserve the low-level and high-level information from STWN layer and combine them together as the input of the output layer. This function is achieved by skip connections, which is defined as:

$$h_O = \sum_{K=0}^{K} h_S^k \tag{13}$$

where h_S^k denotes the representation from the k_{th} STWN layer and h_O denotes the final representation from the stacked STWN layer. Actually, the skip connections are conducted to sum the output from different STWN layers. Then the input tensors of the output layer h_{Skip} should be reshaped as $[H, W, L \cdot C]$ to meet the requirement of CNNs. In the output layer, the input tensors are passed to two stacked CNNs, which is defined as:

$$\widehat{X}_{t+T} = ReLU(CNN(ReLU(CNN(h_O)))) \tag{14}$$

4.7. Loss Function and Pseudo Code

To stabilize the training process, we apply Huber loss in this case, which is defined as:

$$L\left(X_{t+T}, \widehat{X}_{t+T}\right) = \begin{cases} 0.5 * \left(X_{t+T} - \widehat{X}_{t+T}\right)^2 & \text{if } |X_{t+T} - \widehat{X}_{t+T}| < 1 \\ |X_{t+T} - \widehat{X}_{t+T}| - 0.5 & \text{otherwise} \end{cases} \tag{15}$$

5. Experiments and Analysis

The training details are displayed in this section, and some experimental results are presented to show the performance of our model.Table 1.

5.1. Dataset Processing

In this paper, three datasets in different domains are adopted to demonstrate the effectiveness of our model, Uber order dataset, urban crime dataset and urban fire dataset respectively. The urban crime and fire datasets in this paper are collected from a public safety data repository managed by San Francisco government[1], and the focused area is defined in San Francisco city. The Uber order dataset is from New York's Uber official statistics[2]. To prevent the noise caused by some outliers, we limit the spatial scope of the raw data. To ensure the data abundance in each time slot, we set one hour as a time slot in Uber dataset, one day as a time slot in urban crime and fire datasets. And considering the balance between fine-grained prediction and data sparsity, we divide the spatial range in these three datasets as 20*10, 20*20, 20*16 grid map respectively. The details about these three datasets are displayed in Table 2. In all subsequent experiments, we divide the training dataset, validation dataset and testing dataset according to the ratio of 7: 2: 1.

[1] https://datasf.org/opendata/
[2] https://www1.nyc.gov/nyc-resources/agencies.page

Table 1
Details about three different dataset.

Dataset	Spatial range (Lat*Lon)	Temporal range	Time slots	grid map
Uber	$[40.628, 40.830] * [-74.05, -73.88]$	2014/04/01 – 2014/08/31	3628	20*10
Crime	$[37.71, 37.80] * [-122.51, -122.38]$	2003/01/01 – 2018/06/30	5600	20*20
Fire	$[37.75, 37.80] * [-122.46, -122.38]$	2014/06/31 – 2019/06/31	1803	20*16

Table 2
Performance comparison of all methods on three datasets in terms of RMSE MAE and MAPE.The best and second best results are bolded and starred in the table respectively.

Algorithms	Merics	Uber	Crime	Fire
ARIMA	RMSE	1.2461	0.1934	0.3217
	MAE	0.3102	0.0673	0.1028
	MAPE	0.0759	0.0421	0.0377
	R2	0.9067	0.9124	0.9336
XGBoost	RMSE	0.8135	0.1572	0.2610
	MAE	0.2707	0.0514	0.0866
	MAPE	0.0568	0.0278	0.0286
	R2	0.9320	0.9487	0.9678
GRU	RMSE	0.7626	0.1295	0.2332
	MAE	0.2462	0.0443	0.0753
	MAPE	0.0494	0.0210	0.0228
	R2	0.9546	0.9677	0.9803
Conv-LSTM	RMSE	0.7458	0.1146	0.2201
	MAE	0.2336	0.0421	0.0702
	MAPE	0.0481	0.0196	0.0210
	R2	0.9678	0.9765	0.9873
PredRNN	RMSE	0.7378	0.1083	0.2174
	MAE	0.2289	0.0401	0.0694
	MAPE	0.0466	0.0190	0.0199
	R2	0.9715	0.9801	0.9892
ST-ResNet	RMSE	0.7425	0.1104	0.2189
	MAE	0.2316	0.0406	0.0697
	MAPE	0.0471	0.0193	0.0204
	R2	0.9702	0.9790	0.9887
DMVST-Net	RMSE	0.7306	0.1032	0.2108
	MAE	0.2268	0.0394	0.0678
	MAPE	0.0456	0.0189	0.0197
	R2	0.9756	0.9843	0.9921
ST-Transformer	RMSE	0.7182*	0.0984*	0.2011*
	MAE	0.2173*	0.0372*	0.0645*
	MAPE	0.0418*	0.0184*	0.0192*
	R2	0.9782	0.9879	0.9948
DCRNN	RMSE	0.7526	0.1187	0.2254
	MAE	0.2401	0.0433	0.0725
	MAPE	0.0487	0.0201	0.0214
	R2	0.9702	0.9810	0.9891
T-GCN	RMSE	0.7563	0.1201	0.2304
	MAE	0.2425	0.0442	0.0734
	MAPE	0.0489	0.0204	0.0216
	R2	0.9693	0.9781	0.9884
Graph WaveNet	RMSE	0.7268	0.0992	0.2076
	MAE	0.2204	0.0378	0.0662
	MAPE	0.0426	0.0190	0.0197
	R2	0.9764	0.9858	0.9935
ADVW-Net	RMSE	**0.6730**	**0.0922**	**0.1892**
	MAE	**0.2014**	**0.0337**	**0.0610**
	MAPE	**0.0381**	**0.0172**	**0.0181**
	R2	**0.9827**	**0.9932**	**0.9981**

5.2. Training Details and Comparison Algorithm

For our model, the hidden dimension of embedding layer, Gated TCN layer, GCN layer are all set as 32. The input time step of our model is set as 12 and the output time step is 1, the dimension of learnable parameters in adaptive graph generator is set as 36 and the missing rate threshold δ is set as 0.6 by default. The Adam optimizer is used in training process of ADVW-Net. The batch size is set as 16 in all subsequent experiments. We applied the technique of exponential decay learning rate

with the initial learning rate of 0.005. And the early stop strategy is used in training process to avoid over-fitting if the validation loss begins to keep increasing.

To verify the performance of our model, eleven algorithms are compared with ADVW-Net, including ARIMA [26], XGBoost [48], GRU [49], Conv-LSTM [8], PredRNN [34], ST-ResNet [36], DMVST-Net [36], ST-Transformer [50], DCRNN [19], T-GCN [37] and GrapWaveNet [39]. Some details of these state-of-art models are set as:

1. **ARIMA:** ARIMA is a classical time series regression model. We assume that the value in next time slot is only related to the value in the last time slot. Hence, we set the auto-regressive term as 1, the integrated term as 0, the moving average term as 0, respectively. In this case, ARIMA is actually a first-order autoregressive model.
2. **XGBoost:** XGBoost is a state-of-art ensemble statistical learning model, which is integrated by tree models. Multiple tree models can effectively extract diverse features automatically from data. The number of iterations is set as 100, the maximum tree depth is set as 10.
3. **GRU:** GRU is a popular variant of recurrent neural networks, which has been widely used in deep sequence learning tasks. The number of GRU layers is set as 2, the number of hidden unit is set as [64,64], the input length is set as 12 and the learning rate is set as 0.005.
4. **Conv-LSTM:** Conv-LSTM is the first deep learning model that couples spatial and temporal information by replacing traditional LSTM units with convolution operations, which has been successfully applied in precipitation prediction. The number of Conv-LSTM layers is set as 2, the number of filters is set as [32,32], the size of filters are set as 3*3, the input length is set as 12 and the learning rate is set as 0.005.
5. **PredRNN:** PredRNN can be seen as a improved version of Conv-LSTM. This model can solve the problem of high-level information loss in historical sequences by adding spatiotemporal memory units. The number of PredRNN layers is set as 2, the number of filters is set as [32,32], the size of filters are set as 3*3, the input length is set as 12 and the learning rate is set as 0.005.
6. **ST-ResNet:** ST-ResNet is a simple but efficient CNN-based spatial–temporal prediction model. Spatial–temporal features in different time period can be captured by ResNet, the superior variant of CNN. Considering the universality of spatial–temporal prediction tasks, we only apply the nearest temporal features in this case. The number of CNN layers is set as 5, the size of filters are all set as 3*3, the input length is set as 3 and the learning rate is set as 0.001.
7. **DMVST-Net:** DMVST-Net is a hybrid model that combines spatial, temporal and semantic information. For spatial view, the scope of local CNN is set as 5*5, size of filters are set as 3*3, and dimension of the output is set as 32. For the temporal view, the input length is set as 12. For semantic view, we use Pearson correlation coefficient to construct graph and the size of graph embedding is set as 16. The learning rate of the whole model is set as 0.001.
8. **ST-Transformer:** ST-Transformer is a model that integrates CNN and Transformer. The number of CNN layers is set as 2, the number of filters is set as [32,32], the size of filters are all set as 3*3, the dimension of feed-forward layer is set as 64, the number of attention head is set as 4, the input length is set as 12 and the learning rate is set as 0.001.
9. **DCRNN:** DCRNN is a model that integrates GCN and GRU. In this case, we convert the grid maps into adjacency graphs according to the region neighbor relationship, each region as a node and the connection weight of adjacent regions is 1, otherwise it is 0. The number of diffusion step is set as 2, the size of graph embedding is set as 32, the dimension of GRU hidden layer is set as 64, the input length is set as 12 and the learning rate is set as 0.001.
10. **T-GCN:** T-GCN is a model that integrates GCN and GRU. In this case, we do the same operation as in DCRNN. The number of GCN layers is set as 2, the size of graph embedding is set as 32, the dimension of GRU hidden layer is set as 64, the input length is set as 12 and the learning rate is set as 0.001.
11. **Graph WaveNet:** Graph WaveNet is a model that integrates GCN and WaveNet. In this case, we do the same operation as in DCRNN. The number of Graph WaveNet layers is set as 4, the size of graph embedding is set as 32, the dimension of TCN layers is set as 32, the input length is set as 12 and the learning rate is set as 0.001.

5.3. Result Analysis

We evaluate the effectiveness of proposed framework ADVW-Net in this section. Four metrics are used in evaluation the prediction performance of each algorithm, respectively as the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percent Error (MAPE) and R-Squared (R2) whose definitions are as follows:

$$RMSE = \left(\frac{1}{n}\Sigma_{i=1}^{n}\left(X_{t+T} - \widehat{X}_{t+T}\right)^2\right)^{\frac{1}{2}} \tag{16}$$

$$MAE = \frac{1}{n}\Sigma_{i=1}^{n}\left|X_{t+T} - \widehat{X}_{t+T}\right| \tag{17}$$

$$MAPE = \frac{1}{n}\Sigma_{i=1}^{n}\frac{\left|X_{t+T} - \widehat{X}_{t+T}\right|}{|X_{t+T}|} \tag{18}$$

$$R2 = 1 - \frac{\Sigma_{i=1}^{n}\left(X_{t+T} - \widehat{X}_{t+T}\right)^2}{\Sigma_{i=1}^{n}\left(\bar{X}_{t+T} - \widehat{X}_{t+T}\right)^2} \tag{19}$$

Since the prediction task in our paper is actually a regression task, the classical regression evaluation metric RMSE and MAE are applied, showing the absolute error between the predicted value and the real value. The purpose of MAPE is to measure the percentage error between the predicted value and the real value. For the three metrics, the lower RMSE, MAE and MAPE are, the better the model performs. Note that, MAPE needs to mask the region with a label of 0 when evaluating the prediction performance. The purpose of R2 is to show how well the model fits the data. Note that, the higher R2, the better the model fits.

5.3.1. Overall Performance

In Table 2, the comparison results on ARIMA, XGBoost, GRU, Conv-LSTM, PredRNN, ST-ResNet, DMVST-Net and ST-Transformer are displayed. We conducted 10 independent experiments and the best results of each metric on the testing datatset are in boldface.

From Table 2, we find that the classic time series prediction model ARIMA and statistical learning model XGBoost still have some gaps with deep learning methods in three metrics. The reason may be that they have limited capabilities to capture the complex and dynamic features from the spatial and temporal scales simultaneously. Furthermore, the performance of GRU is significantly better than that of ARIMA and XGBoost, but it is significantly worse than other hybrid deep learning models. This implies that spatial information plays an important role in spatio-temporal prediction tasks, therefore modeling only on the time scale is not enough.

Obviously, ADVW-Net achieves better performance than other methods in terms of all three metrics. Specifically, our model outperforms at least 6.29%, 7.31%, 8.85% in RMSE, MAE and MAPE on Uber Order dataset, the improvement ratios of RMSE, MAE and MAPE are at least 6.30%, 8.92%, 6.52% on Urban crime dataset and the improvement ratios of RMSE, MAE and MAPE are at least 5.91%,5.42%, 5.72% of improvements on Urban Fire dataset, which further confirms that our proposed model ADVW-Net is a practically effective solution for urban events prediction. Such significant improvements can be explained from two perspectives. First, in spatial perspective, some deep learning models like Conv-LSTM, PredRNN and ST-ResNet, only focus on the effect of geographic distance on the correlations between spatial regions. However, there are many latent correlations at the regional level that cannot be represented in Euclidean space. Thus, the model without considering the these latent correlations cannot achieve a higher accuracy. Similarly, for some GCN-based methods such as T-GCN and DCRNN, the performance is even lower than that of CNN-based methods such as Conv-LSTM and PredRNN, which shows that CNN's local weight sharing mechanism is better than GCN's global weight sharing mechanism on region-wise spatial prediction tasks. The performance of Graph WaveNet with adaptive dynamic graphs is significantly improved compared with other GCN-based methods. Second, in temporal perspective, we consider multi-level temporal information simultaneously by stacked STWN layers, so the Gated TCN model is more efficient and stable than RNN-based models and self-attention-based models in training process.

5.3.2. Case Study For Global Prediction

In order to reveal the effectiveness of our model in a more intuitive way, we randomly choose six consecutive time slots from dataset to show the prediction performance of ADVW-Net. To enhance the credibility of performance in visualization,

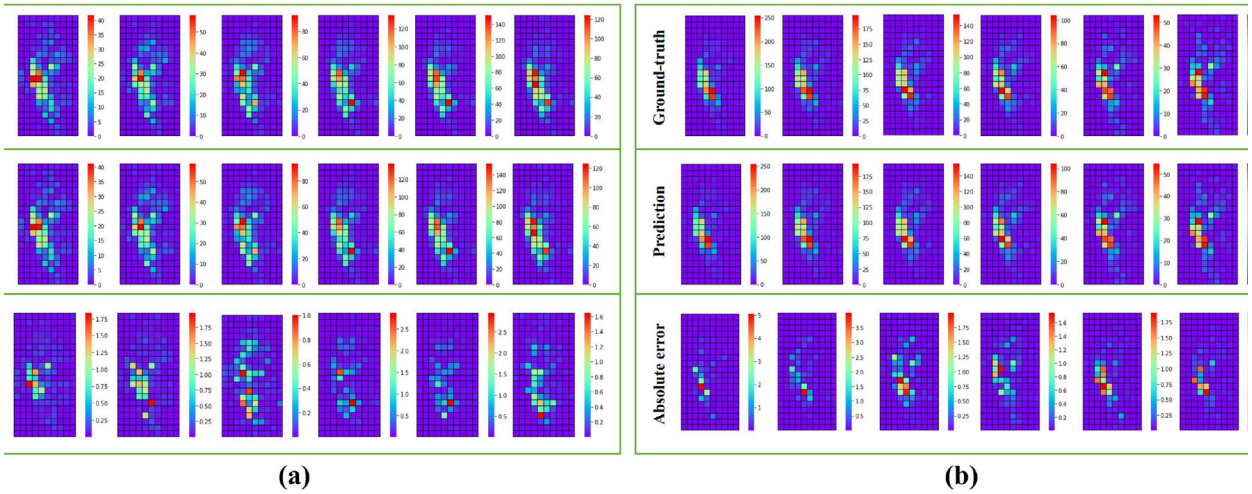

Fig. 5. The ground truth map, predicted map and absolute error map of selected six consecutive time slots from two groups on Uber dataset..

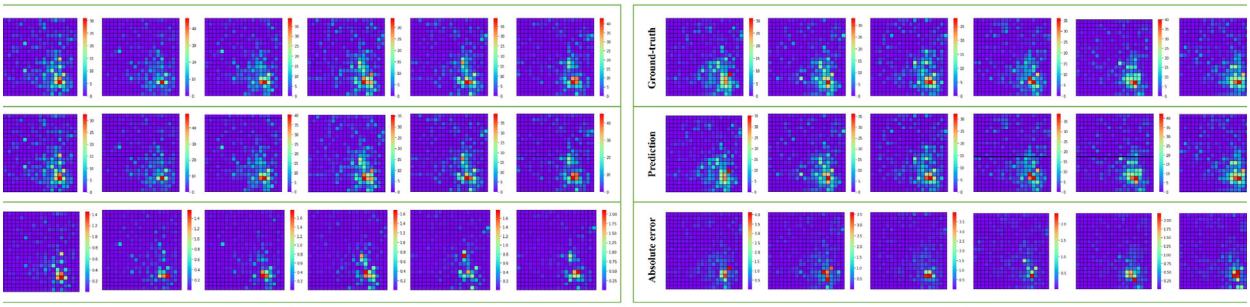

Fig. 6. The ground truth map, predicted map and absolute error map of selected six consecutive time slots from two groups on Crime dataset..

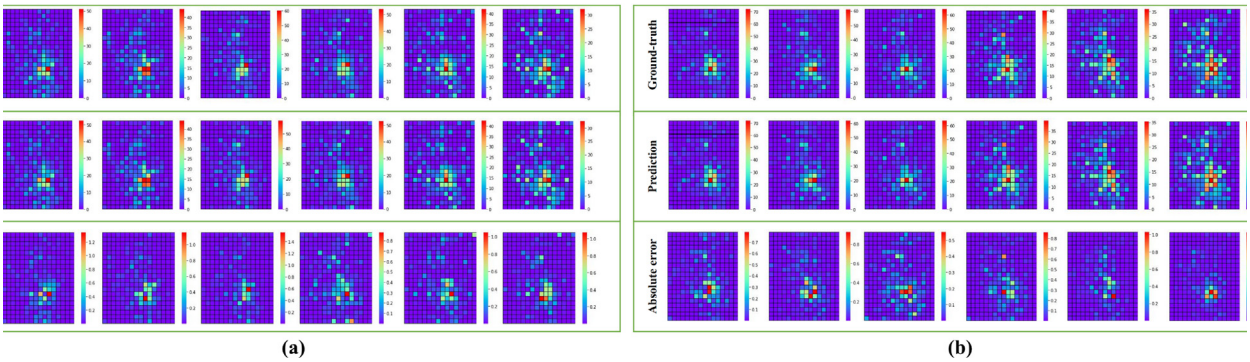

Fig. 7. The ground truth map, predicted map and absolute error map of selected six consecutive time slots from two groups on Fire dataset..

we provide two random groups of visualization in this case. The visualization in Fig. 5, Fig. 6 and Fig. 7 are for Uber dataset, Crime dataset and Fire dataset respectively. Each figure contains two sub-figures: sub-figure (a) represents the first group and sub-figure (b) represents the second group. For the first group, the time period of Uber is from 10 am, June 1st, 2014 to 15 am, June 1st, 2014, the time period of Crime is from September 1st, 2016 to September 7th, 2016 and the time period of Fire is from October 1st, 2018 to October 7th, 2018. For the second group, the time period of Uber is from 6 pm, August 10th, 2014 to 11 pm, August 10th, 2014; the time period of Crime is from February 5th, 2017 to February 10th, 2017; the time period of Fire is from July 1st, 2019 to July 7st, 2019. In each sub-figure, from top to bottom are successively ground truth, predicted results and absolute errors. Surprisingly, our proposed model ADVW-Net can achieve a high accuracy of spatial–temporal events prediction on different datasets.

5.3.3. Case Study For Adaptive Adjacency Matrix

For these three datasets, we select top 20 regions with dense data to show their sub adaptive matrix adaptive adjacency matrix and draw the heat map based on the normalized weights of the sub adjacency matrices as shown in Fig. 8. Especially, the density of the data is sorted in descending order from column 0 to column 19. We select the weights of column 0 for observation, and mark the top 5 regions with higher values on the grid maps. Note that, for the three datasets, the densest regions are marked by red and the top 5 regions with higher weights are marked by green.

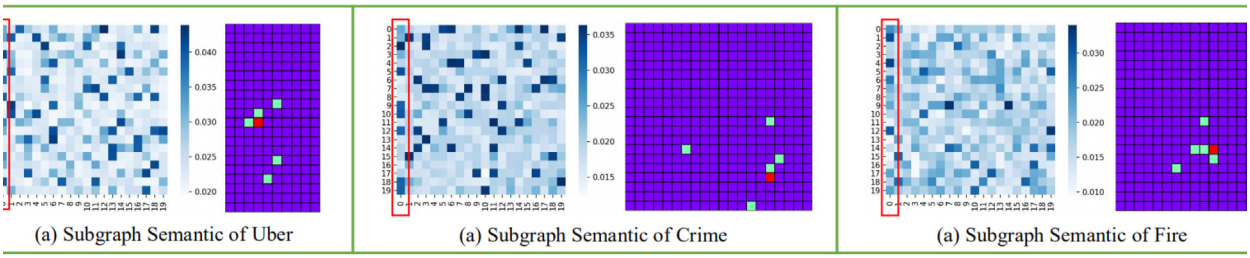

Fig. 8. The visualization of sub adaptive adjacency matrix for three datatsets and their geographical location marked on grid maps.

Table 3
Performance comparison of different variants on three datasets in terms of RMSE MAE and MAPE.

Variants	Merics	Uber	Crime	Fire
Only graph-level view	RMSE	0.7241	0.1021	0.2105
	MAE	0.2175	0.0389	0.0694
	MAPE	0.0432	0.0193	0.0201
Only pixel-level view	RMSE	0.7031	0.1013	0.2015
	MAE	0.2123	0.0389	0.0648
	MAPE	0.0419	0.0186	0.0194
Graph-level view with all regions	RMSE	0.6912	0.0982	0.1935
	MAE	0.2108	0.0380	0.0646
	MAPE	0.0398	0.0181	0.0187
Graph-level view with random initialization	RMSE	0.6958	0.0992	0.1954
	MAE	0.2126	0.0378	0.0665
	MAPE	0.0405	0.0182	0.0185
Traditional WaveNet	RMSE	0.7328	0.1058	0.2162
	MAE	0.2276	0.0394	0.0697
	MAPE	0.0454	0.0195	0.0206
ADVW-Net	RMSE	0.6730	0.0922	0.1892
	MAE	0.2014	0.0337	0.0610
	MAPE	0.0381	0.0172	0.0181

From Fig. 8, we can find that the most semantically similar regions are not just some neighboring regions, but also some regions with relative larger distances. The adaptive graph can establish the correlations between these regions effectively.

5.3.4. Effect of Adaptive Dual-View Module

To verify the effectiveness of Adaptive Dual-View Module in our proposed model, we conduct experiments with ADVW-Net using five different variants. The details of these four variants are described as below:

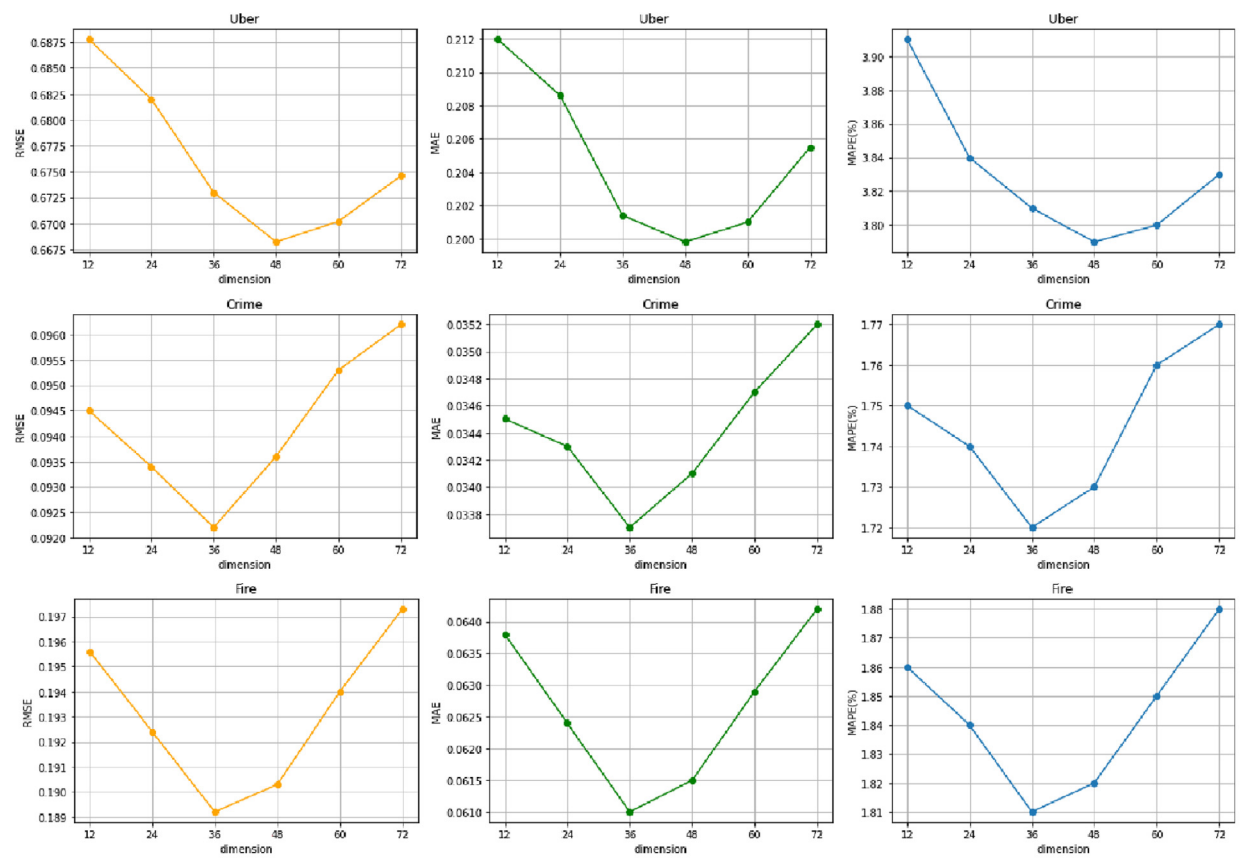

Fig. 9. The change trend of three metrics with *d*.

1. **Only graph-level view:** This variant only reserves the adaptive GCN model in Adaptive Dual-View Module. This means that the prior influence of geographic distance on spatial correlation is not considered, but only the variant is expected to be able to adaptively learn the spatial latent correlations.
2. **Only pixel-level view:** This variant only reserves the CNN model in Adaptive Dual-View Module. This means that the model only considers the impact of geographic distance on spatial correlation, and does not consider other latent correlations.
3. **Graph-level view with all regions:** This variant adopts all regional nodes to construct the adaptive graph without considering the missing rate of temporal information.
4. **Graph-level view with random initialization:** This variant adopts random initialization approach for adaptive graph embedding, which does not depend on the input data.
5. **Traditional WaveNet:** This variant abandons the Adaptive dual-view module, so that it simply captures the temporal dynamics of each region without considering the spatial correlation between the different regions.

Table 3 shows the average score of MAE, RMSE, and MAPE of different variants. We find that our complete model works better than the Graph-level view with random initialization model and Graph-level view with all regions model. This means that learning the graph structure from the existing data instead of random embedding can better represent the inner correlations of the data. Meanwhile, filter out some regions with sparse data to construct graph structure can avoid the interference of sparse signals. Also, we can find that the models with dual views can work better than Only graph-level view model and Only pixel-level view model. It indicates that pixel-level view and graph-level view can supplement each other with some useful information to obtain a superior representation. Among all the variants, Traditional WaveNet achieved the worst effect. This implies the importance of simultaneously considering spatial–temporal dynamics for spatio-temporal prediction tasks.

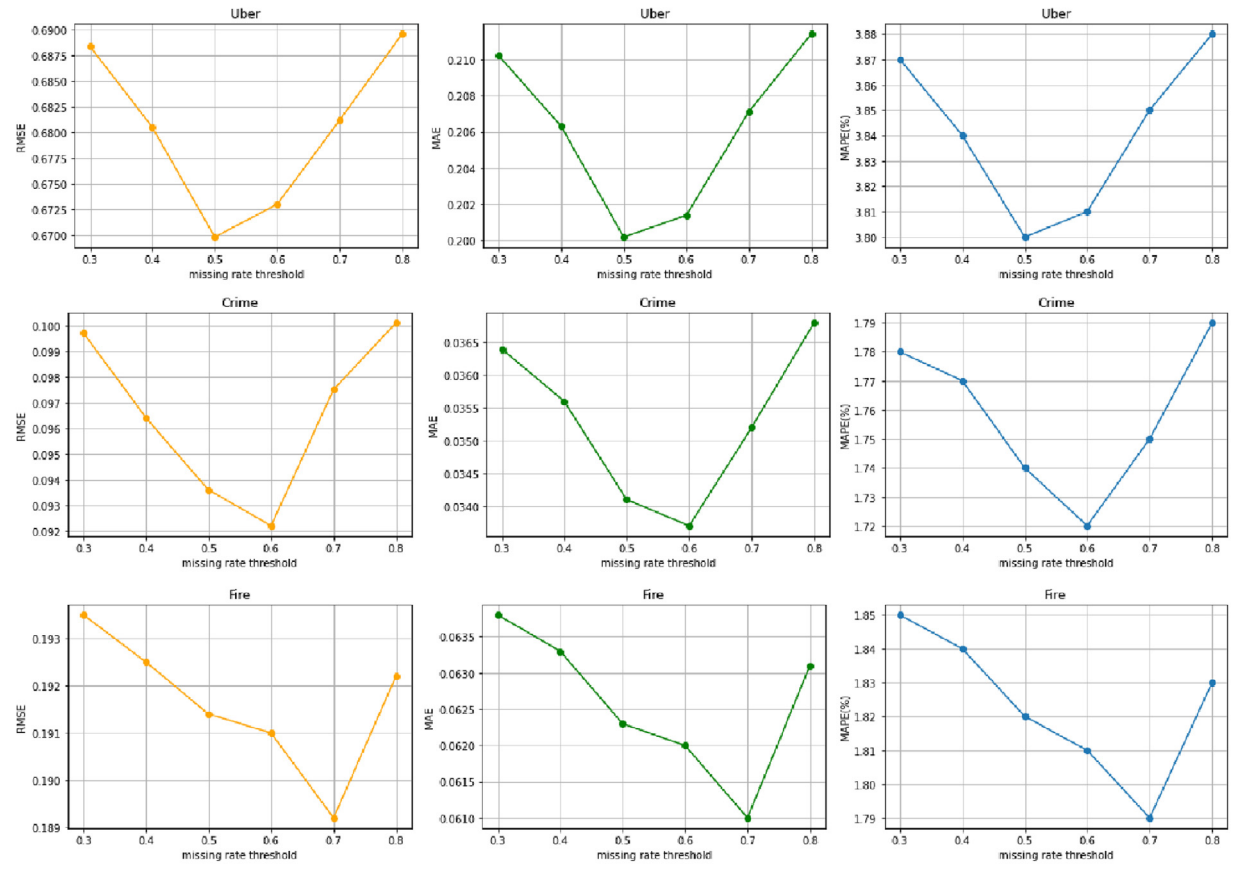

Fig. 10. The change trend of three metrics with δ.

5.3.5. Parameter Sensitivity Analysis

To further show the effectiveness of our proposed model, we conduct the experiments under different parameter setting, including the dimension of the learnable parameters in adaptive graph generator (denoted by d) and threshold of regional temporal information missing rate (denoted by δ).

First, we fix δ as the default value and change d in the range of [12,24,36,48]. Note that, d is an integer multiple of the input sequence length. The results are displayed in Fig. 9. When d is equal to 48, the model obtains the best value on the Uber dataset. When d is equal to 36, the model obtains the best value on the Crime dataset and Fire dataset.

Second, we fix d as the default value and change δ in the range of [0.3, 0.4, 0.5, 0.6, 0.7, 0.8]. The results are displayed in Fig. 10. When δ is equal to 0.5, the model obtains the best value on the Uber dataset. When δ is equal to 0.6, the model obtains the best value on the Crime dataset. When δ is equal to 0.7, the model obtains the best value on the Fire dataset.

6. Conclusion

In this paper, we gain a deeper insight into the urban spatial–temporal event prediction and propose a novel deep learning framework named ADVW-Net. The experimental results of ADVW-Net are remarkable compared with other traditional state-of-the-art models, which demonstrates that integration of pixel-level view and graph-level view enhances spatial representation to improve the performance of prediction. The visualization on different datasets also reveals that our model has achieved extraordinarily high accuracy for predicting the event data, whether from numerical perspective or spatial distribution perspective. However, this work has some limitations: (a) The video-like modeling approach for spatial–temporal events does not consider the real regional function or correlation, which may lead to the lack of practical significance for the urban event prediction from spatial perspective. (b) Although the spatial–temporal wavenet layer in our model has achieved superior performance in capturing spatial–temporal dynamics, the representation learning of space and time dimensions is separate, which is hard to capture some coupling dynamics in spatial–temporal scale. In the future, the promising direction is to make further improvements on ADVW-Net by proposing more effectiven adaptive graph learning methods or designing novel deep learning models that can capture spatial and temporal dynamics synchronously.

CRediT authorship contribution statement

Guangyin Jin: Conceptualization, Methodology, Software, Visualization, Writing - original draft. **Chenxi Liu:** Conceptualization, Methodology, Software. **Zhexu Xi:** Writing - original draft, Writing - review & editing. **Hengyu Sha:** Conceptualization, Software. **Yanyun Liu:** Writing - review & editing. **Jincai Huang:** Supervision.

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.

References

[1] Yu, Zheng, Licia, Capra, Ouri, Wolfson, Hai, Yang, Urban Computing: Concepts, Methodologies, and Applications, Acm Transactions on Intelligent Systems & Technology Special..
[2] G.O. Mohler, M.B. Short, P.J. Brantingham, F.P. Schoenberg, G.E. Tita, Self-exciting point process modeling of crime, Journal of the American Statistical Association 106 (493) (2011) 100–108.
[3] N. Zarei, M.A. Ghayour, S. Hashemi, Road traffic prediction using context-aware random forest based on volatility nature of traffic flows, in: Asian Conference on Intelligent Information and Database Systems, Springer, 196–205, 2013..
[4] Y. Lecun, Y. Bengio, Convolutional networks for images, speech, and time series, Handbook of Brain Theory & Neural Networks..
[5] T. Cheng, J. Wang, Application of a dynamic recurrent neural network in spatio-temporal forecasting (2007) 173–186..
[6] H. Yao, X. Tang, H. Wei, G. Zheng, Y. Yu, Z. Li, Modeling spatial-temporal dynamics for traffic prediction, arXiv preprint arXiv:1803.01254..
[7] G. Jin, Q. Wang, X. Zhao, Y. Feng, Q. Cheng, J. Huang, Crime-GAN, A Context-based Sequence Generative Network for Crime Forecasting with Adversarial Loss, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 1460–1469.
[8] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-C. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting, in: Advances in neural information processing systems, 802–810, 2015..
[9] S. Wang, J. Cao, P.S. Yu, Deep learning for spatio-temporal data mining: A survey, arXiv preprint arXiv:1906.04928..
[10] G. Jin, Q. Wang, C. Zhu, Y. Feng, J. Huang, X. Hu, Urban Fire Situation Forecasting: Deep sequence learning with spatio-temporal dynamics, Applied Soft Computing 97 (2020) 106730.
[11] H. Peng, H. Wang, B. Du, M.Z.A. Bhuiyan, H. Ma, J. Liu, L. Wang, Z. Yang, L. Du, S. Wang, et al, Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting, Information Sciences 521 (2020) 277–290.
[12] A. Ali, Y. Zhu, M. Zakarya, Exploiting dynamic spatio-temporal correlations for citywide traffic flow prediction using attention based neural networks, Information Sciences 577 (2021) 852–870.
[13] G. Jin, H. Sha, Y. Feng, Q. Cheng, J. Huang, GSEN: An ensemble deep learning benchmark model for urban hotspots spatiotemporal prediction, Neurocomputing 455 (2021) 353–367.
[14] G. Jin, C. Zhu, X. Chen, H. Sha, X. Hu, J. Huang, UFSP-Net: a neural network with spatio-temporal information fusion for urban fire situation prediction, in: IOP Conference Series: Materials Science and Engineering, vol. 853, IOP Publishing, 012050, 2020b..
[15] F. Li, J. Feng, H. Yan, G. Jin, D. Jin, Y. Li, Dynamic Graph Convolutional Recurrent Network for Traffic Prediction: Benchmark and Solution, arXiv preprint arXiv:2104.14917..
[16] G. Jin, Y. Cui, L. Zeng, H. Tang, Y. Feng, J. Huang, Urban ride-hailing demand prediction with multiple spatio-temporal information fusion network, Transportation Research Part C: Emerging Technologies 117 (2020) 102665.

[17] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, Y. Liu, Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 3656–3663.
[18] G. Jin, Q. Wang, C. Zhu, Y. Feng, J. Huang, J. Zhou, Addressing Crime Situation Forecasting Task with Temporal Graph Convolutional Neural Network Approach, in: 2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), IEEE, 474–478, 2020d..
[19] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting, in: International Conference on Learning Representations, 2018..
[20] Y. Zhang, T. Cheng, Graph deep learning model for network-based predictive hotspot mapping of sparse spatio-temporal events, Computers, Environment and Urban Systems 79 (2020) 101403.
[21] G. Jin, M. Wang, J. Zhang, H. Sha, J. Huang, STGNN-TTE: Travel time estimation via spatial–temporal graph neural network, Future Generation Computer Systems 126 (2022) 70–81.
[22] G. Jin, H. Yan, F. Li, J. Huang, Y. Li, Spatial-Temporal Dual Graph Neural Networks for Travel Time Estimation, arXiv preprint arXiv:2105.13591..
[23] Q. Wang, G. Jin, X. Zhao, Y. Feng, J. Huang, CSAN: A neural network benchmark model for crime forecasting in spatio-temporal scale, Knowledge-Based Systems 189 (2020) 105120.
[24] G. Jin, H. Yan, F. Li, Y. Li, J. Huang, Hierarchical Neural Architecture Search for Travel Time Estimation, in: Proceedings of the 29th International Conference on Advances in Geographic Information Systems, 91–94, 2021c..
[25] Y. Zhou, J. Li, H. Chen, Y. Wu, J. Wu, L. Chen, A spatiotemporal hierarchical attention mechanism-based model for multi-step station-level crowd flow prediction, Information Sciences 544 (2021) 308–324.
[26] B.M. Williams, L.A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results, Journal of transportation engineering 129 (6) (2003) 664–672.
[27] V. Radziukynas, A. Klementavicius, Short-term wind speed forecasting with ARIMA model, in: 2014 55th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON), IEEE, 145–149, 2014..
[28] Y. Pan, M. Zhang, Z. Chen, M. Zhou, Z. Zhang, An ARIMA based model for forecasting the patient number of epidemic disease, in: 2016 13th International Conference on Service Systems and Service Management (ICSSSM), IEEE, 1–4, 2016..
[29] Y. Ogata, R.S. Matsu'ura, K. Katsura, Fast likelihood computation of epidemic type aftershock-sequence model, Geophysical research letters 20 (19) (1993) 2143–2146.
[30] A. Mcgovern, T. Supinie, I. Gagne, T.N. DJ, M. Collier, R.A. Brown, J. Basara, J. Williams, Understanding severe weather processes through spatiotemporal relational random forests, in: 2010 NASA conference on intelligent data understanding (to appear), Citeseer, 2010..
[31] R. Yu, Y. Yang, L. Yang, G. Han, O.A. Move, RAQ–A random forest approach for predicting air quality in urban sensing systems, Sensors 16 (1) (2016) 86.
[32] X. Li, L. Peng, Y. Hu, J. Shao, T. Chi, Deep learning architecture for air quality predictions, Environmental Science and Pollution Research 23 (22) (2016) 22408–22417.
[33] J. Cui, X. Zhou, Y. Zhu, Y. Shen, A road-aware neural network for multi-step vehicle trajectory prediction, in: International Conference on Database Systems for Advanced Applications, Springer, 701–716, 2018..
[34] Y. Wang, M. Long, J. Wang, Z. Gao, P.S. Yu, Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 879–888.
[35] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017..
[36] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, Z. Li, Deep multi-view spatial-temporal network for taxi demand prediction, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018b..
[37] 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 Transactions on Intelligent Transportation Systems 21 (9) (2019) 3848–3858.
[38] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 3634–3640.
[39] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph WaveNet for Deep Spatial-Temporal Graph Modeling, in: The 28th International Joint Conference on Artificial Intelligence (IJCAI), International Joint Conferences on Artificial Intelligence Organization, 2019.
[40] C. Zheng, X. Fan, C. Wang, J. Qi, Gman: A graph multi-attention network for traffic prediction, arXiv preprint arXiv:1911.08415..
[41] G. Jin, Z. Xi, H. Sha, Y. Feng, J. Huang, Deep multi-view spatiotemporal virtual graph neural network for significant citywide ride-hailing demand prediction, arXiv preprint arXiv:2007.15189..
[42] G. Jin, H. Sha, Y. Feng, Q. Cheng, J. Huang, Modeling Spatiotemporal Geographic-Semantic Dynamics for Urban Hotspots Prediction..
[43] B. Lu, X. Gan, H. Jin, L. Fu, H. Zhang, Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 1025–1034.
[44] M. Henaff, J. Bruna, Y. LeCun, Deep convolutional networks on graph-structured data, arXiv preprint arXiv:1506.05163..
[45] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, arXiv preprint arXiv:1710.10903..
[46] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907..
[47] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, 2017..
[48] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.
[49] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555..
[50] H. Lin, W. Jia, Y. Sun, Y. You, Spatial-Temporal Self-Attention Network for Flow Prediction, arXiv preprint arXiv:1912.07663..