Neural Networks 162 (2023) 340–349

Contents lists available at ScienceDirect

Neural Networks

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

Traffic forecasting with graph spatial–temporal position recurrent network

Yibi Chen [a,b], Kenli Li [a,*], Chai Kiat Yeo [b], Keqin Li [a,c]

[a] College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
[b] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[c] Department of Computer Science, State University of New York, New Paltz, NY, USA

ARTICLE INFO

Article history:
Received 5 July 2022
Received in revised form 28 February 2023
Accepted 7 March 2023
Available online 15 March 2023

Keywords:
Adaptive graph learning
Approximate personalized propagation
Spatial–temporal
Traffic forecasting
Position graph convolution

ABSTRACT

With the development of social economy and smart technology, the explosive growth of vehicles has caused traffic forecasting to become a daunting challenge, especially for smart cities. Recent methods exploit graph spatial–temporal characteristics, including constructing the shared patterns of traffic data, and modeling the topological space of traffic data. However, existing methods fail to consider the spatial position information and only utilize little spatial neighborhood information. To tackle above limitation, we design a Graph Spatial–Temporal Position Recurrent Network (GSTPRN) architecture for traffic forecasting. We first construct a position graph convolution module based on self-attention and calculate the dependence strengths among the nodes to capture the spatial dependence relationship. Next, we develop approximate personalized propagation that extends the propagation range of spatial dimension information to obtain more spatial neighborhood information. Finally, we systematically integrate the position graph convolution, approximate personalized propagation and adaptive graph learning into a recurrent network (i.e. Gated Recurrent Units). Experimental evaluation on two benchmark traffic datasets demonstrates that GSTPRN is superior to the state-of-art methods.

1. Introduction

Currently, Intelligent Transportation Systems (ITS) (Dogra et al., 2022; Ying et al., 2022) are being piloted in both developed and developing countries as a part of the transformation to smart cities. Building ITS requires transport infrastructures (Papageorgiou & Kotsialos, 2002), technology studies, data analysis, etc. ITS can help to manage and optimize city traffic and alleviate traffic accidents given the massive growth of vehicles in the urban areas. According to a report by USA National Safety Council (NSC), deaths from motor vehicles rose 8% in 2020 compared to 2019, with as many as 42,060 fatalities from vehicle crashes. Traffic forecasting can thus ensure safe travel and contributes positively to the sustainable development of ITS and the building of smart cities.

The rapid advancement of deep learning technology has achieved impressive results in processing the spatial–temporal information (Ali, Zhu, & Zakarya, 2022) of Euclidean data. Taking a convolution neural network (CNN) as an example, it mainly learns the spatial pixels of spatial–temporal data (e.g. images and videos). However, high-quality images and videos in the real world are not that readily available and available data can be limited in information content such as remote sensing data. Compared to Euclidean data, graph-structured information (An et al., 2021; Wu et al., 2020) which typifies non-Euclidean data contains richer features and complex relationships.

Graph-based traffic data is irregular as it can establish multiple spatial–temporal characteristics and relationship types for the different regions. Hence, graph-based traffic forecasting can more accurately predict the future situation. Fig. 1 shows the spatial–temporal dependence information of a central region (e.g. region 1) with its surrounding regions as well as its propagation along the spatial and temporal dimensions. Region 1 is affected by the regions of varying distances from it at the same time step and their influence on Region 1 decreases as the distance increases; this is called spatial dependence. A future scenario of a region is predicted according to the situation changes at various historical moments in the same region; this is called temporal dependence. Spatial–temporal dependence combines spatial and temporal dimension of every region causing the change in itself and other regions at any time. The position information of every region and the propagation distance among every region and its surrounding regions both affect the amount of information obtained by these regions, especially for dynamic and real-time changes in the traffic of smart cities. Traffic situation forecasting is thus a great challenge for mining information from the spatial–temporal data.

* Corresponding author.
E-mail addresses: chenyibi@hnu.edu.cn (Y. Chen), lkl@hnu.edu.cn (K. Li), asckyeo@ntu.edu.sg (C.K. Yeo), lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.neunet.2023.03.009
0893-6080/

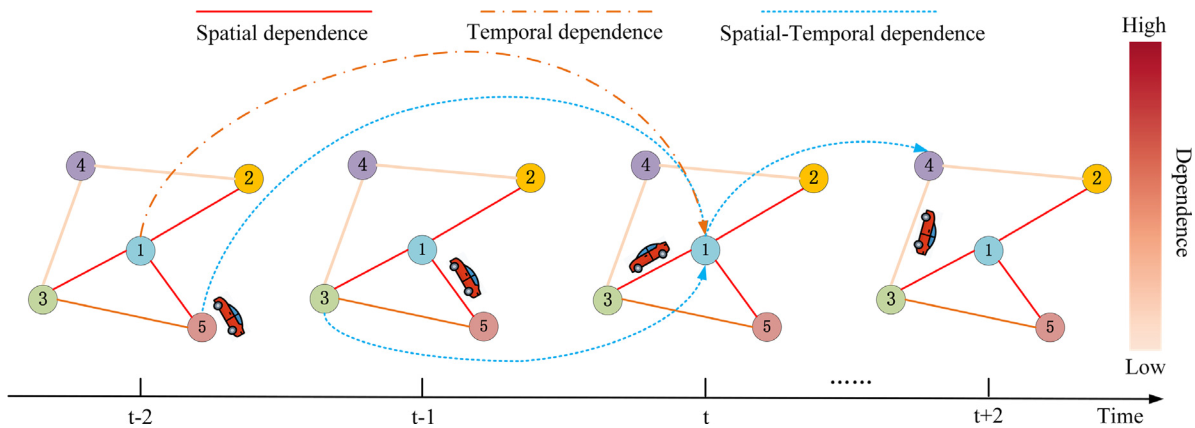


Fig. 1. Spatial–Temporal dependence.

Classical approaches predict the traffic situation based on time series such as Auto-Regressive Integrated Moving Average (ARIMA) (Lee & Fambro, 1999) and Multivariate Structural Time-series (MST) (Ghosh, Basu, & O'Mahony, 2009). Although these methods have achieved good results, they have not extracted the non-linear features. The emergence of CNN effectively addresses the non-linear features problem. ST-ResNet (Zhang, Zheng, & Qi, 2017) utilizes convolutional residual neural network in different branches (i.e. period, trend and closeness). LMST3D-ResNet (Chen et al., 2021) designs a 3D convolutional residual neural network to learn the spatial–temporal features in multiple local regions and considers external related factors (e.g. weather).

Recently, research works in graph neural networks provide a new direction for traffic forecasting. ST-MetaNet (Pan et al., 2019) constructs two network modules whereby the graph attention network (GAT) learns spatial dependency information and the recurrent neural network captures the temporal correlation features. ASTGCN (Guo, Lin, Feng, Song, & Wan, 2019) proposes an attention mechanism and a graph convolutional network (GCN) which extracts the spatial–temporal information simultaneously, and then aggregates the weights to obtain the forecast results. STSGCN (Song, Lin, Guo, & Wan, 2020) proposes a synchronous GCN model to capture the heterogeneous local spatial–temporal characteristics in the different time points. AGCRN (BAI, Yao, Li, Wang, & Wang, 2020) establishes an adaptive GCN mechanism to learn the spatial–temporal correlations via adaptive parameter learning and inter-dependencies. Despite these methods achieving good performance, they fail to consider the spatial position information. Moreover, spatial neighborhood information is limited by the propagation range of the node resulting in it not being fully exploited.

To overcome the above limitation, we present an end-to-end framework called GSTPRN, that models traffic series to obtain high-quality embedding for traffic forecasting. More precisely, we construct a position graph convolution based on self-attention, that utilizes position embedding matrix and computes the dependence strengths among the multiple nodes to capture the spatial dependence relationship. Furthermore, we develop approximate personalized propagation which aggregates an unlimited number of neighborhood propagation layers to extend the range of node neighborhood propagation, thereby capturing more spatial neighborhood information. We also employ power iteration to reduce the computational complexity. We further integrate the position graph convolution module, approximate personalized propagation and adaptive graph learning into a recurrent network (i.e. Gated Recurrent Units), whereby the adaptive graph learning utilizes adaptive node parameters and graph generation to enhance the GCN. We apply GSTPRN and multiple baseline methods on two real-world datasets for the traffic forecasting task. Extensive experimental results show that our model outperforms the state-of-art (SOTA) baselines.

We summarize the main contributions as follows:

- We present a novel architecture called GSTPRN, that systematically integrate position graph convolution, approximate personalized propagation and adaptive graph learning into a recurrent network.
- In GSTPRN, we construct a position graph convolution module based on self-attention to capture spatial dependence relationship among nodes.
- Our approach adopts approximate personalized propagation to obtain more spatial neighborhood information, and only requires a few parameters.
- The experimental results on two real-world datasets demonstrate that our GSTPRN is superior to the SOTA methods.

2. Related work

2.1. Time-series forecasting

Traffic forecasting based on time-series still attracts the attention of researchers despite having been studied for a while. Linear feature learning methods represented by historical average (HA) and vector autoregressive (VAR) (Zivot & Wang, 2006) consider the internal dependencies in multiple time series. Dial (2006) proposed the user-equilibrium traffic assignment model to choose the optimal path. Nuzzolo and Russo (1996) designed a random utility and space–time network models to handle low frequency transit services. Cascetta and Cantarella (1991) established a doubly dynamic assignment framework to model different days and different subperiods in one day. Di Gangi and Croce (2005) integrated dynamic traffic assignment and Kalman filtering for short time flow prediction. To improve their poor performance, machine learning-based approaches (e.g. Wu, Ho, & Lee, 2004 and Van Lint & Van Hinsbergen, 2012) capture complex relationship through effectively handcrafted characteristics.

From the non-linear feature perspective, deep learning-based methods extract features from the spatial–temporal data. Tang et al. (2020) designed local dependencies and global temporal dynamics to improve useful global temporal distribution by memory network. ST-ResNet (Zhang et al., 2017) proposes a combination of 2D CNN and residual network to capture the spatial dependence of any two regions. STDN (Yao et al., 2018) adopts long short-term (LSTM) and local 2D CNN to extract the temporal and spatial information of local regions of a city, respectively. LMST3D-ResNet (Chen et al., 2021) designs a 3D convolutional residual neural network to learn spatial–temporal features in multiple local regions and considers external related factors (e.g. weather). The above-mentioned CNN-based approaches model Euclidean traffic data by 2D or 3D convolution, while failing to learn road topology information from non-Euclidean data.

2.2. GCN-based spatial–temporal forecasting

Recently, GCN-based methods to tackle non-Euclidean traffic data have become mainstream. DCRNN (Li, Yu, Shahabi, & Liu, 2017) establishes bidirectional random walks to learn spatial correlation information, and designs an encoder–decoder structure to capture the temporal information. MRA-BGCN (Chen et al., 2020) proposes a GCN to obtain the interaction of nodes and edges in a graph in both node-wise and edge-wise manner. ST-GCN (Yan, Xiong, & Lin, 2018) develops an automatic learning model to establish dynamic skeletons to address the limitations of hand-crafted features in spatial–temporal prediction. DeepSTN+ (Lin, Feng, Lu, Li, & Jin, 2019) aims to obtain long-range spatial correlation information via ConvPlus component and PoI prior knowledge. STDN (Yao, Tang, Wei, Zheng, & Li, 2019) proposes an attention mechanism and a gating module to capture temporal shifting and spatial correlation information between different locations. AGCRN (BAI et al., 2020) proposes node adaptive parameter learning and data adaptive graph generation for each traffic sequence and different traffic sequences without predefined graphs. DSTGNN (Huang et al., 2022) creates a spatial dependence graph and utilizes the inhomogeneous Poisson process to learn dynamical relationship and infer intensity, respectively. STGODE (Fang, Long, Song, & Xie, 2021) constructs an ODE and two TCN blocks with residual structure to learn traffic information. T-GCN (Zhao et al., 2019) integrates the modules of GCN and GRU to extract complex spatial dependencies and dynamic temporal dependencies, respectively. ASTGCN (Guo et al., 2019) employs an attention mechanism and graph convolutional network to form a spatial–temporal network to learn the dynamic spatial–temporal correlations information.

2.3. Attention mechanism

The attention mechanism assigns different weights to the various information in order to extract key information such that the model can achieve more accurate judgments. It is widely applied in many fields (e.g. image classification Cai & Wei, 2020 and pattern recognition Wang, Zhang, Kan, Shan, & Chen, 2020). DSANet (Huang, Wang, Wu, & Tang, 2019) designs two parallel convolutional components and self-attention mechanism to jointly learn the dynamic periodic time series. GMAN (Zheng, Fan, Wang, & Qi, 2020) proposes an encoder–decoder and adopts an attention mechanism to model the relationships of the history and future time steps based on different road locations. ASTGNN (Guo, Lin, Wan, Li, & Cong, 2021) presents a multi-head self-attention module to obtain global receptive information and dynamic temporal features to learn data heterogeneity and solve the limitation of long-term prediction. GALSTM (Wei & Sheng, 2020) integrates attention mechanism and LSTM to capture spatial–temporal information. BuildSenSys (Fan et al., 2020) describes two attention mechanisms which are combined with a recurrent neural network to capture correlation information of nearby traffic from building sensing data. ST-GRAT (Park et al., 2020) aims to model multiple road situations from spatial dependence attention, temporal dependence attention, and spatial sentinel vectors. Wang, Zhu, Sun, and Tian (2021) proposed an LSTM to learn dynamic traffic information and capture temporal relationship and an attention mechanism to extract temporal relationship.

In summary, our proposed GSTPRN is quite different from the above-mentioned. GSTPRN can capture spatial dependence relationship among nodes and more spatial neighborhood information. Moreover, GSTPRN systematically integrates different modules (i.e. position graph convolution, approximate personalized propagation and adaptive graph learning) into a recurrent network to fully learn spatial–temporal information.

3. Preliminaries

Traffic Spatial–Temporal Graph Definition. For a traffic spatial–temporal graph $G = (V, E, A)$, in which $|V| = N$ is the number of nodes (e.g. N traffic observation detectors), E represents the set of edges, $A \in \mathbb{R}^{N\times N}$ is an adjacency matrix. The historical observation data of each node at different time is represented as $X = \{X_{:,0}, X_{:,1}, \dots, X_{:,t}\}$. The traffic spatial–temporal graph signal matrix is expressed as $X_{:,t} = \{x_{1,t}, \dots, x_{i,t}, \dots, x_{N,t}\}^T \in \mathbb{R}^{N\times C}$, that is the situation of N nodes at time t, and C is the number of features.

Problem Definition. We aim to predict the future situation of the regions based on the historical observed traffic data. Given traffic spatial–temporal graph signal matrix based on historical observation data, predicting traffic situation is expressed as: $X_{:,t+1}, \dots, X_{:,t+\tau}$. For traffic spatial–temporal forecasting, we describe the problem as learning a function f_θ to predict the future τ steps situation from the historical record data in the past T steps:

$$\{X_{:,t+1}, \dots, X_{:,t+\tau}\} = f_\theta(X_{t-T+1}, X_{t-T+2}, \dots, X_{:,t}; G) \tag{1}$$

4. Methodology

In this section, we introduce our proposed GSTPRN framework to learn spatial–temporal features for traffic forecasting. The motivation of this architecture is to obtain spatial dependence relationship through position graph convolution and to capture more spatial neighborhood information through approximate personalized propagation, to achieve high-quality embedding. Therefore, we systematically integrate position graph convolution, approximate personalized propagation and adaptive graph learning into the recurrent network.

The overall GSTPRN architecture is shown in Fig. 2. Denoting the historical traffic observation data by $X = (X_{t-T+1}, X_{t-T+2}, \dots, X_{:,t}) \in \mathbb{R}^{N\times C\times T}$, we employ the linear projection of the initial embedding layer to transform X into a high-dimensional representation $X' \in \mathbb{R}^{N\times d\times T}$, and then perform loop input in temporal step size, where $d \gg C$. For example, the first input is the spatial–temporal features at Time 0. To obtain position embedding matrix $E^{sp} \in \mathbb{R}^{N\times d}$, we introduce an additional embedding vector for each node while preserving the graph structure characteristics. The process of extracting spatial–temporal features includes position graph convolution, approximate personalized propagation and adaptive graph learning. Position graph convolution based on self-attention captures the spatial dependence information by calculating the spatial dependence strengths among the nodes. Approximate personalized propagation extends the propagation range of the node spatial information to obtain more spatial neighborhood information. Adaptive graph learning utilizes adaptive node parameter and adaptive graph generation to enhance GCN and then obtains the node embedding matrix. The hidden state (i.e. $h'_0, h'_1, \dots, h'_{t-1}$) in recurrent network indicates that the spatial–temporal information of this step is passed to the spatial–temporal information of the next step. We fuse spatial–temporal features of each step, and then adopt an aggregation layer to consolidate these features to generate the prediction.

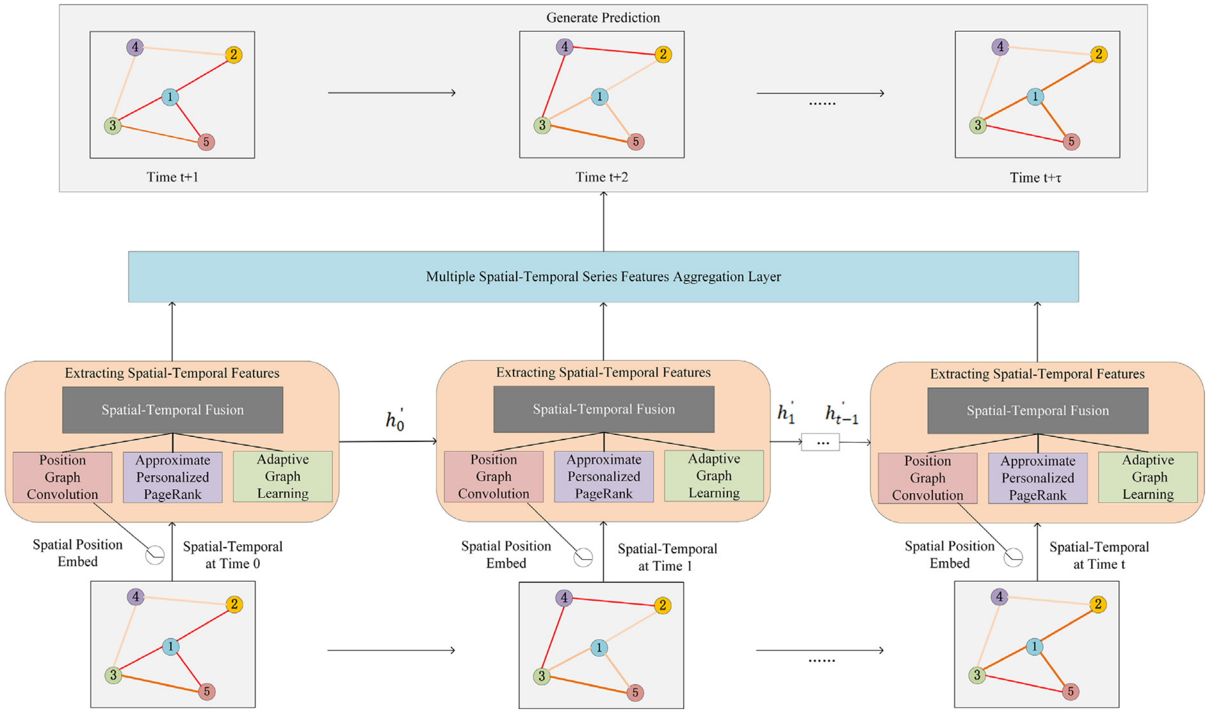


Fig. 2. GSTPRN architecture.

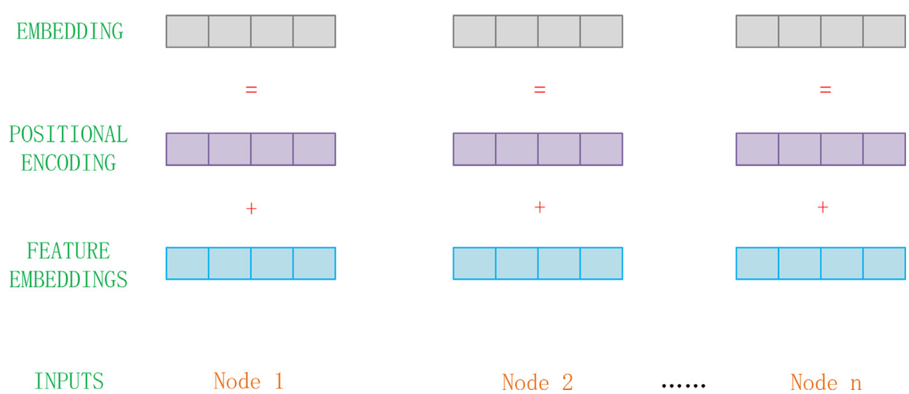


Fig. 3. Spatial position embedding.

4.1. Position graph convolution network

The GCN utilizes message passing among nodes and aggregates node neighborhood information to learn the matrix which is represented by the traffic section information. However, spatial position information and relationship strength of multiple nodes fail to be considered in GCN, resulting in the inability to capture high-level spatial information.

Given the above limitation, we construct position graph convolution based on self-attention, utilizing spatial position matrix and then calculate the dependence strengths of the spatial dimension to capture the spatial dependence relationship. In Fig. 3, we take t time as an example: each node is represented by feature embedding based on the spatial–temporal feature matrix $X'_{:,t} \in \mathbb{R}^{N\times d}$. For position encoding, we introduce an additional embedding vector for each node and then adopt a list of node indices to produce the corresponding position embedding (i.e. position embedding matrix $E^{sp}_{:,t}$). $E^{sp}_{:,t}$ then adds $X'_{:,t}$ to it to form the final embedding $X'^{sp}_{:,t} \in \mathbb{R}^{N\times d}$ as the input of the position graph convolution. We take the matrix $X'^{sp}_{:,t}$, and then the spatial dependence weight matrix $S^{p}_{:,t}$ represents the spatial dependence strengths among the nodes by self-attention. $S^{p}_{:,t}$ is obtained as follows:

$$S^{p}_{:,t} = softmax\left(\frac{X'^{sp}_{:,t} \cdot X'^{spT}_{:,t}}{\sqrt{d}}\right) \in \mathbb{R}^{N\times N} \tag{2}$$

S^{ab} reflects the dependence strength between nodes a and b in $S^{p}_{:,t}$. A big value represents a stronger dependency while a small value indicates weaker dependence. Next, we utilize spatial dependence weight matrix $S^{p}_{:,t}$ to adjust the static matrix (i.e. $\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}} = I_n + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$). The position graph convolution is expressed as follows:

$$S'_{:,t} = \sigma((\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}} \cdot S^{p}_{:,t})X'^{sp}_{:,t}W). \tag{3}$$

where, σ is a nonlinear activation function (i.e. ReLU), I_n denotes an unit matrix, $\widetilde{A} = I_n + A$ represents the adjacency matrix with introduced self-connection, $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$ is the diagonal degree matrix, and $W \in \mathbb{R}^{d\times f}$ is a projection matrix.

Our proposed model obtains $S^{p}_{:,t}$ to better adapt to the dynamic changes of the spatial–temporal traffic data. The reasons for better adaptation to the traffic situation are as follows:

(1) The recurrent network interacts with the information in the hidden state, so that the spatial position information embedding can identify the core or non-core regions, thereby helping in the diversion of traffic flow efficiently. (2) The spatial position information embedding contributes valuable semantic information to our model, such as the change of traffic flow from the residential region to the working region in the morning and from the working region to the residential region in the evening.

4.2. Approximate personalized propagation

In traffic graph data, a region (i.e. a node) affects its nearby and further regions, especially the center region. Therefore, the aggregation and exchange of information among regions is limited by the range of propagation. The multiple GCN layers aggregate node neighborhood information within a number of few-hop neighbors. However, GCN will result in over-smoothing and over-fitting as the number of layers increases, leading to performance degradation.

To tackle the limitation of node neighborhood propagation in traffic graph data, we develop approximate personalized propagation that extends the propagation range to obtain more spatial neighborhood information. More precisely, approximate personalized propagation is to aggregate an unlimited number of node neighborhoods through personalized PageRank, and then utilize approximate method to reduce the number of calculations. Fig. 4 shows the neighborhood information propagation scheme.

Personalized PageRank first obtains each root node based on the teleport (or restart vector) (i.e. one-hot vector), such that the root node $b \in N$ is generated by the teleport vector k_b. Personalized PageRank describes the root node b through a recurrent equation:

$$\tau_{ppr}(k_b) = (1-\alpha)\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}\tau_{ppr}(k_b) + \alpha k_b, \tag{4}$$

where $\alpha \in (0, 1]$ is the teleport probability which can control the root nodes' propagation range to obtain the different neighborhood information. Next, we solve Eq. (4) to get:

$$\tau_{ppr}(k_b) = \alpha(I_n - (1-\alpha)\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}})^{-1} \cdot k_b. \tag{5}$$

The influence score between nodes b and c is expressed as $K(b, c)$, which is proportional to the cth element of $\tau_{ppr}(k_b)$. Fully personalized PageRank matrix is achieved by replacing the indicator vector k_b with an unit matrix I_n, and the equation is described as follows:

$$\Pi_{ppr} = \alpha\left(I_n - (1-\alpha)\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}\right)^{-1}, \tag{6}$$

$K(b, c) \propto \Pi_{ppr}^{(bc)}$ is the influence score of (bc) pair, where different node pairs produce different influence scores.

In our propagation scheme, we encode the different spatial–temporal feature matrix (e.g. $X'_{:,t}$) by GCN (Kipf & Welling, 2016) to capture the node embedding matrix (e.g. $H_{:,t} : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times f}$). Then, fully personalized PageRank propagates each node with its own features to faraway neighbors to aggregate more neighborhood information as follows:

$$X^s_{:,t} = \alpha\left(I_n - (1-\alpha)\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}\right)^{-1} H_{:,t}. \tag{7}$$

From Eq. (7), we observe that fully personalized propagation computes the dense matrix $\mathbb{R}^{N\times N}$ with memory requirement $O(n^2)$. Therefore, approximate personalized propagation employs power iteration for fully personalized PageRank, and utilizes the sparse structure form of the graph (i.e. $\widetilde{A}_{sparse}$) to avoid the matrix $\mathbb{R}^{N\times N}$. This scheme can result in linear computation complexity. Unlike the power iteration of normal PageRank which employs a regular random walk, the personalized PageRank is related to a restart random walk. Each iteration of approximate personalized propagation is computed as follows:

$$\begin{aligned} M^{(0)}_{:,t} &= X^s_{:,t}, \\ M^{(j)}_{:,t} &= (1-\alpha)\widetilde{D}^{-\frac{1}{2}}\widetilde{A}_{sparse}\widetilde{D}^{-\frac{1}{2}}M^{(j-1)}_{:,t} + \alpha X^s_{:,t}, \\ M_{:,t} &= M^{(J)}_{:,t}, \end{aligned} \tag{8}$$

$X^s_{:,t}$ is both an initial vector and a set of teleport vector, and J is the number of power iteration ($j \in [0, J-2]$).

Compared with the GCN method which adds the additional layers and learns more parameters, approximate personalized propagation captures more neighborhood information via a few parameters and without the need to establish additional layers. Due to the propagation scheme utilizing the unlimited neighborhood aggregation layers, the gradient flows during backpropagation thus bring about more information to improve model accuracy.

4.3. Adaptive Graph Learning

Adaptive Graph Learning aims to improve GCN's ability via adaptive node parameter and adaptive graph generation. Adaptive node parameter decomposes the parameters in GCN, and then the specific parameters of the node are obtained from the shared set of weights and bias in all the nodes based on the node embedding. Adaptive graph generation infers node embedding from the data to produce the graph during training.

The GCN of one order Chebyshev produces the node embedding matrix, and reduces the number of parameters by sharing parameters among all the nodes. Thus, different spatial–temporal feature matrix (e.g. $X'_{:,t}$) is described as follows:

$$\overline{H}_{:,t} = (I_n + \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}})X'_{:,t}\Theta + b, \tag{9}$$

where, $X'_{:,t} \in \mathbb{R}^{N\times d}$ and $\overline{H}_{:,t} \in \mathbb{R}^{N\times f}$ represent the input and output of the GCN respectively, $\Theta \in \mathbb{R}^{d\times f}$ is the weights and $b \in \mathbb{R}^f$ is the bias.

In traffic series, the interaction among nodes are affected by external factors (e.g. weather, event) and show different situations at different times. It is an effective method to establish a specific-parameter space for each node, but arranging parameters for each node needs a large number of parameters (i.e. $\Theta \in \mathbb{R}^{N\times d\times f}$) to be optimized, which leads to over-fitting. Therefore, adaptive node parameter utilizes the idea of matrix factorization to enhance GCN. Taking weight as an example, $\Theta = E_G \cdot W_G$ is produced by the node embedding matrix $E_G \in \mathbb{R}^{N\times F}$ and a large shared weight pool $W_G \in \mathbb{R}^{F\times d\times f}$, in which $F \ll N$. For a single node (e.g. node i), node embedding E^i_G extracts parameter Θ^i from W_G, which can be viewed as finding a set of specific parameters for node learning. Likewise for bias b. Adaptive node parameter improves GCN's ability as follows:

$$\overline{H}_{:,t} = (I_n + \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}})X'_{:,t}\Theta + E_G b_G, \tag{10}$$

Adaptive graph generation can automatically capture implicit dependencies from the data. Firstly, all nodes are randomly initialized to generate the node embedding (i.e. $E_A \in \mathbb{R}^{N\times d_e}$), in which each row of E_A denotes a node embedding and d_e is the node embedding dimension. Next, we calculate E_A to multiply E_A^T to infer the dependency between each pair of nodes as shown:

$$\overline{A} = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}} = softmax(ReLU(E_A \cdot E_A^T)) \tag{11}$$

By replacing $\widetilde{A}$, the matrix $\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$ is produced to reduce the unnecessary recalculation during training. E_A can automatically capture the hidden dependencies among nodes during the training process to capture the adaptive matrix, shown as follows:

$$\overline{H}_{:,t} = (I_n + \overline{A})X'_{:,t}\Theta \tag{12}$$

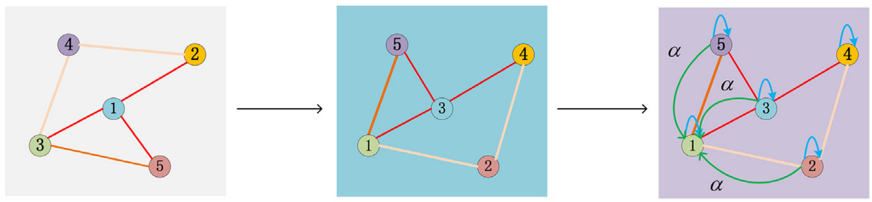


Fig. 4. Flowchart of neighborhood information propagation.

Finally, we combine adaptive node parameter (i.e. Eq. (10)) and adaptive graph generation (i.e. Eq. (12)) to obtain adaptive graph learning:

$$\overline{H}_{:,t} = (I_n + \overline{A})X'_{:,t}\Theta + E_G b_G \tag{13}$$

4.4. Traffic prediction

In this section, we integrate position graph convolution, approximate personalized propagation and adaptive graph learning into the recurrent network (i.e. Gated Recurrent Units), described as follow:

$$\begin{aligned}
\widetilde{S} &= S'[X'_{:,t}, h'_{t-1}],\\
\widetilde{M} &= M[X'_{:,t}, h'_{t-1}],\\
\widetilde{H} &= \overline{H}[X'_{:,t}, h'_{t-1}],\\
r_t &= \sigma(\widetilde{S}_r + \widetilde{M}_r + \widetilde{H}_r),\\
z_t &= \sigma(\widetilde{S}_z + \widetilde{M}_z + \widetilde{H}_z),\\
\overline{k}_t &= tanh(S'[X'_{:,t}, r \odot h'_{t-1}]\\
&\quad + M'[X'_{:,t}, r \odot h'_{t-1}]\\
&\quad + H'[X'_{:,t}, r \odot h'_{t-1}]),\\
k_t &= z \odot h'_{t-1} + (1-z) \odot \overline{k}_t.
\end{aligned} \tag{14}$$

where the input and the output at time t are $X'_{:,t}$ and k_t, respectively. The reset gate and the update gate are r and z. $\odot$ is the operation of Hadamard Product. σ is the sigmoid activation function.

In GSTPRN framework, we choose L1 loss function to obtain our training objective, and employ Adam optimizer to optimize all the parameters. The loss function is expressed as follows:

$$L(\theta_{parameters}) = \sum_{t+1}^{t+\tau} |X_{:,i} - X'_{:,i}| \tag{15}$$

where $\theta_{parameters}$ is all the parameters in GSTPRN, $X_{:,i}$ and $X'_{:,i}$ express the ground truth and model prediction, respectively.

5. Experiment

To evaluate our proposed GSTPRN, we conduct a series of experiments on two real datasets (i.e. PeMSD4 and PeMSD8) from Caltrans Performance Measurement System (PeMS). PeMS data are collected in real-time from nearly 40,000 individual detectors spanning the freeway system across all major metropolitan areas of California. The datasets collect highway data in real-time every 30 s, and the traffic data are aggregated from the original data in 5-min interval. The datasets contain three data types (i.e total flow, average occupancy and average speed).

5.1. Dataset

PeMSD4: This dataset contains the traffic data of 307 traffic observation detectors in the San Francisco Bay region. The detectors collect data on the traffic flow of the highway from 1/1/2018 to 2/28/2018.

Table 1
Baseline comparison on traffic forecasting.

Method	PeMSD4			PeMSD8		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
VAR	24.67	38.77	17.35%	19.31	30.24	12.61%
GRU-ED	23.49	38.69	16.72%	21.79	35.12	12.99%
DSANet	22.68	34.88	15.98%	17.05	26.64	11.23%
DCRNN	21.57	33.98	15.14%	16.98	26.91	10.98%
MSTRPGNN	21.98	34.21	15.49%	17.67	27.51	12.14%
ASTGCN	21.91	34.09	15.36%	17.55	27.24	12.01%
STSGCN	21.43	33.72	14.83%	17.31	27.03	11.51%
STODE	21.17	32.98	14.74%	16.81	25.94	10.66%
AGCRN	20.08	32.84	13.42%	16.98	26.99	10.90%
GSTPRN	**19.45**	**31.91**	**12.96%**	**15.68**	**24.96**	**10.09%**

PeMSD8: This dataset contains the traffic data of 170 traffic observation detectors in the San Bernardino region. The detectors collect data on the traffic flow of the highway from 7/1/2016 to 8/31/2016.

5.2. Settings

We divide the two datasets into training set, validation set, and test set with a ratio of 6:2:2. We adopt three widely used evaluation metrics, namely, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) to compare the performance of the baseline methods.

We adopt one-hour historical observation data to forecast next one-hour traffic situation, in which the input is 12 steps (i.e. 12 horizons) of historical observation data and the output is next 12 steps (i.e. 12 horizons). Hyperparameter settings used in our model are : learning rate is 0.001; hidden dimension is 64; power iteration step is 10, teleport probability is 0.1. Early stopping algorithms are applied to our model, and we set the patience threshold to 15.

5.3. Baseline methods

- VAR (Zivot & Wang, 2006) considers the internal dependencies in multiple time series.
- GRU-ED (Cho et al., 2014) constructs two recurrent neural networks and combines encoder–decoder architecture to learn traffic time series features.
- DSANet (Huang et al., 2019) designs two parallel convolutional components and self-attention mechanism to jointly learn the dynamic periodic spatial–temporal features.
- DCRNN (Li et al., 2017) utilizes RNN structure, which establishes bidirectional random walks to learn the spatial correlation information. It also designs an encoder–decoder structure to capture the temporal information.
- MSTRPGNN (Cascetta & Cantarella, 1991; Nuzzolo & Russo, 1996) extracts multiple spatial–temporal information by random parameters in graph neural network.

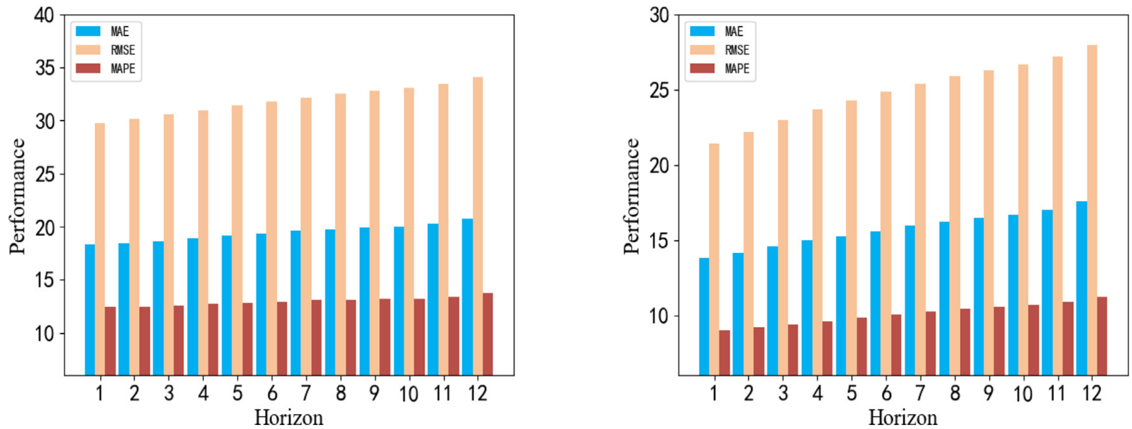
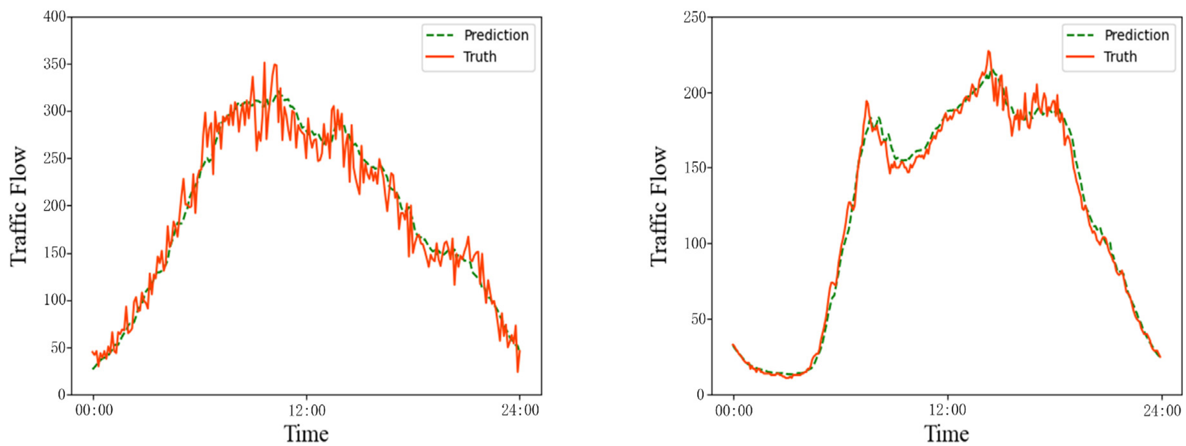


Fig. 5. PEMSD4 (Left) and PEMSD8 (Right).



(a) Node 40 (b) Node 198

Fig. 6. Visualization of prediction and truth.

• ASTGCN (Guo et al., 2019) employs the attention mechanism and graph convolutional network to form a spatial–temporal network to learn dynamic spatio-temporal correlations information.

• STSGCN (Song et al., 2020) proposes a synchronous GCN model to capture heterogeneous local spatial–temporal characteristics during the different time points.

• STGODE (Fang et al., 2021) constructs an ODE and two TCN blocks with residual structure to learn traffic information.

• AGCRN (BAI et al., 2020) proposes node adaptive parameter learning and data adaptive graph generation for each traffic sequence and different traffic sequences without pre-defined graphs.

5.4. Performance comparison and analysis on traffic prediction

In Table 1 , our proposed GSTPRN is compared with multiple baseline methods, and the experimental results show that our method is superior to the SOTA methods on two traffic datasets. We observe that the methods based on spatial–temporal dimension (e.g. DSANet, DCRNN, ASTGCN, STSGCN, STGODE, AGCRN, GSTPRN) achieve better performance than the methods based on temporal dimension (e.g. VAR, GRU-ED), because the spatial dimension information reflects the influence among regions at different distances, especially in smart cities.

Therefore, we conduct further analysis on the above spatial–temporal dimension methods. DSANet, DCRNN, ASTGCN, STSGCN and STGODE employ shared patterns of parameters to learn traffic graph data, while they fail to achieve node-specific patterns. AGCRN determines the parameters and generates the graph adaptatively but it does not utilize the high-level information.

Our proposed GSTPRN can obtain high-level information and constructs position graph convolution to utilize position embedding matrix to capture the spatial dependence relationship among multiple nodes. GSTPRN also adopts approximate personalized propagation to obtain more spatial neighborhood information. Thus, GSTPRN achieves better performance.

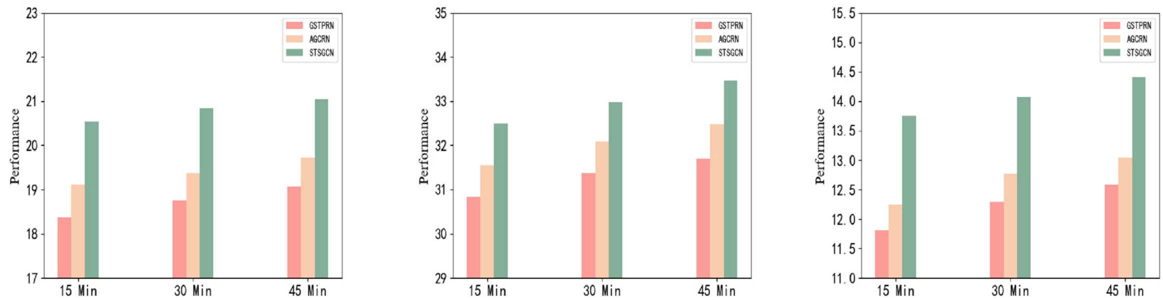
5.5. Prediction performance analysis at each horizon

We further validate GSTPRN's performance for different horizons (i.e. different steps) using the two datasets and the results are shown in Fig. 5. We observe that the prediction performance decreases (i.e. MAE, RMSE, MAPE increases) when the horizon increases. However, the performance change is small from the one horizon to the next horizon, showing that our proposed GSTPRN is stable.

Overall, the gradual prediction performance trajectory and small performance changes across horizons show that our model is effective.

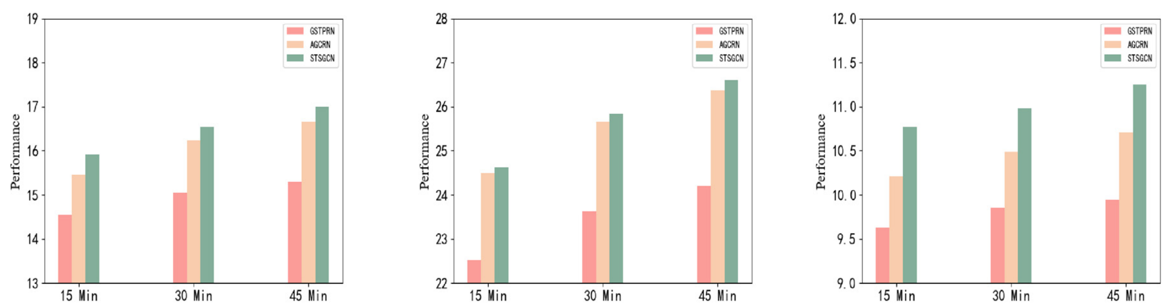
5.6. Visualization analysis

Fig. 6 shows the traffic flow across time to reflect the prediction versus the truth. From Fig. 6(a), we observe that the traffic flow increases gradually before 11:00, and then decreases. The



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

Fig. 7. Performance of multiple time intervals on PEMSD4.



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

Fig. 8. Performance of multiple time intervals on PEMSD8.

traffic flow fluctuates more obviously around 12:00. In Fig. 6(b), the traffic flow registers a slight decrease and then increases rapidly. A peak is reached after a period of 12:00, followed by a decline.

Compared to Fig. 6(b), Fig. 6(a) shows that the truth has high and fluctuating traffic flow. Correspondingly, the prediction also fluctuates accordingly. The visualization of the prediction versus the truth illustrates that our GSTPRN is effective.

5.7. Performance of multiple time intervals

We evaluate the effect of several methods (i.e. GSTPRN, AGCRN, STSGCN) on multiple time intervals, as shown in Figs. 7 and 8.

On two datasets, each method consistently improves in performance as the time decreases. From Fig. 7, we observe that STSGCN declines significantly in MAE and MAPE compared to GSTPRN and AGCRN. In Fig. 8, the performance of STSGCN and AGCRN is close in terms of MAE and RMSE.

Overall, our proposed GSTPRN outperforms AGCRN and STSGCN in multiple time intervals by the two datasets.

5.8. Ablation study

To further illustrate the performance of GSTPRN, we conduct ablation experiments to reflect the effects of the different modules.

- GSTPRN-noAGL: this method removes the adaptive graph learning module to study the benefits of specific parameters and data inference.
- GSTPRN-noAPP: this method deletes the approximate personalized propagation module to study the advantages of extending node neighborhood information.

Table 2
Performance of different modules on PeMSD4.

Method	PeMSD4		
	MAE	RMSE	MAPE
GSTPRN	19.45	31.91	12.96%
GSTPRN-noAGL	19.75	32.42	13.17%
GSTPRN-noAPP	19.87	32.65	13.31%
GSTPRN-noPGC	19.94	32.75	13.48%

Table 3
Performance of different modules on PeMSD8.

Method	PeMSD8		
	MAE	RMSE	MAPE
GSTPRN	15.68	24.96	10.09%
GSTPRN-noAGL	15.93	25.69	10.31%
GSTPRN-noAPP	16.18	25.88	10.47%
GSTPRN-noPGC	16.35	26.34	10.69%

- GSTPRN-noPGC: this method does not consider position graph convolution module to study the importance of position information and spatial dependence.

From the experimental results in Table 2, we can see that GSTPRN-noAGL, GSTPRN-noAPP and GSTPRN-noPGC all registered performance degradation compared to GSTPRN. We observe that removing the position graph convolution (i.e. GSTPRN-noPGC) has a greater impact on performance, which shows that position information and spatial dependence are more sensitive to the model. Moreover, GSTPRN-noAGL achieves better performance than GSTPRN-noAPP, which shows that expanding the propagation range of nodes neighborhood is more effective than the adaptive graph learning module.

We also evaluate the performance on PeMSD8 dataset, which is shown in Table 3. Specifically, we find that GSTPRN-noAGL achieves better performance than GSTPRN-noAPP and GSTPRN-noPGC, which shows that position information, spatial dependence and expansion of neighborhood propagation range are more important in traffic prediction. Furthermore, when comparing GSTPRN-noAPP and GSTPRN-noPGC, we observe that position information and spatial dependencies bring more benefits than node neighborhood information.

The results of the ablation experiments show that the individual modules do contribute towards GSTPRN's improvements, resulting in a very effective architecture for traffic prediction.

6. Conclusion

In this paper, we aim to analyze and tackle the problems of spatial position information, spatial dependence and the propagation of node neighborhood information in traffic forecasting.

Our proposed GSTPRN architecture is an end-to-end structure and jointly integrate position graph convolution, approximate personalized propagation and adaptive graph learning into the recurrent network (i.e. Gated Recurrent Units). More precisely, we construct position graph convolution based on self-attention, that utilizes position embedding matrix and computes spatial dependence strengths to capture the spatial dependence among multiple nodes. Moreover, our architecture incorporates approximate personalized propagation which aggregates an unlimit number of neighborhood propagation layers to extend the range of node neighborhood propagation to capture more neighborhood information. For a comprehensive comparison against the various baseline methods, we also design multiple experiments (e.g. different modules and time intervals on PeMSD4 and PeMS08) for a more complete performance analysis. The extensive experimental results show that our model outperforms the SOTA baselines.

CRediT authorship contribution statement

Yibi Chen: Conceptualization, Methodology, Software. **Kenli Li:** Supervision, Formal analysis, Writing - review & editing. **Chai Kiat Yeo:** Formal analysis, Writing - review & editing. **Keqin Li:** Supervision, Validation, Software.

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.

Acknowledgments

This research was supported by the Key Program of the National Natural Science Foundation of China (Grant Nos. 61133005, 61432005) and Postgraduate Scientific Research Innovation Project of Hunan Province (Grant Nos. CX20220413).

References

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

An, J., Guo, L., Liu, W., Fu, Z., Ren, P., Liu, X., et al. (2021). IGAGCN: Information geometry and attention-based spatiotemporal graph convolutional networks for traffic flow prediction. *Neural Networks*, *143*, 355–367.

BAI, L., Yao, L., Li, C., Wang, X., & Wang, C. (2020). Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in Neural Information Processing Systems*, *33*.

Cai, W., & Wei, Z. (2020). Remote sensing image classification based on a cross-attention mechanism and graph convolution. *IEEE Geoscience and Remote Sensing Letters*.

Cascetta, E., & Cantarella, G. E. (1991). A day-to-day and within-day dynamic stochastic assignment model. *Transportation Research Part A: General*, *25*(5), 277–291.

Chen, W., Chen, L., Xie, Y., Cao, W., Gao, Y., & Feng, X. (2020). Multi-range attentive bicomponent graph convolutional network for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 04* (pp. 3529–3536).

Chen, Y., Zou, X., Li, K., Li, K., Yang, X., & Chen, C. (2021). Multiple local 3D CNNs for region-based prediction in smart cities. *Information Sciences*, *542*, 476–491.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Di Gangi, M., & Croce, A. (2005). Combining simulative and statistical approach for short time flow forecasting. In *Proceedings of ETC 2005, Strasbourg, France 18–20 September 2005-transport policy and operations-traffic engoneering and street management-intergrated traffic management II*.

Dial, R. B. (2006). A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transportation Research, Part B (Methodological)*, *40*(10), 917–936.

Dogra, R., Rani, S., Babbar, H., Verma, S., Verma, K., & Rodrigues, J. J. (2022). DCGCR: Dynamic clustering green communication routing for intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*.

Fan, X., Xiang, C., Chen, C., Yang, P., Gong, L., Song, X., et al. (2020). BuildSenSys: Reusing building sensing data for traffic prediction with cross-domain learning. *IEEE Transactions on Mobile Computing*, *20*(6), 2154–2171.

Fang, Z., Long, Q., Song, G., & Xie, K. (2021). Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 364–373).

Ghosh, B., Basu, B., & O'Mahony, M. (2009). Multivariate short-term traffic flow forecasting using time-series analysis. *IEEE Transactions on Intelligent Transportation Systems*, *10*(2), 246–254.

Guo, S., Lin, Y., Feng, N., Song, C., & Wan, H. (2019). Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01* (pp. 922–929).

Guo, S., Lin, Y., Wan, H., Li, X., & Cong, G. (2021). Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*.

Huang, S., Wang, D., Wu, X., & Tang, A. (2019). Dsanet: Dual self-attention network for multivariate time series forecasting. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 2129–2132).

Huang, F., Yi, P., Wang, J., Li, M., Peng, J., & Xiong, X. (2022). A dynamical spatial-temporal graph neural network for traffic demand prediction. *Information Sciences*, *594*, 286–304.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Lee, S., & Fambro, D. B. (1999). Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation Research Record*, *1678*(1), 179–188.

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926.

Lin, Z., Feng, J., Lu, Z., Li, Y., & Jin, D. (2019). Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. In *Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01* (pp. 1020–1027).

Nuzzolo, A., & Russo, F. (1996). Stochastic assignment models for transit low frequency services: Some theoretical and operative aspects. In *Advanced methods in transportation analysis* (pp. 321–339). Springer.

Pan, Z., Liang, Y., Wang, W., Yu, Y., Zheng, Y., & Zhang, J. (2019). Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1720–1730).

Papageorgiou, M., & Kotsialos, A. (2002). Freeway ramp metering: An overview. *IEEE Transactions on Intelligent Transportation Systems*, *3*(4), 271–281.

Park, C., Lee, C., Bahng, H., Tae, Y., Jin, S., Kim, K., et al. (2020). ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 1215–1224).

Song, C., Lin, Y., Guo, S., & Wan, H. (2020). Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 01* (pp. 914–921).

Tang, X., Yao, H., Sun, Y., Aggarwal, C., Mitra, P., & Wang, S. (2020). Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 04* (pp. 5956–5963).

Van Lint, J., & Van Hinsbergen, C. (2012). Short-term traffic and travel time prediction models. *Artificial Intelligence Applications to Critical Transportation Issues*, *22*(1), 22–41.

Wang, Y., Zhang, J., Kan, M., Shan, S., & Chen, X. (2020). Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12275–12284).

Wang, J., Zhu, W., Sun, Y., & Tian, C. (2021). An effective dynamic spatiotemporal framework with external features information for traffic prediction. *Applied Intelligence*, *51*(6), 3159–3173.

Wei, C., & Sheng, J. (2020). Spatial-temporal graph attention networks for traffic flow forecasting. In *IOP conference series: Earth and environmental science, vol. 587, no. 1*. IOP Publishing, Article 012065.

Wu, C. H., Ho, J. M., & Lee, D. T. (2004). Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems*, *5*(4), 276–281.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24.

Yan, S., Xiong, Y., & Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*.

Yao, H., Tang, X., Wei, H., Zheng, G., & Li, Z. (2019). Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01* (pp. 5668–5675).

Yao, H., Tang, X., Wei, H., Zheng, G., Yu, Y., & Li, Z. (2018). Modeling spatial-temporal dynamics for traffic prediction. arXiv preprint arXiv:1803.01254.

Ying, Z., Cao, S., Liu, X., Ma, Z., Ma, J., & Deng, R. H. (2022). PrivacySignal: Privacy-preserving traffic signal control for intelligent transportation system. *IEEE Transactions on Intelligent Transportation Systems*.

Zhang, J., Zheng, Y., & Qi, D. (2017). Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence, vol. 31, no. 1*.

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., et al. (2019). T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, *21*(9), 3848–3858.

Zheng, C., Fan, X., Wang, C., & Qi, J. (2020). Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 01* (pp. 1234–1241).

Zivot, E., & Wang, J. (2006). Vector autoregressive models for multivariate time series. In *Modeling Financial Time Series with S-Plus®* (pp. 385–429). Springer.