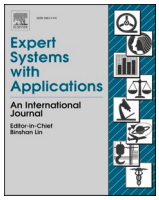
Expert Systems With Applications 239 (2024) 122449

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Location and time embedded feature representation for spatiotemporal traffic prediction

Wei Li [a], Xin Liu [a], Wei Tao [b], Lei Zhang [c], Junhua Zou [a], Yu Pan [d], Zhisong Pan [a,*]

[a] Command and Control Engineering College, Army Engineering University, Nanjing 210007, PR China
[b] Strategic Assessments and Consultation Institute, Academy of Military Science, Beijing 100091, PR China
[c] Academy of Military Science, Beijing 100091, PR China
[d] National University of Defense Technology, Changsha 410073, PR China

ARTICLE INFO

Keywords:
Traffic Prediction
Spatiotemporal Sequence
TransD Algorithm
Embedding
Graph Convolutional Network

ABSTRACT

As a fundamental spatiotemporal sequence forecasting problem, traffic prediction is pivotal in transportation management and urban computing. Nonetheless, the intricate and dynamic nature of spatiotemporal correlations presents significant obstacles in acquiring precise forecasts. Existing techniques utilize graph convolutional networks in conjunction with temporal modules, such as recurrent neural networks or transformer-based structures, to effectively extract spatiotemporal features. Unfortunately, current approaches struggle with outliers and fail to capture potential global correlations between different timestamps. In this study, we propose an innovative **S**patio-**T**emporal **G**raph **C**onvolution **N**etwork with **E**mbedded location and time features (STEGCN) for traffic prediction problems, which can generate precise and prompt predictions. STEGCN effectively captures the complex interdependencies among location, time, and traffic volume by leveraging the TransD algorithm to embed their representations. For each timestamp, a graph convolution module is exploited to capture the spatial features, merged with the embeddings of location and time that serve as global external information. Then, we leverage a temporal module composed of 1-D convolutions to capture the spatiotemporal patterns. The traffic volume embedding is employed to constrain predictions within a reasonable range. Extensive experiments and rigorous analysis show that our STEGCN model outperforms state-of-the-art baselines, demonstrating exceptional performance and potential for practical application.

1. Introduction

As a typical spatiotemporal sequence forecasting problem, traffic prediction has emerged as the vital component of intelligent city construction and the indispensable cornerstone of the Intelligent Transportation System (ITS) (Wang, 2010). Precise traffic volume estimation has proven its importance and necessity in many fields, such as traffic congestion prediction (Akhtar and Moridpour, 2021), travel time estimating (Qiu et al., 2019), taxi dispatching (Xu et al., 2017), urban planning (Lin et al., 2019), pedestrian flow safety (Galanis et al., 2017); and so on. Hence, traffic prediction has become a leading domain of interest among researchers.

Traffic data presents a typical illustration of spatiotemporal data, encapsulating information on vehicle travel captured by various sensors situated in different locations. Each sensor's observation record constitutes a spatiotemporal data point, encompassing dynamic and interdependent spatial and temporal patterns. In traffic data, temporal information typically manifests as a continuous time series, which can be utilized to portray vehicle speed and density, among other features. In contrast, spatial information is acquired from multiple sensor positions to reflect the trajectory and road network topology. As shown in Fig. 1, as sensor 1 and sensor 2 are located close to each other in space, the feature patterns of the data they collect are similar. Due to its complicated spatiotemporal patterns hidden behind data, traffic prediction has remained a highly challenging matter, demanding considerable attention and expertise over an extended period.

Currently, the combination of Graph Neural Networks (GNN) (Kipf and Welling, 2016) (Jiang and Luo, 2022) with Recurrent Neural Networks (RNN) (Rumelhart et al., 1986) (Abdoos and Bazzan, 2021) has become the prevailing and extensively adopted technique for traffic

* Corresponding author at: No. 1 Haifu Lane, Guanghua Road, Qinhuai District, Nanjing, Jiangsu, PR China.
E-mail addresses: liwei@aeu.edu.cn (W. Li), liuxin@aeu.edu.cn (X. Liu), wtao_plaust@163.com (W. Tao), zhanglei@aeu.edu.cn (L. Zhang), jh_zou@aeu.edu.cn (J. Zou), panyu0511@nudt.edu.cn (Y. Pan), panzhisong@aeu.edu.cn (Z. Pan).

https://doi.org/10.1016/j.eswa.2023.122449
Received 20 April 2023; Received in revised form 2 September 2023; Accepted 2 November 2023
Available online 4 November 2023

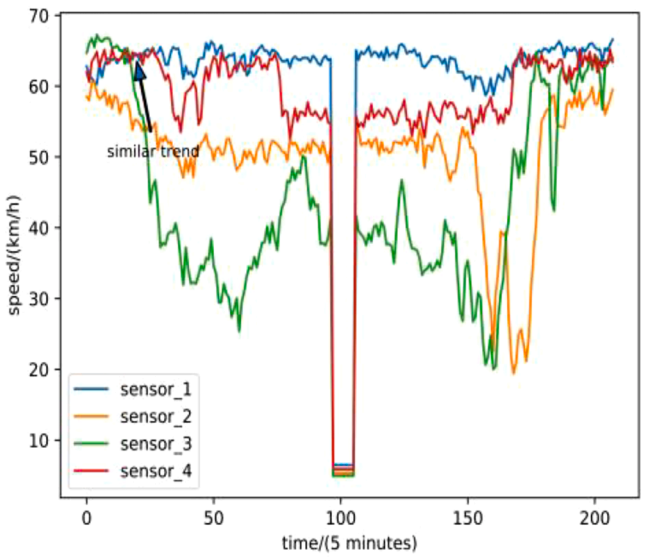


Fig. 1. Similar trend between neighborhood sensors in traffic data.

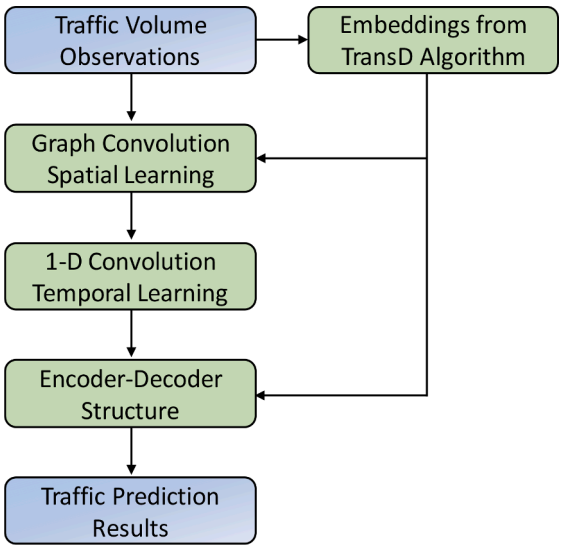


Fig. 2. A brief concept map of our proposed approach.

prediction, which effectively captures spatiotemporal patterns. The integration technique between the spatial and temporal components can be broadly classified into three categories: sequential, parallel, and fused methods. The representative of the sequential methods is STGCN (Yu et al., 2018), which stacks spatial and temporal features in sequential order. GMAN (Zheng et al., 2020) is a typical parallel method that utilizes the spatial and temporal modules simultaneously. The fusion mechanism aims at modifying the inner structure of the RNN module, such as replacing the multiplication with the convolution operation (DCRNN (Li et al., 2018). Besides, some approaches (ST-ResNet (Zhang et al., 2017), DMVST-Net (Yao et al., 2018), STNN (He et al., 2020) leverage external information, including holidays, weather, and events, to promote the extraction of dynamic spatiotemporal patterns. Nonetheless, these approaches still suffer from the following shortcomings:

- During the training procedure, the entire spatiotemporal series is segmented into individual samples using sliding windows without considering the global relationship among location, time, and traffic volume, thus neglecting the global correlations between different time intervals.
- The prediction accuracy is largely affected by the anomalies, which can cause unnecessary parameter shifts in the model, resulting in overfitting of the training data and overly optimistic accuracy evaluations. This can lead to poor performance of the model in handling new and unseen data due to significant differences between the outliers and the training data.
- RNN structures are widely utilized in traffic prediction models, with drawbacks such as high computational complexity, large time consumption, and difficulty handling long sequences.

To overcome the concerns mentioned above, we present an innovative Spatio-Temporal Graph Convolution Network with Embedded location and time features (STEGCN) for traffic volume prediction, consisting of four parts. As illustrated in Fig. 2, we utilize the TransD algorithm (Ji et al., 2015) to obtain the interconnected feature representation by embedding location, time, and traffic volume into a new space. The embeddings learned are exploited as global external information to assist in extracting spatiotemporal patterns. With enhanced spatiotemporal statistics, a graph convolution module captures the spatial correlations among different regions. To alleviate the pressure of computation cost, we leverage a 1-D convolution (Kiranyaz et al., 2015) module to model the temporal properties with simplified parameters, which are more interpretable and can be run in parallel. Subsequently, an encoder-decoder structure comprising fully connected layers is proposed to handle the extracted spatiotemporal features. With the traffic volume embedding representation from the TransD algorithm, the module can constrain prediction values within a reasonable range. In summary, the main contributions of this paper can be listed as follows:

- We are the first to employ knowledge graph algorithms to learn the entity and relation embeddings as external information for traffic volume prediction. The embeddings serve as the global external information and restrict the forecast in a reasonable range. This idea can also be extended to other spatiotemporal sequence prediction problems such as nowcasting and crop yield prediction.
- We propose a temporal block composed of a 1-D convolution and mask mechanism. By utilizing the 1-D convolution module, we effectively capture temporal correlations, enabling the encoding of patterns across various scales at a faster speed. Additionally, the integration of a masking mechanism ensures model consistency throughout both the training and testing stages.
- The efficacy of the proposed model is evaluated across six publicly available real-world datasets. Based on experimental results, it is evident that STEGCN achieves superior performance compared to other baselines. Additionally, we conducted an ablation study to assess each component's efficacy within our model.

The following content of the paper is arranged as follows: In Section 2, this study provides a literature review of the relevant work and retrospectively analyzes the research process of traffic flow prediction while introducing the current research status in this field. We provide some preliminaries in Section 3 to facilitate comprehension of this research. In Section 4, the detailed implementations of the architecture are introduced. Section 5 describes the experimental settings and results which proved the effectiveness of our model. Section 6 concisely summarizes this work and furnishes prospects for future research.

2. Related work

Traffic prediction serves as the core problem of intelligent urban computing and construction, while it remains a challenge because of the intricate and dynamic spatiotemporal patterns among traffic data. Before deep learning arose, researchers used statistical methods and traditional machine learning approaches to tackle traffic prediction problems. The representatives of the statistical methods are Autoregressive Integrated Moving Average (ARIMA) (Abellana, 2021), Support

Vector Regression (SVR) (Barbour et al., 2018), Gradient Boosting Decision Tree (GBDT) (Wu et al., 2020), Vector Auto-Regression (VAR) (Polson and Sokolov, 2017), and Linear Conditional Gaussian Bayesian Networks (LCGBN) (Zhu et al., 2016). It should be noted that the effectiveness of the methods mentioned above relies on strong assumptions regarding the stationarity, linearity, and non-periodicity of the series, which is unrealistic when dealing with real-world data (Xie et al., 2020). Furthermore, while ignoring the spatial relationships between different regions, these models are primarily designed for small datasets and hence need more capacity to handle the exponentially expanding transportation data.

Deep learning has facilitated rapid advancement in various fields, encompassing traffic prediction, by providing an innovative predictive model that can handle the complexity and heterogeneity of transportation data (Fadlullah et al., 2017). To extract the spatial relations among regions, Zhang et al. (2016) first represented the traffic volume data as a series of images and applied a convolutional neural network (CNN) (LeCun et al., 1998). Considering the temporal correlations, Jin et al. (2018) proposed a model named STRCN, which introduces Long Short-term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) along with the CNN structure to extract spatiotemporal features simultaneously. In the work of Gong et al. (2020), three spatiotemporal models (OLS-AO, OLS-DT, and OLS-MR) were used to predict the network-wide crowd flow distribution based on the online latent space (OLS) strategy. Subsequently, a series of similar works (GraphCNN-LSTM (Bogaerts et al., 2020); CLM (Cao et al., 2020); ST-DCCNAL (Li et al., 2019)) are proposed at this angle. However, modeling traffic data as a grid (Fig. 3a) would require sacrificing significant underlying structural information, leading to poor prediction accuracy. As shown in Fig. 3b, traffic data sensors are often irregularly deployed, making it challenging to represent the data using traditional grid-based approaches. In recent years, GNNs have garnered significant attention in traffic prediction due to their notable success in effectively tackling non-Euclidean structure data. Many studies have integrated graph convolution with various architectures to address traffic forecasting problems. For instance, some studies have combined graph convolution with the gated recurrent unit (DCRNN (Li et al., 2018), while others used a combination of graph convolution and 1-D convolution (STGCN (Yu et al., 2018). Ou et al. (2022) proposed a model named STP-TrellisNets+, which combines the temporal convolutional framework TrellisNet with diffusion graph convolutional networks to predict multi-step passenger flow in the metro system. Some approaches leverage an adaptive adjacency matrix of the graph (Graph WaveNet (Wu et al., 2019)) or graph structure learning with temporal convolution (MTGNN (Wu et al., 2020)). Another approach fuses graph convolution layers from different perspectives (STFGNN (Li and Zhu, 2021 Wang et al., 2023)). However, these methods typically handle traffic data sample by sample without considering global correlations among location, time, and traffic volume. Meanwhile, as the intricacy of the model architecture and the scale of model parameters escalate, a corresponding decrement in training and execution velocity is encountered. This prevalent phenomenon must be carefully addressed as a critical concern that demands attention. Our proposed STEGCN model comprehensively incorporates location, time, and traffic volume information, making use of global information to establish linkages. This approach results in predicted values that are constrained within a reasonable interval, thereby enabling more accurate predictions of traffic volume. As a result, our model provides robust support for traffic management planning.

3. Preliminaries

3.1. Notation

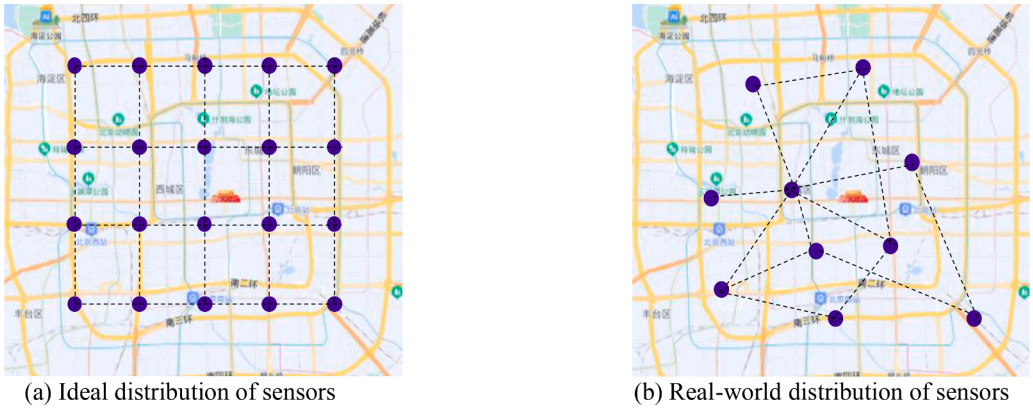
To reduce confusion for the reader regarding the specific symbols or terms used to describe our research, we have compiled a table of symbols or terms in this paper. The main notations that appeared in this paper are summarized in Table 1.

3.2. Problem definition

Assuming that there are N sensors deployed to collect traffic volume data in their respective areas on the road network. In light of practical

Table 1
Notations in this paper.

Name	Description
G	An undirected graph
V, v_k	Nodes of a graph, the k-th node. $\|V\| = N$
E, e_{ij}	Edges of a graph, the edge between v_i and v_j
A	The adjacency matrix of a graph
D	The degree matrix of a graph
I_n	Identity matrix with dimension n
W	Convolution kernel
b	Bias
h, H	Hidden layer representations in graph convolution
$\sigma(\cdot)$	A non-linear activation function
f, θ	Model function and its parameters
$Concat(\cdot)$	A function utilized to concatenate tensors
$X^t \in \mathbb{R}^{N \times d}$	Traffic volume at time interval t with dimension d
$X^{P:Q}$	Abbreviation of $\{X^P, X^{P+1}, ..., X^Q\}$
h, t, r	The head entity, tail entity, and relation
$E_L, E_T, E_V \in \mathbb{R}^m$	Embedding of location, time, and traffic volume with dimension m trained by the transit algorithm



(a) Ideal distribution of sensors (b) Real-world distribution of sensors

Fig. 3. Different modeling methods for the distribution of sensors on road networks. (a) The ideal distribution of sensors exhibits a regular and grid-like pattern, which can be modeled as images. (b) In practical deployments, the arrangement of sensors often follows an irregular and non-Euclidean structure, which is better suited for modeling using graphs. In both scenarios (a) and (b), the dashed lines illustrate the schematic network topology between sensors.

considerations, the road network is always regarded as a graph $G = < V, E, A >$ rather than a regular grid-like network. We utilize $X_t \in \mathbb{R}^{N \times d}$ to represent the total traffic observations of dimension d at each time interval t. Given the historical observation $X_{t-P+1:t}$ in the past P time intervals, traffic prediction aims to learn a model function f to predict the traffic volumes in the next Q time interval, which can be defined as follows:

$$X_{t+1:t+Q} = f(X_{t-P+1:t}, \theta), \tag{1}$$

where θ denotes the model parameters.

3.3. Graph convolutional network

Graph neural network (GNN) is a deep learning framework that can effectively process graph-structured data. Unlike traditional deep learning models focusing on grid-like data, GNNs allow for processing non-Euclidean data such as social networks, molecules, and trajectories. There are two main categories of GNN: spatial-based graph convolution (Velickovic et al., 2017;Defferrard et al., 2016) and spectral-based graph convolution (Bruna et al., 2013;Hechtlinger et al., 2017). In the spatial-based graph convolution, feature information from neighborhood nodes is aggregated in the spatial domain to form the higher representation of a node. It can be summarized as a weighted average pooling operation based on the adjacency matrix, which calculates the output feature by linearly computing the input feature. The whole process can be defined as follows:

$$h_i^{(l+1)} = \sigma\left(\sum_{j=1}^{N} W_{ij}^{(l)} h_j^{(l)}\right), i = 1, 2, ..., N, l = 0, 1, 2, \cdots \tag{2}$$

where h_i^l denotes the feature vector of node i and $W_{ij}^{(l)}$ represents the convolution kernel in the l-th layer.

The spectral-based graph convolution is an approach to convolutional operations that effectively handles graph-structured data by leveraging the Fourier transform. The technique involves several steps, beginning with decomposing the normalized Laplacian matrix of the graph into its spectral components using eigenvectors and eigenvalues. Next, convolution is applied in the spectral domain using the same filters. Non-linear activation functions are then implemented to filter the output signal, followed by pooling operations for down-sampling. The formulation is shown below:

$$H = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}XW\right), \tag{3}$$

where the $\widetilde{A} = A + I_n$ is the adjacency matrix with self-loops, $\widetilde{D}$ denotes the degree matrix of $\widetilde{A}$, and W is the convolution kernel to be learned.

3.4. TransD algorithm

The knowledge graph embedding representation algorithm is a method used to map the entities and relations within a knowledge graph with the intention of better comprehending and analyzing the structural and semantic information present. The aim of these algorithms is to represent entities and relations as continuous vectors, thus resulting in entities with similar semantics existing close together within vector space. Additionally, through vector operations, new entities and relations may be inferred. The common knowledge graph representation algorithms include TransE (Bordes et al., 2013); TransH (Wang et al., 2014); TransR (Lin et al., 2015), and TransD (Ji et al., 2015). Among these algorithms, TransD stands out as it offers several advantages over the previous three methods. These advantages encompass the handling of many-to-many relationships, improved semantic representation capabilities, and enhanced reasoning abilities. Therefore, we have selected TransD as our choice for learning the embedding representation of location, time, and traffic.

Specifically, for each relation r and entity e, TransD introduces two corresponding transformation matrices M_r and M_e, which are initialized with the identity matrix. The entity vector e and relation vector r are first transformed by their corresponding transformation matrix to obtain a new entity vector e' and relation vector r' (namely $e' = e^*M_e$, $r' = r^*M_r$). Give the head and tail entities representation h' and t', the scoring function can be denoted as follows:

$$score = \|h' + r' - t'\| \tag{4}$$

which computes the distance between the sum of the transformed head entity and relation embeddings and the transformed tail entity embedding.

During the training process, TransD aims to minimize the margin-based ranking function L which comprises the correct and incorrect triplet, which can be denoted as follows:

$$L = \sum\left[max(0, \gamma + \text{score}(h, r, t) - \text{score}(h', r, t')\right] + \sum\left[max(0, \gamma + score(h, r, t) - score(h, r, t')\right] \tag{5}$$

where γ denotes the hyperparameter of the margin. By using optimization algorithms such as stochastic gradient descent (SGD), we can obtain the final embedding of the entity and relation vectors, which can be used in relationship reasoning, entity classification, and other tasks.

4. Methodology

In this section, we explicitly elaborate on the implantation of the proposed model, Spatio-Temporal Graph Convolution Network with Embedded location and time features (STEGCN). As shown in Fig. 4, the whole model consists of four parts. The spatial correlation extracting

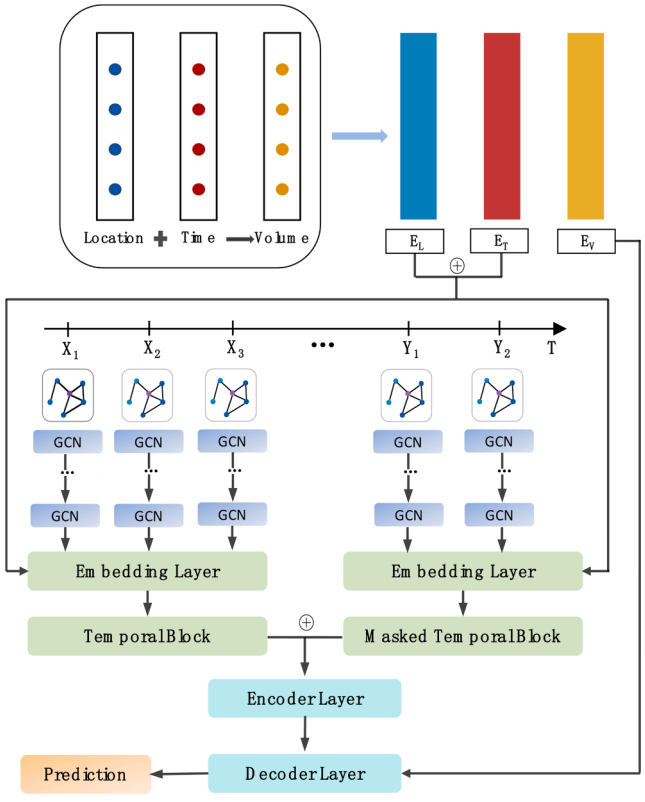


Fig. 4. The overall architecture of STEGCN. Here, the symbol "⊕" denotes the concatenation of two tensors. GCN represents a single layer of the Graph Convolutional Network.

part takes the traffic observations as input, utilizing a graph convolutional layer to extract the spatial correlations between different nodes (each node corresponds to a specific region in the real world) at each time interval. It is worth mentioning that the spatial correlations in prediction time intervals are also extracted during the training process. We deploy the TransD algorithm to train the correlated embeddings of the three key factors to obtain the global relations between location, time, and traffic volume. Then, the spatial and temporal embeddings are merged with the spatial features and then sent to a temporal block (with a masking mechanism during the validation or testing procedure). Subsequently, the two temporal block outputs are sent to an encoder-decoder along with the traffic volume embeddings and generate the predictions. A thorough description of each stage will be presented in the subsections below.

4.1. Spatial correlation extracting

This section shows how to extract the spatial correlations among regions at every time interval. To increase authenticity, we have opted to represent the traffic network as a graph $G =< V, E, A >$ instead of a regular grid map. Spectral-based graph convolution is applied to model the spatial relations, and we leverage two layers of GCN in the practical operation. The adjacency matrix A can be represented as follows:

$$A_{ij} = \begin{cases} 1, & if\, e_{ij} \in E \\ 0, & otherwise \end{cases}, \tag{6}$$

and $\widetilde{A} = A + I_n$ is the adjacency matrix with self-loop, $\widetilde{D}$ denotes the degree matrix of $\widetilde{A}$. The information propagation of the signal on the graph can be defined as:

$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W\right), \tag{7}$$

where $H^{(l)}$ denotes the node features in the l-th layer. For the traffic volumes at each time interval t, the corresponding spatial features are extracted via the two-layer GCN modules. Given the traffic observations $X_{in} \in \mathbb{R}^{P\times N\times d}$, the subsequent spatial representation can be characterized as $X_{s\_in} \in \mathbb{R}^{P\times N\times d_1}$ (d_1 is the dimension of the spatial features). In the training phase, predictions are ascribed with spatial representations labeled as $X_{s\_pre} \in R^{P\times N\times d_1}$ analogously; nevertheless, identical representations are not available during the evaluation phase.

4.2. Embeddings pre-training

As shown in section 3.4, the TransD algorithm aims to capture the semantic relations between entities and relationships in the knowledge graph by embedding vectors of entities and relationships. The incorporation of learned embeddings serves as a powerful source of global external information, resulting in a notable boost in the model's ability to make predictions accurately. Given the training set, we partition traffic volume data into discrete segments based on their numerical values. The time interval is divided into 168 discrete units, which correspond to one-hour segments over a seven-day week and 24-hour day, and locations are represented using a one-hot vector with a dimension of N corresponding to the number of unique locations in the dataset. These three vectors are mapped to a space with the same dimensionality m via a fully connected layer, resulting in a triplet. Then the triplet containing location, time, and traffic volume is sent into the TransD algorithm to extract their correlation. Subsequently, we can get the embedding vectors $E_L, E_T, E_V \in \mathbb{R}^m$, corresponding to location, time, and traffic volume, respectively.

For time interval k, the spatial feature can be represented as $X^k \in \mathbb{R}^{N\times d_1}$. X^k is then merged with its location and time embeddings to generate the embedded feature $X^k_{embeded}$, which is then sent into the temporal block. The process can be formulated as follows:

$$X^k_{embedded} = Concat\left(X^k, E_L, E_T\right),\ X_{embedded} \in \mathbb{R}^{P\times N\times d_2}\left(d_2 = d_1 + \frac{(N+1)m}{NP}\right), \tag{8}$$

which can be represented as $X^k_{embedded\_in}$ and $X^k_{embedded\_pre}$ for the input sequence and predictions, respectively, during the training process.

After incorporating three interrelated pre-training embedding representations, namely location, time, and traffic volume, our model effectively constrains the predicted traffic volume for a specific region and time within a more reasonable interval. This constraint significantly reduces the adverse impact of outlier points on the overall prediction performance of the model.

4.3. Temporal block with mask mechanism

To efficiently capture the temporal patterns in the temporal block, 1-D convolution is exploited to extract features from the sequence by performing convolutional operations on the time series with a faster training speed than RNNs. Given the input $X_{embedded\_in} \in \mathbb{R}^{P\times N\times d_2}$ (from the inputs or the predictions during the training stage), it is firstly reshaped into $X_{t\_in} \in \mathbb{R}^{P\times Nd_2}$. Then a convolution kernel $W_t \in \mathbb{R}^{P\times Nd_2}$ is utilized to extract the features of the input. The output of the 1-D convolution layers X_{t_out} can be denoted as follows:

$$X_{t\_out} = \sigma(W_t X_{t\_in} + b),\ X_{t\_out} \in R^{P\times Nd_2}. \tag{9}$$

A masking mechanism must be introduced during validation and testing as the predicted values are unknown. The outputs of these two temporal blocks are merged, to the later of which a masking matrix is applied. The final product of this part X_{st} can be denoted as follows:

$$X_{st} = Concat\left(X_{t\_out}, X_{t\_out\_pre} \odot M\right), \tag{10}$$

where $X_{t\_out}, X_{t\_out\_pre}$ denote the spatiotemporal properties of the inputs and predictions respectively, $\odot$ is the Hadamard product operation, and M represents the masking matrix with the same shape of $X_{t\_out\_pre}$. To accomplish feature fusion, we opt to concatenate different dimensions together. This allows for better integration of multiple features into a cohesive representation.

4.4. Encoder and decoder

The encoder module, made up of a fully connected layer, takes the spatiotemporal feature X_{st} as input. Suppose X_e is the output of the encoder, which can be formulated as follows:

$$X_e = \sigma_e(W_e X_{st} + b_e), \tag{11}$$

where W_e, b_e are trainable parameters and σ_e is an activation function.X_e is then concatenated with the traffic volume embeddings E_V and sent to the decoder, which is also composed of a fully connected layer. By incorporating traffic volume embeddings, the model can effectively limit its predictions to a realistic range, minimizing the impact of anomalous values on the results. The resultant X_d of the decoder can be formulated as:

$$X_d = \sigma_d(W_d Concat(X_e, E_V) + b_d), \tag{12}$$

where W_d, b_d are trainable parameters, and σ_d denotes an activation function. Followed by a reshaping layer, X_d is finally mapped to the expected predictions.

5. Experiments

In this section, we present an extensive experimentation of our model, STEGCN, on six distinct traffic datasets to meticulously validate its efficiency. Firstly, a fair comparison between STEGCN and various

baselines is conducted to validate its effectiveness. Then, the effect of the vital component of our model is verified via an ablation study, where we compare the performance of STEGCN and its variants with some parts deleted. Subsequently, an extensive ablation study is performed where the efficacy of the critical components of STEGCN is evaluated by comparing it with its variants wherein specific components have been removed. To gain a deeper understanding of the impact of various hyperparameters, we conduct a sensitivity analysis on several critical parameters. Lastly, we provide a detailed analysis of our model's space and time complexity.

5.1. Datasets and metrics

We perform all experiments using six traffic datasets obtained from the Caltrans Performance Measurement System (PeMS), which are categorized based on the types of traffic observations. METR-LA and PEMS-BAY are two traffic speed datasets released by Li *et al.* (Li et al., 2018) and publicly accepted by academia. PEMS03, PEMS04, PEMS07, and PEMS08, released by Song *et al.* (Song et al., 2020), are four traffic flow datasets constructed from four districts in California, respectively. All the datasets mentioned above are road network data of non-Euclidean structures and are aggregated into a window of 5 min. The detailed statistics of the six datasets are shown in Table 2. We evaluate the model performance by three standard regression error metrics consistent with Li *et al.* (Li and Zhu, 2021): Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). The formulas of the metrics above can be computed as follows:

$$MAE = \frac{1}{N_t NQ}\sum_{i=1}^{N_t}\sum_{j=1}^{N}\sum_{k=1}^{Q}\left|y_{ij}^{k} - \hat{y}_{ij}^{k}\right|, \tag{13}$$

$$RMSE = \sqrt{\frac{1}{N_t NQ}\sum_{i=1}^{N_t}\sum_{j=1}^{N}\sum_{k=1}^{Q}\left(y_{ij}^{k} - \hat{y}_{ij}^{k}\right)^2}, \tag{14}$$

$$MAPE = \frac{1}{N_t NQ}\sum_{i=1}^{N_t}\sum_{j=1}^{N}\sum_{k=1}^{Q}\frac{|y_{ij}^{k} - \hat{y}_{ij}^{k}|}{y_{ij}^{k} + \varepsilon}, \tag{15}$$

where N_t, N, Q denote the number of test samples, number of sensors, and the prediction horizon, respectively. y_{ij}^{k} is the ground truth and $\hat{y}_{ij}^{k}$ represents the predicted value. A decrease in the values of all three metrics indicates an improvement in model performance. Notably, the inclusion of a small positive number ε in the denominator of formula (15) not only confers significance to the formula but also bolsters the stability and robustness of the model.

5.2. Baseline methods

We compare our proposed model STEGCN with the following baselines:

5.2.1. Traditional models

- **VAR** (Polson and Sokolov, 2017): Vector Auto-Regression is a traditional statistical model used to analyze multivariate data, which models the relationship between multiple variables over time, allowing for prediction to be made based on past patterns.
- **ARIMA** (Abellana, 2021): Autoregressive Integrated Moving Average is a classic time series prediction model that combines auto-regression and moving averages. We regard the data collected from each sensor as a separate time series and exploit ARIMA to model.
- **SVR** (Barbour et al., 2018): Support Vector Regression is a popular regression algorithm, the core idea of which is to utilize a kernel function to map the input into a high-dimension space and find a maximum margin hyperplane to fit the training data.

5.2.2. Deep learning models

- **FC-LSTM** (Sutskever et al., 2014): Fully Connected Long Short-Term Memory is a typical recurrent neural network with a fully connected layer to make predictions.
- **DCRNN** (Li et al., 2018): Diffusion Convolutional Recurrent Neural Network integrates recurrent neural networks with a diffusion process on the graph to model the spatiotemporal dependencies in an encoder-decoder framework.
- **STGCN** (Yu et al., 2018): Spatio-Temporal Graph Convolutional Network employs a series of ST-Conv blocks, which consist of temporal-gated convolution and spectral-based graph convolution, along with residual connections to capture spatial and temporal patterns.
- **Graph WaveNet** (Wu et al., 2019): Graph WaveNet merges a stacked dilated convolution and graph convolution with an adaptive dependency matrix to capture the dynamic spatiotemporal features.
- **STSGCN** (Song et al., 2020): Spatial-Temporal Synchronous Graph Convolutional Network contains multiple localized spatiotemporal subgraphs for different periods to obtain the spatiotemporal heterogeneities.
- **STFGNN** (Li and Zhu, 2021): Spatial-Temporal Fusion Graph Neural Network fuses various spatial and temporal graphs to capture the spatiotemporal relationships in parallel. The fused graph is integrated with a gated convolution block to handle long series.
- **SCINet** (Liu et al., 2022): Sample Convolution and Interaction Network leverages a downsample-convolve-interact structure to learn temporal patterns at various resolutions. Subsequently, the network extracts features from each subsequence using unique convolutional filters.

5.3. Experimental setups

To be consistent with the baselines mentioned above for a fair comparison environment, we partitioned the METR-LA and PEMS-BAY datasets into training, validation, and test sets, adopting a ratio of 7:1:2. Meanwhile, the remaining four datasets (PEMS03, PEMS04, PEMS07, and PEMS08) were divided with ratio 6:2:2. With a sample rate of 5 min, we set the historical horizon P to 12 and keep the prediction horizon $Q = 12$ (In other words, we used traffic observations from the previous hour to predict the traffic volume for the upcoming hour).

To accelerate the training process, we employed the libcity framework (Wang et al., 2021) based on PyTorch[1] as the primary tool of implementation. All the experiments are executed on a 64-bit Ubuntu server with an Intel(R) Xeon(R) CPU @ 2.20 GHz and eight NVIDIA Titan Xp GPU cards. The embedding dimension of location, time, and traffic volume is searched over {50, 100, 200, 400, 1000}. We utilize the dropout mechanism in fully connected layers with a dropout ratio of 0.3 to alleviate the overfitting of the mode. We set the maximum training epoch to 200 and implemented an early stopping mechanism that halts training if the validation loss does not decrease for 30 consecutive epochs. The Adam optimizer (Kingma and Ba, 2015) is used to optimize the entire model with a learning rate of 0.001, while all other optimizer parameters are set to their default values.

5.4. Main results

We validate our model and compare it to other baselines on two traffic volume prediction problems. The main results of traffic volume (traffic speed, traffic flow) prediction are illustrated in Table 3 and Table 4, respectively. It is worth noting that the value of the metric MAPE in both tables is a percentage. The prediction horizons on the METR-LA and PEMS-BAY datasets vary within the range of {3, 6, 12}

[1] https://pytorch.org/.

Table 2
The detailed statistics of six datasets.

Datasets	TYPE	Sensors	Edges	Input	Output	Time interval	Time Range	Split Ratio	Missing Ratio	Sample Rate
METR-LA	Speed	207	1515	12	3/6/12	34,272	Mar – Jun 2012	7:1:2	8.109 %	5 min
PEMS-BAY	Speed	325	2369	12	3/6/12	52,116	Jan – May 2017	7:1:2	0.003 %	5 min
PEMS03	Flow	358	547	12	12	26,208	Sep – Nov 2018	6:2:2	0.672 %	5 min
PEMS04	Flow	307	340	12	12	16,992	Jan – Feb 2018	6:2:2	3.182 %	5 min
PEMS07	Flow	883	866	12	12	28,224	May – Aug 2017	6:2:2	0.452 %	5 min
PEMS08	Flow	170	295	12	12	17,856	Jul – Aug 2018	6:2:2	0.696 %	5 min

Table 3
Prediction performance with different prediction horizons on the METR-LA and PEMA-BAY datasets.

Dataset	METR-LA			PEMS-BAY		
Metric Horizon	MAE (3/6/12)	RMSE (3/6/12)	MAPE (3/6/12)	MAE (3/6/12)	RMSE (3/6/12)	MAPE (3/6/12)
VAR	4.42/5.41/6.52	7.8//9.13/10.11	13.00/12.70/15.80	1.74/2.32/2.93	3.16/4.25/5.44	3.60/5.00/6.50
ARIMA	3.77/3.99/6.90	7.66/8.21/13.23	11.20/9.60/17.40	1.45/1.62/3.38	3.19/3.30/4.76	3.14/3.50/8.30
SVR	3.99/5.05/6.72	8.45/10.87/13.67	9.30/12.10/16.70	1.85/2.48/3.28	3.59/5.18/7.08	3.80/5.50/8.00
FC-LSTM	3.44/3.77/4.37	6.30/7.23/8.69	9.60/10.90/13.20	2.05/2.20/2.37	4.19/4.55/4.96	4.80/5.20/5.70
DCRNN	2.77/3.15/3.60	5.38/6.45/7.59	7.30/ 8.80/10.50	1.38/1.74/2.07	2.95/3.97/4.74	2.90/3.90/4.90
STGCN	2.88/3.47/4.59	5.74/7.24/9.40	7.62/9.57/12.70	1.36/1.81/2.49	2.96/4.27/5.69	2.90/4.17/5.79
Graph WaveNet	2.69/3.07/3.53	5.15/6.22/7.37	6.90/ 8.37/10.01	1.30/1.63/1.95	2.74/3.70/4.52	2.73/3.67/4.63
STSGCN	3.31/4.13/5.06	7.62/9.77/11.66	8.06/10.29/12.91	1.44/1.83/2.26	3.01/4.18/5.21	3.04/4.17/5.40
STFGNN	2.68/3.07/3.48	5.21/6.11/7.24	6.98/8.05/9.53	1.29/1.54/1.98	2.69/3.48/4.67	2.62/3.72/4.83
STEGCN	**2.63/2.91/3.03**	**5.13/5.87/6.94**	**6.81/7.33/9.16**	**1.17/1.51/1.78**	**2.46/3.07/4.26**	**2.48/3.49/4.27**

Table 4
Prediction performance of different approaches on the PEMS03, PEMS04, PEMS07, and PEMS08 datasets.

Dataset	PEMS03			PEMS04			PEMS07			PEMS08		
Metric	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
VAR	23.65	38.26	24.51	24.54	38.61	17.24	50.22	75.63	32.22	19.19	29.81	13.10
ARIMA	35.41	47.59	33.78	33.73	48.80	24.18	38.17	59.27	19.46	31.09	44.32	22.73
SVR	21.97	35.29	21.51	28.70	44.56	19.20	32.49	50.22	14.26	23.25	36.16	14.64
FC-LSTM	21.33	35.11	23.33	27.140	41.59	18.20	29.98	45.94	13.20	22.20	34.06	14.20
DCRNN	18.18	30.31	18.91	24.70	38.12	17.12	25.30	38.58	11.66	17.86	27.83	11.45
STGCN	17.49	30.12	17.15	22.70	35.55	14.56	25.38	38.78	11.08	18.02	27.83	11.40
Graph WaveNet	19.85	32.94	19.31	25.45	39.70	17.29	26.85	42.78	12.12	19.13	31.05	12.68
STSGCN	17.48	29.21	16.78	21.19	33.65	13.90	24.26	39.03	10.21	17.13	26.80	10.96
STFGNN	16.77	28.34	16.30	19.83	31.88	13.02	22.07	35.80	9.21	16.64	26.22	10.60
SCINet	**14.98**	**24.08**	**14.11**	18.95	30.89	11.86	21.19	34.03	8.83	15.72	24.76	9.80
STEGCN	15.53	26.07	15.19	**18.47**	**30.25**	**11.16**	**20.88**	**33.21**	**8.42**	**14.98**	**23.71**	**9.44**

(predicting the future traffic volumes in the next 15, 30, and 60 min, respectively). For convenience and comparison with the previous baseline, the prediction horizon on the other four datasets is set to 12 (predicting the future traffic volumes in the next 60 min). Our proposed STEGCN method demonstrates better performance than other baselines in most cases.

Overall, the superior predictive performance of deep learning methods compared to traditional methods (VAR, ARIMA, SVR) in handling large-scale and structurally complex data is well-documented across almost all cases. This consequence may contribute to the strong assumptions of data in the traditional method. For instance, ARIMA requires the time series to be stable and is sensitive to outliers and anomalies. VAR is based on the hypothesis that the variates are linearly correlated, while SVR is sensitive to noise and requires a handcraft kernel function. The relatively poor performance of FC-LSTM indicates that temporal and spatial data characteristics must be considered simultaneously in traffic prediction.

The satisfactory performance of Graph WaveNet and STEGCN validates that the computationally expensive RNN structures are not necessary for spatiotemporal sequence prediction. STFGNN and STSGCN exploit multiple graphs from different perspectives to enhance the ability of spatiotemporal feature extracting, which ignores the global external information and the impact of predicted anomalies far away from the reasonable range. SCINet leverages multiple convolutional filters to capture temporal features from different subsequences while it neglects the dynamic spatial dependencies among regions. The performance of our model exceeds that of the approaches mentioned above. It is noteworthy that, under a careful analysis of the results in Table 3 and Table 4, our model seems more capable of dealing with long-term prediction, owing to the deployments of prediction features extracted in the training part.

5.5. Ablation study

A series of intricate ablation experiments have been conducted on the METR-LA and PEMS03 datasets to demonstrate the effectiveness of individual modules within the proposed model. For the sake of convenience, we name several variants of STEGCN as follows:

- **w/o 1-D Conv**: This variant removes the 1-D convolution layer in the temporal module, which cannot capture the temporal patterns in data.
- **w/o TransD**: Without the TransD algorithm module, the correlated embeddings of location, time, and traffic volume are removed from STEGCN, thus lacking the global external information and unable to constrain the prediction within the reasonable range.

- **w/o TVE**: This variant removes the traffic volume embedding in the encoder layer, which loses the constraints on the predictions.
- **w/ LSTM**: We replace the 1-D convolution block with an LSTM layer in the temporal module.

We set the historical and prediction horizon to 12, keeping other parameters the same as in the main result part. Fig. 5 shows the comparison results of the four variants with STEGCN. From the observations that the variant without temporal block (w/o 1-D Conv) gets the worst performance among all variants, it can be concluded that temporal feature extraction plays an indispensable role in traffic prediction. Meanwhile, we can observe that deploying the embeddings from the TransD algorithm improves the prediction efficiency of the model vastly. As can be derived from Fig. 5, the prediction constraints from the traffic volume embeddings are proven effective, owing to the captured inherent relationships among location, time, and traffic volume. Moreover, when the temporal block is replaced by a single LSTM layer, the performance of the model decreases slightly, indicating that the 1-D convolutional module has better capability to capture the temporal dependencies than normal LSTM structures.

5.6. Parameter sensitivity analysis

We conduct a sensitivity analysis for model hyperparameters on the commonly used METR-LA dataset. Due to the significant number of hyperparameters involved in the training process, we focus our gaze on the impact of the hyperparameters m and k, which denote the embedding dimension and the number of graph convolution layers, respectively. We set the prediction horizon to 12 while keeping the other parameters at their default values, and Table 5 displays the ranges of changes for both variables m and k. According to Fig. 6, the STEGCN model achieves the best performance when $m = 200$ and $k = 2$. As shown in Fig. 6, as the dimensionality m of the mapping space increases, the model can have more parameters to learn enhanced embedding representations for time, location, and traffic volume, thereby improving the model performance. As m increases continuously, both the training time and memory consumption of the model also increase, yet the performance does not improve, indicating a potential consequence of overfitting. The augmentation of graph convolutional layers in GCN results in a deeper feature extraction process that captures more intricate and advanced characteristics within the graph, ultimately leading to an improvement in model performance. However, the performance decreases with the continuous increase of m. The observed phenomenon may be attributed to the persistent convolutions of neighborhood information in multilayer GCN, which may induce an over-averaging effect on such information and consequently homogenize the features of proximate nodes, ultimately resulting in reduced efficacy of feature extraction.

Table 5
Parameter sensitivity analysis settings on METR-LA.

Hyperparameter	Range
m	{50, 100, 200, 400, 1000}
k	{1, 2, 3, 4}

Furthermore, to analyze the impact of the input sequence length (P) on the predictive performance of the model, we conducted corresponding experiments on the METR-LA dataset while keeping the output sequence length Q equal to 6. We set the value range of P to {3, 6, 9, 12, 30, 60} and performed corresponding experiments on three models: FC-LSTM, STGCN, and STEGCN. The experimental results are presented in Fig. 7. Overall, it can be observed that as P increases, the prediction performance initially improves and then declines. The initial increase in prediction accuracy may be attributed to the incorporation of more historical information, which enables the model to capture timing features more comprehensively. However, the decline in performance during the latter half of the model may be caused by overfitting resulting from an excessive number of parameters or the introduction of excessive noise.

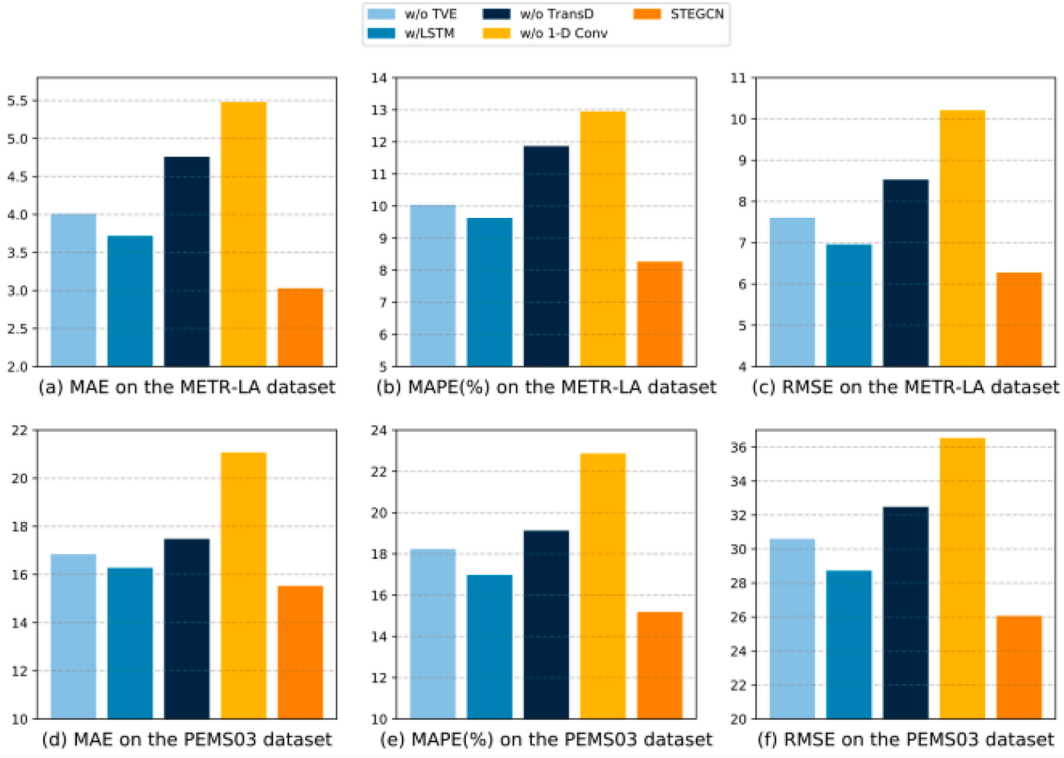


Fig. 5. Ablation study on METE-LA and PEMS03 datasets.

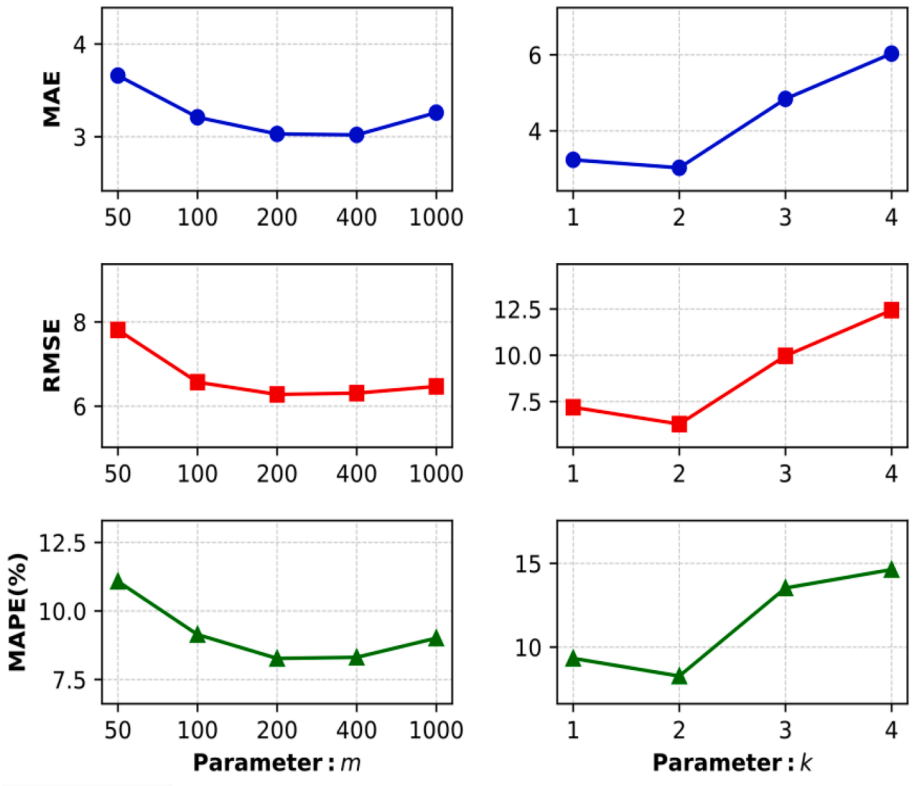


Fig. 6. Parameter sensitivity analysis of STEGCN on METR-LA.

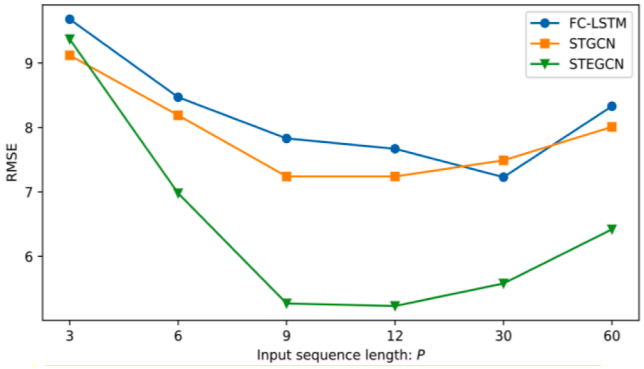


Fig. 7. Analysis of the impact of the input sequence length *P*.

5.7. Run time and total parameters

Apart from the predictive ability, runtime and model size are fundamental factors that contribute to evaluating the overall quality of the structured model. In this section, we exhibit the runtime and parameters of some representative baselines and STEGCN on the PEMS-BAY dataset. As we can conclude from Table 6, FC-LSTM is the fastest model with the least parameters because of its simple structure with the absence of spatial correlations, which can only provide a worse prediction, as shown in Table 3 and Table 4. STEGCN is faster than DCRNN and STSGCN. Graph WaveNet and STEGCN have relatively few parameters and shorter runtime due to the replacement of the RNN structure. However, when the runtime and model size are similar, the predictive accuracy of Graph WaveNet is inferior to that of STEGCN. All in all, STEGCN runs faster than common deep learning approaches because of the 1-D convolution structure instead of RNN structures, and its forecast performance is also superior.

6. Conclusion and future work

This manuscript introduces an innovative spatiotemporal graph convolution network named STEGCN that embeds both location and time features to predict traffic patterns. The proposed model incorporates the TransD algorithm to derive correlated embeddings of location, time, and traffic volume. These embeddings are then utilized to introduce the global information into the model and ensure that the predicted value remains within a reasonable range. Furthermore, the graph convolution layers are utilized to extract the spatial patterns, while two 1-D convolutional temporal blocks, employing a masking mechanism, are applied to address the temporal properties of the data. The experimental results have demonstrated the effectiveness of the model. To facilitate further research and exploration, the source code and data associated with our model have been made publicly available on a widely accessible repository.[2]

In future work, we aspire to develop a comprehensive end-to-end framework by integrating the training process of the TransD algorithm with the primary spatiotemporal prediction methodology. This integrated approach is expected to generate more suitable embeddings for the prediction tasks. Moreover, we plan to extend the scope of our proposed framework to other relevant spatiotemporal prediction scenarios, such as meteorological outlook, grain yield, and other related fields.

CRediT authorship contribution statement

Wei Li: Methodology, Software, Writing – original draft. **Xin Liu:** Writing – review & editing. **Wei Tao:** Data curation, Formal analysis. **Lei Zhang:** Validation. **Junhua Zou:** Conceptualization. **Yu Pan:** Visualization, Investigation. **Zhisong Pan:** Supervision, Project administration.

[2] https://github.com/liwei/stegcn.

Table 6
Runtime and parameters on PEMS-BAY.

Efficiency Assessment	FC-LSTM	DCRNN	STGCN	Graph WaveNet	STSGCN	STEGCN
Training (s/epoch)	22	682	57	57	326	61
Inference (s)	1	46	19	8	17	11
Parameters	66,949	372,353	395,489	278,960	1,180,876	293,275

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.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No. 62076251 and 62106281). The views and opinions expressed in this paper are those of the authors alone and do not necessarily represent the official positions or perspectives of the funding agencies.

References

Abdoos, M., & Bazzan, A. L. (2021). Hierarchical traffic signal optimization using reinforcement learning and traffic prediction with long-short term memory. *Expert Systems with Applications, 171*, Article 114580.

Abellana, D. P. M. (2021). Short-term traffic flow forecasting using the autoregressive integrated moving average model in Metro Cebu (Philippines). *International Journal of Applied Decision Sciences, 14*(5), 565–587.

Akhtar, M., & Moridpour, S. (2021). A review of traffic congestion prediction using artificial intelligence. *Journal of Advanced Transportation*, 1–18.

Barbour, W., Mori, J. C. M., Kuppa, S., & Work, D. B. (2018). Prediction of arrival times of freight traffic on US railroads using support vector regression. *Transportation Research Part C: Emerging Technologies, 93*, 211–227.

Bogaerts, T., Masegosa, A. D., Angarita-Zapata, J. S., Onieva, E., & Hellinckx, P. (2020). A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. *Transportation Research Part C: Emerging Technologies, 112*, 62–77.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems, 26*.

Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*. https://arxiv.org/abs/1312.6203.

Cao, M., Li, V. O., & Chan, V. W. (2020). A CNN-LSTM model for traffic speed prediction. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020)*, pp. 1-5.

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*, *29*.

Fadlullah, Z. M., Tang, F., Mao, B., Kato, N., Akashi, O., Inoue, T., & Mizutani, K. (2017). State-of-the-art deep learning: evolving machine intelligence toward tomorrow's intelligent network traffic control systems. *IEEE Communications Surveys & Tutorials, 19*(4), 2432–2455.

Galanis, A., Botzoris, G., & Eliou, N. (2017). Pedestrian road safety in relation to urban road type and traffic flow. *Transportation Research Procedia, 24*, 220–227.

Gong, Y., Li, Z., Zhang, J., Liu, W., & Zheng, Y. (2020). Online spatio-temporal crowd flow distribution prediction for complex metro system. *IEEE Transactions on Knowledge and Data Engineering, 34*(2), 865–880.

He, Z., Chow, C. Y., & Zhang, J. D. (2020). STNN: A spatio-temporal neural network for traffic predictions. *IEEE Transactions on Intelligent Transportation Systems, 22*(12), 7642–7651.

Hechtlinger, Y., Chakravarti, P., & Qin, J. (2017). A generalization of convolutional neural networks to graph-structured data. arXiv preprint arXiv:1704.08165. https://arxiv.org/abs/1704.08165.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. (2015, July). Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Vol. 1, pp. 687-696.

Jiang, W., & Luo, J. (2022). Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications, 207*, Article 117921.

Jin, W., Lin, Y., Wu, Z., & Wan, H. (2018). Spatio-temporal recurrent convolutional networks for citywide short-term crowd flows prediction. In *In Proceedings of the 2nd International Conference on Compute and Data Analysis* (pp. 28–35).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*. https://arxiv.org/abs/1412.6980v5.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint* arXiv:1609.02907. https://arxiv.org/abs/1609.02907.

Kiranyaz, S., Ince, T., & Gabbouj, M. (2015). Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Transactions on Biomedical Engineering, 63*(3), 664–675.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.

Li, M., & Zhu, Z. (2021). Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, Vol. 35, No. 5, pp. 4189-4196.

Li, Y., Rose, Y.u., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *In Proceedings of the Sixth International Conference on Learning Representations*.

Li, W., Tao, W., Qiu, J., Liu, X., Zhou, X., & Pan, Z. (2019). Densely connected convolutional networks with attention LSTM for crowd flows prediction. *IEEE Access, 7*, 140488–140498.

Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015, February). Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the AAAI conference on artificial intelligence, Vol. 29, No. 1.

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 Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, Vol. 33, No. 01, pp. 1020-1027.

Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L., & Xu, Q. (2022). SCINet: time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems, 35*, 5816–5828.

Ou, J., Sun, J., Zhu, Y., Jin, H., Liu, Y., Zhang, F., … Wang, X. (2022). STP-TrellisNets+: spatial-temporal parallel trellisnets for multi-step metro station passenger flow prediction. *IEEE Transactions on Knowledge and Data Engineering.*

Polson, N. G., & Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies, 79*, 1–17.

Qiu, J., Du, L., Zhang, D., Su, S., & Tian, Z. (2019). Nei-TTE: intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city. *IEEE Transactions on Industrial Informatics, 16*(4), 2659–2666.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533–536.

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 Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, Vol. 34, No. 01, pp. 914-921.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems, 2*, 3104–3112.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *Stata Journal, 1050*(20), 10–48550.

Wang, F. Y. (2010). Parallel control and management for intelligent transportation systems: concepts, architectures, and applications. *IEEE Transactions on Intelligent Transportation Systems, 11*(3), 630–638.

Wang, Z., Ding, D., & Liang, X. (2023). TYRE: A dynamic graph model for traffic prediction. *Expert Systems with Applications, 215*, Article 119311.

Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014, June). Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI conference on artificial intelligence, Vol. 28, No. 1.

Wang, J., Jiang, J., Jiang, W., Li, C., & Zhao, W. X. (2021). Libcity: An open library for traffic prediction. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, pp. 145-148.

Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 1907-1913.

Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., & Zhang, C. (2020). Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 753-763.

Wu, W., Jiang, S., Liu, R., Jin, W., & Ma, C. (2020). Economic development, demographic characteristics, road network and traffic accidents in Zhongshan, China: gradient boosting decision tree model. *Transportmetrica A: Transport science, 16*(3), 359–387.

Xie, P., Li, T., Liu, J., Du, S., Yang, X., & Zhang, J. (2020). Urban flow prediction from spatiotemporal data using machine learning: a survey. *Information Fusion, 59*, 1–12.

Xu, J., Rahmatizadeh, R., Bölöni, L., & Turgut, D. (2017). Real-time prediction of taxi demand using recurrent neural networks. *IEEE Transactions on Intelligent Transportation Systems, 19*(8), 2572–2581.

Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., ... & Li, Z. (2018). Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, Vol. 32, No. 1.

Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *In Proceedings of the 27th International Joint Conference on Artificial Intelligence* (pp. 3634–3640).

Zhang, J., Zheng, Y., & Qi, D. (2017). Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, Vol. 31, No. 1.

Zhang, J., Zheng, Y., Qi, D., Li, R., & Yi, X. (2016). DNN-based prediction model for spatio-temporal data. In *In Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 1–4).

Zheng, C., Fan, X., Wang, C., & Qi, J. (2020). Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, Vol. 34, No. 01, pp. 1234-1241.

Zhu, Z., Peng, B., Xiong, C., & Zhang, L. (2016). Short-term traffic flow prediction with linear conditional gaussian Bayesian network. *Journal of Advanced Transportation, 50*(6), 1111–1123.