Multi-Task Synchronous Graph Neural Networks for Traffic Spatial-Temporal Prediction

He Li  
heLI@xidian.edu.cn  
Xidian University  
Xi’an, Shaanxi, China

Duo Jin  
djin@stu.xidian.edu.cn  
Xidian University  
Xi’an, Shaanxi, China

Xuejiao Li  
xji_521@stu.xidian.edu.cn  
Xidian University  
Xi’an, Shaanxi, China

Jianbin Huang*  
jbian@xidian.edu.cn  
Xidian University  
Xi’an, Shaanxi, China

Jaesoo Yoo  
yjs@cbnu.ac.kr  
Chungbuk National University  
Cheongju, Chungbuk, South Korea

ABSTRACT
Traffic spatial-temporal prediction is of great significance to traffic management and urban construction. In this paper, we propose a multi-task graph synchronous neural network (MTSGNN) to synchronously predict the spatial-temporal data at the regions and transitions between regions. The method of constructing “multi-task graph representation” is proposed to retain the information of regions and transitions that existing works can not reflect. Then our model synchronously captures multiple types of dynamic spatial correlations, models dynamic temporal dependencies and re-weights different time steps to solve the problem of long-term time modeling. In three real data sets, we verify the validity of the proposed model.

CCS CONCEPTS
• Theory of computation → Dynamic graph algorithms; • Computing methodologies → Artificial intelligence.

KEYWORDS
Spatial-Temporal Prediction, Graph Neural Networks, Spatial-Temporal Correlations

1 INTRODUCTION
Spatial-temporal prediction is a fundamental research problem in smart city field. It has a wide range of applications such as weather prediction, taxi demand prediction and so on. In this paper, we study the traffic spatial-temporal prediction problem, which is based on historical observations to predict future traffic data. The accurate prediction of traffic data is of great importance for alleviating the issue of congestion and helping drivers plan the route.

The goal of this task is to simultaneously predict the traffic data of regions and transitions between regions. However, this task is very challenging due to the following reasons: (1) The mutual influences of regions and transitions: In a urban network network, the traffic flow of regions is composed of the traffic flow of transitions. Therefore, the changes in the traffic flow of transitions will cause the changes in the traffic flow of regions. (2) Dynamic spatial-temporal correlations: The traffic condition between different regions will affect each other, and the traffic condition of different regions change dynamically over time, so that the correlations between regions are also dynamically changeable. Besides, the different historical traffic data have an unequal impact on the future traffic data. Traffic data generally shows periodicity and trend, but it is affected by sudden factors such as traffic accident and bad weather. So how to extract the dynamic and non-linear spatial-temporal correlations is a challenging task.

Although the above methods have achieved obvious improvements over traditional approaches, we argue that they are not sufficient to achieve satisfying performance on spatial-temporal prediction. On the one hand, these studies mainly focus on predicting the traffic data in each area[3]. However, in many cases, people need to know the traffic data of transitions between regions. Many existing methods use two independent components to perform two prediction tasks, ignoring the mutual influence between them. On the other hand, these studies do not consider the multiple types of dynamic spatial correlations between regions [7]. However, we observe that both spatial proximity and functional similarity between
We can model different types of spatial dependencies according to the nature of traffic spatial-temporal prediction is to learn a function $f$ to predict the next $Q$ time steps traffic data based on the given history $P$ times step traffic data. Therefore, our final goal is formulated as follows:

$$f(X_{(t-P+1):(t)}, Z_{(t-P+1):(t)}, G) \rightarrow \hat{X}_{(t+1):(t+Q)}, \hat{Z}_{(t+1):(t+Q)} \quad (1)$$

where $X_{(t-P+1):(t)} \in \mathbb{R}^{P \times N \times C}$ and $Z_{(t-P+1):(t)} \in \mathbb{R}^{P \times N \times N \times C}$ are historical traffic data of regions and transitions, respectively. $\hat{X}_{(t+1):(t+Q)} \in \mathbb{R}^{Q \times N \times C}$, $\hat{Z}_{(t+1):(t+Q)} \in \mathbb{R}^{Q \times N \times N \times C}$ are the prediction target.

## 3 METHODOLOGY

MTSGNN consists of three core parts: (1) The construction of multi-task graph representation. (2) Stacked multi-task synchronous graph neural layers (MTSGNLs). (3) A predictor. Each multi-task synchronous graph neural layer is constructed by stacked MTSGNMs, a GAGCN and a TAM.

### 3.1 Overview

In this part, we briefly illustrate the overall framework of MTSGNN. The overview of our model is shown in Figure 1. First of all, we connect the traffic data representation of regions and transitions at each time step to construct the multi-task graph representation. Secondly, we use stacked MTSGNMs with a GAGCN to simultaneously model multiple types of dynamic spatial dependencies. Then, we design a temporal attention mechanism module to capture the dynamic temporal correlations. At last, we feed the extracted spatial-temporal features into the predictor to obtain the target traffic.

### 3.2 Multi-task Graph Representation Construction

The purpose of constructing the multi-task graph representation is to model region-level and transition-level spatial-temporal dependencies simultaneously. The core idea to obtain this representation is to connect the traffic data representation of regions and transitions at each time step (Figure 2). Before the construction operation, we reshape the traffic data representation of transitions at each time step $Z_t$ as $Z'_t \in \mathbb{R}^{N \times N \times C}$ so that it can be connected with the traffic data representation of regions directly. Then, we can simultaneously capture region-level and transition-level spatial-temporal features according to the multi-task graph representation.

Because the number of channels of the multi-task graph representation is up to $(N + 1) \times C$, it is difficult for the model to tackle such high-dimensional data. Thus, we exploit the $1 \times 1$ convolutional operation to reduce the dimension to $d$-dimension.

### 3.3 Multi-task Synchronous Graph Neural Module

In this part, we consider the spatial correlations between regions from two perspectives, including (1) spatial proximity, which treats the city as a neighborhood graph $G_N = (V, E, A_N)$ (2) functional similarity, which treats the city as a functional similarity graph $G_F = (V, E, A_F)$. We need to emphasize that our method can extend to other types of spatial correlations by applying different adjacency matrices.

**Spatial proximity** The spatial neighbors of a region refer to its geographic near regions. We treat the urban network as a grid graph and a region is neighbor to its eight adjacent areas.

$$A_{N,ij} = \begin{cases} 1 & V_i \text{ and } V_j \text{ are connected} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$
where $\epsilon$ is a threshold. The functional similarity between region $V_i$ and $V_j$ can be obtained by calculating the cosine similarity of historical traffic data:

$$Similarity(V_i, V_j) = \cosine_similarity(D_{0-\tau}(V_i), D_{0-\tau}(V_j))$$

where $D_{0-\tau}(V_i)$ denotes the historical traffic data of region $V_i$ from time step 0 to $\tau$. $|\tau|$ represents the number of all available time intervals in the training set, $\cosine_similarity$ denotes the cosine similarity calculation operation.

Graph convolutional layer. There are multiple types of spatial features between regions, different types of spatial features have different influences on traffic data of different regions. Inspired by this, we design a novel graph convolution operation to extract multiple types of spatial features. The graph convolution operation is defined as follows:

$$h^{(l+1)} = W_N \odot \sigma(A_N h^{(l)}W_1 + b_N) + W_F \odot \sigma(A_F h^{(l)}W_2 + b_F)$$

where $h^{(l)}$ denotes the output of the l-th graph convolution layer, $A_N, A_F \in \mathbb{R}^{N \times N}$ denote the adjacency matrices of spatial proximity and functional similarity, respectively. $\sigma$ is an activation function, $W_N, W_F \in \mathbb{R}^{N \times C_{out}}, W_1, W_2 \in \mathbb{R}^{C_{in} \times C_{out}}, b_N, b_F \in \mathbb{R}^{C_{out}}$ are learnable parameters. $C_{in}, C_{out}$ are the number of features of the input and output, respectively. $\odot$ is the Hadamard product. The graph convolutional operation can be used not only to the spatial correlations mentioned above but also to other types of spatial correlations according to different adjacency matrices.

Aggregating operation. After stacking $L$ graph convolution blocks, we use the sum fusion operation to aggregate the features of different hidden layers. There are $P$ MTSGNMs in a MTSGNL, each MTSGNM can extract spatial features of different time periods in parallel. The output of each MTSGNM should be concatenated to be added with the output of the GAGCN.

3.4 Gated Self-adaptive Graph Convolutional Network

Although MTSGNM can extract different types of spatial features, it can not capture dynamic spatial correlations between regions. Thus, we propose a novel gated self-adaptive graph convolutional network to model dynamic spatial dependencies. We use a learnable matrix $A^r \in \mathbb{R}^{N \times N}$ as the adjacency matrix. Gating mechanisms in RNN-based methods are applied in the convolutional network. Given the input $X \in \mathbb{R}^{P \times N \times d}$ the process is defined as following:

$$S = (A^r X W_1 + b_1) \otimes \text{sigmoid}(A^r X W_2 + b_2)$$

where $S \in \mathbb{R}^{P \times N \times d}$ denotes the output, $W_1, W_2 \in \mathbb{R}^{d \times d}, b_1, b_2 \in \mathbb{R}^d$ are learnable parameters, sigmoid is the sigmoid activation function, $\otimes$ is the element-wise product.

3.5 Temporal Attention Module

The future traffic conditions on a road are potentially correlated with the historical data, the different historical traffic condition have an unequal impact on the future traffic data. Attention mechanism has been seen in many tasks to selectively focus on important parts of the input information. Inspired by this, we design the $K$-head temporal attention mechanism module to re-weights the extracted spatial features from different time steps, which addresses the issue of the long-term temporal modeling.

3.6 Predictor

We design a predictor to map the output of the last MTSGCL to the target prediction. In detail, we first use 2-layer convolutional operation to ensure that the output has the same size and dimension as the prediction target. Then we use a cropping operation to cut the output of the last convolutional operation layer into the prediction results of regions and transitions. The specific formula is as follows:

$$Y = \text{Conv}(\text{Conv}(X^S_{(t-\tau):t} + \hat{X}(t+Q))) \in \mathbb{R}^{Q \times N \times (N+1) \times C}$$

$$\hat{X}(t+Q) = Y(\cdot ; 0 : C = 1) \in \mathbb{R}^{Q \times N \times C}$$

$$\hat{Z}(t+Q) = Y(\cdot ; C :) \in \mathbb{R}^{Q \times N \times C}$$

Where $Y$ represents the concatenation of the prediction results of regions and transitions. We reshape $\hat{Z}^{(t+Q)}$ as $\hat{Z}(t+Q) \in \mathbb{R}^{Q \times N \times N \times C}$ to get the final prediction results of transitions.

3.7 Loss

In order to update the parameters of our model, we design the loss function as follows:

$$L(\Theta) = \sum_{t} \sum_{i} \sum_{k} |\hat{X}(t, i, k) - X(t, i, k)|$$

$$+ \sum_{t} \sum_{i} \sum_{j} \sum_{k} |\hat{Z}(t, i, j, k) - Z(t, i, j, k)|$$

where $\Theta$ represents the all learnable parameter of MTSGNN.
The hidden layers in each MTSGNMs is 32-dimension. The learning rate is set to 0.001. Besides, in order to better alleviate the overfitting problem, we set dropout to 0.3.

2. The first two datasets used in this experiment are collected from the Detectors data from July 1st to August 31th, 2016. It is collected from Caltrans Performance Measurement System.

Experimental setup. Our model is implemented using Pytorch, run in a Linux environment, and trained with Adam optimizer with a batch size of 8. We employ $P=12$ historical time steps to predict the traffic data of the next $Q=12$ steps. We use 70% of the data set for the training set, 10% as the verification set, and the remaining 20% as the test set. Our model contains 2 MTSGNLs, each MTSGNL contains 12 MTSGNMs, a temporal attention module and a GAGCN. The hidden layers in each MTSGNMs is 32-dimension. The learning rate is set to 0.001. Besides, in order to better alleviate the overfitting problem, we set dropout to 0.3.

Baselines. We compared our framework with existing classical and deep learning models including HA, ARIMA [4], LSTM, GRU [2], and STResNet [6].

### 4 EXPERIMENTS

In this section, to justify the effectiveness of MTSGNN, we conduct extensive experiments on three datasets about traffic data to evaluate the predictive performance of the proposed model.

Datasets. We use three different datasets as shown in Table 1. The first two datasets used in this experiment are collected and published by Didi Chuxing. Both of them are GPS trajectory datasets. The last one is the detector data from July 1st to August 31th, 2016. It is collected from Caltrans Performance Measurement System.

Experimental setup. Our model is implemented using Pytorch, run in a Linux environment, and trained with Adam optimizer with a batch size of 8. We employ $P=12$ historical time steps to predict the traffic data of the next $Q=12$ steps. We use 70% of the data set for the training set, 10% as the verification set, and the remaining 20% as the test set. Our model contains 2 MTSGNLs, each MTSGNL contains 12 MTSGNMs, a temporal attention module and a GAGCN. The hidden layers in each MTSGNMs is 32-dimension. The learning rate is set to 0.001. Besides, in order to better alleviate the overfitting problem, we set dropout to 0.3.

Baselines. We compared our framework with existing classical and deep learning models including HA, ARIMA [4], LSTM, GRU [2], and STResNet [6].

### Experimental Results. As shown in Table 2 and Table 3, our MTSGNN surpasses the compared models on every metric across the three datasets. This result proves the advantage of MTSGNN in capturing the dynamic spatial-temporal correlations and demonstrates its reliable predictive ability for region-level and transition-level prediction tasks. On the Haikou dataset, GRU outperforms STResNet in terms of MAE and RMSE, which shows that not all methods that utilize spatial-temporal information are better than time series methods.

### 5 CONCLUSIONS

In this paper, we propose a novel model named MTSGNN to predict region-level and transition-level traffic data simultaneously by constructing the multi-task graph representation. By utilizing the multi-task synchronous graph neural modules with a gated self-adaptive GCN, our model can synchronously captures multiple types of dynamic spatial correlations of regions and transitions. Then we use the temporal attention module to model dynamic temporal dependencies. Experimental results conducted on three public datasets prove the superiority of MTSGNN.

### ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (61602354, 61876138), Natural Science Foundation of Shaanxi Province (2019JM-227), National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT). (No. 2019R1A2C2084257), and the MSIT(Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program(IITP-2021-2020-0-01462) supervised by the IITP(Institute for Information communications Technology Planning Evaluation).

### REFERENCES


