Spatio-Temporal Dynamic Graph Relation Learning for Urban Metro Flow Prediction

Peng Xie, Minbo Ma, Tianrui Li, Senior Member, IEEE, Shenggong Ji, Shengdong Du, Zeng Yu and Junbo Zhang, Member, IEEE

Abstract—Urban metro flow prediction is of great value for metro operation scheduling, passenger flow management and personal travel planning. However, the problem is challenging. First, different metro stations, e.g. transfer stations and non-transfer stations have unique traffic patterns. Second, it is difficult to model complex spatio-temporal dynamic relation of metro stations. To address these challenges, we develop a spatio-temporal dynamic graph relational learning model (STDGRL) to predict urban metro station flow. First, we propose a spatio-temporal node embedding representation module to capture the traffic patterns of different stations. Second, we employ a dynamic graph relationship learning module to learn dynamic spatial relationships between metro stations without a predefined graph adjacency matrix. Finally, we provide a transformer-based long-term relationship prediction module for long-term metro flow prediction. Extensive experiments are conducted based on metro data in four cities, China, with experimental results demonstrating the advantages of our method compared over 14 baselines for urban metro flow prediction.

Index Terms—Spatio-temporal Data, Urban Flow Prediction, Graph Neural Networks

1 INTRODUCTION

As an important part of urban public transportation, urban metro occupies a large proportion of urban traffic. Especially for large cities, accurate prediction of urban metro passenger flow is critical to metro operation scheduling [1], passenger flow management [2], and personal travel planning [3]. Urban metro networks are dynamic graphs which have rich spatial and temporal characteristics. Figure 1(a) shows the change of passenger outflow for three different metro stations in Chongqing over the time frame of one day. We can observe that the passenger outflow of station 1 has a small peak between 7:00 and 9:00 in the morning, and there is also a small evening peak period between 17:00 and 19:00. While station 2 also has a relatively small peak in the morning, there is no obvious evening peak, and the overall one-day passenger outflow is smaller than that of station 1. Station 3 has a large peak in passenger outflow in the morning, and then the passenger outflow after 9:00 decreases significantly. Still, the overall passenger flow of station 3 is much larger than those of stations 1 and 2. We can see that these stations have their own different station traffic patterns, not just a simple, fixed spatial connection relationship between stations. Different metro stations are connected and affected each other. This spatial dependency relationship changes dynamically along with time and location as shown in Figure 1(b).

In order to achieve good predictions of passenger flow in metro stations, some research works have been tried and studied [4], [5], [6], [7]. Most of these methods model the flow change trend of metro stations according to inflow and outflow passenger data, metro network topology map, weather, and other external factors. They often use CNN and GNN-based methods to capture spatial dependencies in metro flow data [8], apply RNN-based and Attention-based methods to model the temporal dependencies of metro traffic data [9], and some also take external factors into account [6]. Although these studies have made positive progress, most of them only use a single metro traffic data set or need to predefine the adjacency graph between stations. Others treat different stations in the metro network as the same kind of node. Overall, the generalization performance of these models is insufficient.

In summary, the urban metro flow prediction task faces three major challenges:

1) Modeling unique traffic patterns at different stations: Previous research [8], [10], [11] treated metro stations as equal nodes or divided metro stations into transfer stations and non-transfer stations. The parameters are shared globally or locally when using a static adjacency matrix, and the computational cost is relatively small. Still, it ignores the traffic flow patterns differences between different stations. However, we find that although different stations are directly connected or are all transfer stations, they have unique traffic change patterns, as shown in Figure 1(a). Therefore, it is necessary to model the traffic patterns of different stations separately.

2) Dynamic spatial dependency relations between stations: The spatial dependencies between stations are treated static in existing work [6], [8], [12]. Some of them express
of the difficulty of modeling long-term time series. As the prediction period becomes longer, the influence of uncertain factors will reduce the prediction accuracy, and the dynamic variance of the metro flow itself also increases the uncertainty. In general, compared with short-term prediction, long-term prediction is more difficult but has greater practical application value.

In order to cope with the above challenges, we propose a spatio-temporal dynamic graph relation learning method for metro flow prediction, which can model different traffic patterns at different stations and capture the dynamic spatial dependency relation between stations. At the same time, it can carry out long-term prediction, which can better support traffic management for metro operators and travel decisions for urban residents. The contributions of this paper include four aspects, as follows:

- A node-adaptive parameter learning module is adopted to learn different station-specific spatiotemporal embedding representations to capture the flow patterns of different stations.
- A dynamic graph relation learning module is proposed to learn the dynamic spatial dependencies between stations, which does not require a predefined spatial relationship of station connections, but directly learns the dynamic spatial dependencies between stations from spatiotemporal graph data.
- A long-term temporal relation prediction module based on Transformer is used to predict the long-term metro flow. The predicted results can offer a useful reference for urban metro operation management and personal travel planning.
- Experiments are conducted on 4 different cities’ metro datasets, including Beijing, Shanghai, Chongqing, Hangzhou. Compared with the 14 baseline methods, the experimental results have significantly improved prediction performance.

The remainder of this paper is organized as follows. In Section 2, we present the related work about urban flow prediction and graph neural networks. In Section 3, we introduce some preliminary concepts and formalize the metro flow prediction problem. In Section 4, we show the overall framework of the proposed STDGRL model. The experiment result, visualization and analysis are given in Section 5. We conclude the work in Section 6.

### 2 RELATED WORK

#### 2.1 Urban Flow Prediction

Urban flow prediction is important for traffic management [13], land use [14], public safety [15], etc. The urban flow prediction can be regarded as a spatio-temporal prediction task, which is a kind of research problem that uses spatio-temporal machine learning methods to learn spatio-temporal correlations from spatio-temporal datasets [16]. At present, a large number of researchers have conducted studies on the task of urban flow prediction. Xie et al. [17] divided the urban flow prediction task into crowd flow prediction, traffic flow prediction, and public transport flow prediction and reviewed the classical deep learning methods. With the city’s continuous development, more and more their spatial dependencies directly with the existence or lack of connections between stations. The distance between them and the similarity of traffic flow is regarded as spatial dependencies. But these static methods ignore the fact that the passenger inflow and outflow of a station are not only affected by its upstream, downstream, and nearby stations, but also depend on time, weather, and other external factors. Therefore, it’s a challenge to capture the dynamic spatial dependency relation between stations.

### 3) Long-term temporal prediction

To better support the downstream applications, it is necessary to carry out a long-term metro station flow prediction. Existing research [6] on short-term metro station passenger flow prediction has been carried out. Still, there is a lack of relevant research on long-term accurate metro station flow prediction because...
more people are pouring into the city, and the metro and
other public transportations occupy the main body of the
urban traffic flow. Accurate metro flow prediction is of great
value for urban traffic management, urban public safety, and
residents’ daily travel. In the early work, researchers used
statistical-based methods for urban flow prediction, such as
ARIMA (Autoregressive Integrated Moving Average) [18],
SARIMA (Seasonal Auto-Regressive Integrated Moving
Average) [19] and other methods. Later, some classic machine
learning methods were used for urban flow prediction, such
as SVR (Support Vector Regression) [20], K-NN (K-nearest
neighbor) [21] and other methods. But these methods of-
ten ignored spatiotemporal correlations are hinted in spa-
tiotemporal data, which are crucial for accurate urban flow
prediction.

In recent years, with the development of deep learning,
deep learning methods have been used in the research field
of urban flow prediction. The representative works mainly
include the time series method represented by RNN [22],
the spatial relation method represented by CNN [23], and a
spatiotemporal relationship method combining the two [9],
[24], [25]. Based on RNN and its variant series, these meth-
ods focus on capturing temporal dependencies in spatio-
temporal data, such as closeness, periodicity, trend, etc [15].
These CNN-based methods mainly capture the spatial de-
cendencies in spatiotemporal data, such as spatial distance,
spatial hierarchy, and regional functional similarity [25]. In
addition, such methods combining RNN and CNN consider
both temporal and spatial dependencies and propose hybrid
models to model the spatiotemporal characteristics in traffic
data [27].

Later, due to the rise and continuous development of the
graph neural network [28], [29], [30] and the graph structure
of the road network and rail transit network, more and more
researchers have used GNN-based methods for urban flow
prediction tasks [31], [32], [33] and achieved good results.
For more related papers, you can refer to these overview
papers [34], [35], [36], [37].

2.2 Graph Neural Networks

Graph neural networks can model graph data in non-
Euclidean space, especially the dependencies between
nodes. Graph neural networks research is developing
rapidly, and many research works have emerged [6], [38],
[39], [40]. Wu et al. [38] divided graph neural network meth-
ods into graph convolutional networks, graph attention
networks, graph autoencoders, graph generation networks,
and graph spatiotemporal networks. Applying the graph
neural network to urban flow prediction, traffic forecasting,
and other fields is natural. Since the road network and rail
transit network can be regarded as the road segments and
stations in the graph, the graph spatiotemporal network

However, the previous methods using GNNs for spa-
tiotemporal prediction tasks mostly use a predefined graph
structure or a single fixed graph adjacency matrix [41] or
multiple graph adjacency matrices for fusion [12]. This type
of method regards the spatial dependence in spatiotemporal
data as static and invariant. However, in reality, the spa-
tiotemporal relationship in spatio-temporal data is dynamic.
It is necessary to model the dynamic graph relationship
in spatio-temporal data and capture the spatio-temporal
dynamics. Compared with previous methods, our method
mainly learns the dynamic graph relationship in the spa-
tiotemporal data to obtain more accurate traffic prediction
results.

3 Problem Formulation

This paper proposes a spatio-temporal dynamic graph rel-
ating learning model for flow prediction in metro stations.
Our model does not need a predetermined metro network
topology map, and can directly learn spatial dependencies
from metro flow data, which has broad applicability to
metro flow prediction tasks in different cities.

Before introducing our model in detail, we first define
and represent the metro flow prediction task and related
conceptual notations. At station \( i \), the metro flow of time
period \( t \) can be expressed as \( X_{i,t} \in R^{2} \), which includes
the passenger inflow and outflow. The flow information
of the entire metro network can be expressed as \( X_{i,t} = (X_{1,t}, X_{2,t}, \ldots, X_{N,t}) \in R^{N \times 2} \), where \( N \) means the number
of metro stations. The metro flow in this paper contains two
perspectives, which are passenger inflow and outflow in
metro stations. The metro station flow prediction task can
be defined as, given the historical flow sequence, predicting
the flow sequence for a period of time in the future.

\[
X_{i,t+1}, X_{i,t+2}, \ldots, X_{i,t+m} = F_{\theta} (X_{i,t}, X_{i,t-1}, \ldots, X_{i,t-T+1}),
\]

(1)

where \( \theta \) means all the learnable parameters in the STDGRL
model, \( T \) is the length of the input flow sequence, and \( m \)
means the length of the predicted flow sequence.

4 Methodology

The overall architecture of the model is shown in Figure 2. It
contains a node-specific spatiotemporal embedding module,
a dynamic spatial relationship learning module, a long-term
temporal prediction module and a spatio-temporal fusion
module. First, we propose a node-specific spatio-temporal
embedding module to embed and represent the stations of
the metro spatio-temporal graph. Then we adopt a dynamic
spatial relationship learning module to learn the spatial
dependencies directly from the metro flow data without
relying on a specific metro network topology. Finally, a
Transformer-based long-time-series dependency prediction
module is used to predict the metro flow in a long-term
sequence, making its prediction more suitable for actual
metro dispatch management and daily operation scenarios.
4.1 Node-specific Spatio-Temporal Embedding

The node-specific adaptive parameter learning module (NAPL) is adopted. The classic graph convolution operation is calculated by the following formula:

\[ Z = \left( I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \right) X\Theta + b, \]  

(2)

where \( A \in \mathbb{R}^{N \times N} \) is the adjacency matrix of the graph, \( D \) is the degree matrix, \( I_N \) is the identity matrix, \( X \in \mathbb{R}^{N \times C} \) is the input of the graph convolutional network layer, \( Z \in \mathbb{R}^{N \times C} \) is the output of the graph convolutional network layer, \( C \) and \( F \) both are the embedding dimension respectively, \( \Theta \in \mathbb{R}^{F \times F} \) and \( b \in \mathbb{R}^F \) represent learnable weights and biases, respectively.

In this method, all nodes on the graph share parameters such as weights and biases. According to [42], different nodes have different traffic flow patterns, as shown in Figure 1(a), because they have different attributes, such as POI distribution around the nodes, various weather conditions, and different flow patterns. For more accurate traffic prediction, it is necessary to learn different traffic patterns for different nodes, that is, to learn node-specific patterns by using different learnable parameters rather than globally shared parameters.

In order to learn node-specific patterns, a node-specific adaptive parameter learning module is proposed, which learns the node embedding matrix \( E_G \in \mathbb{R}^{d \times d} \) and weight pool \( W_G \in \mathbb{R}^{d \times C \times F} \). The \( \Theta \) in Formula 2 can be calculated by the node embedding matrix and the weight pool, \( \Theta = E_G \cdot W_G \). Such a computation can be interpreted as learning node-specific patterns from all station time-series patterns. The bias \( b \) can also be calculated in the same way. The parameter module of the final node adaptation can be expressed by Formula 3:

\[ Z = \left( I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \right) XE_GW_G + E_Gb_G. \]  

(3)

4.2 Dynamic Spatial Relation Learning

In a metro network, the connection relationship between stations is fixed and static. However, static connection relationship cannot reflect the dynamic spatial dependence between stations. Moreover, the passenger’ inflow and outflow change over time, so it is necessary to learn this dynamic spatial dependency from spatiotemporal data. Therefore, a dynamic spatial relationship learning module (DSRL) is proposed, which is a representation model with adaptive and spatial structure awareness. Inspired by [42], we first randomly initialize a learnable node embedding dictionary \( E_A \in \mathbb{R}^{d \times d} \) for all nodes. During the model training process, \( E_A \) will be dynamically updated. Each row of \( E_A \) represents the embedding representation of the node, and \( d_e \) represents the dimension of node embedding. Then, the spatial dependency between nodes is calculated by multiplying \( E_A \) and \( E_A^T \). Finally, we can get the generated graph Laplacian matrix as shown in the formula below.

\[ D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = \text{softmax}(\text{ReLU}(E_A \cdot E_A^T)), \]  

(4)

where the softmax function is used to normalize the learned adaptive matrix. The calculation formula of GCN is as follows:

\[ Z = \left( I_N + \text{softmax}(\text{ReLU}(E_A \cdot E_A^T)) \right)X\Theta + b. \]  

(5)

For the nodes at time step \( t \), the operation of a GRU module can be expressed as follows:

\[ \tilde{A} = \text{softmax}\left( \text{ReLU}(E_AE_A^T) \right), \]

\[ z_t = \sigma_z \left( \tilde{A} \left[ X_{:,t}, h_{t-1} \right] EW_z + Eb_z \right), \]

\[ r_t = \sigma_r \left( \tilde{A} \left[ X_{:,t}, h_{t-1} \right] EW_r + Eb_r \right), \]

\[ h_t = \text{tanh}\left( \tilde{A} \left[ X_{:,t}, r \odot h_{t-1} \right] EW_h + Eb_h \right), \]

\[ h_t \equiv z_t \odot h_{t-1} + (1-z_t) \odot h_t, \]  

(6)

where \([\cdot]\) means the concate operation, \(\odot\) denotes the element-wise multiplication, \(E, W_z, W_r, W_h, b_z, b_r, b_h\) are the parameters to be learned, \(X_{:,t}\) and \(h_t\) are input and output at time step \( t \). Finally, the output \( Y_S \) of the component is obtained through a fully connected network.
4.3 Long-Term Temporal Prediction

To capture the long-term global dependencies of metro flow sequences, we propose a long-term temporal prediction module (LTTP). A Transformer-based [43] long-term temporal prediction method is adopted for long-term metro flow prediction. This layer includes a multi-head self-attention layer, a feed-forward neural network layer, and a layer normalization layer. First, the multi-head self-attention layer is introduced. The attention calculation formula is shown in Formula 7. The dot product between all keys and the given queries is calculated, divided by $\sqrt{d_k}$, and then multiplied by $V$. Finally, a softmax function is used to calculate the attention score of each position. These attention scores will be used as weights to aggregate information from different parts. Long-term temporal dependencies are computed in high-dimensional latent subspaces.

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V,$$

where $Q, K \in \mathbb{R}^{T \times d_k}$ and $V \in \mathbb{R}^{T \times d_v}$ mean the query subspace, key subspace and value subspace of all nodes, respectively. A position embedding is added to each position to enable the LTTP layer to perceive the relative position in the entire traffic sequence. The formula of position coding $e_t$ is shown below:

$$e_t = \begin{cases} \sin \left( \frac{t \times 10000^{2i/d_{model}}} {10000} \right), & \text{if } t = 0, 2, 4 \ldots \\ \cos \left( \frac{t \times 10000^{2i/d_{model}}} {10000} \right), & \text{otherwise}. \end{cases}$$

Then, the output calculated by the multi-head self-attention layer is passed to the feedforward neural network layer. Finally, the output $Y_T$ of the LTTP network is obtained through the residual connection [44] and layer normalization.

4.4 Spatio-temporal Fusion

In order to effectively utilize the captured temporal and spatial dependencies, we adopt spatio-temporal fusion module to fuse the learned temporal and spatial dependencies. As shown in the following formula:

$$X_{:,t+1}, X_{:,t+2}, \ldots, X_{:,t+n} = W_S \odot Y_S + W_T \odot Y_T,$$

where $Y_S$ is the output of spatial relation learning module, $Y_T$ is the output of temporal relation learning module, $\odot$ is the Hadamard product, $W_S$ and $W_T$ are the learnable weight parameters.

5 Experiments

In this section, we first introduce the experimental setup, including the description of the dataset, experimental environment, implementation details, and evaluation metrics. Next, we compare our proposed method STDGRL with 14 representative methods. Finally, we conduct extensive experiments and analyze the effectiveness of our model and each module.

5.1 Experiments Settings

1) Dataset description: In this paper, we use 4 metro card swiping datasets: Beijing Metro dataset [6], Shanghai Metro dataset [12], Chongqing Metro dataset, and Hangzhou Metro dataset [12].

BJMetro: This dataset collects the data of Beijing Metro for five consecutive weeks from February 29 to April 3, 2016. It contains 17 metro lines and 276 metro stations, excluding the Airport Express and its stations.

SHMetro: This dataset uses the Shanghai Metro dataset published in [12], and the format of the dataset is consistent with the original paper. The time slice size is 15 minutes, and the time span is from July 1 to September 30, 2016. The Shanghai Metro dataset contains a total of 288 stations.

CQMetro: This dataset is private and obtained by preprocessing the Chongqing metro swiping card data. We divide the data into 15-minute time slices to get the passenger inflow and outflow of the stations within the time slice. The time span is from March 1 to March 31, 2019. The Chongqing Metro dataset contains a total of 170 stations.

HZMetro: This dataset also uses the Hangzhou Metro dataset published in [12]. The format of the dataset is consistent with the original paper. The time slice size is 15 minutes, and it contains 80 stations. The time frame is January 2019, with a total of 25 days.

2) Implementation details: We use the deep learning framework PyTorch [45] to implement the model STDGRL in this paper and the deep learning models in the comparison methods. The experimental equipment uses a GPU card with an NVIDIA Titan V. In the Chongqing Metro data set, the card swiping data between 23:00-06:00 every day is directly deleted. Since this period is not within the operating time range of the metro, no passenger enter or leave the stations. We normalized the dataset in the same way as used in AGCRN [42]. The training set, validation set, and test set of the four datasets are divided in a chronological order according to the ratio of 7:1:2. The batch size is set to 64. The Adam [46] optimizer is used to optimize our model for a maximum of 200 epochs. And we use an early stop strategy with the patience of 50. The learning rate is 0.01. We take the data of the 4 historical time steps as input and the data of the next 4 time steps as output. Although our proposed method does not require a predefined adjacency matrix graph, we use the predefined adjacency matrix graph method as a contrasting method.

3) Evaluation metrics: We use three metrics commonly used in spatiotemporal prediction tasks, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE), to evaluate the performance of the method.

- Mean Absolute Error (MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|.$$

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}.$$
JOURNAL OF LTEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2015

- Mean Absolute Percentage Error (MAPE)

\[
MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right|
\]

where \(n\) is the number of test samples, \(\hat{y}_i\) and \(y_i\) mean the predicted passenger flow and the actual passenger flow, respectively. \(\hat{y}_i\) and \(y_i\) are transformed into the scale of the original value by inverse Z-score normalization.

5.1.1 Baselines

In this section, we compare the proposed STDGRL model with 14 baseline models, as shown in Table 1. These models can be divided into five categories, including (1) two traditional time series models, (2) two single deep learning models, (3) eight graph spatiotemporal network models for traffic prediction or multivariate time series forecasting proposed in recent years, (4) one Transformer-based traffic prediction model, and (5) one recently proposed graph neural network model for metro passenger flow prediction. These models are described in detail as follows:

- **Historical Average (HA)** [47]: This model obtains the current traffic by averaging the historical traffic in the same time slice. This method is calculated for a single time series each time.
- **Support Vector Regression (SVR)** [48]: This machine learning model serves as a classic baseline model for a class of time series forecasting, using linear support vector machines for time series forecasting tasks. It is often used as a comparison method in time series forecasting tasks.
- **Long Short-Term Memory (LSTM)** [49]: This is a classic deep learning method for time series that captures the temporal correlations of spatiotemporal sequences.
- **Gated Recurrent Unit (GRU)** [50]: As a variant model of RNN, it can also capture the time-series correlation in the spatiotemporal sequence, but it cannot learn the spatial correlation. It is a time series forecasting method based on deep learning.
- **T-GCN** [51]: It is a traffic prediction model based on graph convolutional network, which can capture spatiotemporal dependencies in spatiotemporal sequence data. It combines a graph convolutional neural network and a gated recurrent neural network.
- **DCRNN** [52]: To capture the complex spatial dependencies and nonlinear temporal dynamics of road networks, a diffusion convolutional recurrent neural network is proposed for traffic prediction. It is one of the classic methods for spatiotemporal sequence prediction in graph neural network-based methods.
- **STGCN** [53]: This is a spatiotemporal graph convolutional network based on convolutional structure, and it is used for the traffic prediction task. It has a faster training speed and fewer parameters.
- **AGCRN** [54]: This method does not require a predefined spatial graph and is an adaptive graph convolutional network that can learn spatiotemporal dependencies from spatiotemporal data.
- **Graph WaveNet** [55]: It uses a node embedding method to learn the adaptive spatial graph structure, a spatiotemporal graph network method combining graph convolution and dilated causal convolution is proposed.
- **STTN** [52]: It is a Transformer-base spatio-temporal model for traffic prediction.
- **Multi-STGCnet** [8]: It is a combined model containing graph convolutional network and LSTM for metro passenger flow prediction.
- **GMAN** [55]: This is a graph multi-attention encoder-decoder model for long-term traffic prediction.
- **MTGNN** [54]: It is a graph neural network framework for multivariate time series forecasting, which can capture the spatial and temporal dependencies in spatio-temporal data.
- **ASTGCN** [55]: It is an attention based spatial temporal graph convolutional network for traffic flow forecasting, the model contains spatial-temporal attention mechanism and spatio-temporal convolution modules.
- **STDGRL (ours)**: The proposed spatiotemporal prediction network based on spatiotemporal dynamic graph relationships for traffic forecasting in metro stations. Compared with the previous methods, our method does not require a predefined spatial graph on the one hand and can perform long-term metro flow prediction on the other hand.

| Model         | Temporal Relation | Spatial Relation | Node Embedding | ST Fusion |
|---------------|-------------------|------------------|----------------|-----------|
| HA            | ✓                 |                  |                |           |
| SVR           | ✓                 |                  |                |           |
| LSTM          | ✓                 |                  |                |           |
| GRU           | ✓                 |                  |                |           |
| T-GCN         | ✓                 | ✓                | ✓              |           |
| DCRNN         | ✓                 | ✓                | ✓              |           |
| STGCN         | ✓                 | ✓                | ✓              |           |
| AGCRN         | ✓                 |                  | ✓              |           |
| Graph WaveNet | ✓                 |                  | ✓              |           |
| STTN          | ✓                 |                  | ✓              |           |
| Multi-STGCnet | ✓                 |                  | ✓              |           |
| GMAN          | ✓                 |                  | ✓              |           |
| MTGNN         | ✓                 |                  | ✓              |           |
| ASTGCN        | ✓                 |                  | ✓              |           |
| STDGRL (ours) | ✓                 |                  | ✓              |           |

| Model  | Total Training Time (s) | Training Time (s) Per Epoch |
|--------|-------------------------|----------------------------|
| STDGRL | 658.2                   | 3.291                      |
| ASTGCN | 995.4                   | 9.954                      |
| MTGNN  | 1212.6                  | 12.126                     |
| GMAN   | 8986.4                  | 112.33                     |

5.2 Overall Performance

Table 3 to Table 6 show the overall prediction performance of our method and 14 comparative methods on the Beijing, Shanghai, Chongqing, and Hangzhou Metro datasets. In the prediction interval of the next hour, three evaluation indicators MAE, RMSE, and MAPE are used for evaluation. We can see that the results of the classical machine learning-based time series forecasting method are worse than the...
deep learning-based methods such as LSTM, GRU methods, indicating that the modeling of non-linear data dependencies in the spatiotemporal data is crucial when making traffic predictions. In addition, we also find that the performance of the traffic prediction models based on graph neural network proposed in recent years are better than LSTM and GRU methods. The reason is that they can capture the spatio-temporal dependence in spatio-temporal graph data better than deep learning models.

On the SHMetro dataset, our method STDGRL completely surpasses the most related three methods GMAN, MTGNN, and ASTGCN in terms of MAE and MAPE. Moreover, we also recorded the training time of the three models and ours. We find that the total training time of our method is 658.2s, which is smaller than the three methods (995.4s, 1212.6s, and 8986.4s, respectively); and the average training time per epoch of our method is also smaller. So it is much faster to train our model. Detailed time are shown in Table 2. On the CQMetro dataset, the MAPE value of our method outperforms GMAN, MTGNN, and ASTGCN for the next 15 minutes prediction. We also beat MTGNN and ASTGCN for the next 30 minutes, 45 minutes, and 60 minutes prediction. As for BJMetro and HZMetro datasets, the improvements of our method are relatively smaller or even behind others, but our model still performs very competitively. In general, it is not our goal to develop a "all-win" model that can beat all other methods on all datasets (neither do other methods). Rather, we see the pros & cons of each method, which has its best use cases in different settings. Given there are significant differences between metro networks and traffic patterns in different cities, our method, overall, has attained an excellent prediction performance and fast training speed. Figure 3 shows the inflow and outflow prediction performance at one day in the SHMetro dataset.

### Table 3: Performance comparison of baseline methods on BJMetro dataset.

| Model        | MAE (15min) | MAE (30min) | MAE (45min) | MAE (60min) |
|--------------|-------------|-------------|-------------|-------------|
| HA           | 95.7779     | 207.2597    | 0.7318      | 95.7779     |
| SVR          | 133.3139    | 313.8002    | 2.1439      | 143.1395    |
| LSTM         | 99.2410     | 243.2237    | 1.9165      | 115.4021    |
| GRU          | 96.3814     | 237.3694    | 1.7907      | 98.0139     |
| T-GCN        | 97.1880     | 157.4064    | 1.8642      | 126.7785    |
| DCRNN        | 32.4452     | 67.2272     | 0.2681      | 47.0715     |
| STGCN        | 32.1576     | 62.6209     | 0.3366      | 37.8307     |
| AGCRN        | 25.1688     | 47.8686     | 0.2397      | 25.3167     |
| STTN         | 35.6133     | 78.8141     | 0.3647      | 32.7436     |

### Table 4: Performance comparison of baseline methods on SHMetro dataset.

| Model        | MAE (15min) | MAE (30min) | MAE (45min) | MAE (60min) |
|--------------|-------------|-------------|-------------|-------------|
| HA           | 76.9445     | 169.6002    | 0.9358      | 76.9445     |
| SVR          | 89.4518     | 230.2805    | 1.2532      | 94.6976     |
| LSTM         | 58.1613     | 108.2152    | 0.6381      | 57.8482     |
| GRU          | 31.2478     | 65.8625     | 0.3176      | 32.5833     |
| T-GCN        | 74.6344     | 124.6865    | 1.3138      | 76.1906     |
| DCRNN        | 27.9394     | 54.2426     | 0.2633      | 37.2232     |
| STGCN        | 28.2697     | 52.5552     | 0.3136      | 36.9222     |
| AGCRN        | 24.0087     | 47.1056     | 0.2316      | 27.0434     |
| STTN         | 29.0291     | 56.2013     | 0.2661      | 30.2127     |
| Graphwavernet| 26.2299     | 50.3182     | 0.2448      | 38.1380     |
| Multi-STGCNet| 49.6580     | 128.6207    | 0.9756      | 50.4986     |
| GMAN         | 25.7015     | 48.1071     | 0.3227      | 26.1412     |
| MTGNN        | 24.4736     | 46.1361     | 0.3183      | 27.8870     |
| ASTGCN       | 48.1161     | 87.3258     | 0.7461      | 53.7781     |
| STDGRL       | 23.7239     | 46.8692     | 0.2143      | 24.3754     |

**Authorized licensed use limited to: SOUTHWEST JIATONG UNIVERSITY. Downloaded on July 02, 2023 at 03:17:46 UTC from IEEE Xplore. Restrictions apply. © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.**
TABLE 5: Performance comparison of baseline methods on CQMetro dataset.

| Model     | 15min MAE | 15min RMSE | 15min MAPE | 30min MAE | 30min RMSE | 30min MAPE | 45min MAE | 45min RMSE | 45min MAPE | 60min MAE | 60min RMSE | 60min MAPE |
|-----------|-----------|------------|------------|-----------|------------|------------|-----------|------------|------------|-----------|------------|------------|
| HA        | 56.6874   | 120.7926   | 0.6848     | 56.6874   | 120.7926   | 0.6848     | 56.6874   | 120.7926   | 0.6848     | 56.6874   | 120.7926   | 0.6848     |
| SVR       | 60.0164   | 143.7840   | 1.2217     | 61.4924   | 145.5862   | 1.2169     | 64.1068   | 149.3066   | 1.2537     | 67.5121   | 154.3567   | 1.3186     |
| LSTM      | 15.1076   | 29.2919    | 0.8762     | 14.9974   | 28.5219    | 0.9348     | 15.4466   | 29.1093    | 1.0224     | 15.8549   | 29.6383    | 1.2524     |
| GRU       | 14.5013   | 28.4447    | 0.8418     | 14.3555   | 27.7498    | 0.8976     | 14.5529   | 28.0491    | 0.9320     | 14.6878   | 28.1582    | 1.0298     |
| T-GCN     | 20.1979   | 33.4217    | 1.3637     | 21.1046   | 34.6047    | 1.5492     | 23.0371   | 37.3405    | 1.8133     | 24.7309   | 40.4497    | 2.2657     |
| DCRNN     | 15.3833   | 28.6454    | 0.8490     | 15.9655   | 29.1312    | 0.9072     | 17.1855   | 32.3489    | 0.9258     | 18.3593   | 35.7147    | 0.9949     |
| STGCN     | 14.8434   | 26.5124    | 0.9370     | 13.0715   | 23.3458    | 0.8382     | 13.1600   | 23.3890    | 0.8548     | 13.4021   | 23.9076    | 0.9473     |
| AGCRN     | 12.8426   | 23.2149    | 0.7358     | 12.3304   | 22.2855    | 0.6931     | 12.4552   | 22.6841    | 0.6900     | 12.5081   | 22.6701    | 0.7402     |
| STTN      | 15.0992   | 27.9610    | 0.8255     | 14.9527   | 27.4131    | 0.8817     | 14.9681   | 26.8947    | 0.8621     | 15.6465   | 28.1648    | 1.0059     |
| Graphwavenet | 14.3624 | 25.8309    | 0.7889     | 14.5080   | 25.3433    | 0.8629     | 15.0909   | 26.5043    | 0.9230     | 15.7601   | 27.4222    | 1.1251     |
| Multi-STGCN | 17.5820 | 36.3206    | 0.8167     | 17.4633   | 35.7817    | 0.8414     | 17.4939   | 35.9225    | 0.8347     | 17.5682   | 36.0584    | 0.8753     |
| GMAN      | 12.2238   | 20.6095    | 0.7700     | 12.1508   | 20.7265    | 0.7563     | 12.2014   | 20.8506    | 0.7606     | 12.3904   | 21.1702    | 0.7761     |
| MTGNN     | 12.5330   | 22.8966    | 0.6737     | 48.9423   | 77.2453    | 5.4796     | 49.2734   | 77.6676    | 5.9827     | 50.7017   | 80.6448    | 6.7085     |
| ASTGCN    | 27.0901   | 40.6978    | 2.7829     | 28.7492   | 43.0850    | 3.4224     | 30.2731   | 46.1285    | 4.3312     | 31.7619   | 49.6536    | 5.6773     |
| STDGRL(ours) | 12.4831 | 23.0040    | 0.6421     | 12.3304   | 22.2855    | 0.6931     | 12.4552   | 22.6841    | 0.6900     | 12.5081   | 22.6701    | 0.7402     |

This dataset contains 288 stations more than other cities stations like Beijing, Chongqing, and Hangzhou. It shows that our proposed method performs well on a small number of stations and also achieves good experimental results on a large number of stations.

Fig. 3: Inflow and outflow prediction visualization on the SHMetro dataset.

![Inflow](image1)

![Outflow](image2)

Fig. 4: Ablation study performance on the SHMetro dataset.

5.3 Ablation Study

We design a comprehensive ablation study to evaluate the sub-modules of STDGRL. The baseline model of our ablation study is GCGRU(T-GCN). This model is a classical traffic forecasting method, which combines GCN and GRU for capturing spatio-temporal dependencies. And we remove the NAPL component from the STDGRL model to construct STDGRL-NAPL. STDGRL-Transformer and STDGRL-GRU-Transformer are the variants of our STDGRL respectively,
which remove GRU module, GRU and Transformer module from STDGR-L model. The experimental result on the four datasets are illustrated in Table 7 to Table 10.

We also show the ablation study performance on the SHMetro dataset in Figure 8. We can observe that: 1) The results in the Table show that the performance of GCGRU (T-GCN) is not as good as that of the other three comparison models, which may be due to its use of pre-defined graphs and difficulty in capturing complex spatial dependencies between nodes. 2) Compared with the STDGR-L model, the performance of the STDGR-NAPL model decreases by a large proportion and is inferior to STDGR-Transformer and STDGR-GRU-Transformer, indicating that it is necessary to capture node-specific traffic patterns in the STDGR-L model. 3) After Transformer and GRU modules are removed from the STDGR-L model, the performance is lower than that of the STDGR-L model, but better than that of the STDGR-NAPL model, indicating the necessity of using short-term and long-term time series prediction modules in the STDGR model. And it also demonstrates learning the specific traffic patterns of nodes are more important than learning temporal dependencies.

Overall, NAPL, DSRL and temporal learning modules jointly boost the prediction performance of the STDGR-L model.

In summary, the experiment result demonstrates that STDGR-L can learn the spatial and temporal relation from the metro spatio-temporal graph of different scales and achieve promising prediction performance.
6 Conclusion

We proposed STDGRL, a novel spatio-temporal dynamic graph relationship learning model, for predicting multi-step passenger inflow and outflow in urban metro stations. STDGRL can capture the traffic patterns of different metro stations and the dynamic spatial dependencies between metro stations. In addition, STDGRL can capture long-term temporal relationship dependencies for long-term metro flow prediction. We validated our model on real metro datasets in 4 cities and experimental results achieved significant performance improvements over 14 baselines. In future work, we plan to research the influence of weather, events and POI on the change of metro passenger flow, and the detection and prediction of sudden large passenger flow in metro stations.

Acknowledgment

This research was supported by the National Key R&D Program of China (2019YFB2101801) and the National Natural Science Foundation of China (No. 62176221, No. 62276215). The authors would like to thank Zhipeng Luo for his insightful discussions.

References

[1] Y. Gong, Z. Li, J. Zhang, W. Liu, and Y. Zheng, “Online spatio-temporal crowd flow distribution prediction for complex metro system,” IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 2, pp. 865–880, 2022.
[2] Y. Gong, Z. Li, J. Zhang, W. Liu, and J. Yi, “Potential passenger flow prediction: A novel study for urban transportation development,” in Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 4020–4027.
[3] Q. Cheng, W. Deng, and M. A. Raza, “Analysis of the departure time choices of metro passengers during peak hours,” IET Intelligent Transport Systems, vol. 14, no. 8, pp. 866–872, 2020.
[4] L. Tang, Y. Zhao, J. Cabrera, J. Ma, and K. L. Tsui, “Forecasting short-term passenger flow: An empirical study on shenzhen metro,” IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 10, pp. 3631–3622, 2018.
[5] Y. Liu, Z. Liu, and R. Jia, “Deeppf: A deep learning based architecture for metro passenger flow prediction,” Transportation Research Part C: Emerging Technologies, vol. 101, pp. 18–34, 2019.
[6] J. Zhang, F. Chen, Z. Cui, Y. Guo, and Y. Zhu, “Deep learning architecture for short-term passenger flow forecasting in urban rail transit,” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 11, pp. 7004–7014, 2020.
[7] P. Chen, X. Fu, and X. Wang, “A graph convolutional stacked bidirectional uni-directional lstm neural network for metro ridership prediction,” IEEE Transactions on Intelligent Transportation Systems, no. 99, pp. 1–13, 2021.
[8] J. Ye, J. Zhao, K. Ye, and C. Xu, “Multi-stgcn: A graph convolution based spatial-temporal framework for subway passenger flow forecasting,” in Proceedings of the 2020 International Joint Conference on Neural Networks, 2020, pp. 1–8.
[9] X. Ma, J. Zhang, B. Du, C. Ding, and L. Sun, “Parallel architecture of convolutional bi-directional lstm neural networks for network-wide metro ridership prediction,” IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 6, pp. 2278–2288, 2018.
[10] J. Ou, J. Sun, Y. Zhu, H. Jin, Y. Liu, F. Zhang, J. Huang, and X. Wang, “Stp-trellisnets: Spatial-temporal parallel trellisnets for metro station passenger flow prediction,” in Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 1185–1194.
[11] L. Liu, J. Chen, H. Wu, J. Zhen, G. Li, and L. Lin, “Physical-virtual collaboration modeling for intra-and inter-station metro ridership prediction,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 4, pp. 3377–3391, 2020.
[12] J. Chen, L. Liu, H. Wu, J. Zhen, G. Li, and L. Lin, “Physical-virtual collaboration graph network for station-level metro ridership prediction,” arXiv preprint arXiv:2001.04889, 2020.
[13] Q. Chen, X. Song, H. Yamada, and R. Shibasaka, “Learning deep representation from big and heterogeneous data for traffic accident inference,” in Proceedings of the AAAI Conference on Artificial Intelligence, 2016, pp. 338–344.
[14] K. Jayarajah, A. Tan, and A. Misra, “Understanding the inter-dependency of land use and mobility for urban planning,” in Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers, 2018, pp. 1079–1087.
[15] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in Proceedings of the 2017 ACM Conference on Artificial Intelligence, 2017, pp. 1655–1661.
[16] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, “Dnn-based prediction model for spatio-temporal data,” in Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2016, pp. 921–924.
[17] P. Xie, T. Li, J. Liu, S. Du, X. Yang, and J. Zhang, “Urban flow prediction from spatiotemporal data using machine learning: A survey,” Information Fusion, vol. 59, pp. 1–12, 2020.
[18] B. M. Williams, P. K. Durvasula, and D. E. Brown, “Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models,” Transportation Research Record, vol. 1644, no. 1, pp. 132–141, 1998.
Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Seasonal autoregressive integrated moving average and support vector machine models: prediction of short-term traffic flow on freeways,” Transportation Research Record, vol. 2215, no. 1, pp. 85–92, 2011.

M. Lippi, M. Bertini, and P. Frasconi, “Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning,” IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 2, pp. 871–882, 2013.

F. G. Habtemichael and M. Cetin, “Short-term traffic flow rate forecasting based on identifying similar traffic patterns,” Transportation Research Part C: Emerging Technologies, vol. 66, pp. 61–78, 2016.

Z. Zhao, W. Chen, X. Wu, H. Chen, and J. Liu, “LSTM network: a deep learning approach for short-term traffic forecast,” IET Intelligent Transportation Systems, vol. 11, no. 2, pp. 68–75, 2017.

W. Jiang and L. Zhang, “Geospatial data to images: A deep-learning framework for traffic forecasting,” Tsinghua Science and Technology, vol. 24, no. 1, pp. 52–64, 2018.

Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, “A hybrid deep learning based traffic flow prediction method and its understanding,” Transportation Research Part C: Emerging Technologies, vol. 90, pp. 166–180, 2018.

Z. Duan, Y. Yang, K. Zhang, Y. Ni, and S. Bijajin, “Improved deep hybrid networks for urban traffic flow prediction using trajectory data,” IEEE Access, vol. 6, pp. 31820–31827, 2018.

J. Zhang, Y. Zheng, J. Sun, and D. Qi, “Flow prediction in spatio-temporal networks based on multitask deep learning,” IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 3, pp. 468–478, 2019.

H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, “Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction,” in Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 5668–5675.

J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and deep locally connected networks on graphs,” in Proceedings of the 2nd International Conference on Learning Representations, 2014.

M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in Advances in Neural Information Processing Systems, 2016, pp. 3837–3845.

T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proceedings of the 5th International Conference on Learning Representations, 2017.

W. Jiang and J. Luo, “Graph neural network for traffic forecasting: A survey,” arXiv preprint arXiv:2101.11174, 2021.

Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” in Proceedings of the 6th International Conference on Learning Representations, 2018.

B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting,” in Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 2018, pp. 3634–3640.

D. A. Tedspjurmomo, Z. Bao, B. Zheng, F. Choudhury, and A. K. Qin, “A survey on modern deep neural network for traffic prediction: Trends, methods and challenges,” IEEE Transactions on Knowledge and Data Engineering, 2020.

J. Ye, J. Zhao, K. Ye, and C. Xu, “How to build a graph-based deep learning architecture in traffic domain: A survey,” IEEE Transactions on Intelligent Transportation Systems, 2020.

A. Boubrache, Y. Tao, and P. Sun, “Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems,” Computer networks, vol. 182, p. 107484, 2020.

K. N. Bui, J. Cho, and H. Yi, “Spatio-temporal graph neural network for traffic forecasting: An overview and open research issues,” Applied Intelligence, vol. 52, no. 5, pp. 2763–2774, 2022.

Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, “A comprehensive survey on graph neural networks,” IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4–24, 2020.

Y. Zhu, W. Xu, J. Zhang, Y. Du, J. Zhang, Q. Liu, C. Yang, and S. Wu, “A survey on graph structure learning: Progress and open challenges,” arXiv e-prints, pp. arXiv–2103, 2021.

F. Dai, T. Zhao, H. Zhu, J. Xu, Z. Guo, H. Liu, J. Tang, and S. Wang, “A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability,” arXiv preprint arXiv:2204.08570, 2022.
Minbo Ma received B.S. degree from Chongqing Three Gorges University, Chongqing, China in 2017. He received M.S. degree from Southwest Jiaotong University, Chengdu, China in 2019. He is working toward the PhD degree in Southwest Jiaotong University. His research interests include spatio-temporal data mining, urban computing, information extraction, machine learning, etc.

Tianrui Li received the B.S., M.S., and Ph.D. degrees from the Southwest Jiaotong University, Chengdu, China in 1992, 1995, and 2002, respectively. He was a postdoctoral researcher with SCK CEN, Belgium from 2005 to 2006, and a visiting professor with Hasselt University, Belgium, in 2008, the University of Technology, Sydney, Australia, in 2009, and the University of Regina, Canada, 2014. He is currently a Professor and the Director of the Key Laboratory of Cloud Computing and Intelligent Techniques, Southwest Jiaotong University. He has authored or coauthored more than 300 research papers in refereed journals and conferences. His research interests include big data, cloud computing, data mining, granular computing and rough sets. He is a fellow of IRSS and senior member of ACM and IEEE.

Shengdong Du received the Ph.D. degrees from the Southwest Jiaotong University, Chengdu, China in 2020. He is now an associate professor at the school of computing and artificial intelligence, southwest jiaotong university. His research interests include data mining and machine learning.

Zeng Yu is an Assistant Professor in the School of Computing and Artificial Intelligence, Southwest Jiaotong University. He received the BS and MS degrees from the School of Mathematics, China University of Mining and Technology in 2008 and 2011, and the PhD degree from the School of Information Science and Technology, Southwest Jiaotong University in 2018, respectively. He was a visiting PhD student with Georgia State University, USA, from 2014 to 2016. His research interests include data mining, machine learning and computer vision.

Junbo Zhang is a Senior Researcher of JD Intelligent Cities Research. He is leading the Urban AI Product Department of JD iCity at JD Technology, as well as AI Lab of JD Intelligent Cities Research. Prior to that, he was a researcher at Microsoft Research Asia (MSRA). He has published over 50 papers in spatio-temporal data mining and AI, urban computing, deep learning, and federated learning. He serves as an Associate Editor of ACM Transactions on Intelligent Systems and Technology. He received a number of honors, including the Second Prize of the Natural Science Award of the Ministry of Education in 2021, the 22nd China Patent Excellence Award in 2021, the ACM Chengdu Doctoral Dissertation Award in 2016, the Chinese Association for Artificial Intelligence (CAAI) Excellent Doctoral Dissertation Nomination Award in 2016. He is a senior member of CCF, a member of IEEE and ACM.