Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Transfer learning with a spatiotemporal graph convolution network for city flow prediction*

Binkun LIU[1,2,3], Yu KANG[1,3,4], Yang CAO[†‡1,3,4], Yunbo ZHAO[1,3,4], Zhenyi XU[†‡2,3,4]

[1]*Department of Automation, University of Science and Technology of China, Hefei 230026, China*

[2]*Key Laboratory of System Control and Information Processing, Ministry of Education, Shanghai 200240, China*

[3]*Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China*

[4]*Institute of Advanced Technology, University of Science and Technology of China, Hefei 230088, China*

[†]E-mail: forrest@ustc.edu.cn; xuzhenyi@mail.ustc.edu.cn

Received Aug. 23, 2023; Revision accepted Dec. 11, 2023; Crosschecked Nov. 15, 2024; Published online Dec. 27, 2024

Abstract: Recently, deep learning based city flow prediction has been extensively used in the establishment of smart cities. These methods are data-hungry, making them unscalable to areas lacking data. Although transfer learning can use data-rich source domains to assist target domain cities in city flow prediction, the performance of existing methods cannot meet the needs of actual use, because the long-distance road network connectivity is ignored. To solve this problem, we propose a transfer learning method based on spatiotemporal graph convolution, in which we construct a co-occurrence space between the source and target domains, and then align the mapping of the source and target domains' data in this space, to achieve the transfer learning of the source city flow prediction model on the target domain. Specifically, a dynamic spatiotemporal graph convolution module along with a temporal encoder is devised to simultaneously capture the concurrent spatiotemporal features, which implies the inherent relationship among the road network structures, human travel habits, and city bike flow. Then, these concurrent features are leveraged as cross-city invariant representations and nonlinearly spanned to a co-occurrence space. The target domain features are thereby aligned with the source domain features in the co-occurrence space by using a Mahalanobis distance loss, to achieve cross-city bike flow prediction. The proposed method is evaluated on the public bike flow datasets in Chicago, New York, and Washington in 2015, and significantly outperforms state-of-the-art techniques.

Key words: Transfer learning; City flow prediction; Spatiotemporal graph convolution
https://doi.org/10.1631/FITEE.2300571 **CLC number:** TP311; U495

1 Introduction

In recent years, many countries and regions have been studying the construction of smart cities (Tascikaraoglu, 2018), hoping to grasp real-time data such as urban air quality (Kang et al., 2018) and urban flow (Xu et al., 2023) to predict the future state and achieve efficient management of the city. City flow prediction is an important part of a smart city, and can help government departments effectively control traffic congestion and pollutant emissions. The main task of city flow prediction is to accurately predict the corresponding state in the future using the given historical traffic flow.

The existing city flow prediction methods are divided mainly into the parametric model and the non-parametric model. Parametric models (Van Der

‡ Corresponding authors

* Project supported by the National Natural Science Foundation of China (Nos. 62103124 and 62033012), the Major Special Science and Technology Project of Anhui Province, China (No. 202003a07020009), and the Open Project Program of Key Laboratory of Ministry of Education of System Control and Information Processing, China (No. SCIP20230109)

ORCID: Binkun LIU, https://orcid.org/0000-0002-6812-876X; Yang CAO, https://orcid.org/0000-0002-2891-4379; Zhenyi XU, https://orcid.org/0000-0002-5804-882X

Voort et al., 1996) refer to a model based on certain assumptions, usually with a fixed structure, and parameters are calculated based on empirical data. Non-parametric models (Mallick et al., 2019; Zhao et al., 2020) are represented by deep learning models, which can effectively capture nonlinear spatiotemporal features. They both have a strong dependence on historical data. When historical data are very scarce, these models often perform poorly. Considering that not every city has a large amount of historical data, when the target city has only a small amount of historical data, the means to use data-rich source cities to help target cities improve the accuracy of flow prediction is a problem worthy studying. The main challenge involved in tackling this problem comprises the fact that the source-domain and target-domain cities have different spatial structures, resulting in different spatiotemporal distributions of their flow, with the result that the source domain cannot directly assist in the prediction of the target domain.

Previous studies on city flow transfer learning prediction have invariably adopted parameter sharing schemes as shown in Fig. 1. The source and target domains are usually divided into grids or subgraphs of the same size. The grid (Lv et al., 2015; Wang LY et al., 2019) makes it difficult to express the natural non-Euclidean geometric characteristics of flow. The natural solution is to use the graph to model the spatial structure (Huang YJ et al., 2021; Mallick et al., 2021), wherein subgraphs are used to model local information. The above mentioned researches ignore the connectivity of the long-distance road network, leading to the loss of spatial information. In this case, it is difficult to learn the common characteristics of the source and target domains, which will adversely affect the transfer effect.

To solve the challenge, we propose a transfer learning method based on spatiotemporal graph convolution network (TL-STGCN). Considering that the traffic flow is related to the road network and human travel habits, we devise a dynamic spatiotemporal graph convolutional module and a temporal encoder to explore cross-city invariant representations. Specifically, the dynamic spatiotemporal graph convolutional module consists of a dynamic graph convolution layer and two temporal convolution layers. The dynamic graph convolution learns the interactive relationship between the graph nodes to adaptively explore the cross-city similarity of the road network. Then, the temporal convolution and temporal encoder jointly explore the cross-city similarity relationship between human travel habits and city flow. Thus, these concurrent spatiotemporal features are extracted and leveraged as cross-city invariant representations. Because of the differences in road networks, these features belong to different feature spaces. Therefore, we construct a co-occurrence space between the source and target domains. This co-occurrence space is nonlinearly spanned by these concurrent features. The nonlinear spanning is composed of point-wise convolution and a multilayer fully connected network with a nonlinear activation

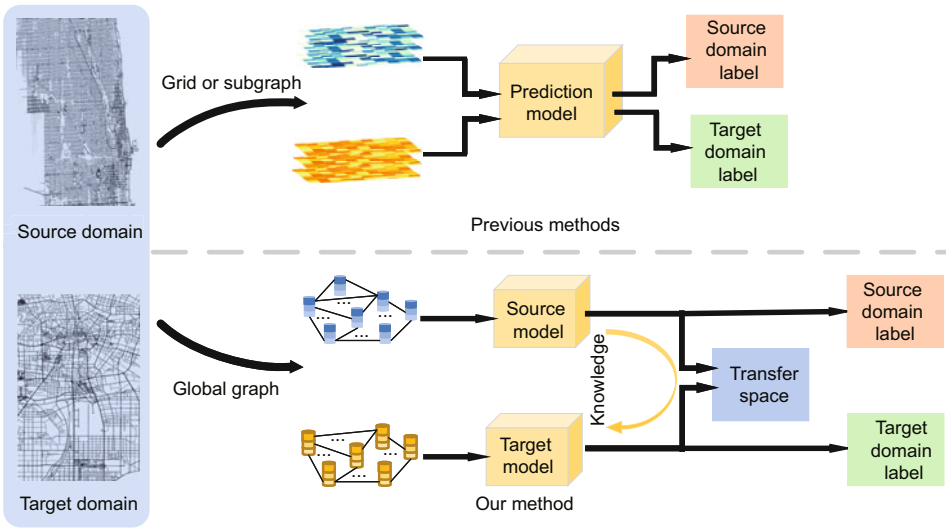

Fig. 1 Different transfer prediction strategies

function LeakyReLU. In this co-occurrence space, we devise Mahalanobis distance (MD) loss measured by a simplified MD (De Maesschalck et al., 2000) to describe the difference in feature distribution between the source and target domains, thereby achieving feature alignment. Through joint optimization of prediction loss and MD loss, knowledge transfer from the source domain to the target domain is achieved.

The main contributions of the proposed method are summarized as follows:

1. We propose TL-STGCN, which achieves knowledge transfer from the source domain city to the target domain city.

2. The dynamic spatiotemporal graph convolution network (STGCN) and temporal encoder are devised to capture the inherent relationship among the road network, human travel habits, and city flow to obtain the shared features of the source and target domains.

3. The proposed method is evaluated on public bike flow datasets in Chicago, Washington, and New York in 2015, and achieves the best results.

2 Related works

2.1 City flow prediction

City flow prediction in smart cities is a concern. The approaches used for carrying out city flow prediction can be categorized into parametric and nonparametric models.

2.1.1 Parametric models

A parametric model refers to a model with certain assumptions which usually has a fixed structure and calculates parameters based on empirical data. Parametric models include Markov random field (Lippi et al., 2010), Markov chain (Huang DR et al., 2017), historical average (HA) (Liu and Guan, 2004), and autoregressive integrated moving average (ARIMA) (Ahmed and Cook, 1979) and its variants (Williams and Hoel, 2003). The most widely used model among the above parameter models is the ARIMA model, which relies on the assumption of stationarity, which is explained as the mean, variance, and autocorrelation not being subject to change. Therefore, the ARIMA model is particularly suitable for stable traffic flow prediction. When there are a large amount of uninterrupted data, the ARIMA model can achieve relatively high accuracy, but the model parameters are not portable. However, in actual situations, data are often missing due to various reasons, and the assumption of stationarity cannot always be satisfied. Under such circumstances, the prediction accuracy of the ARIMA model will be significantly reduced. In addition, the ARIMA model performs forecasting from the perspective of time-series analysis and does not consider the spatial correlation of data.

2.1.2 Nonparametric models

With the rapid development of deep learning, deep neural networks can capture the nonlinear feature of traffic flow, and an increasing number of researchers apply deep learning networks to the prediction of traffic flow. The present research focuses mainly on the extraction of spatiotemporal features. The methods of spatial feature extraction can be divided into the ones based on standard grid data and those based on graph structure data.

To capture the spatial correlation, data are usually constructed into standard grid data (Ma et al., 2017) or graph structure data (Hu et al., 2022; Shao W et al., 2022; Yao et al., 2023). Grid-based methods are used mainly to model spatial correlation by dividing the urban road network into grids. Ma et al. (2017) converted traffic flow into images describing spatiotemporal relationships through a two-dimensional spatiotemporal matrix. The spatial dimension is converted to the vertical axis of the image, and the temporal dimension is converted to the horizontal axis of the image. The graph-based models (Yu et al., 2018; Seng et al., 2021; Wang BW and Wang, 2022) focus on modeling the urban road network as a graph structure to capture spatial correlations. Considering that crowd flow is highly correlated with the origin–destination (OD) location of the flow trajectory, Miao et al. (2023) proposed an adversarial multi-task learning framework based on Bayesian enhancement that can predict crowd flow and flow OD simultaneously. To integrate the local and global spatial features of the flow images and semantic spatiotemporal graphs, a Bayesian-based enhanced heterogeneous spatiotemporal attention network is constructed and mitigates the effects of data uncertainty. Due to the dynamic spatiotemporal relationship between stations in static traffic road networks and historical traffic flows, Peng

et al. (2020) proposed a spatiotemporal correlation dynamic graph neural network framework for urban traffic passenger flow prediction. The dynamic traffic station relationships over time are first modeled as a spatiotemporal correlation dynamic graph structure based on historical traffic passenger flow. Then, a dynamic graph recurrent convolutional neural network is designed to learn the spatiotemporal feature representation of urban transportation network topology and transportation hubs.

2.2 Transfer learning

The core of transfer learning (He et al., 2023) is to find the similarities between domains, to transfer knowledge and solve the problem of lack of data. Transfer learning is divided mainly into three categories: sample-based transfer learning, feature-based transfer learning, and model-based transfer learning. Sample-based transfer learning adopts mainly the method of assigning different weights to different samples for transfer; that is, the greater the similarity of the samples, the higher the weight. The main method of feature-based transfer learning is to transform the features, so that the originally dissimilar features have similar characteristics after transformation. Model-based transfer learning refers to building models with shared parameters.

There is a relative scarcity in the literature dealing with the transfer prediction of city flows based on global graphs. Wang B et al. (2018) proposed transferable traffic deep learning (TT-DL) that first trains the source domain network, uses its weight as the initial weight of the target domain network, and finally uses the target domain data to fine-tune its weight. This method will achieve better results when the source and target domains have high similarity, but this method cannot handle the situation where the source and target domains have different data sizes. Wang LY et al. (2019) proposed RegionTrans. First, the source and target domains were divided into a set of grids, and then the grid matching function was used to match each target city grid with a similar source city grid. Through parameter transfer, the transfer prediction from the source domain to the target domain was realized. Wang SZ et al. (2022) proposed a deep attention adaptive network model called ST-DAAN for transferring cross-domain spatiotemporal knowledge for urban crowd flow prediction. Raw spatiotemporal data from source and target domains were first gridded and then mapped to the common embedding space. Next, domain adaptation was applied on several domain-specific layers to match the average embedding of the two domain distributions. However, these methods do not consider non-Euclidean geometric characteristics.

Existing traffic flow transfer prediction (Li et al., 2020; Huang YJ et al., 2021; Mallick et al., 2021) usually uses subgraphs to model local information, and thus the long-distance road network connectivity is ignored. Our proposed method uses dynamic graph convolution to adaptively extract global information to obtain the shared feature of the road network across cities, and it uses temporal convolution and temporal encoder to extract the shared features of human travel habits. These shared features are nonlinearly spanned to a co-occurrence space, the feature alignment is achieved by MD loss in this co-occurrence space, and the transfer prediction from the source domain to the target domain is achieved.

3 Preliminaries

1. Spatial graph. We construct undirected graphs $G^{\mathrm{s}}=(V^{\mathrm{s}},E^{\mathrm{s}},\boldsymbol{A}^{\mathrm{s}})$ and $G^{\mathrm{t}}=(V^{\mathrm{t}},E^{\mathrm{t}},\boldsymbol{A}^{\mathrm{t}})$ according to the spatial structure of the source and target cities, respectively, where graph nodes V^{s} and V^{t} are the stations of the source and target cities, respectively, $|V^{\mathrm{s}}|=N^{\mathrm{s}}$ and $|V^{\mathrm{t}}|=N^{\mathrm{t}}$, N^{s} and N^{t} are the numbers of stations in the source and target cities, respectively, and E^{s} and E^{t} are the edge sets of the source and target cities, respectively, indicating the connectivity between the stations. Here, any two stations are connected, and $\boldsymbol{A}^{\mathrm{s}}\in\mathbb{R}^{N^{\mathrm{s}}\times N^{\mathrm{s}}}$ and $\boldsymbol{A}^{\mathrm{t}}\in\mathbb{R}^{N^{\mathrm{t}}\times N^{\mathrm{t}}}$ are the adjacency matrices of the source and target cities, respectively.

2. Problem. City flow in each station on the graphs G^{s} and G^{t} is divided according to the time interval Δt, to obtain the time series of each station. $x_t^{\mathrm{s},i}\in\mathbb{R}$ and $x_t^{\mathrm{t},j}\in\mathbb{R}$ represent the traffic flows of the i^{th} and j^{th} stations in the source and target cities, respectively, at time t. $\boldsymbol{X}_t^{\mathrm{s}}=(x_t^{\mathrm{s},1},x_t^{\mathrm{s},2},\dots,x_t^{\mathrm{s},N^{\mathrm{s}}})^{\mathrm{T}}\in\mathbb{R}^{N^{\mathrm{s}}\times 1}$ and $\boldsymbol{X}_t^{\mathrm{t}}=(x_t^{\mathrm{t},1},x_t^{\mathrm{t},2},\dots,x_t^{\mathrm{t},N^{\mathrm{t}}})^{\mathrm{T}}\in\mathbb{R}^{N^{\mathrm{t}}\times 1}$ represent the traffic flows of all nodes in the source and target cities, respectively, at time t. $\boldsymbol{\mathcal{X}}_{T_0}^{\mathrm{s}}=\left(\boldsymbol{X}_1^{\mathrm{s}},\boldsymbol{X}_2^{\mathrm{s}},\dots,\boldsymbol{X}_{T_0}^{\mathrm{s}}\right)^{\mathrm{T}}\in\mathbb{R}^{T_0\times N^{\mathrm{s}}}$ and $\boldsymbol{\mathcal{X}}_{T_1}^{\mathrm{t}}=\left(\boldsymbol{X}_1^{\mathrm{t}},\boldsymbol{X}_2^{\mathrm{t}},\dots,\boldsymbol{X}_{T_1}^{\mathrm{t}}\right)^{\mathrm{T}}\in\mathbb{R}^{T_1\times N^{\mathrm{t}}}$ represent the traffic flows of all nodes on the T_0-length time series of the source city and the T_1-length time series of the target

city, respectively. $\boldsymbol{Y}_t^{\mathrm{s}} = (y_t^{\mathrm{s},1}, y_t^{\mathrm{s},2}, \ldots, y_t^{\mathrm{s},N^{\mathrm{s}}})^{\mathrm{T}} \in \mathbb{R}^{N^{\mathrm{s}}\times 1}$ and $\boldsymbol{Y}_t^{\mathrm{t}} = (y_t^{\mathrm{t},1}, y_t^{\mathrm{t},2}, \ldots, y_t^{\mathrm{t},N^{\mathrm{t}}})^{\mathrm{T}} \in \mathbb{R}^{N^{\mathrm{t}}\times 1}$ represent the traffic flows of all the nodes in the source and target cities at the future time point t, respectively. Given the sufficient city flow $\boldsymbol{\mathcal{X}}_{T_0}^{\mathrm{s}}$ of the source city and the little city flow $\boldsymbol{\mathcal{X}}_{T_1}^{\mathrm{t}}$ of the target city, we are enabled to predict the target city flow $\boldsymbol{\mathcal{Y}}_Q^{\mathrm{t}} = (\boldsymbol{Y}_1^{\mathrm{t}}, \boldsymbol{Y}_2^{\mathrm{t}}, \ldots, \boldsymbol{Y}_Q^{\mathrm{t}}) \in \mathbb{R}^{N^{\mathrm{t}}\times Q}$ at Q time steps in the future.

For example, the source city may last several years (T_0), but the target city may have only a few days (T_1) because it has just begun to be recorded. As shown in Fig. 2, when we want to predict the future flow of the target city, a source city with sufficient data can be introduced to assist the target city prediction.

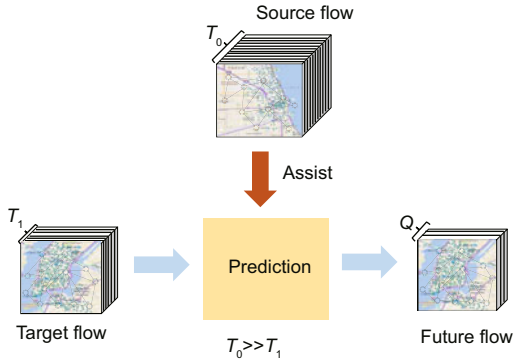

Fig. 2 An example of flow transfer prediction

4 Method

Fig. 3 shows the overall framework of the TL-STGCN model proposed in this paper. It consists of the backbone network and the transfer module. The road network and human travel habits affect mainly the spatiotemporal distribution of the city flow in space and time, respectively. Dynamic STGCN consists of the dynamic adjacency matrix (DAM) and STGCN block to adaptively extract spatiotemporal features. DAM of the traffic flow is calculated to assist the STGCN block to extract spatiotemporal features. The temporal encoder is constructed by a multilayer fully connected layer to extract static human travel time information and fuse it with the spatiotemporal feature. By nonlinear transformation, the shared features between the source and target domains are spanned to a co-occurrence space, and the difference is measured and minimized. Finally, the predictions corresponding to the source and target domains are outputted through the prediction block. The detailed information of each part is shown in Fig. 3.

4.1 Backbone network

The purpose of the backbone network is to simultaneously capture the features of the road network and human travel habits; it describes the features of human travel habits and the road network shared by the source and target domains. Knowledge transfer between the source and target domains can be achieved only when these shared features are extracted.

4.1.1 Road network feature extraction

1. DAM. The static adjacency matrix relies mainly on human experience, and it is difficult to find the cross-domain similarity of the interaction relationship between the graph nodes. Therefore, to explore the similarity of the spatial structure and improve the adaptability of the global spatial dependence information, we use a DAM instead of a static adjacency matrix to construct a graph.

We use parameter matrices $\boldsymbol{W}^{\mathrm{s}} \in \mathbb{R}^{N^{\mathrm{s}}\times N^{\mathrm{s}}}$ and $\boldsymbol{W}^{\mathrm{t}} \in \mathbb{R}^{N^{\mathrm{t}}\times N^{\mathrm{t}}}$ to obtain the symmetric matrix, whose Hadamard product with the initial adjacency matrix is derived. $\boldsymbol{W}^{\mathrm{s}}$ and $\boldsymbol{W}^{\mathrm{t}}$ are learnable parameters that are updated with the data of the source and target domains, respectively. Considering that the elements of $\boldsymbol{W}^{\mathrm{s}}$ and $\boldsymbol{W}^{\mathrm{t}}$ should be non-negative, the nonlinear function σ is used to constrain $\boldsymbol{W}^{\mathrm{s}}$ and $\boldsymbol{W}^{\mathrm{t}}$.

$$\begin{cases} \tilde{\boldsymbol{A}}^{\mathrm{s}} = \boldsymbol{A}^{\mathrm{s}} \odot \dfrac{\sigma\left(\boldsymbol{W}^{\mathrm{s}}\right)+\sigma\left((\boldsymbol{W}^{\mathrm{s}})^{\mathrm{T}}\right)}{2}, \\ \tilde{\boldsymbol{A}}^{\mathrm{t}} = \boldsymbol{A}^{\mathrm{t}} \odot \dfrac{\sigma\left(\boldsymbol{W}^{\mathrm{t}}\right)+\sigma\left((\boldsymbol{W}^{\mathrm{t}})^{\mathrm{T}}\right)}{2}, \end{cases} \tag{1}$$

$$\sigma = \begin{cases} 10, & x \geq 10, \\ 10^{-5}, & x \leq 0, \\ x, & \text{otherwise}, \end{cases} \tag{2}$$

where $\odot$ represents the Hadamard product and x is the input value of σ.

2. Graph convolution. In our research, all station data at each time step are converted into a graph, and the station data at each time step can be considered as features of the corresponding node. We use graph convolution based on the spectrogram theory to perform convolution operation on each graph,

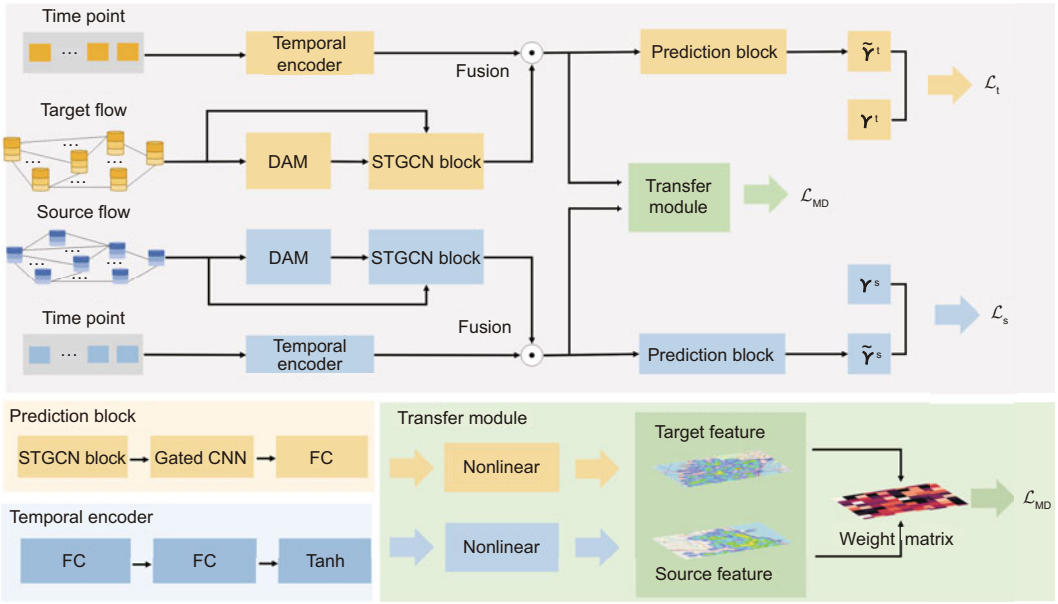

Fig. 3 Structure of the proposed TL-STGCN framework. DAM: calculation of the dynamic adjacency matrix to find similar spatial road network structures across domains; time point: encoding vector of prediction time point; temporal encoder: learning of human travel time information by multilayer fully connected layer; STGCN block: learning the flow spatiotemporal features using the spatial graph structure information; gated convolutional neural network (CNN): extraction of temporal features; FC: fully connected layer; fusion: fusion of the flow spatiotemporal features and time point information

and the operator is defined as $*_G$. In the spectrogram analysis, the corresponding graph is represented by the Laplacian matrix. The graph Laplacian matrices of the source and target domains are defined as $\boldsymbol{L}^{\mathrm{s}} = \boldsymbol{D}^{\mathrm{s}} - \tilde{\boldsymbol{A}}^{\mathrm{s}} \in \mathbb{R}^{N^{\mathrm{s}} \times N^{\mathrm{s}}}$ and $\boldsymbol{L}^{\mathrm{t}} = \boldsymbol{D}^{\mathrm{t}} - \tilde{\boldsymbol{A}}^{\mathrm{t}} \in \mathbb{R}^{N^{\mathrm{t}} \times N^{\mathrm{t}}}$, respectively, and $\boldsymbol{L}^{\mathrm{s}} = \boldsymbol{I}_{N^{\mathrm{s}}} - (\boldsymbol{D}^{\mathrm{s}})^{-\frac{1}{2}} \tilde{\boldsymbol{A}}^{\mathrm{s}} (\boldsymbol{D}^{\mathrm{s}})^{-\frac{1}{2}}$ and $\boldsymbol{L}^{\mathrm{t}} = \boldsymbol{I}_{N^{\mathrm{t}}} - (\boldsymbol{D}^{\mathrm{t}})^{-\frac{1}{2}} \tilde{\boldsymbol{A}}^{\mathrm{t}} (\boldsymbol{D}^{\mathrm{t}})^{-\frac{1}{2}}$ are obtained after standardization. $\tilde{\boldsymbol{A}}^{\mathrm{s}}$ and $\tilde{\boldsymbol{A}}^{\mathrm{t}}$ are dynamic adjacency matrices, $\boldsymbol{I}_{N^{\mathrm{s}}}$ and $\boldsymbol{I}_{N^{\mathrm{t}}}$ are the identity matrices, and the diagonal matrices $\boldsymbol{D}^{\mathrm{s}} \in \mathbb{R}^{N^{\mathrm{t}} \times N^{\mathrm{t}}}$ $(D^{\mathrm{s}}_{ii} = \sum_j \tilde{A}^{\mathrm{s}}_{ij})$ and $\boldsymbol{D}^{\mathrm{t}} \in \mathbb{R}^{N^{\mathrm{t}} \times N^{\mathrm{t}}}$ $(D^{\mathrm{t}}_{ii} = \sum_j \tilde{A}^{\mathrm{t}}_{ij})$ are the degree matrices of the source and target domains, respectively. The eigenvalue decompositions of the Laplacian matrix are $\boldsymbol{L}^{\mathrm{s}} = \boldsymbol{U}^{\mathrm{s}} \boldsymbol{\Lambda}^{\mathrm{s}} (\boldsymbol{U}^{\mathrm{s}})^{\mathrm{T}}$ and $\boldsymbol{L}^{\mathrm{t}} = \boldsymbol{U}^{\mathrm{t}} \boldsymbol{\Lambda}^{\mathrm{t}} (\boldsymbol{U}^{\mathrm{t}})^{\mathrm{T}}$, where $\boldsymbol{\Lambda}^{\mathrm{s}} \in \mathbb{R}^{N^{\mathrm{s}} \times N^{\mathrm{s}}}$ and $\boldsymbol{\Lambda}^{\mathrm{t}} \in \mathbb{R}^{N^{\mathrm{t}} \times N^{\mathrm{t}}}$ are the diagonal matrices composed of eigenvalues, and $\boldsymbol{U}^{\mathrm{s}} \in \mathbb{R}^{N^{\mathrm{s}} \times N^{\mathrm{s}}}$ and $\boldsymbol{U}^{\mathrm{t}} \in \mathbb{R}^{N^{\mathrm{t}} \times N^{\mathrm{t}}}$ are the Fourier bases composed of eigenvectors. Then, the graph convolution of the source-domain and target-domain space-time flows $\boldsymbol{X}^{\mathrm{s}}_t$ and $\boldsymbol{X}^{\mathrm{t}}_t$ at time t is expressed as

$$\begin{cases} \boldsymbol{g}^{\mathrm{s}}_\theta *_G \boldsymbol{X}^{\mathrm{s}}_t = \boldsymbol{g}^{\mathrm{s}}_\theta (\boldsymbol{L}^{\mathrm{s}}) \boldsymbol{X}^{\mathrm{s}}_t = \boldsymbol{U}^{\mathrm{s}} \boldsymbol{g}^{\mathrm{s}}_\theta (\boldsymbol{\Lambda}^{\mathrm{s}}) (\boldsymbol{U}^{\mathrm{s}})^{\mathrm{T}} \boldsymbol{X}^{\mathrm{s}}_t, \\ \boldsymbol{g}^{\mathrm{t}}_\theta *_G \boldsymbol{X}^{\mathrm{t}}_t = \boldsymbol{g}^{\mathrm{t}}_\theta (\boldsymbol{L}^{\mathrm{t}}) \boldsymbol{X}^{\mathrm{t}}_t = \boldsymbol{U}^{\mathrm{t}} \boldsymbol{g}^{\mathrm{t}}_\theta (\boldsymbol{\Lambda}^{\mathrm{t}}) (\boldsymbol{U}^{\mathrm{t}})^{\mathrm{T}} \boldsymbol{X}^{\mathrm{t}}_t, \end{cases} \tag{3}$$

where $\boldsymbol{g}^{\mathrm{s}}_\theta$ and $\boldsymbol{g}^{\mathrm{t}}_\theta$ are the convolution kernels of the source and target domains, respectively. Due to the high computational complexity of this convolution, the Chebyshev polynomial is used to approximate this problem:

$$\begin{cases} \boldsymbol{g}^{\mathrm{s}}_\theta *_G \boldsymbol{X}^{\mathrm{s}}_t = \boldsymbol{g}^{\mathrm{s}}_\theta (\boldsymbol{L}^{\mathrm{s}}) \boldsymbol{X}^{\mathrm{s}}_t \approx \sum_{k=0}^{K-1} \theta^{\mathrm{s}}_k \boldsymbol{T}_k (\tilde{\boldsymbol{L}}^{\mathrm{s}}) \boldsymbol{X}^{\mathrm{s}}_t, \\ \boldsymbol{g}^{\mathrm{t}}_\theta *_G \boldsymbol{X}^{\mathrm{t}}_t = \boldsymbol{g}^{\mathrm{t}}_\theta (\boldsymbol{L}^{\mathrm{t}}) \boldsymbol{X}^{\mathrm{t}}_t \approx \sum_{k=0}^{K-1} \theta^{\mathrm{t}}_k \boldsymbol{T}_k (\tilde{\boldsymbol{L}}^{\mathrm{t}}) \boldsymbol{X}^{\mathrm{t}}_t, \end{cases} \tag{4}$$

where $\boldsymbol{T}_k(\tilde{\boldsymbol{L}}^{\mathrm{s}}) \in \mathbb{R}^{N^{\mathrm{s}} \times N^{\mathrm{s}}}$ and $\boldsymbol{T}_k(\tilde{\boldsymbol{L}}^{\mathrm{t}}) \in \mathbb{R}^{N^{\mathrm{t}} \times N^{\mathrm{t}}}$ are the k^{th} Chebyshev polynomials calculated by the scaled Laplacian $\tilde{\boldsymbol{L}}^{\mathrm{s}} = \frac{2}{\lambda^{\mathrm{s}}_{\max}} \boldsymbol{L}^{\mathrm{s}} - \boldsymbol{I}_{N^{\mathrm{s}}}$ and $\tilde{\boldsymbol{L}}^{\mathrm{t}} = \frac{2}{\lambda^{\mathrm{t}}_{\max}} \boldsymbol{L}^{\mathrm{t}} - \boldsymbol{I}_{N^{\mathrm{t}}}$, respectively. $\lambda^{\mathrm{s}}_{\max}$ and $\lambda^{\mathrm{t}}_{\max}$ are the maximum eigenvalues of the source-domain and target-domain Laplacian matrices, respectively. θ^{s}_k and θ^{t}_k are the learnable parameters of the source and target domains, respectively.

4.1.2 Human travel habit feature extraction

After the spatial road network features are obtained through the graph convolution operation, the human travel habit features need to be extracted. To better learn the relationship between human travel habits and flow, it is necessary to learn the dynamic and static features of traffic flow, that is, the trend

of traffic flow changing over time and the impact of time points on traffic flow. As shown in Fig. 4, flow peaks are often at 8:00 a.m. and 5:00 p.m., while the flow at midnight is often low. It can be seen that time points and flow are also strongly correlated.

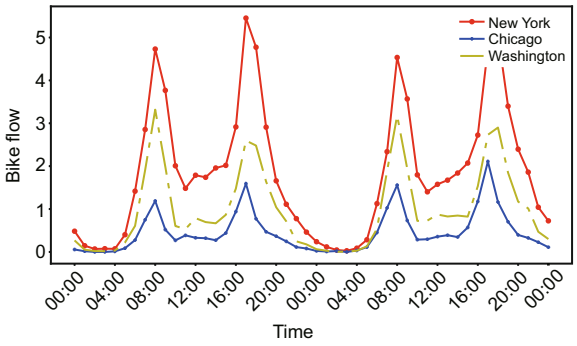

Fig. 4 Changes in bike flow in three cities within 48 h

1. Dynamic flow feature. In the time dimension, one-dimensional (1D) convolution is used to capture the dynamic flow feature, and its kernel width is K_t. Each graph node can be regarded as a time sequence with length T_0 and the number of input channels C_i. We use the gated CNNs to extract the dynamic flow feature $\boldsymbol{F}_{T_\mathrm{d}} \in \mathbb{R}^{(T_0-K_\mathrm{t}+1)\times C_\mathrm{o}}$, where C_o is the number of output channels.

The STGCN block consists of two gated CNN layers and a spatial graph convolutional layer. The spatial graph convolutional layer is wrapped by two temporal convolutional layers to form an STGCN block similar to a sandwich structure. The STGCN block can capture the spatial temporal feature of flow. The r^{th} layer of the STGCN block can be described as follows:

$$\boldsymbol{\mathcal{X}}^{(r)} = f_{\mathrm{st}}^{(r)}(\boldsymbol{\mathcal{X}}^{(r-1)}) \in \mathbb{R}^{N\times C_\mathrm{o}^{(r)}\times(T_0-2r(K_\mathrm{t}-1))}, \quad (5)$$

where $\boldsymbol{\mathcal{X}}^{(r-1)} \in \mathbb{R}^{N\times C_\mathrm{o}^{(r-1)}\times(T_0-2(r-1)(K_\mathrm{t}-1))}$ represents the $(r-1)^{\mathrm{th}}$ layer. In the input of the STGCN block, $f_{\mathrm{st}}^{(r)}$ represents the abstract function of the STGCN block of the r^{th} layer.

2. Static flow feature. To obtain the static flow feature, that is, the impact of time point on flow, specifically, the time point is encoded to generate 1D time vectors $\boldsymbol{T}_\mathrm{p}^\mathrm{s}$ and $\boldsymbol{T}_\mathrm{p}^\mathrm{t}$ with a length of 24; then, the time vectors $\boldsymbol{T}_\mathrm{p}^\mathrm{s}$ and $\boldsymbol{T}_\mathrm{p}^\mathrm{t}$ serve as the input of the temporal encoder to obtain the static flow feature $\boldsymbol{F}_{T_\mathrm{p}}$. The input dimension of the fully connected network is 24; the output dimensions for the source and target domains are N^s and N^t, respectively.

$$\begin{cases} \boldsymbol{F}_{T_\mathrm{p}}^\mathrm{s} = f(\boldsymbol{W}_{T_\mathrm{p}}^\mathrm{s}\boldsymbol{T}_\mathrm{p}^\mathrm{s} + \boldsymbol{b}_{T_\mathrm{p}}^\mathrm{s}), \\ \boldsymbol{F}_{T_\mathrm{p}}^\mathrm{t} = f(\boldsymbol{W}_{T_\mathrm{p}}^\mathrm{t}\boldsymbol{T}_\mathrm{p}^\mathrm{t} + \boldsymbol{b}_{T_\mathrm{p}}^\mathrm{t}), \end{cases} \quad (6)$$

where $\boldsymbol{W}_{T_\mathrm{p}}^\mathrm{s}$, $\boldsymbol{b}_{T_\mathrm{p}}^\mathrm{s}$ and $\boldsymbol{W}_{T_\mathrm{p}}^\mathrm{t}$, $\boldsymbol{b}_{T_\mathrm{p}}^\mathrm{t}$ are the learnable parameters of the temporal encoder in the source and target domains, respectively, f represents the nonlinear activation function, and we choose tanh in the present study.

Finally, the static flow feature is fused with the spatiotemporal feature of flow to obtain the influence of the time point on the flow:

$$\begin{cases} \tilde{\boldsymbol{\mathcal{X}}}_t^{\mathrm{s},(1)} = \boldsymbol{F}_{T_\mathrm{p}}^\mathrm{s} \odot \boldsymbol{\mathcal{X}}_t^{\mathrm{s},(1)} + \boldsymbol{\mathcal{X}}_t^{\mathrm{s},(1)}, \\ \tilde{\boldsymbol{\mathcal{X}}}_t^{\mathrm{t},(1)} = \boldsymbol{F}_{T_\mathrm{p}}^\mathrm{t} \odot \boldsymbol{\mathcal{X}}_t^{\mathrm{t},(1)} + \boldsymbol{\mathcal{X}}_t^{\mathrm{t},(1)}, \end{cases} \quad (7)$$

where $\boldsymbol{\mathcal{X}}_t^{\mathrm{s},(1)}$ and $\boldsymbol{\mathcal{X}}_t^{\mathrm{t},(1)}$ represent the features extracted from the first layer of the source and target domains, respectively. We replace the input of the 2^{nd} STGCN block from $\boldsymbol{\mathcal{X}}_t^{\mathrm{s},(1)}$ and $\boldsymbol{\mathcal{X}}_t^{\mathrm{t},(1)}$ with $\tilde{\boldsymbol{\mathcal{X}}}_t^{\mathrm{s},(1)}$ and $\tilde{\boldsymbol{\mathcal{X}}}_t^{\mathrm{t},(1)}$ to simulate the impact of time on flow.

4.2 Transfer module

In the above, the backbone network captures the inherent relationship among the road network, human travel habits, and traffic flow. Next, the main difficulty lies in ascertaining the means of achieving feature alignment.

1. Nonlinear transformation. For the transfer module with non-shared parameters, the numbers of convolution channels of the source and target models may be inconsistent. To make our method more general, we use point-wise convolution r_θ to linearly transform the target domain $r_\theta(\boldsymbol{\mathcal{X}}^{\mathrm{t},(r)})$ to keep it consistent with that of the source domain.

Considering that the source and target domains have different road network structures, we ensure that the source domain spatiotemporal feature $\boldsymbol{\mathcal{X}}^{\mathrm{s},(r)} \in \mathbb{R}^{N^\mathrm{s}\times C_\mathrm{o}^{\mathrm{s},(r)}\times(T_0-2r(K_\mathrm{t}-1))}$ and the target domain spatiotemporal feature $\boldsymbol{\mathcal{X}}^{\mathrm{t},(r)} \in \mathbb{R}^{N^\mathrm{t}\times C_\mathrm{o}^{\mathrm{t},(r)}\times(T_0-2r(K_\mathrm{t}-1))}$ belong to different feature spaces. These features between the source and target domains are nonlinearly spanned to a co-occurrence space $\mathbb{S}$; thereafter, we describe the difference in this co-occurrence space.

In this research, we use a fully connected network with nonlinear activation function LeakyReLU and batch normalization (BN) layer to achieve

nonlinear transformation. Each fully connected layer is followed by a nonlinear activation function LeakyReLU, and the last fully connected layer is followed by a BN layer to standardize the output. It can be written as follows:

$$\begin{cases} \boldsymbol{H}^{\mathrm{s}} = \mathrm{BN}(\boldsymbol{W}_{\mathrm{n}}^{\mathrm{s},(2)} f(\boldsymbol{W}_{\mathrm{n}}^{\mathrm{s},(1)} \tilde{\boldsymbol{\mathcal{X}}}_t^{\mathrm{s},(1)} + \boldsymbol{b}_{\mathrm{n}}^{\mathrm{s},(1)}) \\ \qquad + \boldsymbol{b}_{\mathrm{n}}^{\mathrm{s},(2)}), \\ \boldsymbol{H}^{\mathrm{t}} = \mathrm{BN}(\boldsymbol{W}_{\mathrm{n}}^{\mathrm{t},(2)} f(\boldsymbol{W}_{\mathrm{n}}^{\mathrm{t},(1)} r_\theta(\tilde{\boldsymbol{\mathcal{X}}}_t^{\mathrm{t},(1)}) + \boldsymbol{b}_{\mathrm{n}}^{\mathrm{t},(1)}) \\ \qquad + \boldsymbol{b}_{\mathrm{n}}^{\mathrm{t},(2)}), \end{cases} \tag{8}$$

where $\boldsymbol{W}_{\mathrm{n}}^{\mathrm{s},(1)}$, $\boldsymbol{b}_{\mathrm{n}}^{\mathrm{s},(1)}$ and $\boldsymbol{W}_{\mathrm{n}}^{\mathrm{t},(1)}$, $\boldsymbol{b}_{\mathrm{n}}^{\mathrm{t},(1)}$ are the learnable parameters of the first layer of the source and target domains, respectively, and $\boldsymbol{W}_{\mathrm{n}}^{\mathrm{s},(2)}$, $\boldsymbol{b}_{\mathrm{n}}^{\mathrm{s},(2)}$ and $\boldsymbol{W}_{\mathrm{n}}^{\mathrm{t},(2)}$, $\boldsymbol{b}_{\mathrm{n}}^{\mathrm{t},(2)}$ are the learnable parameters of the second layer of the source and target domains, respectively. The output dimension dim of the fully connected network is a hyperparameter, and finally the source domain feature $\boldsymbol{H}^{\mathrm{s}} \in \mathbb{R}^{\mathrm{dim} \times C_{\mathrm{o}}^{\mathrm{s},(1)} \times (T_0 - 2(K_{\mathrm{t}}-1))}$ and the target domain feature $\boldsymbol{H}^{\mathrm{t}} \in \mathbb{R}^{\mathrm{dim} \times C_{\mathrm{o}}^{\mathrm{t},(1)} \times (T_0 - 2(K_{\mathrm{t}}-1))}$ in the same feature space $\mathbb{S}$ are obtained.

2. Domain adaption. To characterize the difference between the source domain $\boldsymbol{H}^{\mathrm{s}}$ and the target domain $\boldsymbol{H}^{\mathrm{t}}$, we design MD loss which uses the simplified MD (De Maesschalck et al., 2000; Xiang et al., 2008) as the metric. The weight matrix $\boldsymbol{M}' \in \mathbb{R}^{\mathrm{dim} \times \mathrm{dim}}$ is obtained after normalization of the diagonal matrix of learnable parameters $\boldsymbol{M} \in \mathbb{R}^{\mathrm{dim} \times \mathrm{dim}}$. Due to the uneven spatial distribution of urban flows, different nodes contribute differently to transfer outcomes. After mapping the source and target cities to a co-occurrence space, the learnable parameter matrix $\boldsymbol{M}'$ can dynamically focus on dimensions that have a greater impact on distribution differences and improve the transfer prediction.

$$M'_{ij} = \begin{cases} 0, & i \neq j, \\ \frac{\exp(M_{ij})}{\sum_{i=1}^{\mathrm{dim}} \exp(M_{ij})}, & i = j. \end{cases} \tag{9}$$

The MD loss can be written as

$$\begin{aligned} \mathcal{L}_{\mathrm{MD}} = & \frac{1}{NT} \sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{c=1}^{C} \left(\boldsymbol{H}_{n,t,c}^{\mathrm{s}} - \boldsymbol{H}_{n,t,c}^{\mathrm{t}}\right) \\ & \cdot \boldsymbol{M}' \left(\boldsymbol{H}_{n,t,c}^{\mathrm{s}} - \boldsymbol{H}_{n,t,c}^{\mathrm{t}}\right)^{\mathrm{T}}, \end{aligned} \tag{10}$$

where N represents the number of samples, T represents the number of time steps, and C represents the number of convolution channels. By minimizing the transfer loss $\mathcal{L}_{\mathrm{MD}}$, the difference in feature distribution between the source and target domains is reduced, to achieve knowledge transfer between the source and target domains.

The total loss function of the model can be written as

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{t}}(\tilde{\boldsymbol{\Upsilon}}_Q^{\mathrm{t}}, \boldsymbol{\Upsilon}_Q^{\mathrm{t}}) + \alpha \mathcal{L}_{\mathrm{s}}(\tilde{\boldsymbol{\Upsilon}}_Q^{\mathrm{s}}, \boldsymbol{\Upsilon}_Q^{\mathrm{s}}) + \beta \mathcal{L}_{\mathrm{MD}} + \gamma \|\theta\|_2, \tag{11}$$

where α, β, and γ are balance coefficients, θ is the parameter set of the model, $\|\theta\|_2$ represents the complexity of the model, and $\|\cdot\|_2$ represents the 2-norm.

$\mathcal{L}_{\mathrm{t}}$ and $\mathcal{L}_{\mathrm{s}}$ are the regression loss of the source and target domains, respectively; they are formalized as follows:

$$\mathcal{L}_{\mathrm{t}}(\tilde{\boldsymbol{\Upsilon}}_Q^{\mathrm{t}}, \boldsymbol{\Upsilon}_Q^{\mathrm{t}}) = \frac{1}{N} \sum_{n=1}^{N} (\tilde{\boldsymbol{\Upsilon}}_{Q,n}^{\mathrm{t}} - \boldsymbol{\Upsilon}_{Q,n}^{\mathrm{t}})^2, \tag{12}$$

$$\mathcal{L}_{\mathrm{s}}(\tilde{\boldsymbol{\Upsilon}}_Q^{\mathrm{s}}, \boldsymbol{\Upsilon}_Q^{\mathrm{s}}) = \frac{1}{N} \sum_{n=1}^{N} (\tilde{\boldsymbol{\Upsilon}}_{Q,n}^{\mathrm{s}} - \boldsymbol{\Upsilon}_{Q,n}^{\mathrm{s}})^2, \tag{13}$$

where $\tilde{\boldsymbol{\Upsilon}}_{Q,n}^{\mathrm{t}}$ and $\tilde{\boldsymbol{\Upsilon}}_{Q,n}^{\mathrm{s}}$ are the model outputs of the source and target domains, respectively.

5 Experiments

5.1 Setting

5.1.1 Dataset

We use bike flow data in three cities New York, Chicago, and Washington, for the experiment. We use N, C, and W to refer to these three cities. In all datasets, we first delete some stations with missing latitude and longitude, and then use Z-score standardization to process the data. We take bike stations as graph nodes and the reciprocal of the distance between stations as edge weight to construct a spatial graph. To evaluate the effect of the model, we choose one city as the source domain and the two other cities as the target domain for transfer, so there are six transfer scenarios.

Details of the datasets are shown as follows: The numbers of stations are 300, 331, and 283 in Chicago, New York, and Washington, respectively. The data time range for all the three cities is from Jan. 1, 2015 to Mar. 31, 2015. The data length for each city is 2160 timesteps. The time intervals are all 1 h. The source domain uses 80% of the data (i.e., data from Jan. 1, 2015 to Mar. 13, 2015) as the training data.

The target domain uses 8% of the data (i.e., data from Mar. 7, 2015 to Mar. 13, 2015) or 16% (i.e., data from Feb. 28, 2015 to Mar. 13, 2015) as the training data. Twenty percent of the data, i.e., from Mar. 14, 2015 to Mar. 31, 2015, are used as the test data. The input timestep is 12, and the output timestep is 1.

5.1.2 Metrics

The evaluation metrics are root mean square error (RMSE) and mean absolute error (MAE). Considering that the Z-score normalization process is used for data during network training, the model output is first unnormalized when the evaluation metrics are calculated.

$$\mathrm{RMSE}=\sqrt{\frac{1}{N}\sum_{n=1}^{N}(\tilde{\boldsymbol{\Upsilon}}_{Q,n}^{\mathrm{t}}-\boldsymbol{\Upsilon}_{Q,n}^{\mathrm{t}})^2}, \tag{14}$$

$$\mathrm{MAE}=\frac{1}{N}\sum_{n=1}^{N}\left|\tilde{\boldsymbol{\Upsilon}}_{Q,n}^{\mathrm{t}}-\boldsymbol{\Upsilon}_{Q,n}^{\mathrm{t}}\right|. \tag{15}$$

5.1.3 Hyperparameters

The batch size is 50, the epoch number is 100, and the adaptive moment estimation optimizer (ADAM) is selected as the optimization algorithm. The learning rate is 0.001 and the learning rate of matrix $\boldsymbol{M}$ is 0.005, and γ is 0.001. For the hyperparameters a, β, and dim, the optimal hyperparameters are obtained by the grid method as shown in Table 1. C→N indicates that the source and target domains are Chicago and New York, respectively.

Table 1 Experimental hyperparameter settings

Parameter	C→N	W→N	W→C	N→C	N→W	C→W
a	0.05	0.50	1.50	0.25	0.25	1.50
β	1.00	0.01	0.05	0.75	1.00	0.05
dim	250	220	250	250	250	200

C: Chicago; N: New York; W: Washington

5.1.4 Network implementation

Details of the network structure are presented in Table 2. The floating point operations per second (FLOPs) of TL-STGCN is 411 673 328 and the number of model parameters is 599 988 when dimension is 250.

5.1.5 Baselines

To compare the performance of the model discussed in the present research with that of TL-STGCN, we choose the following 10 baselines. All baseline methods are trained using the same target domain data as the transfer method, i.e., 8% or 16%. Non-transfer methods use only target domain data to make predictions on the target domain.

HA (Liu and Guan, 2004): The average of historical data is used to predict future traffic flow.

SVR (Smola and Schölkopf, 2004): The training dataset is used to train the support vector regression model, a linear kernel is used, and the penalty term is 0.001.

ARIMA (Ahmed and Cook, 1979): Parameter models are used to fit historical data to predict future flow, where p=1, d=0, and q=0.

STGCN (Yu et al., 2018): The model has two spatiotemporal feature extraction modules, which separately extract spatial features of urban flow using spectral convolution and temporal features of urban flow using 1D convolution. The spatiotemporal feature extraction module has 32 channels. The learning rate of the model is 0.001.

T-GCN (Zhao et al., 2020): This is a flow-prediction method combining graph convolution and gated recurrent unit (GRU). The model uses 64 GRUs with a learning rate of 0.001.

ASTGCN (Guo et al., 2019): This is a flow-prediction method that introduces a spatiotemporal attention mechanism based on STGCN. Considering the lack of training data, ASTGCN is pruned and

Table 2 Experimental network details

Network layer	Source	Target
STGCN block_1	Output channel number=64	Output channel number=32
	Spatial channel number=32	Spatial channel number=16
STGCN block_2	Output channel number=64	Output channel number=32
	Spatial channel number=32	Spatial channel number=16
Gated CNN	Output channel number=32	Output channel number=32
FC	Input dimension=64, output dimension=1	Input dimension=64, output dimension=1

only the recent branch is retained. The batch size is 64 with a learning rate of 0.001.

STID (Shao ZZ et al., 2022): This is a simple multilayer perceptron (MLP) with additional spatial and temporal identity information; the batch size is 32 with a learning rate of 0.001.

STMGAT (Wang BW and Wang, 2022): This is a spatiotemporal multihead graph attention network that captures spatial dimensional features using a multihead attention graph network. The batch size is 64 with a learning rate of 0.001.

RegionTrans (Wang LY et al., 2019): This is a spatiotemporal transfer prediction method based on grid division.

STDAAN (Wang SZ et al., 2022): This is a deep attention adaptive network model based on grid and maximum mean difference for cross-city crowd flow prediction; the batch size is 32 with a learning rate of 0.0001.

Table 3 Experimental results of MAE and RMSE of the non-transfer baselines using 8% or 16% of the target domain data

Method	Metric	N		C		W	
		8%	16%	8%	16%	8%	16%
HA	MAE	3.01	3.01	1.24	1.24	1.99	1.99
	RMSE	4.86	4.86	2.45	2.45	3.88	3.88
ARIMA	MAE	3.00	2.79	1.29	1.16	1.83	1.73
	RMSE	3.86	3.91	1.83	1.84	2.57	2.62
SVR	MAE	1.98	1.94	0.93	0.90	1.37	1.37
	RMSE	3.46	3.52	1.94	1.97	2.81	2.85
STGCN	MAE	1.55	1.52	0.80	0.77	1.05	1.03
	RMSE	2.71	2.64	1.73	1.67	2.20	2.13
T-GCN	MAE	2.03	1.98	0.97	0.92	1.49	1.43
	RMSE	3.61	3.60	2.02	1.99	3.07	3.00
ASTGCN	MAE	1.79	1.70	0.90	0.88	1.18	1.12
	RMSE	2.95	2.89	1.84	1.88	2.36	2.31
STMGAT	MAE	1.64	1.61	0.82	0.81	1.15	1.14
	RMSE	2.89	2.85	1.76	1.73	2.44	2.41
STID	MAE	1.59	1.55	0.83	0.76	1.11	1.07
	RMSE	2.72	2.70	1.68	1.60	2.34	2.25

C: Chicago; N: New York; W: Washington

5.2 Results

5.2.1 Comparison with existing methods

Tables 3 and 4 show the comparison of MAE and RMSE between TL-STGCN and baselines. Tables 3 and 4 show the results of the non-transfer baselines and the transfer baselines, respectively. Compared with non-transfer methods such as STGCN, TL-STGCN can use rich-data source domain to learn the common spatiotemporal feature, thereby effectively overcoming the adverse effects caused by data shortage.

Compared with transfer methods such as RegionTrans and STDAAN, TL-STGCN uses graph data to model the spatial structure of the flow, fully considers the non-Euclidean geometric characteristics of the urban spatial structure, and retains the spatial information of the flow to the maximum extent, which is conducive to the extraction of common features between the source and target domains. At the same time, we can see that the source domain also has a certain impact on the transfer effect of the target domain. In summary, it proves the validity of TL-STGCN for spatialtemporal flow prediction when the amount of data is small.

Fig. 5 shows the prediction curve of bike flow from Mar. 15, 2015 to Mar. 16, 2015 with our method and comparative methods using only 16% of the target domain training data. Compared to other methods, our method fits the ground truth, especially the peak flow, much better. This shows that our method has learned the knowledge transferred from the source domain to the target domain, thereby improving the prediction accuracy.

Table 4 Experimental results of MAE and RMSE of the transfer baselines using 8% or 16% of the target domain data

Method	Metric	C→N		W→N		W→C		N→C		N→W		C→W	
		8%	16%	8%	16%	8%	16%	8%	16%	8%	16%	8%	16%
RegionTrans	MAE	1.65	1.62	1.64	1.60	0.88	0.84	0.94	0.86	1.32	1.25	1.37	1.31
	RMSE	2.98	2.84	2.95	2.87	1.93	1.82	1.96	1.79	3.08	2.94	3.10	2.91
STDAAN	MAE	1.62	1.61	1.63	1.62	0.82	0.81	0.82	0.81	1.20	1.19	1.21	1.20
	RMSE	2.81	2.79	2.78	2.69	1.76	1.74	1.75	1.74	2.83	2.74	2.88	2.77
TL-STGCN	MAE	**1.49**	**1.44**	**1.50**	**1.45**	**0.76**	**0.74**	**0.76**	**0.74**	**1.02**	**1.01**	**1.01**	**0.99**
	RMSE	**2.55**	**2.50**	**2.59**	**2.53**	**1.65**	**1.57**	**1.63**	**1.59**	**2.13**	**2.09**	**2.14**	**2.09**

C: Chicago; N: New York; W: Washington. Best values are in bold

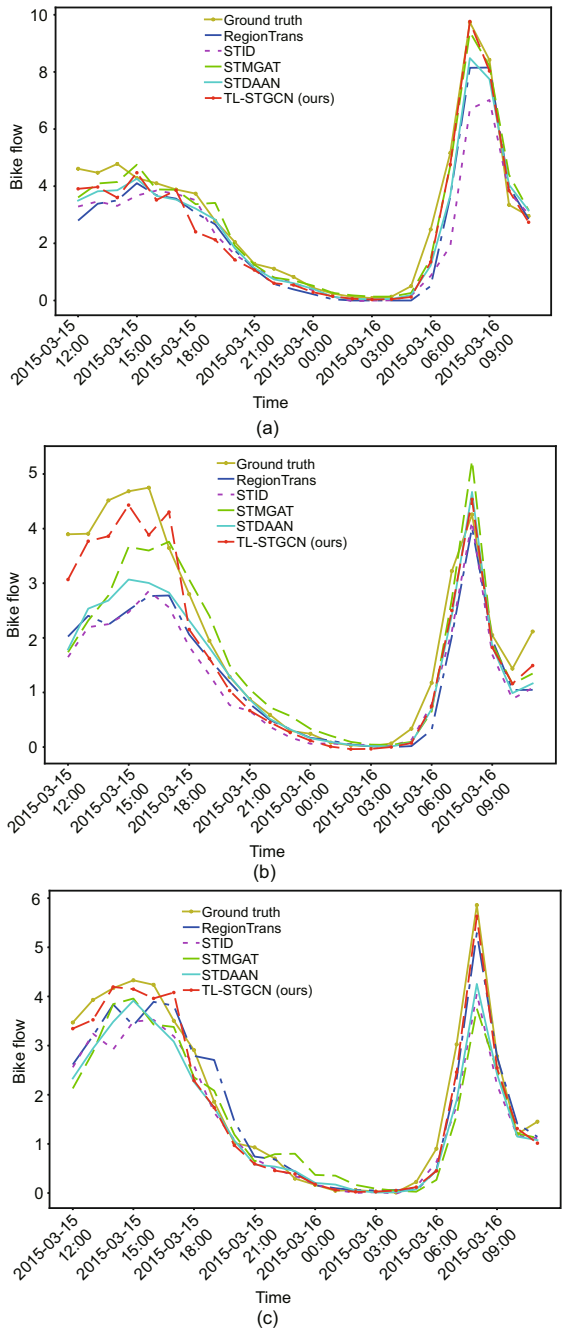

Fig. 5 Prediction of bike flow from Mar. 15, 2015 to Mar. 16, 2015 when using 16% of the target training data: (a) from Chicago to New York; (b) from Washington to Chicago; (c) from Chicago to Washington

Fig. 6 Heatmap of bike flow prediction in a day from Washington to Chicago at 0:00 a.m. (a), 8:00 a.m. (b), 12:00 a.m. (c), from Chicago to Washington at 0:00 a.m. (d), 8:00 a.m. (e), 12:00 a.m. (f), and from Chicago to New York at 0:00 a.m. (g), 8:00 a.m. (h), 12:00 a.m. (i)

Fig. 6 shows a visualization of the change in bike flow throughout the day. For New York, Chicago, and Washington, the bike flow is concentrated in the city center or near the transportation hub at 0:00 a.m., and the flow is low, which is in line with people's daily behavior habits. At 8:00 a.m., people start their day's work, leading to a significant increase in bike flow in suburbs and urban areas. At 12:00 noon, the flow of bikes is relatively reduced. This is because the short break at noon makes it difficult to travel by bike. These results prove that TL-STGCN uses only 16% of the target domain training data to capture the spatialtemporal changes of bike flow.

5.2.2 Ablation studies

To explore in depth how our method functions, the following six variants are designed to facilitate comparison with our method:

BackBone: It is a spatiotemporal prediction method that introduces DAM and time point information based on STGCN.

Finetuning: It involves using the source domain data to train BackBone and using the trained parameters as the initialization parameters of the target domain model, together with the use of zero padding to ensure that the same size is realized for source and

target domains' data.

TL-Local: It involves division of the road network into subgraphs that match the local structure of the source and target domains to minimize distributional differences and enable the performance of transfer prediction.

TL-DAM: It involves removing the DAM and keeping the adjacency matrix constructed based on geographic correlation, with other parts remaining unchanged.

TL-Gated CNN: It involves the use of 1D CNN instead of gated CNN, with the rest of the parameters and structure remaining unchanged.

TL-Loss: It involves replacing the learnable matrix in the MD loss with the unit matrix, while keeping the rest unchanged.

From Tables 5 and 6, we ascertain that compared to BackBone, our method effectively uses data from the source domain. Compared with Finetuning, which uses the source domain model parameters as the initialization parameters of the target domain model for training, TL-STGCN can explicitly measure the difference between the source and target domains, thereby performing targeted optimization and achieving superior results. By comparing TL-Local and TL-STGCN, we can find that the interaction between local regions is important for the transfer effect. This is because the city exists as a whole, which illustrates the importance of global information. Our method considers the impact of road network structure on transfer as a whole, which helps for learning of more general features. Compared with TL-DAM, we can find that DAM helps discover the similar site spatial dependencies of the source and target cities. The experimental results demonstrate that, in comparison with TL-Gated CNN as well as the simple 1D CNN, the gated CNN structure demonstrates greater effectiveness in extracting the temporal features of city flow. Compared with TL-Loss, TL-STGCN can achieve certain advantages because MD loss based on learnable matrices can automatically focus on the dimensions that have a greater impact on the transfer effect, whereas unit matrix can only treat each dimension equally.

Table 5 Ablation results of MAE and RMSE of the non-transfer baseline using 8% or 16% of the target domain data

Method	Metric	N		C		W	
		8%	16%	8%	16%	8%	16%
BackBone	MAE	1.65	1.70	0.83	0.84	1.13	1.10
	RMSE	2.95	3.20	1.83	1.85	2.40	2.39

C: Chicago; N: New York; W: Washington

5.2.3 Data size sensitivity analysis

We test the robustness of TL-STGCN by changing the percentage of training data. When the percentage of training data in the target domain is 8% from Chicago to New York, we change the percentage of training data in the source domain during transfer. It can be seen from Figs. 7 and 8 that when the percentage of source domain training data is higher than 30%, the model performance is stable. Therefore, our method has relatively good robustness to different percentages of source domain training data.

Table 6 Ablation results of MAE and RMSE of the transfer baselines using 8% or 16% of the target domain data

Method	Metric	C→N		W→N		W→C		N→C		N→W		C→W	
		8%	16%	8%	16%	8%	16%	8%	16%	8%	16%	8%	16%
Finetuning	MAE	1.75	1.73	1.77	1.74	0.87	0.87	0.87	0.87	1.21	1.19	1.21	1.18
	RMSE	3.18	3.14	3.17	3.15	1.83	1.91	1.95	1.92	2.66	2.56	2.64	2.61
TL-Local	MAE	1.75	1.65	1.84	1.68	0.90	0.85	0.90	0.82	1.17	1.11	1.23	1.12
	RMSE	3.13	2.87	3.29	3.00	1.90	1.80	1.85	1.72	2.48	2.36	2.52	2.36
TL-DAM	MAE	1.50	1.49	1.52	1.50	0.78	0.75	0.77	0.77	1.04	1.03	1.04	1.02
	RMSE	2.62	2.61	2.61	2.63	1.69	1.66	1.66	1.65	2.18	2.17	2.20	2.15
TL-Gated CNN	MAE	1.53	1.48	1.53	1.46	0.82	0.78	0.84	0.78	1.06	1.02	1.06	1.02
	RMSE	2.58	**2.50**	2.60	**2.50**	1.69	1.61	1.70	**1.58**	**2.13**	**2.09**	**2.13**	**2.08**
TL-Loss	MAE	1.50	1.45	1.51	**1.45**	0.79	**0.74**	0.77	**0.74**	1.03	1.02	1.02	1.00
	RMSE	2.57	**2.50**	**2.59**	2.53	1.66	**1.57**	1.65	1.59	**2.13**	2.10	2.15	2.09
TL-STGCN	MAE	**1.49**	**1.44**	**1.50**	**1.45**	**0.76**	**0.74**	**0.76**	**0.74**	**1.02**	**1.01**	**1.01**	**0.99**
	RMSE	**2.55**	**2.50**	**2.59**	2.53	**1.65**	**1.57**	**1.63**	1.59	**2.13**	**2.09**	2.14	2.09

C: Chicago; N: New York; W: Washington. Best values are in bold

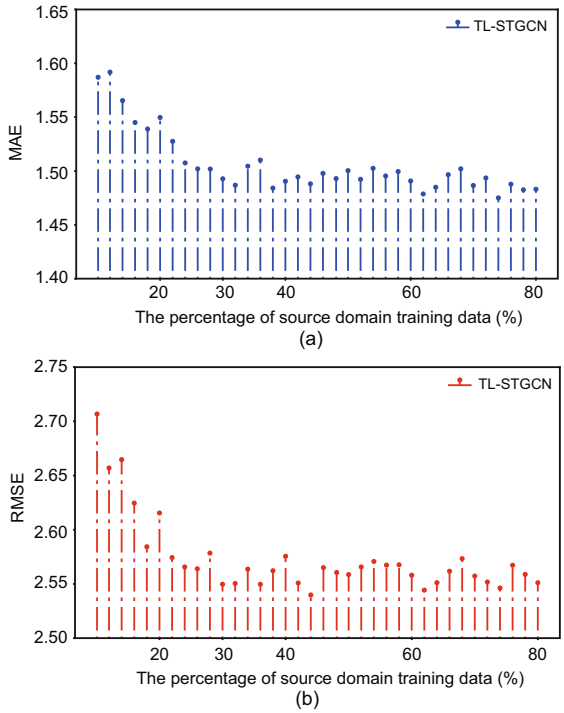

Fig. 7 From Chicago to New York, the percentage of source domain training data is changed when using 8% target domain training data: (a) MAE; (b) RMSE

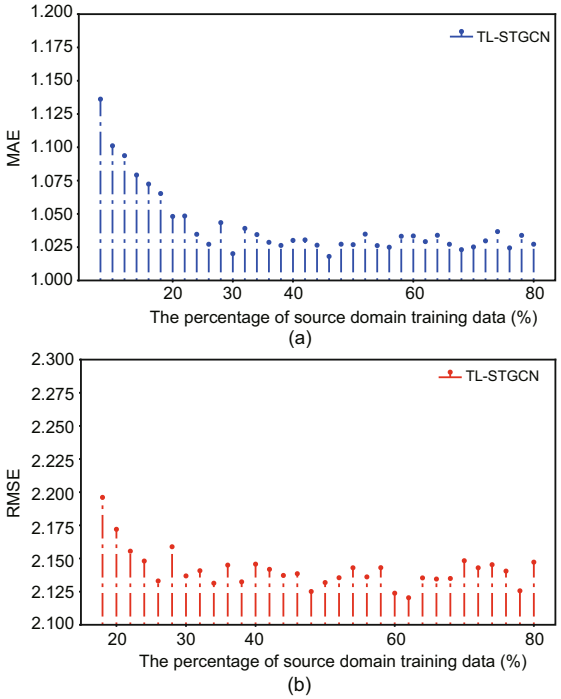

Fig. 8 From New York to Washington, the percentage of source domain training data is changed when using 8% target domain training data: (a) MAE; (b) RMSE

6 Conclusions

In this paper, we propose TL-STGCN that can use data-rich source domains to help data-poor target domains improve the prediction performance. We project the spatiotemporal data of the source and target domains into a common feature space through nonlinear mapping, minimize the distribution difference between the source and target domains, and achieve knowledge transfer from the source domain to the target domain. Finally, we conduct experiments with real bike flow data from Chicago, New York, and Washington. Experimental results show that our proposed method is better than baselines.

In the future, we will extend TL-STGCN to other spatialtemporal transfer prediction problems, such as exhaust gas prediction, air quality index prediction, and people flow prediction.

Contributors

Binkun LIU and Zhenyi XU designed the research. Binkun LIU processed the data and drafted the paper. Yang CAO helped organize the paper. Yu KANG and Yunbo ZHAO helped in data control and project management. Yang CAO and Zhenyi XU revised and finalized the paper. Yu KANG and Zhenyi XU provided the funding acquisition.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding authors upon reasonable request.

References

Ahmed MS, Cook AR, 1979. Analysis of freeway traffic time-series data by using Box-Jenkins techniques. *Transp Res Rec*, 773(722):1-9.

De Maesschalck R, Jouan-Rimbaud D, Massart DL, 2000. The Mahalanobis distance. *Chemom Intell Lab Syst*, 50(1):1-18. https://doi.org/10.1016/S0169-7439(99)00047-7

Guo SN, Lin YF, Feng N, et al., 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. 33rd AAAI Conf on Artificial Intelligence, p.922-929. https://doi.org/10.1609/aaai.v33i01.3301922

He ZW, Li Y, Zhang Y, et al., 2023. Ensemble-transfer-learning-based channel parameter prediction in asymmetric massive MIMO systems. *Front Inform Technol Electron Eng*, 24(2):275-288. https://doi.org/10.1631/FITEE.2200169

Hu J, Lin XH, Wang C, 2022. DSTGCN: dynamic spatial-temporal graph convolutional network for traffic prediction. *IEEE Sens J*, 22(13):13116-13124. https://doi.org/10.1109/JSEN.2022.3176016

Huang DR, Deng ZP, Zhao L, et al., 2017. A short-term traffic flow forecasting method based on Markov chain and grey Verhulst model. 6th Data Driven Control and Learning Systems, p.606-610. https://doi.org/10.1109/DDCLS.2017.8068141

Huang YJ, Song XZ, Zhang SY, et al., 2021. Transfer learning in traffic prediction with graph neural networks. IEEE Int Intelligent Transportation Systems Conf, p.3732-3737. https://doi.org/10.1109/ITSC48978.2021.9564890

Kang GK, Gao JZ, Chiao S, et al., 2018. Air quality prediction: big data and machine learning approaches. *Int J Environ Sci Dev*, 9(1):8-16. https://doi.org/10.18178/ijesd.2018.9.1.1066

Li JY, Guo FC, Wang YB, et al., 2020. Short-term traffic prediction with deep neural networks and adaptive transfer learning. IEEE 23rd Int Conf on Intelligent Transportation Systems, p.1-6. https://doi.org/10.1109/ITSC45102.2020.9294409

Lippi M, Bertini M, Frasconi P, 2010. Collective traffic forecasting. European Conf on Machine Learning and Knowledge Discovery in Databases, p.259-273. https://doi.org/10.1007/978-3-642-15883-4_17

Liu J, Guan W, 2004. A summary of traffic flow forecasting methods. *J Highway Transp Res Devel*, 21(3):82-85.

Lv YS, Duan YJ, Kang WW, et al., 2015. Traffic flow prediction with big data: a deep learning approach. *IEEE Trans Intell Transp Syst*, 16(2):865-873. https://doi.org/10.1109/TITS.2014.2345663

Ma XL, Dai Z, He ZB, et al., 2017. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4):818. https://doi.org/10.3390/s17040818

Mallick T, Balaprakash P, Rask E, et al., 2019. Graph-partitioning-based diffusion convolutional recurrent neural network for large-scale traffic forecasting. *Transp Res Rec*, 2674(9):473-488. https://doi.org/10.1177/036119812093001

Mallick T, Balaprakash P, Rask E, et al., 2021. Transfer learning with graph neural networks for short-term highway traffic forecasting. 25th Int Conf on Pattern Recognition, p.10367-10374. https://doi.org/10.1109/ICPR48806.2021.9413270

Miao H, Shen JX, Cao JN, et al., 2023. MBA-STNet: Bayes-enhanced discriminative multi-task learning for flow prediction. *IEEE Trans Knowl Data Eng*, 35(7):7164-7177. https://doi.org/10.1109/TKDE.2022.3179781

Peng H, Wang HF, Du BW, et al., 2020. Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Inform Sci*, 521:277-290. https://doi.org/10.1016/j.ins.2020.01.043

Seng DW, Lv FS, Liang ZY, et al., 2021. Forecasting traffic flows in irregular regions with multi-graph convolutional network and gated recurrent unit. *Front Inform Technol Electron Eng*, 22(9):1179-1193. https://doi.org/10.1631/FITEE.2000243

Shao W, Jin ZL, Wang S, et al., 2022. Long-term spatio-temporal forecasting via dynamic multiple-graph attention. 31st Int Joint Conf on Artificial Intelligence, p.2225-2232. https://doi.org/10.24963/ijcai.2022/309

Shao ZZ, Zhang Z, Wang F, et al., 2022. Spatial-temporal identity: a simple yet effective baseline for multivariate time series forecasting. Proc 31st ACM Int Conf on Information & Knowledge Management, p.4454-4458. https://doi.org/10.1145/3511808.3557702

Smola AJ, Schölkopf B, 2004. A tutorial on support vector regression. *Stat Comput*, 14(3):199-222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Tascikaraoglu A, 2018. Evaluation of spatio-temporal forecasting methods in various smart city applications. *Renew Sustain Energy Rev*, 82:424-435. https://doi.org/10.1016/j.rser.2017.09.078

Van Der Voort M, Dougherty M, Watson S, 1996. Combining Kohonen maps with ARIMA time series models to forecast traffic flow. *Transp Res Part C Emerg Technol*, 4(5):307-318. https://doi.org/10.1016/S0968-090X(97)82903-8

Wang B, Yan Z, Lu J, et al., 2018. Road traffic flow prediction using deep transfer learning. Proc 13th Int FLINS Conf, p.331-338. https://doi.org/10.1142/9789813273238_0044

Wang BW, Wang JS, 2022. ST-MGAT: spatio-temporal multi-head graph attention network for traffic prediction. *Phys A*, 603:127762. https://doi.org/10.1016/j.physa.2022.127762

Wang LY, Geng X, Ma XJ, et al., 2019. Cross-city transfer learning for deep spatio-temporal prediction. Proc 28th Int Joint Conf on Artificial Intelligence, p.1893-1899. https://doi.org/10.24963/ijcai.2019/262

Wang SZ, Miao H, Li JY, et al., 2022. Spatio-temporal knowledge transfer for urban crowd flow prediction via deep attentive adaptation networks. *IEEE Trans Intell Transp Syst*, 23(5):4695-4705. https://doi.org/10.1109/TITS.2021.3055207

Williams BM, Hoel LA, 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results. *J Transp Eng*, 129(6):664-672.

Xiang SM, Nie FP, Zhang CS, 2008. Learning a Mahalanobis distance metric for data clustering and classification. *Patt Recog*, 41(12):3600-3612. https://doi.org/10.1016/j.patcog.2008.05.018

Xu ZY, Kang Y, Cao Y, 2023. High-resolution urban flows forecasting with coarse-grained spatiotemporal data. *IEEE Trans Artif Intell*, 4(2):315-327. https://doi.org/10.1109/TAI.2022.3153750

Yao ZX, Xia SC, Li Y, et al., 2023. Transfer learning with spatial-temporal graph convolutional network for traffic prediction. *IEEE Trans Intell Transp Syst*, 24(8):8592-8605. https://doi.org/10.1109/TITS.2023.3250424

Yu B, Yin HT, Zhu ZX, 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. Proc 27th Int Joint Conf on Artificial Intelligence, p.3634-3640. https://doi.org/10.24963/ijcai.2018/505

Zhao L, Song YJ, Zhang C, et al., 2020. T-GCN: a temporal graph convolutional network for traffic prediction. *IEEE Trans Intell Transp Syst*, 21(9):3848-3858. https://doi.org/10.1109/TITS.2019.2935152