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 spatiotemporal graph convolution network for city flow prediction*

Binkun LIU[1,2,3], Yu KANG[1,3], 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

unedited

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 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:**

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 Kaur et al. (2018) and urban flow Xu et al. (2023) predict the future state to achieve efficient management of the city. The prediction of city-flow is an important part of a smart city, and can help government departments to 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 mainly divided into the parametric model method and the non-parametric model method. Parametric models Van Der Voort et al. (1996) refer to a model based on certain assumptions, usually with a fixed structure, and calculating parameters based on em-

‡ Corresponding author

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

© Zhejiang University Press 2024

pirical 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 of study.

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 Wang et al. (2019); Lv et al. (2015) 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, examples of this approach can be encountered in the researches of, for instance, Mallick et al. (2021); Huang 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(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 similar 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 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. All in all, the main contributions of the proposed method are summarized as follows:

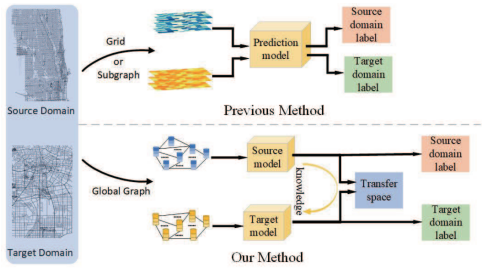


Fig. 1 Different transfer prediction strategies.

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

(2) The dynamic spatiotemporal graph convolution (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 feature 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.

unedited

2 Related Literature

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 the parametric and nonparametric model methods.

2.1.1 Parametric Model

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 et al. (2017), HA Wei (2004), and ARIMA Ahmed and Cook (1979) and its variants Williams and Hoel (2003). The most widely used 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 is 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 predicts from the perspective of time series analysis and does not consider the spatial correlation of data.

2.1.2 Nonparametric Model

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 mainly focuses 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, the data are usually constructed into standard grid data Ma et al. (2017) or graph structure data Yao et al. (2023); Hu et al. (2022); Shao et al. (2022a). Grid-based methods are mainly used to model spatial correlation by dividing the urban road network into grids. Ma et al. (2017) converts traffic flow into images describing spatio-temporal 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); Wang and Wang (2022); Seng et al. (2021) 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) proposes an adversarial multi-task learning framework based on Bayesian enhancement that can co-predict both crowd flow and flow OD simultaneously. To integrate the local and global spatial features of the flow images and semantic spatio-temporal graphs, a Bayesian-based enhanced heterogeneous spatiotemporal attention network is constructed and mitigates the effects of data uncertainty. Due to the dynamic spatio-temporal relationship between stations in static traffic road networks and historical traffic flows, Peng et al. (2020) proposes 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 mainly divided into three categories: sample-based transfer learning, feature-based transfer learning, and model-based transfer learning. Sample-based transfer learning mainly adopts 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 dissim-

unedited

ilar features have similar characteristics after transformation; model-based transfer refers to building models with shared parameters.

There is a relative scarcity in the literature on papers dealing in the transfer prediction of city flows based on global graphs. Lv et al. (2015) proposes TT-DL that first trains the source domain network, then 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 in the source and target domaind' data have different sizes. Wang et al. (2019) proposes the RegionTrans. First, the source and the target domains are divided into a set of grids, and then the grid matching function is 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 is realized. Wang et al. (2022) proposes a deep attentional 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 are first gridded and then mapped to the common embedding space. Next, domain adaptation is 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 studies in the literature Mallick et al. (2021) Huang et al. (2021) Li et al. (2020) usually use 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 feature of human travel habits. These shared features are nonlinearly spanned to a co-occurrence space, and then 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.

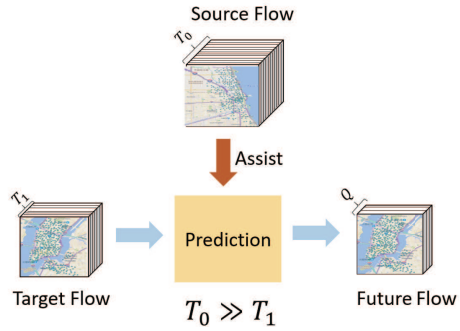


Fig. 2 An example of flow transfer prediction .

3 Preliminary

Definition 3.1. **Spatial Graph**: We construct undirected graphs $G^s = (V^s, E^s, \mathbf{A}^s)$ and $G^t = (V^t, E^t, \mathbf{A}^t)$ 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; $|V^s| = N^s$ and $|V^t| = N^t$, N^s and N^t are the number of stations in the city; and E^s and E^t are the edge sets of the source and target cities, indicating the connectivity between the stations. Here, any two stations are connected; and $\mathbf{A}^s \in \mathbb{R}^{N^s \times N^s}$ and $\mathbf{A}^t \in \mathbb{R}^{N^t \times N^t}$ are the adjacency matrix of the source and target cities.

Problem: The 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. $\mathbf{x}_t^{s,i} \in \mathbb{R}$ and $\mathbf{x}_t^{t,j} \in \mathbb{R}$, respectively, represent the traffic flow of the i-th and j-th stations in the source and target cities, at time t. $X_t^s = (\mathbf{x}_t^{s,1}, \mathbf{x}_t^{s,2}, \ldots, \mathbf{x}_t^{s,N^s})^T \in \mathbb{R}^{N^s \times 1}$ and $X_t^t = (\mathbf{x}_t^{t,1}, \mathbf{x}_t^{t,2}, \ldots, \mathbf{x}_t^{t,N^t})^T \in \mathbb{R}^{N^t \times 1}$ represent the traffic flow of all nodes in the source and the target cities, at time t. $\mathcal{X}_{T_0}^s = \left(X_1^s, X_2^s, \ldots, X_{T_0}^s\right)^T \in \mathbb{R}^{N^{Ss} \times 1 \times T_0}$ and $\mathcal{X}_{T_1}^t = \left(X_1^t, X_2^t, \ldots, X_{T_1}^t\right)^T \in \mathbb{R}^{N^t \times 1 \times T_1}$ represent the traffic flow 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. $Y_t^s = (\mathrm{y}_t^{s,1}, \mathrm{y}_t^{s,2}, \ldots, \mathrm{y}_t^{s,N^s})^T \in \mathbb{R}^{N^s \times 1}$ and $Y_t^t = (\mathrm{y}_t^{t,1}, \mathrm{y}_t^{t,2}, \ldots, \mathrm{y}_t^{t,N^t})^T \in \mathbb{R}^{N^t \times 1}$ represent the traffic flow of all nodes in the source and target cities at the future time point t. Given the sufficient city flow $\mathcal{X}_{T_0}^s$ of source city and the little city flow $\mathcal{X}_{T_1}^t$ of target city, we are enabled to predict the target city flow $\mathcal{Y}_Q^t = (Y_1^t, Y_2^t, \ldots, Y_Q^t) \in \mathbb{R}^{N^t \times 1 \times Q}$ at Q time steps in the future.

For example, the source city may last for several years (T_0), but the target city may only have a

unedited

few days(T_1) because it has only 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.

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 mainly affect the spatiotemporal distribution of city flow in space and time, respectively. Dynamic STGCN consists of Dynamic Adjacency Matrix (DAM) and STGCN block to adaptively extract spatiotemporal features. DAM calculates the traffic flow dynamic adjacency matrix to assist STGCN Block to extract spatiotemporal features. The temporal encoder is constructed by a multi-layer fully connected layer to extract static human travel time information and fuse with the spatiotemporal feature. By nonlinear transformation, the shared features between the source and target domains shared features 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 output through the prediction block. The detailed information of each part is shown in Fig.3.

unedited

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

Dynamic Adjacency Matrix (DAM): The static adjacency matrix mainly relies on human experience to design, 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 matrixes $W^s \in \mathbb{R}^{N^s \times N^s}$ and $W^t \in \mathbb{R}^{N^t \times N^t}$ to obtain the symmetric matrix, whose Hadamard product with the initial adjacency matrix is derived. W^s and W^t are learnable parameters that are updated with the data of the source and target domains. Considering that the elements of W^s and W^t should be non-negative, the nonlinear function σ is used to constrain W^s and W^t.

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

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

where x is input value of σ.

Graph Convolution: In our research, each time step data element of the spatiotemporal flow is correspondingly transformed into a graph, and each sampled value can be regarded as the feature of the corresponding node. We use graph convolution based on spectrogram theory to perform convolution operation on each graph, and the operation operator is defined as $*_G$. In spectrogram analysis, the corresponding graph is represented by the Laplacian matrix. The graph Laplacian matrices of the source and target domains are defined as $L^s = D^s - \tilde{A}^s \in \mathbb{R}^{N^s \times N^s}$ and $L^t = D^t - \tilde{A}^t \in \mathbb{R}^{N^t \times N^t}$, respectively, and $L^s = I_{N_S} - D^{s^{-\frac{1}{2}}} \tilde{A}^s D^{s^{-\frac{1}{2}}}$ and $L^t = I_{N_t} - D^{t^{-\frac{1}{2}}} \tilde{A}^t D^{t^{\frac{1}{2}}}$ are obtained after standardization. $\tilde{A}^s$ and $\tilde{A}^t$ are dynamic adjacency matrices, I_{N_s} and I_{N_t} are identity matrices, and the diagonal matrices $D^s_{ii} = \sum_j \tilde{A}^s_{ij} \in \mathbb{R}^{N^t \times N^t}$ and $D^t_{ii} = \sum_j \tilde{A}^t_{ij} \in \mathbb{R}^{N^t \times N^t}$ are the degree matrixes of the source and target domains. The eigenvalue decomposition of the Laplacian matrix is $L^s = U^s \Lambda^s U^{s^T}$ and $L^t = U^t \Lambda^t U^{t^T}$, where $\Lambda^s \in \mathbb{R}^{N^s \times N^s}$ and $\Lambda^t \in \mathbb{R}^{N^t \times N^t}$ are diagonal matrixes composed of eigenvalues, and $\mathrm{U}^s \in \mathbb{R}^{N^s \times N^s}$ and $\mathrm{U}^t \in \mathbb{R}^{N^t \times N^t}$ are Fourier bases composed of eigenvectors. Then the graph convolution of the source-domain and target-domain space-time flows X^s_t and X^t_t at time t is expressed as:

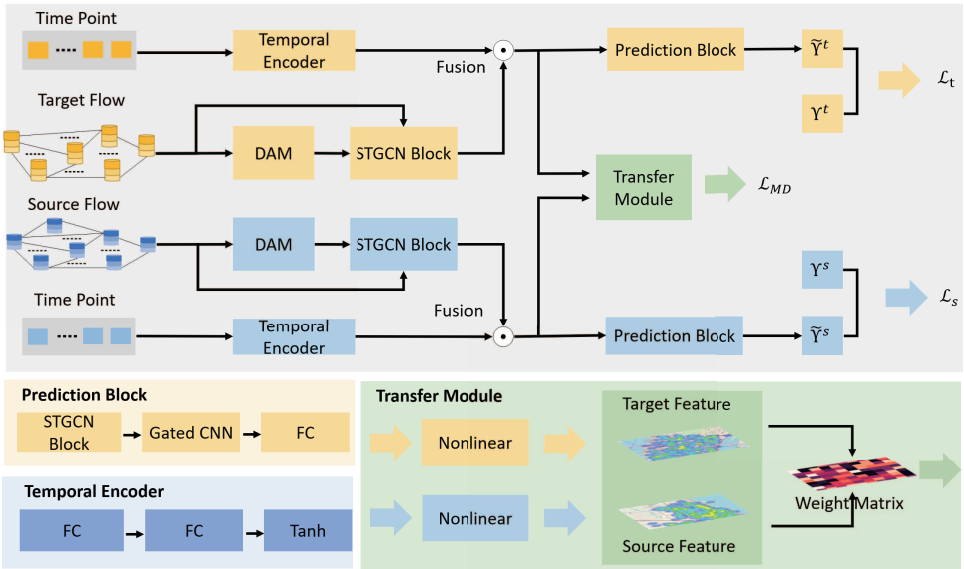


Fig. 3 The structure of proposed TL-STGCN framework. DAM: calculation of the DAM to find similar spatial road network structures across domains; Time Point: encoding vector of prediction time point; Temporal Encoder: learning of human travel time information by multilayer fully connected layer; STGCN Block: learning the flow spatiotemporal features using the spatial graph structure information; Gated CNN: extraction of temporal features; FC: fully connected layer; Fusion: fusion of the flow spatiotemporal features and time point information. information.

unedited

$$\begin{cases} g_\theta^s *_G X_t^s = g_\theta^s(L^s) X_t^s = \mathrm{U}^s g_\theta^s(\Lambda^s) \mathrm{U}^{sT} X_t^s \\ g_\theta^t *_G X_t^t = g_\theta^t(L^t) X_t^t = \mathrm{U}^t g_\theta^t(\Lambda^t) \mathrm{U}^{t^T} X_t^t \end{cases}, \tag{3}$$

where g_θ^s and g_θ^t are the convolution kernels of the source and target domains. Due to the high computational complexity of this convolution, the Chebyshev polynomial is used to approximate this problem.

$$\begin{cases} g_\theta^s *_G X_t^s = g_\theta^s(L^s) X_t^s \approx \sum_{k=0}^{K-1} \theta_k^s T_k(\tilde{L}^s) X_t^s \\ g_\theta^t *_G X_t^t = g_\theta^t(L^t) X_t^t \approx \sum_{k=0}^{K-1} \theta_k^t T_k(\tilde{L}^t) X_t^t \end{cases}, \tag{4}$$

where $T_k(\tilde{L}^s) \in \mathbb{R}^{N^s \times N^s}$ and $T_k(\tilde{L}^t) \in \mathbb{R}^{N^t \times N^t}$ are the k-order Chebyshev polynomials calculated by the scaled Laplacian $\tilde{L}^s = \frac{2}{\lambda_{\max}^s} L^s - I_{N_s}$ and $\tilde{L}^t = \frac{2}{\lambda_{\max}^t} L^t - I_{N_t}$. $\lambda_{\max}^s$ and $\lambda_{\max}^t$ are the maximum eigenvalue of the source-domain and target-domain Laplacian matrix.

4.1.2 Human Travel Habits Feature Extraction

After the spatial road network features are obtained through the graph convolution operation, the human travel habits 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 changes over time and the impact of time points on traffic flow. As shown in Fig.4, 8:00 a.m. and 5:00 p.m. are often flow peaks, while the flow at midnight is often low. It can be seen that time points and flow are also strongly correlated.

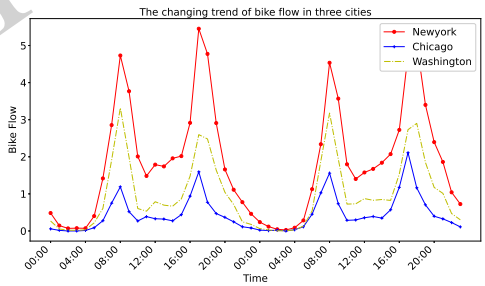


Fig. 4 Changes in bike flow in three cities within 48 hours.

Dynamic Flow Feature: In the time dimension, one-dimensional 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 of length T_0 and channel C_i. We use the Gated CNNs to extract the dynamic flow feature $F_{T_{sq}} \in \mathbb{R}^{(T_0-K_t+1)\times C_o}$. And 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 STGCN block can be described as follows:

$$\mathcal{X}^{(r)} = f_{st}^{(r)}(\mathcal{X}^{(r-1)}) \in \mathbb{R}^{N\times C_0^{(r)}\times(T_0-2\times r\times(K_t-1))}, \tag{5}$$

$\mathcal{X}^{(r-1)} \in \mathbb{R}^{N\times C_i^{(r-1)}\times(T_0-2\times(r-1)\times(K_t-1))}$ represents the r-th layer. In the input of the STGCN block, $f_{st}^{(r)}$ represents the abstract function of the STGCN block of the r-th layer.

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 one-dimensional time vectors $\boldsymbol{T_p^s}$ and $\boldsymbol{T_p^t}$ with a length of 24; then, the time vectors $\boldsymbol{T_p^s}$ and $\boldsymbol{T_p^t}$ serve as the input of the temporal encoder to enable obtaining the static flow feature F_{T_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.

$$\begin{cases} \boldsymbol{F}_{T_p}^S = f(\boldsymbol{W}_{T_p}^s T_p^s + \boldsymbol{b}_{T_p}^s) \\ \boldsymbol{F}_{T_p}^t = f(\boldsymbol{W}_{T_p}^t \boldsymbol{T}_p^t + \boldsymbol{b}_{T_p}^t) \end{cases}, \tag{6}$$

$\boldsymbol{W_{T_p}^s}$, $\boldsymbol{b}_{T_p}^s$ and $\boldsymbol{W_{T_p}^t}$, $\boldsymbol{b}_{T_p}^t$ are the learning parameters of the temporal encoder in the source and target domains, 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.

$$\begin{cases} \tilde{\mathcal{X}}_t^{s,(1)} = F_{T_p}^s \odot \mathcal{X}_t^{s,(1)} + \mathcal{X}_t^{s,(1)} \\ \tilde{\mathcal{X}}_t^{t,(1)} = F_{T_p}^t \odot \mathcal{X}_t^{t,1} + \mathcal{X}_t^{t,(1)} \end{cases}, \tag{7}$$

where $\mathcal{X}_t^{s,(1)}$ and $\mathcal{X}_t^{t,(1)}$ represent the features extracted from the first layer of the source and target domains, and $\odot$ represents the Hadamard product. We replace the input of the 2nd STGCN block from $\mathcal{X}_t^{s,(1)}$ and $\mathcal{X}_t^{t,(1)}$ with $\tilde{\mathcal{X}}_t^{s,(1)}$ and $\tilde{\mathcal{X}}_t^{t,(1)}$ 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.

Non-linear Transformation: For the transfer module with non-shared parameters, the number 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 $r_\theta(\mathcal{X}^{t,(r)})$ to keep it consistent with the source domain.

Considering that the source and target domains have differing road network structures, we ensure that the source domain spatiotemporal feature $\mathcal{X}^{s,(r)} \in \mathbb{R}^{N^s\times C_o^{s,(r)}\times(T_0-2\times r\times(K_t-1))}$ and target domain spatiotemporal feature $\mathcal{X}^{t,(r)} \in \mathbb{R}^{N^t\times C_o^{t,(r)}\times(T_0-2\times r\times(K_t-1))}$ 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 BatchNormal 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.

$$\begin{cases} H^s = BN(\boldsymbol{W}_n^{s,(2)} f(\boldsymbol{W}_n^{s,(1)} \widetilde{x}_t^{s,(1)} + \boldsymbol{b}_n^{s,(1)}) \\ \qquad + \boldsymbol{b}_n^{s,(2)}) \\ H^t = BN(\boldsymbol{W}_n^{t,(2)} f(\boldsymbol{W}_n^{t,(1)} r_\theta(\tilde{x}_t^{t,(1)}) + \boldsymbol{b}_n^{t,(1)}) + \boldsymbol{b}_n^{t,(2)}) \end{cases}, \tag{8}$$

where $\boldsymbol{W}_n^{s,(1)}$, $\boldsymbol{b}_n^{s,(1)}$ and $\boldsymbol{W}_n^{t,(1)}$, $\boldsymbol{b}_n^{t,(1)}$ are the learnable parameters of the first layer of the source and target domains; $\boldsymbol{W}_n^{s,(2)}$, $\boldsymbol{b}_n^{s,(2)}$ and $\boldsymbol{W}_n^{t,(2)}$, $\boldsymbol{b}_n^{t,(2)}$ are the learnable parameters of the second layer of the source and target domains; and f is the nonlinear activation function LeakyRelu. The output dimension dim of the fully connected network is a hyperparameter, and finally the source domain feature $H^s \in \mathbb{R}^{\dim\times C_o^{s,(1)}\times(T_0-2\times(K_t-1))}$ and the target domain feature $H^t \in \mathbb{R}^{\dim\times C_o^{s,(1)}\times(T_0-2\times(K_t-1))}$ in the same feature space $\mathbb{S}$ are obtained.

Domain Adaption: To characterize the difference between the source domain H^s and the target domain H^t, we design MD loss which uses the simplified MD De Maesschalck et al. (2000); Xiang et al. (2008) as the metric. The weight ma-

unedited

trix $\mathrm{M}' \in \mathbb{R}^{\mathrm{dim} \times \mathrm{dim}}$ is obtained after normalization of the diagonal matrix of learnable parameters $\mathrm{M} \in \mathbb{R}^{\mathrm{dim} \times \mathrm{dim}}$. 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 M' can dynamically focus on dimensions that have a greater impact on distribution differences and improve the transfer prediction.

$$M'_{i,j} = \begin{cases} 0 & \text{if } i \neq j \\ \frac{\exp(\mathrm{M}_{i,j})}{\sum_i^{\mathrm{dim}} \exp(\mathrm{M}_{i,j})} & \text{if } i = j \end{cases}, \tag{9}$$

The MD loss can be written as:

$$\mathcal{L}_{MD} = \frac{1}{N \times T} \sum_n \sum_t \sum_c \left(H^s_{n,t,c} - H^t_{n,t,c}\right) \mathrm{M}' \left(H^s_{n,tc} - H^t_{n,t,c}\right)^T, \tag{10}$$

where $n \in \{1, 2, \cdots, N\}$ represents the number of samples, $\mathrm{t} \in \{1, 2, \cdots, T\}$ represents the time step, and $c \in \{1, 2, \cdots, C\}$ represents the number of convolution channels. By minimizing the transfer loss $\mathcal{L}_{MD}$, the difference in feature distribution between the source and target domains is reduced, to achieve the knowledge transfer between the source and target domains.

The total loss function of the model can be written as:

$$\mathcal{L}_{total} = \mathcal{L}_t(\widetilde{\Upsilon}^t_Q, \Upsilon^t_Q) + \alpha \mathcal{L}_s(\widetilde{\Upsilon}^s_Q, Y^s_Q) + \beta \mathcal{L}_{\mathrm{M\ D}} + \gamma \|\theta\|^2, \tag{11}$$

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

$\mathcal{L}_t$ and $\mathcal{L}_s$ are the regression loss of the source and target domains, respectively; they are formalized as follows:

$$\mathcal{L}_t(\widetilde{\Upsilon}^t_Q, \Upsilon^t_Q) = \frac{1}{N} \sum_n (\tilde{Y}^t_{Q,n} - \Upsilon^t_{Q,n})^2, \tag{12}$$

$$\mathcal{L}_s(\widetilde{\Upsilon}^s_Q, \Upsilon^s_Q) = \frac{1}{N} \sum_n (\tilde{Y}^s_{Q,n} - \Upsilon^s_{Q,n})^2, \tag{13}$$

where $n \in \{1, 2, \cdots, N\}$ represents the sample, and $\tilde{Y}^t_{Q,n}$ and $\tilde{Y}^s_{Q,n}$ are the model outputs of source and target domains, respectively.

5 Experiment

5.1 Setting

5.1.1 Dataset

We use bike flow data in three cities for the experiment conducted as part of the present research, New York, Chicago, and Washington. 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 weights to construct a spatial graph. To evaluate the effect of the model, we choose one city as the source domain and another city as the target domain for transfer, and so there are six transfer scenarios. The details of the dataset are shown in Table 1.

5.1.2 Metrics

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

$$RMSE: \sqrt{\frac{1}{N} \sum_n (\tilde{Y}^t_{Q,n} - \Upsilon^t_{Q,n})^2}, \tag{14}$$

$$MAE: \frac{1}{N} \sum_n \left(\left|\tilde{Y}^t_{Q,n} - \Upsilon^t_{Q,n}\right|\right), \tag{15}$$

5.1.3 Hyperparameter

The batch size is 50, the epoch is 100, and Adaptive Moment Estimation optimizer (ADAM) is selected as the optimization algorithm. The learning rate is 0.001 and the learning rate of matrix 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 2. C→N indicates that the source and target domains are Chicago and New York, respectively.

5.1.4 Network Implementation

Details of the network structure are presented in Table 3. The FLOPs of TL-STGCN is 411673328

unedited

Table 1 Experimental datasets details

	Chicago	New York	Washington
Time span	2015.01.01-2015.03.31		
Amount of data	2160		
Time interval	1 h		
Number of stations	300	331	283
Training dataset of source domain	80%(1 Jan. - 13 Mar.)		
Training dataset of target domain	8% (7 Mar. - 13 Mar.)	16% (28 Feb. - 13 Mar.)	
Test dataset	20%(14 Mar. - 31 Mar.)		
Input step	12		
Output step	1		

Table 3 Experimental network details

	Source	Target
STGCN Block_1	out_channels=64 spatial_channels=32	out_channels=32 spatial_channels=16
STGCN Block_2	out_channels=64 spatial_channels=32	out_channels=32 spatial_channels=16
Gated CNN	out_channels=32	out_channels=32
FC	input_dim=64 out_dim=1	input_dim=64 out_dim=1

unedited

Table 2 Experimental hyperparameter settings

	C→N	W→N	W→C	N→C	N→W	C→W
a	0.05	0.5	1.5	0.25	0.25	1.5
β	1	0.01	0.05	0.75	1	0.05
dim	250	220	250	250	250	200

and the number of model parameters is 599988 when dim is 250.

5.1.5 Baseline

To compare with the performance of the model discussed in the present research with that of TL-STGCN, we chose the following six 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 Wei (2004): The average of historical data is used to predict future traffic flow.

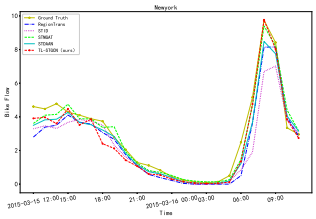
SVR Smola and Schölkopf (2004): The training dataset is used to train the Support Vector Regression model, use a linear kernel, and the penalty term is 0.001.

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.

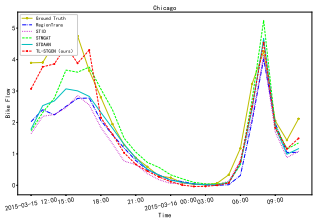
STGCN Yu et al. (2018): Devised as a flow-prediction method based on STGCN, this model contains two layers of STGCN Block with 32 channels and a learning rate of 0.001.

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

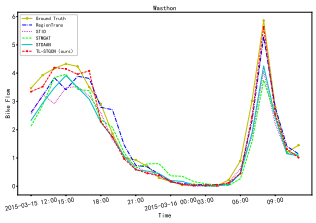
ASTGCN Guo et al. (2019): This is a flow-prediction method that introduces a spatiotemporal attention mechanism based on STGCN. Considering that the training data are less, the ASTGCN is pruned and only the recent branch is retained. The



(a) from Chicago to Newyork



(b) from Washington to Chicago



(c) from Chicago to Washington

Fig. 5 Prediction of bike flow from 2015-03-15 to 2015-03-16 when using 16% target training dataset.

batch size is 64 with a learning rate of 0.001.

STID Shao et al. (2022b): 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 and Wang (2022): This is a spatiotemporal multi-head graph attention network that captures spatial dimensional features using a multi-head attention graph network. The batch size is 64 with a learning rate of 0.001.

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

STDAAN Wang et al. (2022): This is a deep attentional 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.

5.2 Result

5.2.1 Comparison with Existing Methods

Table 4 shows the comparison of MAE and RMSE between TL-STGCN and other baselines. To present the experimental results more clearly, we split Table 4 into two parts, top and bottom, showing the results of the transfer baselines and the non-transfer baselines, respectively. Compared with non-transfer methods such as STGCN, TL-STGCN can use rich-source domain data 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 spatial-temporal flow prediction when the amount of data is small.

Fig.5 shows the prediction curve of bicycle flow from 2015-03-15 to 2015-03-16 with our method and other comparative methods using only 16% of the target domain training data. The curve with yellow * is the ground truth. 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.

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 a.m., and the flow is low, which is in line with people's daily behavior habits. At 8 a.m., people started their day's work, which led to a significant increase in bike flow in suburbs and urban areas. At 12 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 only uses 16% of the target domain training data to capture the spatial-temporal changes of bike flow.

5.2.2 Ablation

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

unedited

Table 4 Experimental results of MAE and RMSE

	Method	Metrics	N		C		W	
			8%	16%	8%	16%	8%	16%
Non-transfer	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

	Method	Metrics	C→N		W→N		W→C		N→C		N→W		C→W	
			8%	16%	8%	16%	8%	16%	8%	16%	8%	16%	8%	16%
Transfer	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**

unedited

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, together with keeping the rest unchanged.

From Table 5, we ascertain that compared to BackBone, our method effectively utilizes the 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 more superior results. By comparing TL-Local and TL-STGCN, we can find that the interaction between local regions is also very important for the transfer effect. This is because the city exists as a whole, which also 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 the DAM helps to discover the similar site spatial dependency of the source and target cities. The experimental results demonstrate that, in comparison with the TL-Gated CNN as well as the simple 1D CNN, the Gated CNN structure demonstrates a greater effectiveness in extracting the temporal features of

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

unedited

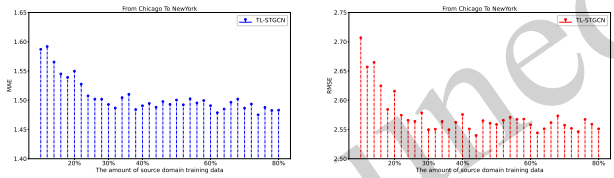


Fig. 7 From Chicago to New York, the percentage of source domain training data is changed when using 8% target domain training data.

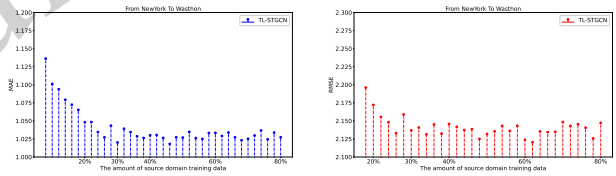


Fig. 8 From New York to Washington, the percentage of source domain training data is changed when using 8% target domain training data.

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.

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 to observe the impact on the percentage effect. 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.

6 Conclusions

In this paper, we propose a TL-STGCN that solves the problem of ascertainment of the means to use adequate data in the source domain to help the target domain lacking data to improve prediction accuracy. 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

Table 5 Ablation results of MAE and RMSE

	Method	Metrics	N		C		W	
			8%	16%	8%	16%	8%	16%
Non-transfer	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

	Method	Metrics	C→N		W→N		W→C		N→C		N→W		C→W	
			8%	16%	8%	16%	8%	16%	8%	16%	8%	16%	8%	16%
Transfer	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

Chicago, New York, and Washington. The experiments showed that our proposed method is better than other baselines.

In the future, we will extend TL-STGCN to other spatial-temporal transfer prediction problems, such as exhaust gas prediction, AQI prediction, and people flow prediction.

Contributors

Binkun Liu and Zhenyi Xu designed the research. Binkun Liu and Renjun Wang processed the data. Binkun Liu drafted the manuscript. Yang Cao helped organize the manuscript. Yu Kang and Yunbo Zhao helped the data control and project management. Yang Cao and Zhenyi Xu revised and finalized the paper. Yu Kang and Zhenyi Xu provided the funding acquisition.

Compliance with ethics guidelines

Binkun Liu, Yu Kang, Yang Cao, Yunbo Zhao, and Zhenyi Xu declare that they have no conflict of interest.

References

Ahmed MS, Cook AR, 1979. Analysis of freeway traffic time-series data by using box-jenkins techniques. *Transportation Research Record*, .

De Maesschalck R, Jouan-Rimbaud D, Massart D, 2000. The mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1):1-18.

Guo S, Lin Y, Feng N, et al., 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. AAAI Conference on Artificial Intelligence.

He Z, Li Y, Zhang Y, et al., 2023. Ensemble-transfer-learning-based channel parameter prediction in asymmetric massive mimo systems. *Frontiers of Information Technology & Electronic Engineering*, 24(2):275-288.

Hu J, Lin X, Wang C, 2022. Dstgcn: Dynamic spatial-temporal graph convolutional network for traffic prediction. *IEEE Sensors Journal*, 22(13):13116-13124.

Huang D, Deng Z, Zhao L, et al., 2017. A short-term traffic flow forecasting method based on markov chain and grey verhulst model. 2017 6th Data Driven Control and Learning Systems (DDCLS), p.606-610.

Huang Y, Song X, Zhang S, et al., 2021. Transfer learning in traffic prediction with graph neural networks. 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), p.3732-3737.

Kaur G, Gao J, Chiao S, et al., 2018. Air quality prediction: Big data and machine learning approaches. *International Journal of Environmental Science and Development*, 9:8-16.

Li J, Guo F, Wang Y, et al., 2020. Short-term traffic prediction with deep neural networks and adaptive transfer learning. 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), p.1-6.

Lippi M, Bertini M, Frasconi P, 2010. Collective traffic forecasting. Machine Learning and Knowledge Discovery in Databases, Berlin, Heidelberg, p.259-273.

Lv Y, Duan Y, Kang W, et al., 2015. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865-873.

Ma X, Dai Z, He Z, et al., 2017. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17:818.

Mallick T, Balaprakash P, Rask E, et al., 2019. Graph-partitioning-based diffusion convolutional recurrent neural network for large-scale traffic forecasting. *Transportation Research Record*, 2674:473 - 488.

unedited

Mallick T, Balaprakash P, Rask E, et al., 2021. Transfer learning with graph neural networks for short-term highway traffic forecasting. 2020 25th International Conference on Pattern Recognition (ICPR), p.10367-10374.

Miao H, Shen J, Cao J, et al., 2023. Mba-stnet: Bayes-enhanced discriminative multi-task learning for flow prediction. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):7164-7177.

Peng H, Wang H, Du B, et al., 2020. Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Information Sciences*, 521:277-290.

Seng D, Lv F, Liang Z, et al., 2021. Forecasting traffic flows in irregular regions with multi-graph convolutional network and gated recurrent unit. *Frontiers of Information Technology & Electronic Engineering*, 22(9):1179-1193.

Shao W, Jin Z, Wang S, et al., 2022a. Long-term spatio-temporal forecasting via dynamic multiple-graph attention. 31st International Joint Conference on Artificial Intelligence, IJCAI 2022, p.2225-2232.

Shao Z, Zhang Z, Wang F, et al., 2022b. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. Proceedings of the 31st ACM International Conference on Information & Knowledge Management, New York, NY, USA, p.4454âĂŞ4458.

Smola A, Schölkopf B, 2004. A tutorial on support vector regression. *Statistics and Computing*, 14:199-222.

Tascikaraoglu A, 2018. Evaluation of spatio-temporal forecasting methods in various smart city applications. *Renewable and Sustainable Energy Reviews*, 82:424-435.

Van Der Voort M, Dougherty M, Watson S, 1996. Combining kohonen maps with arima time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies*, 4(5):307-318.

Wang B, Wang J, 2022. St-mgat:spatio-temporal multi-head graph attention network for traffic prediction. *Physica A: Statistical Mechanics and its Applications*, 603:127762.

Wang L, Geng X, Ma X, et al., 2019. Cross-city transfer learning for deep spatio-temporal prediction. Proceedings of the 28th International Joint Conference on Artificial Intelligence, p.1893âĂŞ1899.

Wang S, Miao H, Li J, et al., 2022. Spatio-temporal knowledge transfer for urban crowd flow prediction via deep attentive adaptation networks. *IEEE Transactions on Intelligent Transportation Systems*, 23(5):4695-4705.

Wei G, 2004. A summary of traffic flow forecasting methods. *Journal of Highway and Transportation Research and Development*, .

Williams B, Hoel L, 2003. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of Transportation Engineering*, 129:664-672.

Xiang S, Nie F, Zhang C, 2008. Learning a mahalanobis distance metric for data clustering and classification. *Pattern Recognition*, 41(12):3600-3612.

Xu Z, Kang Y, Cao Y, 2023. High-resolution urban flows forecasting with coarse-grained spatiotemporal data. *IEEE Transactions on Artificial Intelligence*, 4(2):315-327.

Yao Z, Xia S, Li Y, et al., 2023. Transfer learning with spatialâĂŞtemporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 24(8):8592-8605.

Yu B, Yin H, Zhu Z, 2018. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. Proceedings of the 27th International Joint Conference on Artificial Intelligence, p.3634âĂŞ3640.

Zhao L, Song Y, Zhang C, et al., 2020. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848-3858.

unedited