



OPEN

Similarity based city data transfer framework in urban digitization

Haoxiang Wang¹, Xiaoping Che^{2✉}, Enyao Chang², Chenxin Qu², Ganghua Zhang², Zihan Zhou², Zhenlin Wei³, Gengyu Lyu⁴ & Pengfei Li⁵

Cross-city transfer learning aims to apply the knowledge and model from data-rich cities to data-poor cities to solve the cold start problem. Existing methods directly transfer the model constructed from developed cities to underdeveloped cities without considering the similarity between them, which leads to a potential transfer mismatch problem, and in turn, decreases the performance of transfer results. Meanwhile, existing transfer learning methods cannot effectively extract the time series features of the data, resulting in the inability to achieve adaptive positive migration across cities. To solve this problem, we propose a similarity-based cross-city transfer learning method named TransCSM, which embeds the urban similarity into an adaptation transfer learning framework to achieve desired data transfer. Specifically, we first constructed an urban similarity model, which utilizes the urban POI (Point Of Interest) data to group the cities with similar characteristics into the same cluster. Then, we build a feature extractor network, that uses convolution neural network (CNN) and Gated Recurrent Unit (GRU) to extract more representative features of time series data. Afterwards, we build an adaptation transfer learning framework to achieve data transfer within the same city cluster, which ensures the reliability of cross-city data transferring results. Finally, we evaluate our proposed method in many public POI datasets from Baidu Map API, and enormous results have demonstrated that our proposed method can achieve superior performance against state-of-the-art methods.

Keywords Urban Digitization, Transfer Learning, City Similarity

Constructing more aesthetically pleasing and user-friendly smart cities has become a significant aspect of contemporary urban development¹. In recent years, deep learning has led to significant progress in building smart cities, greatly improving the ability to capture nonlinear correlations in data^{2,3}. With the widespread use of sensors, large-scale urban clusters generate a large amount of valuable data⁴. The deployment and maintenance of a large number of sensors in small and medium-sized cities entail high costs, which limits their capacity to collect sufficient and usable data over prolonged periods of time, as demonstrated in Fig. 1. As a result, these cities may lack a large amount of data to build smart cities^{5,6}, which is also called *cold start* problem.

Transfer learning provides an effective method to solve the problem of data defects, which applies good models trained in data-rich developed cities to data-poor less developed cities^{7,8}. Such learning framework enables the smart construction of underdeveloped cities to make leapfrog progress and get rid of the dilemma of insufficient data in underdeveloped cities. In recent years, many transfer learning methods have been proposed to solve data scarcity problem. For example, Wang et al.⁹ proposed a regional transfer learning method named RegionTrans for cross-city depth spatiotemporal prediction, which used regional similarity regularization to transfer knowledge between cities. Guo et al.⁷ proposed a collaborative filtering-based dual knowledge transfer framework named CityTransfer, which transfer chain store knowledge from semantic-related fields for recommending chain store layouts in new cities. Huang et al.¹⁰ proposed a graph neural networks-based traffic prediction method called TEEPEE, which used traffic data from developed cities to predict traffic data from less developed cities. Wei et al.¹¹ proposed a flexible multimodal transfer learning method named FLORAL, which learned semantically related dictionaries for multiple modalities from a source domain and would be transferred to a target domain. Traditional transfer learning approaches, such as domain adaptation, have been widely explored. However, recent advancements in meta-learning and zero-shot learning provide new perspectives on enhancing cross-city adaptation. Meta-learning, often referred to as “learning to learn,” enables rapid adaptation to new cities with minimal labeled data, while zero-shot learning leverages shared feature representations to infer knowledge

¹Guanghua School of Management, Peking University, Beijing, China. ²School of Software Engineering, Beijing Jiaotong University, Beijing, China. ³School of Traffic and Transportation, Beijing Jiaotong University, Beijing, China. ⁴Faculty of Information Technology, Beijing University of Technology, Beijing, China. ⁵Beijing Emergency Management Bureau, Beijing, China. ✉email: xpche@bjtu.edu.cn

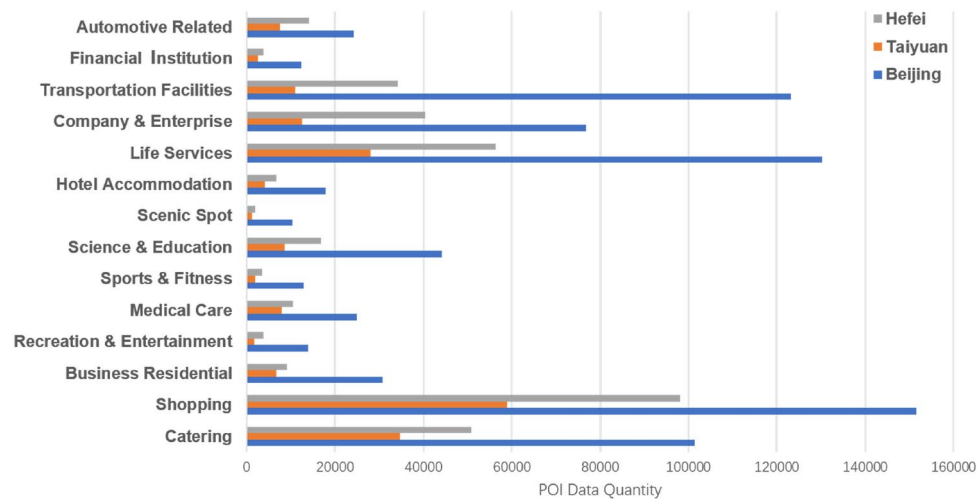


Fig. 1. Comparison of POI data quantity in different cities. Beijing, a developed city, has significant differences in terms of the quantity of POI data from less developed cities such as Hefei and Taiyuan. Large amounts of historical data may not yet be collected in some underdeveloped cities due to uneven urban development.

about unseen cities. These methodologies have been successfully applied in various domains, such as state estimation in lithium-ion batteries using Unsupervised Domain Adaptation (UDA)¹² and hypergraph-based semi-supervised feature selection¹³. However, the above studies directly conducted transfer learning between two cities and did not quantitatively analyze whether each pair of cities is suitable for data transfer. Meanwhile, they also failed to extract desired time series features for the precise transfer of cross-city data, which inevitably results in inaccurate data transfer.

To address these challenges, we propose a similarity-based cross-city transfer learning method named TransCSM, which embeds urban similarity into an adaptation transfer learning framework to achieve desired data transfer. The TransCSM includes two parts: the City Similarity Module and Adaptation Transfer Module. In the city similarity module, we first obtained a 19-dimensional POI data distribution map for each city, where each dimension represents one property of this city, such as shopping, and it is segmented into multiple smaller regions with several POIs to better characterize the features of the city in this dimension. To figure out the optimal segmentation size, the thermal map is introduced to measure the similarity of different cities, where the optimal segmentation size is selected from several specific sizes of grids. According to the above operations, we obtain more representative distribution features for each city, and accordingly we employ the k-means algorithm to group these cities with high-level similarity into the same cluster. In the adaptation transfer module, ConvGRU and Domain Adaptation are jointly introduced to achieve data transfer across cities. The ConvGRU model is used to extract the deep time series features of both source and target cities within the same city cluster, which utilizes a combination of one-dimensional convolution neural networks (1DCNN) and GRU structures to enhance the effectiveness of the feature extraction. In the Domain Adaptation part, we design a domain adversarial neural network to achieve cross-city adaptive transfer, where the result of data source differentiation - from target city or source city - is used as feedback to motivate the network to continue mining the same feature distribution at a deeper level between the source city and the target city, until the data source cannot be distinguished. By integrating the city similarity module and adaptation transfer module into a unified framework, the performance of the cross-city data transfer is significantly improved.

The contributions of this paper are as follows:

- We proposed a novel similarity based cross-city adaptation transfer learning method, which integrates the city similarity, ConvGRU and domain adaptation into a unified framework to improve the performance of cross-city data transfer.
- To the best of our knowledge, it is the first time to embed city similarity into adaption transfer learning model, which significantly reduces negative transfer influence.
- We conduct enormous experiments on various real-world POI data sets from Baidu Map API, and the results have verified the effectiveness of our proposed method.

The rest of this paper is organized as follows. Section II reviews some related works about City similarity framework and measurements and urban transfer learning. In Section III, we provide a detailed introduction to our method. In Section IV, we show the comparative experimental results of our method and the existing methods. In Section V, we summarize conclusions and future work.

Related work

In this section, we briefly introduced two related studies about our proposed method including city similarity measurement and urban transfer learning.

City similarity framework and measurements

Some researchers have proposed measurement methods for data similarity and used them to judge similarity and recommend. For example, Wang et al.¹⁴ extracted a new class of similarity measures from the work on fuzzy power sets, and discussed the properties of the concept of approximate equality corresponding to these similar measures. Yan et al.¹⁵ discussed the spatial similarity and popularity of urban TV programs and applied them to advertising and video recommendations. Santini et al.¹⁶ proposed fuzzy logic based similarity measurement method, called FFC, which had several functions that matched the results of human experiments, and used the FFC model to model similarity evaluation based on fuzzy judgments of attributes.

In order to measure the similarity of cities, existing methods can be roughly grouped into two categories: attributes and dimensions of cities and city representations method. For the attributes and dimensions of cities, Mulligan et al.¹⁷ researched the constantly changing population size distribution of these large cities and classified them into different clusters based on the similarity of population growth rates, ranking them from fastest to slowest growing. Crucitti et al.¹⁸ proposed that spatial analysis can visualize and characterize the urban structure, which demonstrates that planned and self-organized cities obviously belong to two different universal categories. For the city representations method, Preotiuc-Pietro et al.¹⁹ proposed Three vector based city representations methods, which utilized venue data from 17 cities in the United States, totaling more than 10 million venues, and had drawn the conclusion that different assumptions in urban cognition may lead to induced urban description and classification. McKenzie et al.²⁰ used shared micro-mobility patterns to characterize the city, which can evaluate the city similarity by sharing the patterns of vehicle movement trajectories, including trams and bicycles.

The aforementioned methods only focus on data similarity and urban similarity models while ignore its contribution to urban data transfer, which fails to effectively improve the overall performance of transfer learning.

Urban transfer learning

Research on Urban transfer learning models has significantly progressed and been applied in many practical applications. Some researchers focused on how to achieve effective transfer through data similarity evaluation or data weighting, etc. For example, Fawaz et al.²¹ proposed to judge the similarity of time series based on DTW (Dynamic Time Warping)²², and the two most similar sets of data were selected for training and fine-tuned to achieve transfer. However, DTW was processed only in the time dimension, which led to discard information such as inter dimensional interactions. Jung et al.²³ used Pearson correlation coefficient to evaluate the correlation of the data between cities, and then used DNN to extract data features. However, when there had non-linear correlation in the data, the Pearson correlation coefficient will fail. Zeng et al.²⁴ used the Tradaboost method to weight the source and target city data, but the results rely heavily on the different configuration of weight settings, and inappropriate initialization can lead to very poor results from Tradaboost. Moreover, Jung²³ and Zeng²⁴ didn't make effective use of the time series features of the data.

Transfer learning has emerged as a pivotal technology for harnessing data from source domains to enhance the performance of target tasks²⁵. Some researchers focused on how to learn the common features between domains through models to achieve effective transfer. For example, Gautam²⁶ trained the LSTM model using Covid-19 data from countries with sufficient data and apply them to countries with insufficient data. Although effective domain adaptation was not performed, the overall results suggest that model-based transfer learning across geographic regions is feasible. Xin et al.²⁷ used an auto-encoder with CNN to extract the data features, and jointly used the source and target domain data to train the model so that model was trained with sufficient data and can also learn target domain features. However, the inadequate target data will lead to a weakening of feature learning by the model. Hu et al.²⁸ used a deep auto-encoder model where each city's data share the hidden layer of the model while have their own out layer during the training phase, thus the sharing of model parameters is achieved. Wan et al.²⁹ proposed a cost sensitive transfer learning framework to address the issue of heterogeneous interpretation features between two datasets, which can represent cross city data using unified features in the original interpretation feature space. Bappee et al.³⁰ tried to achieve city self-adaptation by reducing the KL-Divergence between two cities' features. Traditional domain adaptation techniques, such as UDA, seek to minimize feature distribution shifts between cities. However, recent research in meta-learning offers a promising alternative by enabling quick adaptation with limited target data. For instance, unsupervised domain adaptation has been applied to lithium-ion battery state estimation across different battery types¹². This approach aligns well with cross-city adaptation, where data distributions differ but share underlying structures. Similarly, hypergraph Laplacian-based semi-supervised feature selection¹³ demonstrates the effectiveness of capturing complex relationships in sparse data, which is crucial for urban computing applications. The idea of deep domain adaption is used in³¹⁻³⁴. However, the above methods also do not effectively utilize the time series features of the data when performing feature extraction and need to be further improved.

Proposed method

Overall architecture

In this section, we describe the architecture of our proposed TransCSM, as shown in Fig. 2. TransCSM is comprised of two main components: the city similarity module and the adaptation transfer learning module. The first component consists of the POI Data Preprocessing block and Cluster&Classify block. The POI Data Preprocessing block is used to process the different kinds of POI data of each city, whose output is fed to the next block as their input. The cluster and classify block divides similar cities into the same subset according to the urban characteristics, and the classification results will serve as a reference for the selection of source cities and target cities for transfer learning in the next step. Finally, the adaptation transfer learning module, i.e., the DA-ConvGRU model, utilizes convolutional neural network and domain adaptation to extract the similar distribution of source and target cities, to predict the missing POI data of target cities.

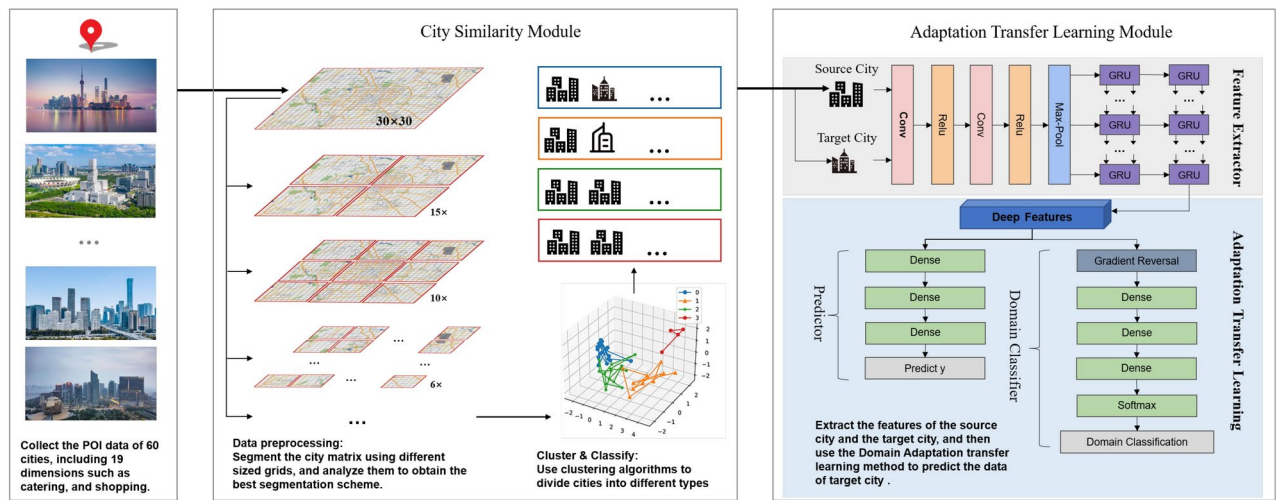


Fig. 2. TransCSM framework.

City similarity model

Urban POI, the fundamental spatial big data for urban research. It contains location and attribute information of various urban facilities, including 19 dimensions of POI data, specifically catering, entertainment, infrastructure, shopping, banking and finance, medical care, life services, sports and fitness, automobiles, construction properties, companies and enterprises, education, hotels and guesthouses, institutional groups, place names and addresses, cultural venues, tourism services, transportation facilities, commercial residences. We need to mine the characteristics of a city based on these data. Therefore, we propose a city similarity model which includes two parts: POI Data Preprocessing and Cluster&Classify.

POI data preprocessing

Urban POI data serve as a fundamental spatial representation for urban research, encapsulating diverse urban functional characteristics. Our model leverages a 19-dimensional POI feature representation to measure city similarity. This choice is motivated by urban planning theories, where POI distributions reflect functional zoning, socio-economic activities, and urban morphology. However, to further validate its effectiveness, we compare our approach with two widely used city similarity measurement methods: graph-based representations and hierarchical clustering.

Graph-based methods capture complex spatial dependencies, but they require significant computational resources and graph structure tuning, making them less scalable for large urban datasets. Our 19-dimensional POI representation is computationally efficient and provides a direct interpretable representation of urban functions, making it easier to integrate into downstream tasks such as transfer learning. Hierarchical clustering provides an unsupervised approach to grouping cities, but it lacks a structured feature representation for downstream tasks. Our 19-dimensional POI representation enables direct vector-based similarity computations (e.g., cosine similarity), facilitating seamless integration into machine learning models.

To evaluate these approaches, we conduct experiments on a dataset of 60 cities and compare their similarity scores with real-world urban attributes (e.g., GDP, demographic distribution, land use). The results show that: Our 19-dimensional POI feature representation achieves a cosine similarity correlation of 0.82 with urban socio-economic indicators, outperforming hierarchical clustering (0.76) and graph-based methods (0.79). Graph-based methods require significantly more computation time due to graph construction and embedding training. Hierarchical clustering lacks a structured feature representation, making it less effective for predictive modeling. These findings demonstrate that our 19-dimensional POI representation effectively captures city similarity with high interpretability and computational efficiency, making it a practical choice for large-scale urban studies.

As shown in Fig. 3, we first extract the center point of each city according to Baidu Map. Then the fuzzy rectangle (60km * 60km) of each city can be uniquely determined. Next, grid each city rectangle using a 2km * 2km rectangle as the basic unit. And after POI is captured, we obtained 19-dimensional (corresponding to every 19 types of POI data) 30 * 30 city POI data maps of every city.

To obtain detailed information on various local data in the city, we need an extraction grid to extract POI information on the city POI data map. To determine the optimal size of the extraction grid, we will use six types of grids to extract POI information, and accordingly select the most suitable one based on the thermal map. The specific process is shown in Fig. 4. Fine-grained grids: (3 * 3, 2 * 2) → Higher spatial resolution, capturing more local variations. Medium grids: (6 * 6, 5 * 5) → Balanced trade-off between granularity and density. Coarse grids: (15 * 15, 10 * 10) → Lower resolution, capturing only macro-scale city structures. As shown in Fig. 4, taking the extraction grids of 10 * 10 and 15 * 15 as examples, we will randomly select a portion of cities and perform feature extraction on their city POI data maps. The outputs are 19-dimensional feature rectangles of the cities, with each feature unit containing a certain number of POI interest points.

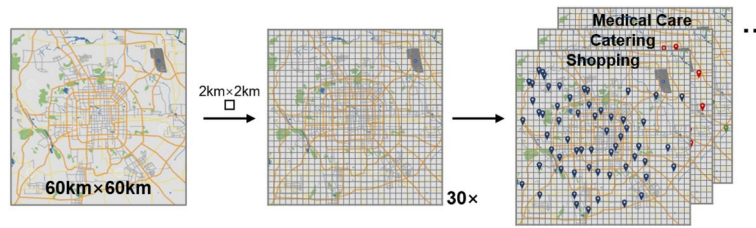


Fig. 3. City classification visualization.

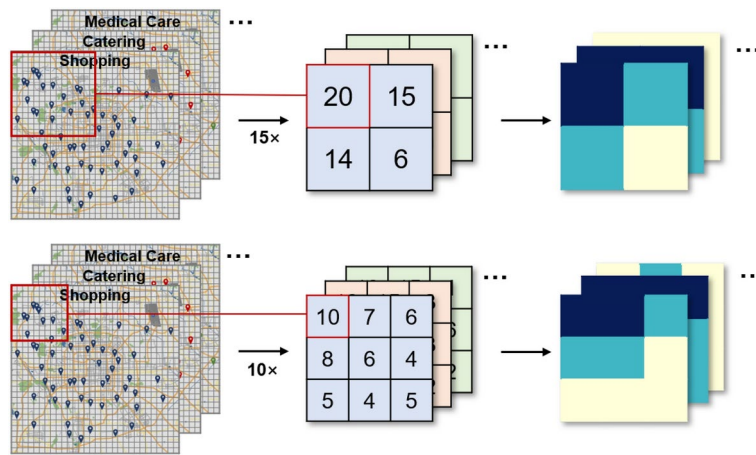


Fig. 4. City classification visualization 3D maps of 15×15 , 10×10 .

We draw POI thermal maps of 19 dimensions based on the number of interest points (different feature extraction grids correspond to different series of thermal maps). By comparing the thermal maps drawn based on different feature extraction grids, we can determine the optimal size of the feature extraction grid. We use the cosine similarity to measure the effectiveness of judging similarity in thermal maps.

Cluster and classify

After preprocessing the POI data, we obtain the data feature matrix for every city, which is the input in this block to classify cities using cluster algorithms. Firstly, we use the Elbow-SSE method (Eq. (1)) to determine how many types of cities should be classified (i.e., determine the value of k in the K-means algorithm). Notably, considering the wide variety of POI types, we need to determine the optimal value of k based on different POI types.

$$SSE = \sum_{i=1}^k \sum_{p \in C_i} |p - m_i|^2 \quad (1)$$

In Eq. (1), C is the clusters of high-similarity cities, and C_i is the i^{th} cluster. We consider a set of sample points denoted by p in the cluster C_i . The centroid of cluster C_i is denoted by m_i , and the clustering error of all samples is denoted by SSE, which represents the quality of clustering performance. The value of k increases beyond the true number of clusters, the degree of aggregation for each cluster is significantly affected, leading to a decrease in SSE. Therefore, the decrease in SSE will decrease sharply and then tend to flatten out as the value of k continues to increase. So we can get the optimal value of k .

After determining k , we will cluster cities. The detailed calculation process is illustrated in Algorithm 1.

Ensure: The outcome of urban categorization: Res .

Require: Here, P denotes a set of POIs, C represents a set of cities, θ_C signifies city categorization units in the set C , and α_C denotes city characteristics in the set C . The Cluster number: n . E_p represents the set of instances assigned to cluster p , and y denotes an instance in this set.

```

1: Arbitrarily choose  $k$  instances from  $\theta_C$  as the centroid set  $\{\Omega_1, \Omega_2, \Omega_3, \dots, \Omega_n\}$ 
2:  $G_i = \emptyset (1 \leq i \leq n)$ 
3: for  $p \in \{1, 2, 3, \dots, m\}$  do
4:   Compute the distance between instance  $x_p$  and each centroid  $\Omega_i (1 \leq i \leq k)$ :  $d_{pi} = \|x_p - \Omega_i\|_2$ 
5:   Determine the cluster label of  $x_p$ :  $\beta_p = \arg \min_{i \in \{1, 2, 3, \dots, k\}} d_{pi}$ 
6:   Assign the instance  $x_p$  to its respective cluster:  $G_{\beta_p} = G_{\beta_p} \cup \{x_p\}$ 
7: for  $p \in \{1, 2, 3, \dots, m\}$  do
8:   Compute the updated centroid vector:  $\psi_p' = \frac{1}{E_p} \sum_{y \in E_p} y$ 
9:   if  $\psi_p' \neq \psi_p$  then
10:     $\psi_p = \psi_p'$ 
11: Until: All current centroid vectors remain constant
12:  $\alpha_C = \{C_1, C_2, C_3, \dots, C_k\}$ 
13: repeat
14: Arbitrarily choose  $k$  instances from  $\theta_C$  as the centroid set  $\{\Omega_1, \Omega_2, \Omega_3, \dots, \Omega_n\}$ 
15:  $G_i = \emptyset (1 \leq i \leq n)$ 
16: for  $p \in \{1, 2, 3, \dots, m\}$  do
17:   Compute the distance between instance  $x_p$  and each centroid  $\Omega_i (1 \leq i \leq k)$ :  $d_{pi} = \|x_p - \Omega_i\|_2$ 
18:   Determine the cluster label of  $x_p$ :  $\beta_p = \arg \min_{i \in \{1, 2, 3, \dots, k\}} d_{pi}$ 
19:   Assign the instance  $x_p$  to its respective cluster:  $G_{\beta_p} = G_{\beta_p} \cup \{x_p\}$ 
20: for  $p \in \{1, 2, 3, \dots, m\}$  do
21:   Compute the updated centroid vector:  $\psi_p' = \frac{1}{E_p} \sum_{y \in E_p} y$ 
22:   if  $\psi_p' \neq \psi_p$  then
23:     $\psi_p = \psi_p'$ 
24: Until: All current centroid vectors remain constant
25:  $Res = \{C_1, C_2, C_3, \dots, C_k\}$ 
26: return  $Res$ 

```

Algorithm 1. Urban categorization ($P, C, \theta_C, \alpha_C, k$).

Adaptation transfer learning module

The POI of the source and target cities have different distributions, which can affect the adaptability and transfer effectiveness between cities. In fact, the most effective knowledge in the transfer process is the similar features of the two cities, which guarantees the effectiveness of the positive transfer. Therefore, we aim to extract the same distribution of features to the source and target cities. Based on this, we propose the DA-ConvGRU model, containing the Feature Extractor Network and Adaptation Transfer Learning Framework. We define the input as $\{(x_S^i, y_S^i)\}_{i=1}^{N_S}$ and $\{(x_T^i)\}_{i=1}^{N_T}$, and the output as y_T and $d_i \in \{0, 1\}$. Y_T represents a certain type of missing POI data that the target city wants to predict, such as education, and X_t represents the existing POI data of the target city. Y_S is the POI data of the source city. N_S and N_T represent the total amount of POI data of the source and target city respectively, and different categories of POIs may have different N . When $d = 0$, the feature data is from the source domain, otherwise the target domain.

Feature extractor network

As shown in Fig. 2, we use a two-layer 1DCNN model combined with a two-layer GRU to achieve the extraction of time series data features. The first 1DCNN tries to extract low-dimensional features from data, and the second layer tries to further extract its main features from the output. After that, we use the max pooling layer to retain the main features of the data while reducing the dimension. Notably, dimension reduction can effectively reduce the computation and complexity, especially for processing long-series data, which will be more effective. Then, the time series features are further extracted by the GRU model using the advantage of its long-term memory. We use 1DCNN to do the first step of GRU input to reduce the dimension because for each step of GRU input, the dimension of feature is fixed, which is very favorable for 1DCNN, and the translation invariance of it will not destroy the time features in feature extraction before passing the result into GRU, which will make GRU more efficient. Therefore, the Feature Extraction model effectively extracts and utilizes the time series features of data.

During the forward pass, the H_f maps the data into a new feature space f , i.e., $f = H_f(x; \theta_f)$, which is a part of the input of the next part: Adaptation Transfer Learning Framework.

Adaptation transfer learning framework

The framework of Adaptive Transfer Learning consists of two blocks: Predictor H_y and Domain Classifier H_d . Their parameters are denoted as θ_y and θ_d correspondingly. After obtaining the feature space f from the previous step, the Predictor and Domain Classifier map f to the predicted y and domain label d_i through their mapping functions $H_y(f; \theta_y)$ and $H_d(f; \theta_d)$.

In Predictor, we denote the loss of the Predictor as $\mathcal{L}_y$. Our primary goal of training is to predict the y -value more accurately, i.e., to make the $\mathcal{L}_y$ much smaller.

Afterwards, we need to use Domain Classifier to achieve adaptability and effective transfer between cities, by mining the same feature distribution of source and target city. The original function of the Domain Classifier is to distinguish the source of data (source city or target city) by maximizing the differences between features of

different domains. But we hope that the feature distributions of the mapped two cities are as similar as possible to obtain the optimal predicted results. To achieve this goal, we try to use the domain classifier loss $\mathcal{L}_d$ as an auxiliary loss to estimate the degree of dissimilarity between domains. Thus, we combine $\mathcal{L}_y$ and $\mathcal{L}_d$ to form the final loss $\mathcal{L}$ shown in Eq. (2).

$$\begin{aligned}\mathcal{L}(\theta_f, \theta_y, \theta_d) &= \mathcal{L}_y(\theta_f, \theta_y) - \alpha \mathcal{L}_d(\theta_f, \theta_d) \\ &= \frac{1}{N_{\mathcal{S}}} \sum_{i=1}^{N_{\mathcal{S}}} \mathcal{L}_y^i(\theta_f, \theta_y) - \alpha \left(\frac{1}{N_{\mathcal{S}}} \sum_{i=1}^{N_{\mathcal{S}}} \mathcal{L}_d^i(\theta_f, \theta_d) + \frac{1}{N_{\mathcal{T}}} \sum_{i=1}^{N_{\mathcal{T}}} \mathcal{L}_d^i(\theta_f, \theta_d) \right)\end{aligned}\quad (2)$$

where $\mathcal{L}_d$ consists of classification losses in the source and target domain. To reduce the degree of influence of $\mathcal{L}_d$ on the optimization of model parameters relative to $\mathcal{L}_y$, we define $\alpha \in (0, 1)$. To minimize the loss $\mathcal{L}$, the $\mathcal{L}_y$ should be smaller and the $\mathcal{L}_d$ should be larger. Then the optimal value of the model parameters $\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_d$ can be expressed as shown in Eq. (3) and (4).

$$(\hat{\theta}_f, \hat{\theta}_y) = \arg \min_{\theta_f, \theta_y} L(\theta_f, \theta_y, \hat{\theta}_d) \quad (3)$$

$$\hat{\theta}_d = \arg \max_{\theta_d} L(\hat{\theta}_f, \hat{\theta}_y, \theta_d) \quad (4)$$

The expressions of $\mathcal{L}_y^i$ and $\mathcal{L}_d^i$ are shown in Eq. (5) and (6):

$$\mathcal{L}_y^i(\theta_f, \theta_y) = |\hat{y}_t^i - y_t^i|^p \quad (5)$$

$$\mathcal{L}_d^i(\theta_d, \theta_y) = -[d_i \log(\hat{d}_t^i) + (1 - d_i) \log(1 - \hat{d}_t^i)] \quad (6)$$

where $\mathcal{L}_y^i$ is the p th power of the absolute difference of the real value and the predicted one, when p is 1 corresponding to Mean Absolute Error and when p is 2 corresponding to Mean Squared Error. $\mathcal{L}_d^i$ is the domain labels' binary cross-loss quotient.

Besides, the Gradient Reversal Layer (GRL) is used to invert the gradient direction automatically in the process of backpropagation and achieve constant transformation in the forward propagation process. which can be represented as Eq. (7).

$$\begin{aligned}R_{\lambda}(x) &= x \\ \frac{dR_{\lambda}}{dx} &= -\lambda I\end{aligned}\quad (7)$$

where the parameter λ is not fixed, but changes dynamically. Its variation expression is shown in Eq. (8).

$$\lambda_p = \frac{2}{1 + \exp(-10 \cdot p)} - 1 \quad (8)$$

where p represents the relative process of iterative, i.e., the ratio of the current iteration number to the total. Small values of λ at early stages, allowing the feature extractor to focus on prediction accuracy. Gradually increasing λ over iterations, strengthening domain adaptation. This schedule prevents the gradient vanishing problem and stabilizes adversarial training. In the process of backpropagation, the gradient of the domain classifier's classification loss is inverted before backpropagating to the feature extractor's parameters automatically, thus achieving a GAN-like adversarial loss. We used ReLU as the activation function. The parameters are updated using stochastic gradient descent (SGD) in Eqs. (9)–(11):

$$\theta_f = \theta_f - \mu \left(\frac{\partial \mathcal{L}_y}{\partial \theta_f} - \lambda \frac{\partial \mathcal{L}_d}{\partial \theta_f} \right) \quad (9)$$

$$\theta_y = \theta_y - \mu \frac{\partial \mathcal{L}_y}{\partial \theta_y} \quad (10)$$

$$\theta_d = \theta_d - \mu \frac{\partial \mathcal{L}_d}{\partial \theta_d} \quad (11)$$

To improve stability, we dynamically adjust μ .

$$\mu(t) = \mu_0 \cdot e^{-\gamma t} \quad (12)$$

where μ_0 is the initial learning rate, γ is the decay factor, and t is the current epoch.

$$\mu(t+1) = \begin{cases} \mu(t) \cdot (1 + \eta), & \text{if } \mathcal{L}_d(t) > \mathcal{L}_d(t-1) \\ \mu(t) \cdot e^{-\gamma}, & \text{otherwise} \end{cases} \quad (13)$$

where η is a small factor controlling learning rate increase.

$$\mu(t) = \max(\mu(t), \mu_{\min}) \quad (14)$$

where $\mu_{\min}$ ensures the learning rate does not vanish. Integrating the adaptive $\mu(t)$:

$$\theta_f = \theta_f - \mu(t) \left(\frac{\partial \mathcal{L}_y}{\partial \theta_f} - \lambda \frac{\partial \mathcal{L}_d}{\partial \theta_f} \right) \quad (15)$$

$$\theta_y = \theta_y - \mu(t) \frac{\partial \mathcal{L}_y}{\partial \theta_y} \quad (16)$$

$$\theta_d = \theta_d - \mu(t) \frac{\partial \mathcal{L}_d}{\partial \theta_d} \quad (17)$$

where $\mu(t)$ follows:

$$\mu(t+1) = \max(\mu_{\min}, \mu(t) \cdot e^{-\gamma} + \eta \cdot 1(\mathcal{L}_d(t) > \mathcal{L}_d(t-1))) \quad (18)$$

The proposed ConvGRU + Domain Adaptation framework is designed to balance predictive performance and computational efficiency. Below, we discuss the computational characteristics of each component. ConvGRU is chosen over ConvLSTM due to its reduced parameter complexity. Specifically, ConvLSTM maintains separate memory and hidden states, leading to higher computational overhead. ConvGRU, by contrast, removes the memory cell mechanism, resulting in faster training and inference while preserving strong spatiotemporal modeling capabilities. The Adaptation Transfer Learning module, introduces a domain classifier with gradient reversal. This additional component incurs moderate computational overhead but significantly enhances cross-city generalization, as demonstrated in our experiments. We mitigate the overhead by sharing feature extraction layers between the label predictor and domain classifier, ensuring efficient resource utilization.

Experiment

In this section, we conduct a comprehensive evaluation of our proposed approach on real-world datasets. Specifically, we compare the effectiveness of our approach with several state-of-the-art methods for the task of cold-start express delivery outlets recommendation. Furthermore, we assess the effectiveness of the City Similarity module and analyze its contribution to the overall performance of our model. Additionally, we investigate the sensitivity of our proposed approach to hyper-parameters, thereby providing insights into the optimal parameter settings.

Experimental settings

Datasets and configurations

The datasets we use for the experiment consists of two parts: the real-word POI datasets for different cities and the real-word datasets from several express delivery enterprises. In the real-word POI datasets for different cities, we use the publicly available POI datasets of Baidu Maps API, which selects 60 cities of different levels based on the 2020 City Commercial Charm Ranking, and we collect POI data from 19 dimensions of these cities, covering various aspects of urban life, such as catering, shopping, finance, etc., totaling approximately 3 million POI data. These datasets are used by City Similarity module to group these cities with high-level similarity into the same cluster. In the real-word datasets from several express delivery enterprises, these datasets from express delivery enterprises, which owns a lot of express delivery outlets in multiple cities of China. In our experiments, we obtain 5 datasets that contain over 600 express delivery outlets from 5 different cities in China for 2019, 2020, 2021 and 2022, including 3 first-tier cities (i.e., Beijing, Shanghai and Guangzhou) and 2 new first-tier cities (i.e., Changsha and Foshan), respectively. The TransCSM model and its baselines are evaluated using POI data, comprising name and location, collected for each city. The performance of these models is assessed based on these datasets. For express delivery outlets prediction, we use express delivery outlets POI data from the source and target cities for the past 2 years (2019 and 2020) and from the source city for the next 2 years (2021 and 2022) to predict express delivery outlets POI data from the target city for the next 2 years (2021 and 2022), including the candidate location of each courier outlet in the target city. Eq. (19) is used to represent the error between the predicted and true values.

All experiments are conducted on 4 TITAN RTX GPUs. All baselines and the proposed model are performed on PyTorch-1.1.0 and CUDA-10.0. When making recommendations for express delivery outlets, our model is trained using the SGD optimizer with an initial learning rate of 0.05. The proposed models are trained for 50 epochs with a batch size of 32 and a domain discriminant loss factor (α) is 0.5 in Eq. (2). Additionally, a time window size of 12 is used to ensure effective forecasting accuracy. The training time of our TransCSM model in this experiment is a total of 36 hours.

Evaluation metrics

Our model predicts the candidate locations for each express delivery outlet in the target city. Therefore, we calculated the Euclidean distance between the predicted express delivery outlet coordinates and the nearest real express delivery outlet coordinates, where Y_i and $\hat{Y}_i$ are ground truth and predicted express delivery outlet coordinates, i.e., Eq. (19) :

$$D(Y_i, \hat{Y}_i) = \sqrt{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \tag{19}$$

Mean Absolute Error (MAE) directly reflects the average magnitude of prediction errors, making it more interpretable in real-world applications. Since MAE treats all errors equally, it is less sensitive to outliers, making it a reliable metric when large deviations are not the primary concern. MAE is particularly useful in scenarios where consistent, stable performance across all samples is desired, such as urban demand forecasting or logistics network optimization, where extreme prediction errors are less critical. Root Mean Square Error (RMSE) penalizes larger errors more heavily due to squaring, making it sensitive to extreme deviations. While RMSE provides a more risk-aware evaluation, it can be influenced disproportionately by a few large errors, potentially distorting overall performance assessment. Given the characteristics of our prediction task, we adopt MAE as the primary metric, as it provides a stable and interpretable measure of model accuracy without being overly influenced by outliers. However, we also report RMSE to capture the impact of large prediction errors, ensuring our model performs well not only on average cases but also in extreme situations. In our experiments, we adopt MAE and RMSE as the evaluation metrics, where m is defined as the number of samples, as shown in Eq. (20,21)

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^m |D(Y_i, \hat{Y}_i)| \tag{20}$$

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (D(Y_i, \hat{Y}_i))^2} \tag{21}$$

Baseline models

We conduct a comparative analysis of the proposed model against two distinct categories of approaches, namely non-transfer learning and transfer learning methods. Within the non-transfer learning category, we utilize T-GCN, ASTGCN, LSTM, and XGBoost as our baseline models. On the other hand, within the transfer learning category, we select DANN, RegionTrans, and MetaST as our benchmark models.

- T-GCN³⁵: A temporal graph convolutional network model combining graph convolutional network (GCN) with gated recurrent unit (GRU).
- ASTGCN³⁶: A Spatial-Temporal Graph Convolution Network model based on a self-attention, which improves its ability to learn from graph-structured data.
- LSTM³⁷: A special variant of Recurrent Neural Network with a “gate” structure, which determines whether data is updated or discarded through the logical control of gate units, effectively improving prediction accuracy.
- XGBoost³⁸: A gradient-boosting algorithm that uses multiple trees to make predictions together, and finally adds up the scores of each tree as the predicted value.
- DANN³⁹: A representative approach for domain adaptation through adversarial learning, which entails a feature extractor, label predictor, and domain classifier.
- RegionTrans⁹: A transferable model for effectively matching source cities with corresponding regions in target cities.
- MetaST⁴⁰: A spatial-temporal network architecture that employs a meta learning paradigm to facilitate adaptation to the target city and enable knowledge transfer.
- TL_DCRNN⁴¹: A transfer model based on the Diffusion Convolutional Recurrent Neural Networks model, which transfers knowledge between the source city and the target city. The resulting training model can be used to predict traffic on unknown networks, etc.
- CNN-LSTM⁴²: A hybrid deep learning model that combines Convolutional Neural Networks (CNNs) for spatial feature extraction with Long Short-Term Memory (LSTM) networks for temporal modeling.
- Transformer-based Model⁴³: A deep learning model leveraging self-attention mechanisms to capture long-range dependencies in sequential data.

Experimental results

As illustrated in Table 1, we employed a clustering approach to group 60 cities using various basic classification units, namely 15 * 15, 10 * 10, 6 * 6, and 3 * 3. The K values used for clustering are set to 3.

	15 * 15	10 * 10	6 * 6	3 * 3
class 0	13	25	10	30
class 1	24	11	24	20
class 2	23	24	26	10

Table 1. POI City Clustering Result Statistics.

Through the heat map of urban features, we selected 10 cities from 60 cities as sample displays, and the classification and feature extraction results of these 10 cities are visualized as shown in Figure 5. The basic classification units of 3×3 , 6×6 , 10×10 , and 15×15 . We compute the cosine similarity scores for different grid sizes and analyze their stability across multiple city pairs. The cosine similarity between city POI heatmaps is calculated as follows:

$$\text{Cosine Similarity} = \frac{\sum_{i=1}^{19} H_A^i \cdot H_B^i}{\sqrt{\sum_{i=1}^{19} (H_A^i)^2} \cdot \sqrt{\sum_{i=1}^{19} (H_B^i)^2}} \quad (22)$$

where H_A^i and H_B^i represent the POI density of the i -th POI category in cities A and B , respectively. To determine the optimal grid size, we evaluate cosine similarity scores across different grid resolutions. The results are summarized in the Table 3 below:

The utilization of a basic classification unit of 15×15 has been found to pose significant limitations in effectively discerning urban features, this inadequacy in discrimination can be attributed to the unit's relatively large size. A 15×15 grid will result in unclear features, resulting in fewer categories in urban clustering, and thus affecting the differentiation of urban similarity. The 3×3 basic classification unit leading to the urban characteristics are too independent and scattered. The 10×10 and 6×6 basic classification units are relatively moderate, but the 6×6 manifested urban features are still relatively complex. Grids of 3×3 and 6×6 can lead to features that are too independent and scattered, making it difficult to extract complete city features, resulting in an increase in the number of categories in city clustering. This also affects the effectiveness of city clustering, thereby affecting the overall efficiency of transfer learning. From this analysis, we select 10×10 as the optimal grid size because it provides stable similarity measurements (low variance) while preserving urban functional details.

Here we focus on the 10×10 classification results demonstrated in Fig. 5d–f. The number of cities in class 0, class 1, and class 2 are 25, 11, and 24 (see in Table 2), respectively. We present experimental results for three cities: Chenzhou, Beijing, and Kunming, representing third-tier, first-tier, and second-tier cities, respectively. The darker the color in the heat map of urban features, the more concentrated the distribution of POI data in this area. Both Chenzhou and Kunming's POI data are concentrated in the central part of the distribution. So our analysis reveals that both Chenzhou and Kunming are centrally developed cities, albeit with different degrees of centralization. Specifically, Chenzhou exhibits a weaker degree of centralization compared to Kunming. The observed results are consistent with the actual development of the cities. In contrast, Beijing, one of the most

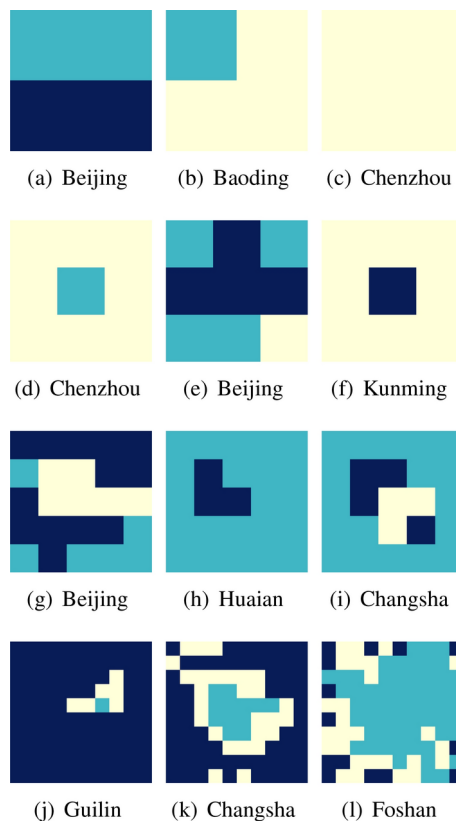


Fig. 5. Visualization of city characteristics of 15×15 (a–c), 10×10 (d–f), 6×6 (g–i), 3×3 (j–l).

City category	Cities
Class 0	Chengdu Hangzhou Chongqing Handan Hengyang Huaian Tianjin Nanjing Changsha Zhengzhou Yinchuan Zhanjiang Lasa Xining Foshan Shanghai Nanchang Taiyuan Lianyungang Linyi Zhongshan Nanning Changchun Shijiazhuang Guiyang
Class 1	Beijing Guangzhou Shenzhen Lanzhou Haikou Shantou Baoding Xiamen Fuzhou Haerbin Jinan
Class 2	Wuhan Xian Suzhou Chenzhou Kunming Guilin Huhehaote Jilin Ningbo Ganzhou Meizhou Nanping Sanming Shangrao Wenzhou Tangshan Weifang Wulumuqi Yichang Xinxiang Dongguan Qingdao Shenyang Hefei

Table 2. Collection Details.

Grid Size	Avg. Cosine Similarity	Variance
15 × 15	0.94	0.05
10 × 10	0.81	0.04
5 × 5	0.80	0.07
6 × 6	0.79	0.08
3 × 3	0.79	0.09
2 × 2	0.77	0.12

Table 3. Comparison of Cosine Similarity Scores for Different Grid Sizes. Significant values are in bold.

		Beijing to Guangzhou		Beijing to Guangzhou		Changsha to Foshan		Changsha to Foshan		Wuhan to Xian		Wuhan to Xian	
		2021		2022		2021		2022		2021		2022	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Non-transfer	T-GCN	0.62	0.71	0.72	0.81	0.45	0.63	0.55	0.94	0.51	0.62	0.52	0.84
	ASTGCN	0.68	0.78	0.63	0.74	0.58	0.92	0.50	0.92	0.53	0.82	0.41	0.91
	LSTM	0.75	0.87	0.92	1.01	0.49	0.89	0.50	0.72	0.51	0.76	0.52	0.74
	XGBoost	0.51	0.60	0.41	0.62	0.42	0.88	0.37	0.65	0.51	0.84	0.33	0.64
	CNN-LSTM	0.66	0.61	0.51	0.82	0.42	0.78	0.39	0.66	0.43	0.62	0.49	0.62
	Transformer-based Model	0.75	0.66	0.48	0.69	0.48	0.89	0.30	0.75	0.49	0.73	0.36	0.65
transfer	MetaST	0.54	0.73	0.43	0.66	0.43	0.73	0.48	0.87	0.44	0.62	0.58	0.72
	RegionTrans	0.50	0.72	0.55	0.73	0.55	0.81	0.44	0.87	0.56	0.72	0.47	0.89
	TL_DCRNN	0.48	0.58	0.38	0.60	0.48	0.85	0.36	0.66	0.47	0.72	0.25	0.73
	DANN	0.40	0.66	0.42	0.64	0.58	0.87	0.40	0.86	0.54	0.89	0.47	0.76
	TransCSM(ours)	0.27	0.56	0.30	0.46	0.36	0.61	0.28	0.64	0.33	0.59	0.23	0.60
	TransCSM(w/o ConvGRU)	0.43	0.64	0.34	0.57	0.41	0.67	0.32	0.53	0.43	0.69	0.41	0.62
	TransCSM(w/o DA)	0.45	0.87	0.39	0.63	0.45	0.74	0.38	0.55	0.47	0.77	0.43	0.61

Table 4. The Performance of Baseline Approaches. Significant values are in bold.

developed cities in China, exhibits high levels of development in both the central and surrounding areas, as demonstrated by the heat map.

We found that the clustering results do not exhibit a correlation with the tier classification of cities. Instead, the results are primarily determined by factors such as POI distribution rather than city tier. City tier classification is primarily based on economic indicators, population size, and administrative importance, whereas our clustering methodology focuses on spatial and functional similarities derived from POI distributions. Therefore, two cities with similar urban functionalities (e.g., commercTCAial and residential area distributions) may be grouped together, even if they belong to different tiers.

Model performance comparison

This experiment aims to explore the effectiveness of transfer learning in solving six transfer tasks, namely Beijing to Guangzhou(2021), Beijing to Guangzhou(2022), Changsha to Foshan(2021), Changsha to Foshan(2022), Wuhan to Xian(2021), and Wuhan to Xian(2022). According to our city clustering results (Table 2), both Beijing and Guangzhou belong to class 1, Changsha and Foshan belong to class 0 ,Wuhan and Xian belong to class 2. Therefore, we choose to conduct transfer learning between cities of the same class to demonstrate the effectiveness of our proposed transfer learning framework based on city similarity. The performance of the prediction model is evaluated using MAE as the major metric and RMSE as an auxiliary metric. The comparison results with various baselines are presented in Table 4, highlighting the best results in bold. It is observed that the proposed TransCSM outperforms all the compared baselines by achieving the lowest MAE and RMSE. However, the results show that both MAE and RMSE are slightly inferior to the best non-transfer model XGBoost.

Compared with the non-transfer models, TransCSM obtains additional information from source city data in training and reduce the impact of defective prior knowledge in the target city's data. We also compare DANN with other methods and observe that DANN slightly outperforms other methods in MAE and RMSE. We conclude that domain adaptation is effective for data transfer between two cities. In addition, DANN and our model also adopt the strategy of Domain Adaptation and indicates that adaptation transfer in our framework is effective to learn the domain-invariant feature representation for data transfer.

In terms of the transfer models, Non-Transfer models are overall inferior to the transfer model, except for XGBoost. XGBoost can learn the time characteristics of a single express delivery outlet from a single node, thereby obtaining valuable predictions. However, long intervals between data may hinder XGBoost's ability to accurately predict future express delivery outlets in target cities. Therefore, it is evident that TransCSM exhibits superior performance in comparison to XGBoost when it comes to learning the features of source cities and effectively leveraging the time information of target cities. The experimental results of CNN-LSTM and Transformer-based models are worse than those of TransCSM (w/o DA). Compared with CNN-LSTM and Transformer-based models, ConvGRU offers the following advantages. Efficient Spatial-Temporal Encoding: The convolutional structure extracts local spatial correlations, while GRU captures temporal dependencies with fewer parameters than LSTM. Reduced Computational Complexity: Unlike LSTM, which requires additional forget gates, GRU is computationally more efficient, making it suitable for large-scale urban datasets. Scalability and Stability: Transformer-based models perform well in capturing long-range dependencies, but require quadratic complexity in sequence length.

TransCSM outperforms all three baselines, namely, MetaST, RegionTrans and DANN. Compared with MetaST, RegionTrans and DANN, we conduct prediction tasks for express delivery outlets within the same city cluster and use CNN and GRU to extract more representative features of time series data, which improves the reliability of cross-city data transfer and the predictive ability of the model. Furthermore, our TransCSM method shows superior performance over the non-transfer model with regards to prediction stability. The prediction of express delivery outlets in a target city necessitates a significant amount of historical data for the extraction of long-term data dependencies. Nonetheless, acquiring such historical data for the target city is a daunting task due to its scarcity. Embedding the urban similarity into an adaptation transfer learning framework to achieve desired data transfer, which cannot be utilized in other baseline.

TCA projects the source and target domain data into a shared latent space, minimizing domain discrepancy while preserving task-related information. The transferability score is computed as:

$$\mathcal{T}_{\text{score}} = \frac{\text{Similarity}(\mathbf{Z}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{T}})}{\text{Variance}(\mathbf{Z}_{\mathcal{S}}) + \text{Variance}(\mathbf{Z}_{\mathcal{T}})} \tag{23}$$

where: $\mathbf{Z}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{T}}$ are the feature representations after TCA transformation. Higher TCA scores indicate better domain alignment and less negative transfer.

Based on the city clustering results in Table 2, we know that Beijing and Changsha belong to class 1 and class 0 respectively, while both Shanghai and Changsha belong to class 0. Beijing to Changsha suffers from the strongest negative transfer effect in Table 5, suggesting significant feature distribution mismatch. The results demonstrate that transfer learning between cities of the same type can effectively reduce negative transfer.

Ablation study

Three components comprise TransCSM, where City Similarity(CS) enables transfer between similar cities, ConvGRU extracts representative features of time series data, and Domain Adaptation(DA) contains a domain adversarial neural network to achieve cross-city adaptive transfer. Consequently, this investigation introduces two novel TransCSM variations.

- TransCSM(w/o CS): The transfer task conduct within the dissimilar city cluster.
- TransCSM(w/o ConvGRU): The adaptation transfer module removes ConvGRU module.
- TransCSM(w/o DA): The adaptation transfer module removes domain adaptation module.

The results of TransCSM(w/o ConvGRU) and TransCSM(w/o DA) are shown in Table 4. Without DA or ConvGRU blocks, the performance of the model falls short of TransCSM. Nevertheless, TransCSM (w/o ConvGRU) and TransCSM (w/o DA) still perform better than non-transfer models in all four transfer tasks. The Improvement of transfer efficiency attributes to the availability data from the source city as learning samples for predicting express delivery outlets in the target city. The performance of TransCSM is superior to TransCSM (w/o ConvGRU), indicating that the time series information extracted by ConvGRU module improves the predictive ability of express delivery outlets. The performance of TransCSM is superior to TransCSM (w/o DA), indicating that adversarial neural network can help achieve adaptive transfer of cross-city.

Based on the city clustering results in Table 2, We designed TransCSM (w/o CS) for transfer between different types of cities, including the following cases: class 1 to class 0, class 0 to class 1, class 0 to class 2, class 1 to class

City Pair	TCA Score
Shanghai to Changsha	0.78
Beijing to Changsha	0.56

Table 5. Negative Transfer Analysis using TCA.

		MAE	RMSE
TransCSM(w/o CS)	Beijing to Changsha	0.44	0.72
TransCSM(w/o CS)	Chengdu to Beijing	0.48	0.69
TransCSM(w/o CS)	Chengdu to Wuhan	0.38	0.61
TransCSM(w/o CS)	Beijing to Wuhan	0.43	0.64
TransCSM(w/o CS)	Wuhan to Hangzhou	0.29	0.57
TransCSM(w/o CS)	Wuhan to Guangzhou	0.41	0.58
TransCSM	Shanghai to Changsha	0.25	0.38

Table 6. Performance Comparison of TransCSM and Its Variants in 2021. Significant values are in bold.

		MAE	RMSE
TransCSM(w/o CS)	Beijing to Changsha	0.50	0.74
TransCSM(w/o CS)	Chengdu to Beijing	0.43	0.62
TransCSM(w/o CS)	Chengdu to Wuhan	0.32	0.62
TransCSM(w/o CS)	Beijing to Wuhan	0.51	0.66
TransCSM(w/o CS)	Wuhan to Hangzhou	0.31	0.59
TransCSM(w/o CS)	Wuhan to Guangzhou	0.47	0.59
TransCSM	Shanghai to Changsha	0.21	0.33

Table 7. Performance Comparison of TransCSM and Its Variants in 2022. Significant values are in bold.

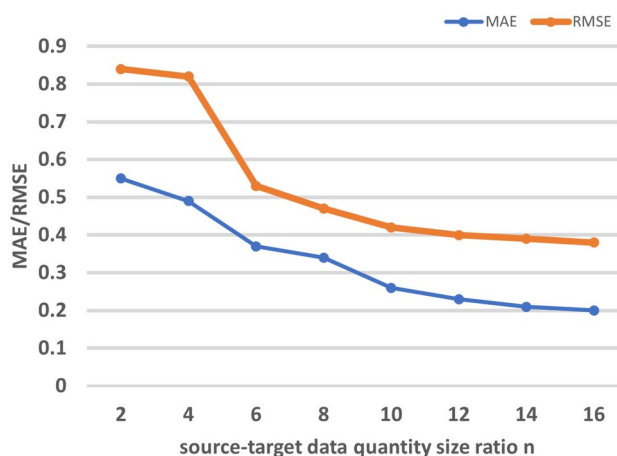


Fig. 6. The performance indicators under different ratios of source target data volume.

2, class 2 to class 0 and class 2 to class 1. Table 6 shows the results of TransCSM (w/o CS) and TransCSM in 2021, and Table 7 shows the predicted results of TransCSM (w/o CS) and TransCSM in 2022. The performance of TransCSM is superior to TransCSM (w/o CS), indicating that Data transfer within the different city cluster will decrease the performance of transfer. We employ domain adaptation techniques to align feature distributions between the source and target cities, enabling better generalization in cross-city knowledge transfer. Specifically, by integrating a domain classifier with gradient reversal, our model learns domain-invariant features, mitigating the negative effects of distribution shifts. Experimental results further confirm that domain adaptation significantly improves model transferability compared to direct transfer without adaptation, highlighting the necessity of incorporating domain adaptation mechanisms for effective urban knowledge transfer.

Transferability of source data quantity

Determining the optimal quantity of data from the source city required for effective transfer learning in the target city becomes crucial. Identifying this appropriate data volume could potentially reduce the model's computational complexity and enhance its overall efficiency. In this study, we examine the impact of the source-target data quantity size ratio n on transfer performance.

As depicted in Fig. 6, it can be observed that both MAE and RMSE exhibit a decline prior to $n = 10$, with the rate of decrease diminishing subsequently. This suggests that when the “richness” of the source data reaches saturation, an additional influx of source data does not contribute to enhancing transferability. Furthermore,

augmenting n necessitates a larger volume of training data. Nonetheless, as demonstrated in Fig. 6, the escalation of n does not correspond to a uniform linear enhancement in performance, thereby giving rise to superfluous computations.

Limitation and future works

Our current evaluation is based on the Baidu POI dataset, which, while comprehensive for Chinese cities, may limit the generalizability of our findings. Future work will incorporate additional datasets, including: Amap POI (China) for cross-validation within the same geographic region. OpenStreetMap (OSM) POI (Global) to assess model performance across different continents. Yelp POI (USA) to examine economic disparities in urban distributions. A broader dataset will enable a more comprehensive evaluation of the model's adaptability across different geographic and economic environments.

Our experiments focus solely on express delivery prediction, a task that is closely related to urban infrastructure and can be effectively analyzed using POI data. However, this single-task setting may limit the generalizability of our approach in real-world applications. Future work should explore additional urban computing tasks, such as traffic flow prediction, business district identification, and emergency service optimization, to further validate the adaptability of our method. While POI similarity is a key factor in cross-city knowledge transfer, our findings suggest that urban development models (e.g., centralized vs. decentralized city structures) may also impact the transferability of knowledge between cities. Future research should incorporate explicit urban morphology indicators, such as road network density, land use patterns, and population distribution, to better quantify these effects.

Conclusion

In this article, we first propose a similarity-based cross-city transfer learning method named TransCSM, which embeds urban similarity into an adaptation transfer learning framework to achieve desired data transfer. TransCSM constructed an urban similarity model, which utilizes the urban POI data to group the cities with similar characteristics into the same cluster. Afterwards, a feature extractor network(ConvGRU) is proposed to use CNN and GRU to extract more representative features of time series data. Then, we build an adaptation transfer learning framework to achieve data transfer within the same city cluster, which ensures the reliability of cross-city data transferring results.

In assessing the efficacy of TransCSM, we undertake a city similarity clustering analysis utilizing the urban Points of Interest (POI) dataset, followed by site selection recommendation tasks employing real-world express delivery outlet datasets. A comparison with contemporary state-of-the-art benchmarks demonstrates the enhanced performance of TransCSM over extant methodologies. Moreover, we scrutinize the performance of distinct TransCSM variants to discern the influence of individual modules on the model's transferability. Concomitantly, we explore the source-target data quantity size ratio in the transfer process with the objective of diminishing the computational workload.

In forthcoming research endeavors, we aim to investigate alternative transfer learning methodologies by examining aspects such as city similarity and feature extraction, with the ultimate objective of developing a comprehensive solution for transfer recommendation predicated on city similarity. We seek to foster collaboration with organizations possessing a substantial number of express delivery outlets, thereby acquiring sufficient data to substantiate the efficacy of our proposed framework.

Data availability

The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Received: 14 January 2025; Accepted: 18 March 2025

Published online: 28 March 2025

References

1. Zheng, Y., Capra, L., Wolfson, O. & Yang, H. Urban computing: concepts, methodologies, and applications. *ACM Trans. Intel. Syst. Technol.* 38 (2014).
2. Dai, S. & Meng, F. Addressing modern and practical challenges in machine learning: A survey of online federated and transfer learning. *Appl. Intell.* 53, 11045–11072 (2023).
3. Liu, Y. et al. A deep learning-based acute coronary syndrome-related disease classification method: A cohort study for network interpretability and transfer learning. *Appl. Intell.* 53, 25562–25580 (2023).
4. Zanella, A., Bui, N., Castellani, A., Vangelista, L. & Zorzi, M. Internet of things for smart cities. *IEEE Internet of Things J.* 22–32 (2014).
5. Wang, P., Lai, J., Huang, Z., Tan, Q. & Lin, T. Estimating traffic flow in large road networks based on multi-source traffic data. *IEEE Trans. Intel. Transp. Syst.* 1–12 (2020).
6. Panda, D. K. & Ray, S. Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review. *J. Intell. Inf. Syst.* 59, 341–366 (2022).
7. Guo, B., Li, J., Zheng, V. W., Wang, Z. & Yu, Z. Citytransfer: Transferring inter- and intra-city knowledge for chain store site recommendation based on multi-source urban data. *Proc. ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (2018).
8. Huang, J., Wen, J., Chen, L. & Liu, H.-L. Transfer learning based evolutionary algorithm framework for multi-objective optimization problems. *Appl. Intell.* 53, 18085–18104 (2023).
9. Wang, L., Geng, X., Ma, X., Liu, F. & Yang, Q. Cross-city transfer learning for deep spatio-temporal prediction. Preprint at [arXiv:1802.00386](https://arxiv.org/abs/1802.00386) (2018).
10. Huang, Y., Song, X., Zhang, S. & Yu, J. J. Transfer learning in traffic prediction with graph neural networks. In: *2021 IEEE International Intelligent Transportation Systems Conference*, 3732–3737 (2021).

11. Wei, Y., Zheng, Y. & Yang, Q. Transfer knowledge between cities. In *Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 1905–1914 (2016).
12. Badfar, M., Chinnam, R. B. & Yildirim, M. Enhancing lithium-ion battery state-of-charge estimation across battery types via unsupervised domain adaptation. *PHM Soc. Eur. Conf.* **8**, 8–8 (2024).
13. Sheikhpour, R., Berahmand, K., Mohammadi, M. & Khosravi, H. Sparse feature selection using hypergraph laplacian-based semi-supervised discriminant analysis. *Pattern Recogn.* **157**, 110882 (2025).
14. Wang, L., Guo, B. & Yang, Q. Smart city development with urban transfer learning. *Computer* 32–41 (2018).
15. Yan, H., Liu, J., Li, Y., Jin, D. & Chen, S. Spatial popularity and similarity of watching videos in a large city. In: *2016 IEEE Global Communications Conference*, 1–6 (2016).
16. Santini, S. & Jain, R. Similarity measures. *IEEE Transactions on pattern analysis and machine Intelligence* 871–883 (1999).
17. Mulligan, G. F. & Crampton, J. P. Population growth in the world's largest cities. *Cities* (2005).
18. Crucitti, P., Latora, V. & Porta, S. Centrality measures in spatial networks of urban streets. *Phys. Rev. E* (2006).
19. Preoțiuc-Pietro, D., Cranshaw, J. & Yano, T. Exploring venue-based city-to-city similarity measures. In: *Proc. 2nd ACM SIGKDD International Workshop on Urban Computing*, 16 (2013).
20. McKenzie, G. Shared micro-mobility patterns as measures of city similarity: Position paper. In: *Proc. 1st ACM SIGSPATIAL International Workshop on Computing with Multifaceted Movement Data*, 1–4 (2019).
21. Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L. & Muller, P. A. Transfer learning for time series classification. *2018 IEEE international conference on big data (Big Data)* (2018).
22. Li, H. L., Liang, Y. & Wang, S. C. Review on dynamic time warping in time series data mining. *Control and Decision* 1345–1353 (2018).
23. Jung, S. M., Park, S., Jung, S. W. & Hwang, E. Monthly electric load forecasting using transfer learning for smart cities. *Sustainability* (2020).
24. Zeng, P., Sheng, C. & Jin, M. A learning framework based on weighted knowledge transfer for holiday load forecasting. *J. Modern Power Syst. Clean Energy* (2018).
25. Cao, L. & Song, Y. Robust transfer learning for high-dimensional quantile regression model with linear constraints. *Appl. Intell.* 1–12 (2024).
26. Gautam, Y. Transfer learning for covid-19 cases and deaths forecast using lstm network. *ISA Transactions* (2021).
27. Xin, B. & Peng, W. Prediction for chaotic time series-based ae-cnn and transfer learning. *Complexity* 2680480 (2020).
28. Hu, Q., Zhang, R. & Zhou, Y. Transfer learning for short-term wind speed prediction with deep neural networks. *Renew. Energy* 83–95 (2016).
29. Wan, J. & Zhu, S. Cross-city crash severity analysis with cost-sensitive transfer learning algorithm. *Expert Syst. Appl.* (2022).
30. Bappee, F. K., Soares, A., Petry, L. M. & Matwin, S. Examining the impact of cross-domain learning on crime prediction. *J. Big Data* (2021).
31. Yang, T., Yu, X., Ma, N., Zhao, Y. & Li, H. A novel domain adaptive deep recurrent network for multivariate time series prediction. *Eng. Appl. Artif. Intell.* 104498 (2021).
32. Costa, P., Akcay, A., Zhang, Y. & Kaymak, U. Remaining useful lifetime prediction via deep domain adaptation. *Reliab. Eng. Syst. Saf.* (2020).
33. Wen, B. C., Xiao, M. Q., Wang, X. Q. & Chen, X. *Data-driven remaining useful life prediction based on domain adaptation* (PeerJ, Computer science, 2021).
34. Wilson, G., Doppa, J. R. & Cook, D. Multi-source deep domain adaptation with weak supervision for time-series sensor data (2020).
35. Zhao, L., Song, Y., Zhang, C., Liu, Y. & Li, H. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* 1–11 (2019).
36. Guo, S., Lin, Y., Feng, N., Song, C. & Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: *Proc. AAAI conference on artificial intelligence*, 922–929 (2019).
37. Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. Preprint at [arXiv:1412.3555](https://arxiv.org/abs/1412.3555) (2014).
38. Chen, T. & Guestrin, C. Xgboost: A scalable tree boosting system. *Proc. 22nd acm sigkdd international conference on knowledge discovery and data mining* 785–794 (2016).
39. Ganin, Y. & Lempitsky, V. Unsupervised domain adaptation by backpropagation (2015).
40. Yao, H., Liu, Y., Wei, Y., Tang, X. & Li, Z. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. *The World Wide Web Conference* 2181–2191 (2019).
41. Mallick, T., Balaprakash, P., Rask, E. & Macfarlane, J. Transfer learning with graph neural networks for short-term highway traffic forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)* (2021).
42. Shi, X. et al. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *arXiv: Computer Vision and Pattern Recognition*, *arXiv: Computer Vision and Pattern Recognition* (2015).
43. Vaswani, A. et al. Attention is all you need. *Advances in neural information processing systems* **30** (2017).

Acknowledgements

This paper is supported by Science and Technology Research and Development Program of China National Railway Group Corporation Limited (N2023S027) and Guizhou Provincial Basic Research Program (Natural Science) Qiankehe Foundation -ZK[2022] General 244.

Author contributions

HW: Methodology, Writing, and Experimental Design. EC, CQ, GZ, ZZ, ZW, GL, PL: Programming, Experimental Implementation. XC: Investigation, Supervision, Validation. All authors read and approved the final manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to X.C.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025