Regional Knowledge Transfer for Urban Traffic Flow Prediction via Satellite Imagery Assisted Contrastive Domain Adaptation

Zhidan Liu, *Senior Member, IEEE*, Zhengze Sun, Junru Zhang, Bolin Zhang, and Panrong Tong

***Abstract*— In traffic flow prediction, the efficacy of deep learning models is largely contingent upon the availability of extensive training datasets, presenting a formidable challenge in data-scarce environments. Transfer learning has emerged as a promising strategy to address this challenge by leveraging abundant data from source cities to enhance predictive accuracy in target cities with limited data. Nonetheless, existing methods frequently neglect the distinct characteristics and interrelationships among various regions within cities, leading to predominantly city-level knowledge transfers that underutilize the potential of transferred information. In this paper, we present SERT, a fine-grained regional knowledge transfer method specifically designed to mitigate data scarcity in traffic flow prediction. SERT initiates the process by establishing relationships between source and target regions through the integration of satellite imagery and Points of Interest (POI) data, effectively capturing region-specific features to create matched region pairs. Subsequently, we propose an innovative contrastive domain adaptation strategy to align the features of these matched regions, thereby facilitating inter-regional knowledge transfer while maximizing the feature distance of unmatched regions to reduce interference from irrelevant data. This approach enables the effective transfer of valuable knowledge from the source cities to its relevant counterparts in the target city. Comprehensive experimental results demonstrate that SERT outperforms existing methods in terms of prediction accuracy while ensuring significant computational efficiency. The code is available at https://github.com/MobiXg/SERT**

***Index Terms*— Transfer learning, traffic flow prediction, satellite imagery, contrastive learning.**

I. Introduction

TRAFFIC flow prediction constitutes a vital component of smart city applications, emphasizing the forecasting of urban traffic patterns for future time periods [1], [2]. With the emergence of deep learning, models such as Convolutional Neural Networks (CNNs) [3], [4], Recurrent Neural Networks (RNNs) [5], [6], and Graph Neural Networks (GNNs) [7], [8] have become increasingly prominent in this domain, primarily owing to their strong feature extraction capabilities. Nevertheless, these models often necessitate extensive traffic data for effective training [9], presenting a significant challenge for urban cities with limited data availability.

To mitigate the challenges posed by data scarcity, transfer learning has emerged as an effective strategy, wherein models are trained using data-rich source domains and subsequently applied to data-deficient target domains [10]. In the context of traffic flow prediction, each city is regarded as a distinct domain, with the target city typically possessing a reduced volume of traffic data relative to the source city. Recent researches [11], [12], [13], [14] have made noteworthy strides in facilitating knowledge transfer for traffic flow prediction across different cities. However, it is important to recognize that cities comprise various regions characterized by unique attributes (e.g., education, entertainment), as illustrated in Fig. 1(a). These attributes give rise to distinct spatial-temporal patterns. For instance, Fig. 1(d) depicts the traffic flow over one week in two regions: one adjacent to a school (left) and the other near a shopping mall (right). It is evident that traffic flow in the educational region peaks on weekdays

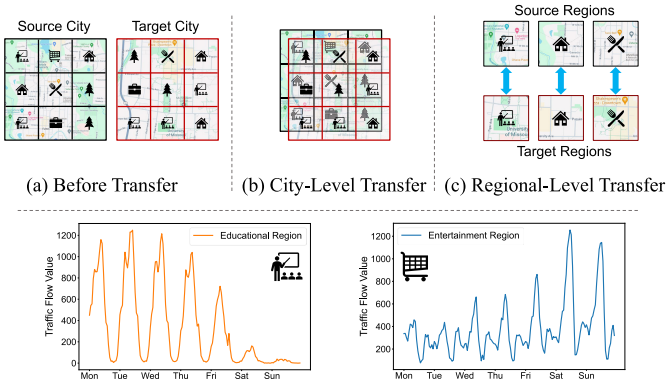


Fig. 1. Regions with varying attributes often exhibit distinct traffic patterns. Different from city-level transfer, regional-level transfer can effectively align the regions with similar attributes.

Received 20 January 2025; revised 1 June 2025; accepted 11 July 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62172284; and in part by Guangdong Provincial Key Laboratory of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things under Grant 2023B1212010007. The Associate Editor for this article was T. G. Molnar. *(Corresponding author: Zhidan Liu.)*

Zhidan Liu is with the INTR Thrust, System Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 510000, China (e-mail: zhidanliu@hkust-gz.edu.cn).

Zhengze Sun and Bolin Zhang are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: sunzhengze2022@email.szu.edu.cn; zhangbolin2023@email.szu.edu.cn).

Junru Zhang is with the Department of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: junruzhang@zju.edu.cn).

Panrong Tong is with Alibaba Cloud Computing, Hangzhou 310030, China (e-mail: panrong.tpr@alibaba-inc.com).

Digital Object Identifier 10.1109/TITS.2025.3589218

and diminishes on weekends, while the entertainment region exhibits an inverse trend. Existing studies [11], [12], [13], [14] predominantly focuses on knowledge transfer at the "city level" (referring to the granularity of the transferred knowledge), neglecting the extraction of region-specific features and the establishment of inter-regional relationships (Fig. 1(b)). This oversight can align regions with entirely different spatial-temporal patterns, leading to two significant issues: (i) *suboptimal transfer*, where valuable knowledge from the source city is misapplied to inappropriate locations within the target city, thereby diminishing its effectiveness; and (ii) *negative transfer*, wherein detrimental knowledge from the source city is inappropriately transferred to the target city, adversely impacting prediction performance.

In light of this analysis, we pose the question: "*Can we optimize the efficacy of transfer learning by facilitating knowledge transfer on a region-to-region basis?*" If feasible, as depicted in Fig. 1(c), knowledge could be transferred between source and target regions exhibiting analogous spatial-temporal patterns. This targeted approach not only leverages valuable knowledge but also filters out irrelevant information, thereby preventing the transfer of counterproductive knowledge. However, the implementation of fine-grained regional knowledge transfer is non-trivial due to two pivotal challenges:

- **Region-specific feature extraction for inter-regional relationship establishment:** Previous studies have utilized check-in data [11] and human mobility data [14] as proxies for traffic flow data in the derivation of city features. However, these auxiliary data sources are not universally available for all cities and may lack generalizability across diverse contexts [15].
- **Effective knowledge transfer between regions:** Fine-tuning methods [11], [12], [14] often encounter performance degradation due to substantial discrepancies in inter-domain data distributions [16]. Domain adaptation techniques [13] often overlook the unique characteristics of intra-domain samples [17] and are susceptible to model collapse [18]. These challenges complicate their application to regional knowledge transfer.

In response to these challenges, we propose a novel transfer learning method for traffic flow prediction, termed SERT (**S**atellite imagery **E**nabled **R**egional **T**ransfer). To tackle the first challenge, we introduce the innovative use of satellite imagery as an auxiliary source for extracting region-specific features. In contrast to traditional auxiliary sources [11], [14], satellite imagery provides comprehensive global coverage, ensuring data availability even in less developed urban areas [19]. Additionally, satellite imagery contains rich information that reflects indicators closely related to traffic conditions, such as population density [20], [21] and economic status [22], [23]. Fig. 2 illustrates the utility of satellite imagery by presenting images of regions from different cities alongside their traffic flow variations over the same week. It is apparent that regions exhibiting similar imagery characteristics demonstrate comparable traffic flow patterns in terms of value range and periodicity, even when situated in different urban locales (our statistical analysis revealed that over 55% of the regions across two cities support this conclusion).

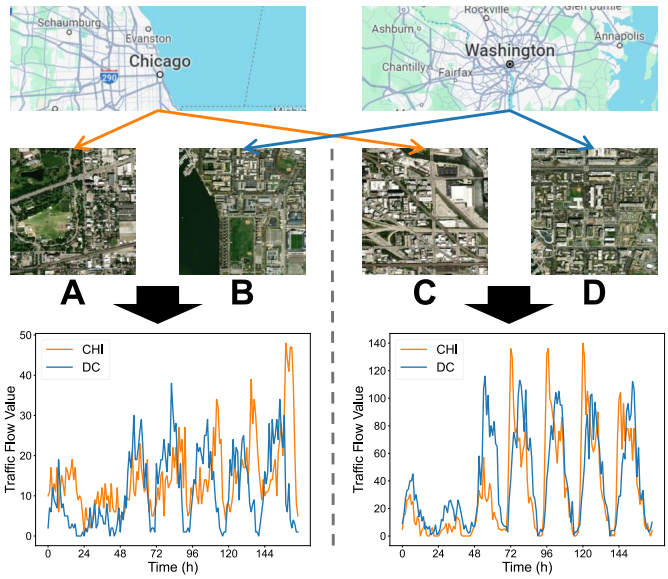


Fig. 2. Satellite images and weekly traffic flow values from regions in Chicago (CHI) and Washington (DC). Regions A and B exhibit a greater presence of green spaces, while regions C and D are characterized by a higher density of buildings. This suggests a positive correlation between imagery characteristics and traffic flow patterns.

Consequently, satellite imagery serves as an effective tool for extracting region-specific features. We also incorporate Points of Interest (POI) data as another auxiliary source, which is also widely accessible [15]. The features extracted from these auxiliary data sources will henceforth be referred to as *auxiliary features*.

To address the second challenge, we propose a contrastive domain adaptation approach for the transfer of regional knowledge. Contrastive learning [24], [25], which aligns positive samples while distinguishing negative ones, effectively extracts discriminative representations and alleviates model collapse, thus presenting a viable solution to the limitations commonly encountered in fine-tuning [11], [12], [14] and domain adaptation [13]. Motivated by this, we employ this methodology to tackle the complexities of regional transfer. Specifically, we identify groups of regions with the most similar auxiliary features from different cities as positive samples, while treating the remaining regions as negative samples. This strategy facilitates finer-grained adaptation between similar regions and mitigates interference from irrelevant regions.

In summary, the contributions of this paper are as follows:

- We introduce SERT, a novel transfer learning method designed to enable region-to-region knowledge transfer, mitigating data scarcity issues in traffic flow prediction.
- By harnessing the generalizability and rich informational content of satellite imagery, we extract region-specific features to establish inter-regional relationships.
- We develop an innovative contrastive learning based knowledge transfer approach, which effectively facilitates the transfer of valuable knowledge between regions while addressing prevalent challenges typically encountered in fine-tuning and domain adaptation.
- Experiments show that SERT outperforms existing methods in traffic flow prediction tasks with notable efficiency.

The remainder of this paper is structured as follows: Section II reviews the related work. Section III introduces the key concept definitions and formally states the problem. Section IV describes the proposed methodology in detail. Section V presents the experimental results, followed by a comparative analysis of our method against existing approaches. Section VI concludes the paper by summarizing the main findings.

II. Related Work

A. Traffic Flow Prediction

Traffic flow prediction aims to forecast traffic patterns across various urban regions over future time periods based on historical data [9]. Traditional approaches typically rely on statistical time series methods such as ARIMA [26] and Holt-Winters [27]. However, these approaches are constrained by their limited capacity for feature extraction, which frequently results in suboptimal predictive performance.

In recent years, deep learning models, including CNNs [3], [28], RNNs [5], [29], and GNNs [7], [30] have garnered increasing attention for traffic flow prediction. For instance, ConvLSTM [31] achieves improved spatial-temporal correlations extraction by extending the fully connected LSTM [32] to incorporate convolutional structures in both the input-to-state and state-to-state transitions. Additionally, PDFormer [33] employs spatial-temporal self-attention mechanisms to effectively capture dynamic long-range dependencies within traffic data. These models demonstrate superior capabilities in modeling complex spatial-temporal dependencies, thereby significantly improving prediction accuracy. Nevertheless, their performance often depends on the availability of large-scale training datasets, which presents a considerable challenge for cities with limited traffic flow data.

B. Transfer Learning for Traffic Flow Prediction

To mitigate the challenge of data scarcity, transfer learning leverages abundant data from source cities to acquire knowledge that is subsequently transferred to target cities, enhancing prediction performance in data-limited environments [34]. Common transfer learning techniques include fine-tuning [11], [12], [14] and domain adaptation [13].

Fine-tuning methods typically involve pre-training a model on the source city, followed by fine-tuning it on the target city. `RegionTrans` [11] firstly pre-trains model on the source city, then transfers knowledge from the entire source city to the target city, leveraging check-in data to construct regional relationships for optimizing the fine-tuning process on the target city. We define this strategy as "*city-to-region*"[1] transfer. In contrast, `MetaST` [12] employs meta-learning [35] to develop a model on source cities that can more rapidly adapt to a target city, described as a "*cities-to-city*" transfer. Similarly, `CrossTReS` [14] utilizes a meta-learning to assign weights to each source region, optimizing the pre-training stage for "*region-to-city*" transfer. Despite these strategies, the conventional two-stage process (pre-training followed by fine-tuning) can introduce gaps in the transfer process between cities [36]. When cities exhibit significantly different data distributions due to factors such as varying development levels, fine-tuning can lead to substantial performance declines in the target city [16].

Domain adaptation [37], [38] addresses data distribution discrepancies by aligning features between source and target domains. `ST-DAAN` [13] utilizes a deep adaptation network [39] for "*city-to-city*" transfer. However, domain adaptation often requires the formulation of complex feature alignment criteria, such as Maximum Mean Discrepancy (MMD) [40], which can result in significant computational complexity, thereby limiting its applicability compared to fine-tuning in related research areas. Furthermore, coarse-grained domain-level alignment methods may overlook the unique characteristics of individual samples within domains, potentially leading to negative transfer [17].

In contrast to existing works that remain at the city-level transfer, we propose a more nuanced "*region-to-region*" knowledge transfer method that takes into account the distinct characteristics of each region to minimize the transfer of detrimental knowledge. Additionally, we introduce an innovative contrastive domain adaptation method that facilitates efficient end-to-end domain feature alignment, thereby mitigating the adverse effects of data distribution differences.

III. Concepts and Problem Statement

We define the essential concepts and notations used in this paper and state the problem aimed to be addressed.

Definition 1 (Region): A city C is divided into $H_C \times W_C$ square grids,[2] where H_C and W_C denote the number of grids along the latitude and longitude, respectively. Each grid corresponds to a specific region r, and the set of all regions in city C is denoted as R_C, where $r \in R_C$.

Definition 2 (Traffic Flow): The traffic flow of an area refers to the statistical count of vehicles (e.g., taxis, bicycles) moving within that area over a certain period of time (e.g., one hour). The time steps for a city C are denoted as $T_C = \{1, 2, \ldots, t_C\}$, where a larger t_C indicates more data for the city. The traffic flow at a specific time step $t \in T_C$ is denoted as x_r^t for region r and as $X_C^t = \left\{x_r^t \mid r \in R_C\right\}$ for all regions in city C. The entire traffic flow dataset for city C is denoted as $\mathcal{X}_C = \left\{X_C^t \mid t \in T_C\right\}$.

Problem Statement (*Transfer Learning for Traffic Flow Prediction*)**.** Given a source city S with abundant traffic data $\mathcal{X}_S$ and a target city T with limited data $\mathcal{X}_T$ (i.e., $t_S \gg t_T$), transfer learning for traffic flow prediction aims to develop a model $F(\cdot)$. This model first extracts spatial-temporal features from both $\mathcal{X}_S$ and $\mathcal{X}_T$. Subsequently, knowledge acquired from

[1]The term "city-to-region" demonstrates a deficiency in fine-grained filtering and optimization of the knowledge sourced from the city, whereas a detailed analysis is conducted on how this transferred knowledge is utilized in the target region. The meanings of other terms, such as "region-to-city", follow the same logic.

[2]Our work primarily focuses on *grid-based* traffic flow data, where flow values are aggregated based on uniformly distributed grid cells. This differs from some *graph-based* approaches [8], [41], where flow data is collected from detection devices distributed unevenly across the road network. As a result, this paper does not extensively cover graph-based methodologies.

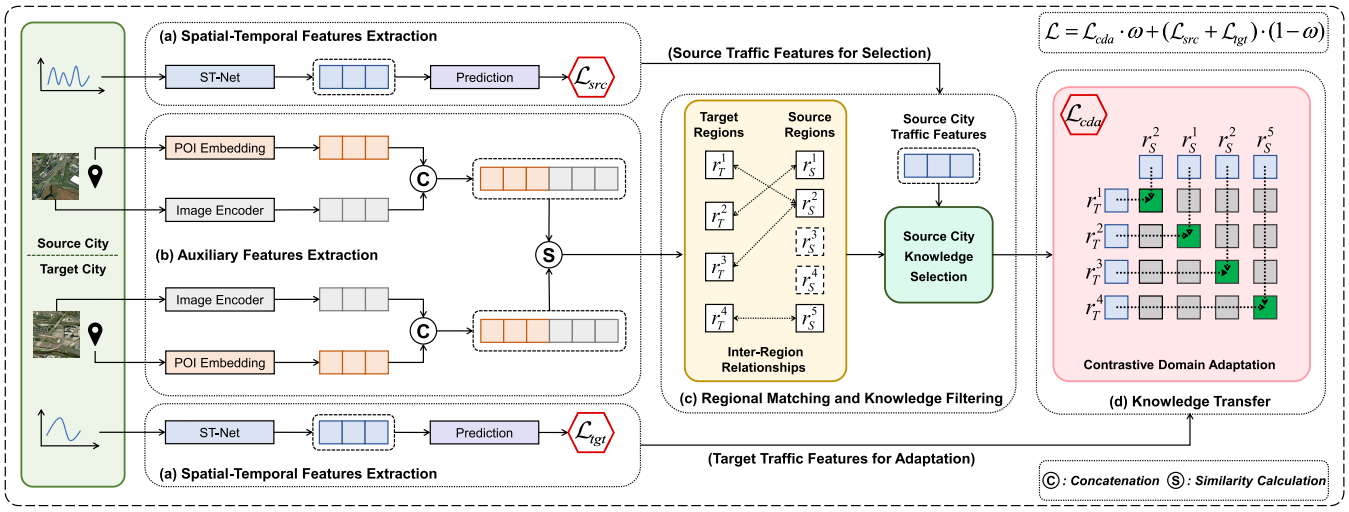


Fig. 3. Overview of SERT. Both the source and target cities include traffic flow data and two auxiliary data sources (i.e., satellite imagery and POI data). The *ST-Net* and *Prediction* modules are responsible for extracting traffic flow features and predicting future traffic values, respectively. The *POI Embedding* and *Image Encoder* modules are tasked with extracting features from POI data and satellite images, respectively. The training process of SERT is guided by a combination of two prediction losses and a transfer loss.

S is transferred to T to enhance traffic flow prediction in T. Finally, $F(\cdot)$ forecasts traffic flow for each region in T for the next time step based on historical data from the preceding k time steps. The objective is to minimize the prediction error in T. This process can be mathematically formulated as:

$$\begin{aligned} &F \leftarrow \text{train}(F, \mathcal{X}_S, \mathcal{X}_T) \\ &\min \sum_{t=k}^{t_T-1} \sum_{r \in R_T} \text{error}\left(x_r^{t+1}, \hat{x}_r^{t+1}\right), \\ &\text{where } \hat{x}_r^{t+1} = F\left(\left[x_r^{t-k+1}, \ldots, x_r^{t}\right]\right). \end{aligned} \tag{1}$$

Here, x_r^{t+1} and $\hat{x}_r^{t+1}$ represent the actual and predicted values for region r in the target city at the subsequent time step, respectively. The prediction error is typically evaluated using metrics such as Mean Squared Error (MSE).

IV. Methodology

A. Framework Overview

We propose a novel framework – SERT, as illustrated in Fig. 3, to address the challenges associated with regional transfer learning for traffic flow prediction. SERT commences by extracting traffic features from the traffic flow data of both source and target cities to predict future traffic flow values and acquire spatial-temporal knowledge. Concurrently, it extracts auxiliary features from satellite imagery and POI data for each region, identifying these as region-specific characteristics. By analyzing the similarity of these auxiliary features, SERT constructs matched pairs of regions between the source and target cities. Ultimately, leveraging these traffic features and matched pairs, SERT facilitates knowledge transfer between corresponding regions through contrastive domain adaptation.

B. Traffic Flow Data Processing

We detail the extraction of spatial-temporal features from traffic flow data and prediction of future traffic flow values for both the source and target cities.

1) Spatial-Temporal Features Extraction: By partitioning each city into square grids, we transform the historical traffic flow data into images as input for the feature extraction network. Each image represents the traffic flow of a city at a specific time step, where the image dimensions $H_C \times W_C$ represent the number of grids. The value of each pixel indicates the traffic flow in a particular region at that time step.

To encode image sequences, we utilize the ST-Net model [12], which integrates CNNs [42] and LSTMs [32] to effectively extract spatial-temporal features. Specifically, a sliding window of length $k+1$ time steps is used to slice the entire sequence along the temporal axis, where the first k time steps serve as historical values and are fed into the model as a single batch, while the $(k+1)$-th time step serves as the prediction target. Initially, the image sequences are fed into CNNs to learn spatial dependencies, capturing the relationships between local neighboring regions. Subsequently, the output from the CNNs is processed by LSTMs to capture temporal dependencies across sequences. The images from both the source and target cities undergo processing through the ST-Net with shared weights, resulting in their respective traffic feature representations, which can be expressed as:

$$\begin{aligned} &\mathbf{z}_C = \text{ST-Net}\left(\left[X_C^{t-k+1}, \ldots, X_C^{t}\right]_{t=k}^{t_C-1}\right), \\ &\text{where } C \in \{S, T\}. \end{aligned} \tag{2}$$

Here, $\mathbf{z}_C \in \mathbb{R}^{|R_C| \times d_{st}}$ represents the spatial-temporal features (i.e., knowledge) of city C, where each row corresponds to the features of a specific region, with a dimensionality of d_{st}.

2) Traffic Flow Prediction: Once spatial-temporal features have been extracted from both the source and target cities, these features are fed into a prediction network to forecast traffic flow for each region at the next time step. This process not only equips the model with traffic prediction capabilities but also enhances its understanding of the data distributions in both cities. We utilize a Multilayer Perceptron (MLP) with

ReLU activation [43] to generate the predictions, where $\mathbf{W}^{(\ell)}$ and $\mathbf{b}^{(\ell)}$ denote the weights and biases of the ℓ-th layer:

$$\hat{X}_C^{t+1} = \mathbf{W}^{(2)}(\text{ReLU}(\mathbf{W}^{(1)}\mathbf{z}_C + \mathbf{b}^{(1)})) + \mathbf{b}^{(2)}, \quad \text{where } C \in \{S, T\}. \tag{3}$$

Subsequently, we calculate the prediction loss for both the source and target cities individually by comparing their predicted values to the ground truth data. The Mean Squared Error (MSE) is utilized to quantify this prediction loss:

$$\mathcal{L}_{src} = \sum_{t=k}^{t_S-1} \text{MSE}(\hat{X}_S^{t+1}, X_S^{t+1}),$$
$$\mathcal{L}_{tgt} = \sum_{t=k}^{t_T-1} \text{MSE}(\hat{X}_T^{t+1}, X_T^{t+1}). \tag{4}$$

Compared to fine-tuning methods [11], [12], [14], which entail distinct pre-training on the source city followed by fine-tuning on the target city, our approach processes traffic data from both cities simultaneously in one integrated stage. This unified strategy facilitates the subsequent establishment of relationships and feature alignment between the source and target regions, mitigating the negative impact of data distribution disparities and promoting more efficient knowledge transfer.

C. Inter-Region Relationships Establishment

To address the first challenge of establishing relationships between source and target regions, we utilize auxiliary data to extract region-specific features, thereby obtaining matched region pairs while filtering out irrelevant source regions.

1) Auxiliary Features Extraction: Due to the limited availability of comprehensive traffic flow data in the target city, which may hinder the extraction of high-quality features, auxiliary data plays a critical role in extracting representative features for each region. We collect satellite images and Points of Interest (POI) data for all regions in both the source and target cities to serve as auxiliary data. For detailed descriptions of these data sources, please refer to Section V-A.1.

To extract high-quality image features from satellite imagery, we employ contrastive learning [20] in a self-supervised manner, minimizing reliance on labeled data. Specifically, for a batch of N satellite images, we calculate the geographic distance between each pair of images using the Haversine formula [44], based on the latitude and longitude of each image's center point. For a given image (the anchor), the image that is geographically closest is identified as its positive sample, while the remaining $N-2$ images are considered negative samples. To extract features, both the anchor image I_i and the positive sample image I_j are processed through a ResNet [45], and their features are projected into a new feature space using an MLP, formulated as follows:

$$\mathbf{h}_i^{img} = \text{ResNet}(I_i), \ \mathbf{h}_j^{img} = \text{ResNet}(I_j),$$
$$\mathbf{p}_i = \text{MLP}(\mathbf{h}_i^{img}), \ \mathbf{p}_j = \text{MLP}(\mathbf{h}_j^{img}). \tag{5}$$

Then, the Normalized Temperature-Scaled Cross Entropy Loss (NT-Xent) [24] is employed to train this contrastive model:

$$\ell_{i,j} = -\log \frac{\exp\left(\text{sim}\left(\mathbf{p}_i, \mathbf{p}_j\right)/\tau\right)}{\sum_{k=1}^{N} \mathbf{1}_{[k\neq i]} \exp\left(\text{sim}\left(\mathbf{p}_i, \mathbf{p}_k\right)/\tau\right)}, \tag{6}$$

where $\text{sim}(\cdot)$ denotes the cosine similarity, τ is a temperature parameter, and $\mathbf{1}_{[k\neq i]}$ is an indicator function that equals 0 when $k = i$, and 1 otherwise. The final loss is aggregated across all positive pairs. After training, the satellite imagery of region r is fed into the trained model, and the output of the ResNet, i.e., $\mathbf{h}_r^{img} \in \mathbb{R}^{d_{img}}$, is treated as the satellite imagery features for region r.

The POI data comprises 14 distinct categories for each region, along with the corresponding count for each category. For a given region r, we represent its POI data as a vector $\mathbf{h}_r^{poi} \in \mathbb{R}^{14}$, where each element indicates the quantity of a particular type of POI. Subsequently, we normalize both the satellite imagery features $\mathbf{h}_r^{img}$ and POI features $\mathbf{h}_r^{poi}$ for region r to mitigate differences in the numerical ranges of the two modalities. To integrate these features while maximizing the retention of their respective information, we concatenate the normalized features to form the auxiliary features for region r as follows:

$$\mathbf{h}_r^{aux} = \text{concat}(\mathbf{h}_r^{img}, \mathbf{h}_r^{poi}), \quad \text{where } r \in \{R_S, R_T\}. \tag{7}$$

2) Region Pairing and Selection: Once the auxiliary features for all regions have been derived, they are utilized to establish relationships between the source and target regions. This is accomplished by calculating the cosine similarity between the auxiliary features of the target city, denoted as $\mathbf{h}_T^{aux} \in \mathbb{R}^{|R_T|\times(d_{img}+14)}$, and those of the source city, denoted as $\mathbf{h}_S^{aux} \in \mathbb{R}^{|R_S|\times(d_{img}+14)}$:

$$D = \text{sim}(\mathbf{h}_T^{aux}, \mathbf{h}_S^{aux}) = \frac{\mathbf{h}_T^{aux} \cdot \mathbf{h}_S^{aux^\top}}{\|\mathbf{h}_T^{aux}\| \cdot \|\mathbf{h}_S^{aux}\|}, \tag{8}$$

where "$\top$" denotes the matrix transpose. As a result, we obtain a similarity matrix $D \in \mathbb{R}^{|R_T|\times|R_S|}$, where $D(i, j)$ indicates the similarity score between the i-th target region and the j-th source region. By identifying the position index of the maximum score in each row of D, we can determine the most similar region in the source city for each target region, thereby forming matched region pairs. Subsequently, using the indices of these matched pairs, we filter and reorganize the spatial-temporal features (i.e., the knowledge) from the source city, expressed as follows:

$$index^* = \text{argmax}(D, \text{axis} = 1),$$
$$\mathbf{z}'_S = \mathbf{z}_S[index^*, :], \tag{9}$$

where $\mathbf{z}_S \in \mathbb{R}^{|R_S|\times d_{st}}$ denotes the initial spatial-temporal features extracted from the source city, which may contain a significant amount of irrelevant or potentially detrimental knowledge, while $\mathbf{z}'_S \in \mathbb{R}^{|R_T|\times d_{st}}$ represents the selected spatial-temporal features of the source city. In knowledge selection, a "top-1" selection strategy is adopted, where only the source region with the highest similarity to each target region is selected for matching. Source regions that are not

selected throughout the process are discarded to filter out irrelevant knowledge and to reduce the computational overhead associated with transferring that knowledge.

Notably, both auxiliary feature extraction and region pairing do not require traffic flow data. This allows it to be independently trained prior to the SERT framework, enabling the pre-obtainment of region matching indices for adaptation to various traffic prediction tasks and further reducing the overall training cost of the model. Additionally, given the advantage of frequent update cycles for satellite imagery and POI data, auxiliary features can be updated in a timely manner to provide more accurate representation of city conditions.

D. Contrastive Domain Adaptation

Having established relationships between source and target regions, we next address the challenge of facilitating effective knowledge transfer between these correspondingly matched regions. Traditional domain adaptation methods [13], [38], [39] often assess domain discrepancy *at the domain level*. For instance, ST-DAAN [13] employs MMD to evaluate the divergence between source and target domain distributions. However, this approach overlooks the differences and connections between regions within each domain [17], complicating direct application to region-to-region transfer. Additionally, existing techniques [11], [46], [47] primarily focus on minimizing domain disparities to facilitate knowledge transfer. For example, RegionTrans [11] minimizes the squared error between the representations of source and target regions. Such approaches can result in *model collapse* [18], where the model excessively aligns features from the source and target domains to a single point in the feature space, driving the alignment loss towards zero and impeding effective training process. To address these challenges, we propose a novel contrastive learning-based domain adaptation method to facilitate finer-grained regional knowledge transfer.

Following the procedures outlined in Sections IV-B.1 and IV-C.2, we derive the spatial-temporal features $\mathbf{z}_T$ for the target city and the selected features $\mathbf{z}'_S$ for the source city. By selecting and reorganizing features from the source city, the row vectors in $\mathbf{z}_T$ and $\mathbf{z}'_S$ that share the same index can be considered as matched pairs of target and source region features. This alignment facilitates the transfer of knowledge from the source region to the corresponding target region. Initially, a shared-weights projection head (i.e., an MLP) is employed to project $\mathbf{z}_T$ and $\mathbf{z}'_S$ into a new feature space:

$$\begin{aligned} \mathbf{e}_S &= \text{project}(\mathbf{z}'_S), \\ \mathbf{e}_T &= \text{project}(\mathbf{z}_T). \end{aligned} \tag{10}$$

Building on the projected features $\mathbf{e}_S$ and $\mathbf{e}_T$, and aligning with the principles of contrastive learning [24], for the i-th region in the target city, we designate its features $\mathbf{e}_T[i,:]$ as the anchor. The features from the source region that are aligned with this target region, $\mathbf{e}_S[i,:]$, are identified as the positive sample. All other row features in $\mathbf{e}_S$, excluding the i-th row, are treated as negative samples. The objective is to minimize the distance between the anchor and the positive sample while maximizing the distance between the anchor and the negative samples in the feature space, thereby promoting effective knowledge adaptation between similar regions and mitigating interference from dissimilar regions.

Algorithm 1 Pseudocode of $\mathcal{L}_{cda}$, PyTorch-Like

```
# e_s, e_t: projected features of source and target cities
# n: number of target regions
# d: regional feature dimensionality
# .t(): matrix transpose
# tem: temperature parameter

# L2 normalization
e_s = normalize(e_s, dim=-1)   # (n, d)
e_t = normalize(e_t, dim=-1)   # (n, d)

# regional feature cosine similarity
logits = matmul(e_s, e_t.t()) * tem   # (n, n)

# loss calculation
labels = arange(n)   # ground truth, an identity matrix
loss_s = cross_entropy(logits, labels)
loss_t = cross_entropy(logits.t(), labels)
loss_cda = (loss_s + loss_t) / 2   # the final transfer loss
```

The transfer loss is crafted around a classification task to train the contrastive framework, ensuring each region identifies the corresponding matched region in another city, as detailed in **Algorithm 1**. Specifically, the framework treats the anchor and its positive sample as belonging to the same category, targeting a mutual prediction result of 1, when negative samples are assigned different labels with a mutual prediction outcome of 0. Actual predictions rely on cosine similarity, with a temperature parameter in the range $(0, 1)$ to modulate the degree of contrast, where lower temperature values enhance separation between positive and negative samples [48]. The label prediction loss is calculated using cross-entropy [49]. Since both source and target regions must predict each other, the final contrastive domain adaptation loss, $\mathcal{L}_{cda}$, is obtained by averaging the two cross-entropy losses.

Through the implementation of the aforementioned contrastive domain adaptation, we can achieve effective regional knowledge transfer. Unlike traditional knowledge transfer techniques [11], [12], [13], [14], our approach leverages the unique characteristics of samples within the domain, facilitating knowledge transfer at the regional level. This enables the filtering of irrelevant knowledge while applying valuable information to the appropriate locations. Additionally, our contrastive domain adaptation method benefits from the balancing effect of negative samples, which mitigates the risk of regions becoming overly clustered or dispersed in the feature space. This balance reduces inter-domain data distribution discrepancies and effectively prevents model collapse. Furthermore, our method employs straightforward cross-entropy loss, eliminating the need for complex feature alignment criteria and significantly reducing training overhead.

Finally, the SERT framework is optimized by simultaneously minimizing three loss functions: the traffic flow

prediction losses for both the source and target cities, denoted as $\mathcal{L}_{src}$ and $\mathcal{L}_{tgt}$ (as defined in (4)), along with the knowledge transfer loss $\mathcal{L}_{cda}$ (as detailed in Algorithm 1). The combined loss function is represented as:

$$\mathcal{L} = \omega\mathcal{L}_{cda} + (1-\omega)(\mathcal{L}_{src} + \mathcal{L}_{tgt}), \tag{11}$$

where ω is a hyperparameter in the range (0, 1) that balances *knowledge transfer* and *traffic flow prediction* objectives.

V. Experiments

We conduct comprehensive evaluations of the model's efficacy in the context of transfer learning for traffic flow prediction. The experimental analysis is structured to assess various aspects of the model's performance, including:

- **Overall Performance**: Evaluating the predictive capabilities of `SERT` in forecasting traffic flows in the target city, particularly under conditions of limited data availability.
- **Ablation Study**: Dissecting the model to determine the contribution of each component, with a focus on the impact of auxiliary data and regional transfer.
- **Sensitivity Analysis**: Examining the robustness of `SERT` to variations in critical hyperparameters to understand their influence on the model's overall performance.

A. Experimental Setup

1) Datasets: In accordance with prior research [14], we conducted experiments using datasets from three major cities: *New York (NY)*[3,4] *Chicago (CHI)*[5,6] and *Washington (DC)*[7,8]. Each dataset includes traffic flow data for two transportation modes: *taxi* and *bike*, which are further divided into *pickup* and *dropoff* subsets, representing passenger boarding and alighting, respectively. The cities were segmented into numerous grid cells measuring 1 km × 1 km. This grid size was selected as it efficiently delineates functional areas without introducing unnecessary complexity to regional features and is commonly adopted in previous studies [11], [14]. Traffic flow data were aggregated by hourly time steps. Detailed statistical information for the datasets is presented in Table I.

In addition to traffic data, we collected satellite imagery and points of interest (POI) data for each region in all cities. Using the open-source geographic information system software QGIS,[9] satellite images for each city were obtained and segmented. Each segmented imagery has a size of 512 × 512 pixels with an approximate ground resolution of 1.95 meters, allowing precise coverage of the geographic area of each 1 km × 1 km region. The POI data, acquired from OpenStreetMap,[10] encompasses 14 categories, including scenic spots, medical and health services, domestic services, residential areas, financial institutions, sports and leisure services, cultural and educational services, shopping, housing services, governments and organizations, corporations, catering, transportation, and public services [14].

Unless otherwise specified, we designate New York and Chicago as the source cities and Washington as the target city, given that the former two have more extensive traffic flow datasets. The data-rich source cities utilize a full year of traffic flow data, divided into an 8-month training set, a 2-month validation set, and a 2-month test set. For the target city, to simulate data scarcity, the training set is limited to only 30 days, 7 days, or 3 days of traffic data, with validation and test sets configured similarly to those of the source cities.

2) Baselines: We compare `SERT` with the following baselines, which are widely utilized in traffic flow prediction tasks or have demonstrated strong knowledge transfer capabilities:

- `ARIMA` [26]: The AutoRegressive Integrated Moving Average (`ARIMA`) model is used for time series analysis, modeling and forecasting by combining autoregression, differencing, and moving average.
- `ST-Net` [12]: `ST-Net` leverages the strengths of both CNNs and LSTMs. CNNs are utilized to capture spatial relationships between regions, while LSTMs are tasked with capturing temporal dependencies of these regions across different time steps.
- `Fine-Tuning`: This process begins by pre-training the ST-Net on data from the source city, followed by fine-tuning the network using data from the target city.
- `RegionTrans` [11]: After pre-training on the source city, `RegionTrans` establishes regional relationships utilizing check-in data. It facilitates knowledge transfer by minimizing the squared error between the matched region features, thereby optimizing the model's fine-tuning stage for the target city.
- `MetaST` [12]: During the pre-training phase on the source city, `MetaST` employs a meta-learning paradigm [35] to derive a well-generalized model initialization, thereby enhancing its adaptability to the target city.
- `ST-DAAN` [13]: This method utilizes a deep adaptation network [39] to project the features of both source and target cities into a common feature space, achieving inter-city feature alignment through MMD.
- `CrossTReS` [14]: During the pre-training phase on the source city, `CrossTReS` leverages human mobility data and road network information to construct regional features and employs meta-learning to learn the influence weights of each source region on the target city.

Among these, `ARIMA` and `ST-Net` are non-transfer methods, trained solely on limited traffic data from the target city. The other baselines are transfer learning methods that leverage source city data to enhance performance in the target city.

3) Implementation Details:

- Unified Configuration: To ensure fair experimental comparisons, all deep learning-based methods utilize a standardized ST-Net architecture for extracting spatial-temporal features from traffic flow data. This ST-Net is composed of three residual blocks, each with 64 output channels, and a single-layer LSTM with a hidden size of 128. The fine-tuning

[3]https://www.nyc.gov
[4]https://citibikenyc.com
[5]https://data.cityofchicago.org
[6]https://divvybikes.com
[7]https://opendata.dc.gov
[8]https://capitalbikeshare.com
[9]https://www.qgis.org/
[10]https://www.openstreetmap.org/

TABLE I
STATISTICS OF TRAFFIC FLOW DATASETS

Cities	Longitude	Latitude	Grids	Time Span	Taxi Trips (million)	Bike Trips (million)
NY	[-74.059, -73.863]	[40.645, 40.848]	20×23		133	13.8
CHI	[-87.740, -87.576]	[41.766, 42.013]	17×28	2016 (Jan 1 – Dec 31)	24.5	3.5
DC	[-77.127, -76.926]	[38.798, 38.969]	21×20		10	2.7

TABLE II
TRAFFIC FLOW PREDICTION ACCURACY IN WASHINGTON (DC). **RED**: BEST RESULTS, **BLUE**: SECOND-BEST RESULTS

Vehicle	Methods	NY						CHI					
		30		7		3		30		7		3	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Taxi	SERT	**3.781**	**1.353**	**3.976**	**1.445**	**4.150**	**1.493**	**3.792**	**1.355**	**3.971**	**1.408**	**4.125**	**1.492**
	± Std. Dev.	0.016	0.021	0.023	0.016	0.009	0.035	0.009	0.023	0.016	0.032	0.016	0.023
	CrossTRes	3.901	1.378	4.159	1.489	4.298	1.586	3.903	1.427	4.130	1.538	4.247	1.611
	ST-DAAN	4.139	1.468	4.274	1.528	4.522	1.736	4.124	1.503	4.246	1.536	4.482	1.627
	MetaST	4.029	1.445	4.301	1.578	4.514	1.679	4.035	1.421	4.284	1.580	4.528	1.698
	RegionTrans	3.988	1.598	4.135	1.498	4.358	1.624	4.091	1.683	4.219	1.517	4.564	1.676
	Fine-Tuning	4.020	1.462	4.228	1.576	4.492	1.719	4.074	1.458	4.307	1.580	4.515	1.667
	ST-Net	4.057	1.439	4.624	1.716	5.500	2.067	4.057	1.439	4.624	1.716	5.500	2.067
	ARIMA	4.771	3.870	4.834	3.928	5.108	4.199	4.771	3.870	4.834	3.928	5.108	4.199
Bike	SERT	**2.114**	**0.954**	**2.245**	**1.000**	**2.341**	**1.033**	**2.150**	**0.949**	**2.269**	**0.986**	**2.346**	**1.005**
	± Std. Dev.	0.007	0.010	0.016	0.013	0.003	0.018	0.015	0.009	0.009	0.015	0.019	0.015
	CrossTRes	2.240	1.000	2.359	1.050	2.489	1.086	2.278	1.005	2.410	1.053	2.495	1.078
	ST-DAAN	2.352	1.061	2.764	1.274	2.832	1.338	2.372	1.060	2.751	1.280	2.784	1.302
	MetaST	2.272	1.028	2.412	1.081	2.500	1.129	2.310	1.034	2.465	1.087	2.563	1.153
	RegionTrans	2.262	1.115	2.454	1.240	2.605	1.356	2.297	1.118	2.524	1.279	2.752	1.458
	Fine-Tuning	2.282	1.043	2.455	1.136	2.570	1.175	2.376	1.052	2.537	1.113	2.634	1.193
	ST-Net	2.297	1.023	2.419	1.068	2.726	1.248	2.297	1.023	2.419	1.068	2.726	1.248
	ARIMA	2.773	2.310	2.788	2.314	2.873	2.386	2.773	2.310	2.788	2.314	2.873	2.386

approaches (Fine-Tuning, RegionTrans, MetaST, CrossTReS) undergo 100 epochs of pre-training followed by 80 epochs of fine-tuning. In contrast, the domain adaptation methods (ST-DAAN, SERT) are trained over 100 epochs. For the traffic flow prediction task, we follow the setup of previous work [14], predicting the next time step's value based on the past $k = 6$ time steps.

- SERT Configuration: The Image Encoder in Section IV-C.1 employs ResNet-18 [45] as its backbone to encode each satellite imagery into a 512-dimensional vector, which is then projected to 64 dimensions using a two-layer MLP with ReLU activation. The projection head for contrastive domain adaptation is a two-layer MLP with ReLU, yielding an output dimension of 256. The experiments utilize the Adam optimizer [50] with a learning rate set to 10^{-3} and a batch size of 32.
- Baselines Configuration: When the source code is available, we adhere to the implementation and parameter settings provided in their code repositories (MetaST,[11] ST-DAAN,[12] CrossTReS[13]). To align the baselines with the experimental setup of this study and to ensure fair comparisons, we made the following modifications: (i) The original check-in data utilized in RegionTrans was replaced with POI data, as the check-in data is not publicly available. (ii) The initial implementation of ST-DAAN did not incorporate labeled data from the target city for training the traffic flow prediction model. To enhance its predictive accuracy, we included the target loss $\mathcal{L}_{tgt}$ as defined in (4) into its loss function.

All experiments were conducted on a single NVIDIA GeForce RTX 3090 GPU with 24GB of memory.

B. Overall Performance

We evaluate the performance of SERT in traffic flow prediction from two perspectives: (i) *prediction accuracy*, quantified by Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), with lower values signifying superior performance; and (ii) *computational efficiency*, assessed in terms of training time and convergence speed. To reduce the impact of randomness, each task is executed five times, with the average results and standard deviations (Std. Dev.) subsequently reported.

1) Prediction Accuracy: Table II presents the traffic flow prediction accuracy results of various methods. Each reported value is the average of the two results from the pickup and dropoff subsets. We have the following three key observations:

- SERT consistently outperforms all other methods across all tasks, achieving an average improvement of 4.4% in RMSE and 5.1% in MAE compared to the strongest baseline. These results highlight SERT's superior capability for knowledge transfer and generalization across different

[11] https://github.com/huaxiuyao/MetaST
[12] https://github.com/MiaoHaoSunny/ST-DAAN
[13] https://github.com/KL4805/CrossTReS

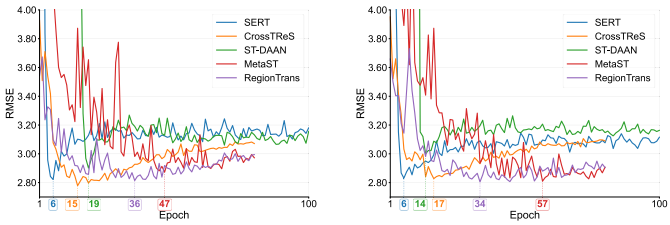
TABLE III
TRAFFIC FLOW PREDICTION ACCURACY IN CHICAGO (CHI) WITH MULTI-SOURCE TRANSFER

Methods		SERT		MetaST		CrossTReS		RegionTrans	
Metrics		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
NY → CHI Bike	7	1.694	0.669	1.761	0.698	1.781	0.681	1.820	0.750
	3	1.829	0.677	1.872	0.741	1.886	0.713	1.867	0.716
DC → CHI Bike	7	1.729	0.654	1.809	0.698	1.796	0.713	1.847	0.737
	3	1.963	0.742	2.004	0.840	2.009	0.776	1.999	0.787
NY + DC → CHI Bike	7	**1.637**	**0.646**	1.727	0.677	/	/	/	/
	3	**1.689**	**0.647**	1.815	0.730	/	/	/	/

cities and transportation modes, providing preliminary validation of our regional knowledge transfer approach.

- CrossTReS consistently achieves the second-best performance across most tasks, attributed to its "selective transfer" mechanism, which effectively reduces the transfer of harmful knowledge from the source city to the target city. However, its design lacks a comprehensive analysis of the target city, resulting in the misapplication of relevant knowledge to inappropriate locations. Notably, SERT outperforms CrossTReS with only 3 days of target training data, compared to CrossTReS's requirement of 7 days, across nearly all datasets. This demonstrates SERT's superior knowledge utilization and highlights its robustness in the face of data scarcity.
- ST-DAAN exhibits the weakest performance among the transfer learning methods, occasionally performing worse than non-transfer approaches. This outcome highlights our concerns regarding the negative transfer phenomenon [51]. The city-to-city transfer design of ST-DAAN facilitates the transfer of substantial detrimental knowledge from the source city to the target city, which not only fails to enhance the prediction accuracy but actually degrades it. This finding emphasizes the necessity for adopting finer-grained, regional-level knowledge transfer strategies to effectively filter out irrelevant information.

2) Multi-Source Transfer Performance: Among the baseline methods, SERT and MetaST uniquely facilitate knowledge transfer from multiple source cities to a single target city. MetaST achieves this by extracting global spatial-temporal patterns from all source cities and applying them to the target city. In contrast, SERT expands its candidate matching set for each target region to include regions from all source cities. In this experiment, we designate NY and DC as source cities and CHI as the target city. For each region in CHI, SERT identifies matching regions in both NY and DC, facilitating knowledge transfer from the two source regions to the target region. Table III presents a performance comparison between single-source and multi-source transfer approaches. The results clearly indicate that multi-source transfer surpasses single-source transfer, as it allows the target city to leverage knowledge from multiple source cities. Moreover, among methods supporting multi-source transfer, SERT delivers superior predictive performance compared to MetaST, owing to its special design that eliminates reliance on iteration-heavy meta-learning.



(a) NY to DC (b) CHI to DC

Fig. 4. Variation of RMSE on the validation set of the target city throughout the model training process.

3) Computational Efficiency: To assess the computational efficiency of various methods, we recorded both the training time and GPU memory usage for each approach. Table IV summarizes the statistics from five independent runs for each method. Here, "*Time*" denotes the average training time across five runs: for fine-tuning methods, this is expressed as "pre-training time + fine-tuning time", whereas for domain adaptation methods, it simply indicates the "training time". For SERT, the reported values do not include the training time of the Image Encoder, as it only needs to be trained once and can be directly used without retraining. "*Epoch*" refers to the epoch at which the validation set for the target city achieves its minimum RMSE during training (excluding the initial 100 epochs of pre-training for fine-tuning methods). The results are based on the *Bike-7days-pickup* dataset.

In terms of absolute training time, SERT exhibits moderate performance. However, given its design for finer-grained region-to-region transfer, which necessitates detailed construction of inter-regional relationships, this efficiency is commendable. Particularly when compared to the coarse-grained transfer approach of ST-DAAN, SERT demonstrates a significant advantage in training time, highlighting the efficiency of the proposed contrastive domain adaptation method. Regarding convergence speed, SERT significantly surpasses other methods. By leveraging its regional-level transfer design, SERT effectively filters out irrelevant knowledge, enabling the transfer of higher-quality knowledge to the target city and facilitating faster model convergence.

To intuitively demonstrate SERT's convergence speed and knowledge filtering capability, Fig. 4 presents the RMSE variation during training for each method (only the fine-tuning phase is shown for fine-tuning methods). The regional

TABLE IV
THE COMPARISON OF TIME AND MEMORY OVERHEAD FOR VARIOUS METHODS DURING TRAINING

Methods	NY → DC		CHI → DC		GPU Memory Usage
	Time (s)	Epoch	Time (s)	Epoch	
SERT	1557.1	[6, 6, 6, 6, 6]	1636.6	[6, 8, 6, 6, 5]	3.3 GB
CrossTReS	1903.6 + 55.8	[24, 15, 17, 16, 25]	2106.0 + 55.6	[28, 12, 14, 25, 20]	9.8 GB
ST-DAAN	2492.6	[14, 5, 12, 16, 10]	2801.2	[31, 15, 30, 16, 18]	2.6 GB
MetaST	1012.6 + 61.4	[59, 60, 60, 48, 54]	1162.2 + 60.8	[50, 66, 43, 57, 44]	7.9 GB
RegionTrans	41.0 + 58.0	[43, 40, 40, 65, 52]	45.0 + 58.4	[61, 38, 49, 48, 57]	2.7 GB

TABLE V
TRAFFIC FLOW PREDICTION PERFORMANCE COMPARISON UNDER 2 KM × 2 KM GRID SIZE

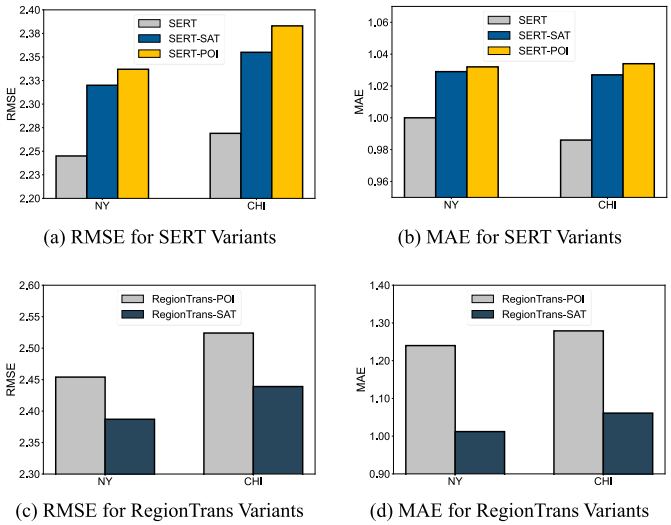
Methods	NY → DC Taxi						CHI → DC Bike					
	30		7		3		30		7		3	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
SERT	**10.585**	**3.936**	**11.023**	**4.179**	**11.517**	**4.263**	**4.468**	**1.724**	**4.690**	**1.858**	**4.839**	**1.847**
CrossTRes	12.884	5.076	13.925	5.601	15.036	6.451	6.617	2.840	7.111	2.960	7.539	3.182
ST-DAAN	12.014	4.578	12.720	5.058	13.071	5.432	5.807	2.297	6.070	2.537	6.864	2.920
MetaST	13.202	5.203	13.828	5.307	14.248	5.780	6.733	3.212	7.304	3.396	7.618	3.602
RegionTrans	12.851	4.920	13.279	5.214	13.755	5.608	7.227	2.820	7.328	2.970	7.595	3.185

transfer design of SERT effectively selects valuable knowledge from the source city and applies it to suitable locations in the target city, enabling faster convergence in the target city. Additionally, ST-DAAN maintains a consistently high RMSE in the early stages (approximately before epoch 20) due to its MMD-based feature alignment approach [40], which leads to local collapse. MetaST, on the other hand, exhibits more pronounced RMSE fluctuations as a result of its reliance on meta-learning [35], making it highly sensitive to hyperparameters such as the learning rate.

In summary, SERT achieves superior predictive accuracy with a more streamlined design and relatively lower training time and memory requirements.

4) Prediction Performance in Larger Regions: By default, each city is divided into 1 km × 1 km regions, as this scale effectively represents functional areas in real-world scenarios. To investigate the impact of region size on the model's prediction performance, we expanded each region to 2 km × 2 km (i.e., merging four adjacent regions into a new one) and conducted experiments.

The results are shown in Table V. Clearly, compared to the prediction results under the 1 km × 1 km grid size in Table II, the prediction error for each method significantly increases under the larger grid size. The reasons for this are twofold: on one hand, larger regions exhibit greater traffic flow values, which increases the base value for prediction, making the increase in prediction error reasonable. On the other hand, oversized regions may encompass areas with diverse functions and characteristics, leading to overly complex regional features that make effective urban representation challenging for the model. Consequently, the performance decline for methods utilizing auxiliary data (e.g., SERT, CrossTRes, RegionTrans) is more pronounced. Nevertheless, SERT still achieves the best results among all methods, demonstrating its generality of excellent performance under different grid division strategies.



(a) RMSE for SERT Variants (b) MAE for SERT Variants

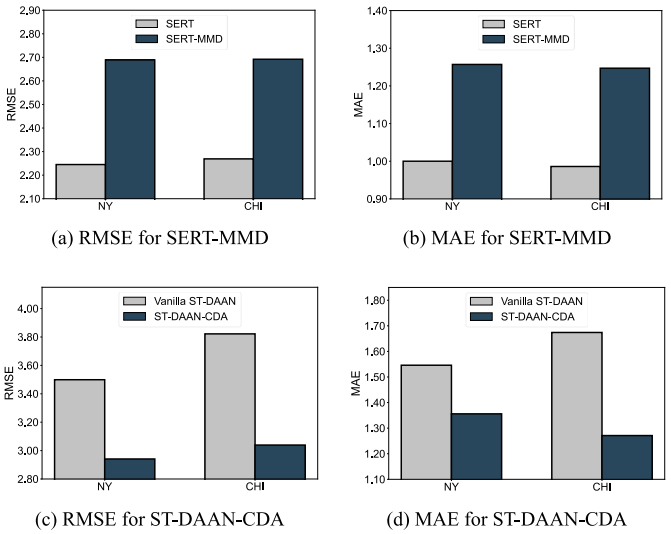
(c) RMSE for RegionTrans Variants (d) MAE for RegionTrans Variants

Fig. 5. Evaluation of the effectiveness of auxiliary data (satellite imagery and POI data) through SERT and RegionTrans variants.

C. Ablation Study

We evaluate the efficacy of the two main design components proposed in this paper: (i) the effectiveness of satellite imagery and POI data as auxiliary data for extracting region-specific features; and (ii) the improvement in prediction performance through regional knowledge transfer via contrastive domain adaptation compared to city-level transfer.

1) Effectiveness of Auxiliary Data: We evaluate the effectiveness of satellite imagery and POI data as auxiliary data, resulting in the development of SERT variants: SERT-POI (using only POI data) and SERT-SAT (using only satellite imagery). Figs. 5(a)-(b) illustrate the prediction performance of SERT and its variants on the *Bike-7days* datasets. When either POI or satellite imagery is used in isolation, prediction performance declines, particularly for SERT-POI. This



(a) RMSE for SERT-MMD (b) MAE for SERT-MMD

(c) RMSE for ST-DAAN-CDA (d) MAE for ST-DAAN-CDA

Fig. 6. Evaluation of the effectiveness of our contrastive domain adaptation approach through SERT and ST-DAAN variants.

highlights the importance of incorporating satellite imagery as an auxiliary data to extract region-specific features. Notably, although SERT-SAT and SERT-POI exhibit slightly inferior performance compared to SERT, their overall performance still surpasses the second-best method, CrossTReS. This demonstrates the reliability of using satellite imagery and POI data individually as auxiliary data, as well as the superiority of SERT's regional-level transfer mechanism.

To further validate the generalizability of our design for extracting region-specific features from satellite imagery, we applied this approach to RegionTrans, creating variants RegionTrans-POI (utilizing only POI data) and RegionTrans-SAT (utilizing only satellite imagery). The results presented in Figs. 5(c)-(d) demonstrate that the inclusion of satellite imagery significantly improves the performance of RegionTrans, suggesting that incorporating satellite imagery as auxiliary data can boost the predictive capabilities of other methods across different cities.

2) Effectiveness of Contrastive Domain Adaptation: To assess the predictive performance improvements achieved through regional transfer, we replaced the contrastive domain adaptation in SERT with the Maximum Mean Discrepancy (MMD)-based alignment method used by ST-DAAN, resulting in a variant named SERT-MMD. This modification downgrades SERT from regional-level transfer to city-level transfer. The MMD criterion aligns spatial-temporal features between the source city and the target city, mathematically expressed as:

$$\mathcal{L}_{mmd} = \left\| \frac{1}{n}\sum_{i=1}^{n}\phi\left(d_S^i\right) - \frac{1}{m}\sum_{j=1}^{m}\phi\left(d_T^j\right) \right\|_{\mathcal{H}}^2, \qquad (12)$$

where ϕ denotes the Gaussian kernel function, $\mathcal{H}$ represents the Hilbert space, and d indicates the data samples [13]. Within SERT-MMD, the original loss term $\mathcal{L}_{cda}$ in (11) is supplanted by $\mathcal{L}_{mmd}$.

Figs. 6(a)-(b) illustrate the performance of SERT-MMD on the *Bike-7days* datasets. The MMD-based domain adaptation method degrades the granularity of knowledge transfer to the city level, introducing harmful source city knowledge and obstructing the application of useful knowledge to appropriate locations in the target city. Consequently, it results in significant performance deterioration.

To further emphasize the necessity of regional transfer, we replaced the MMD-based adaptation in ST-DAAN with our contrastive domain adaptation, termed ST-DAAN-CDA. As shown in Figs. 6(c)-(d) (following the original ST-DAAN paper [13] settings *without using target city data labels for loss calculation*, so the results may be inferior to those in Table II), ST-DAAN-CDA significantly outperforms the original ST-DAAN, benefiting from the refined transfer granularity. This highlights the generalizability of our proposed contrastive domain adaptation across various baselines and confirms the necessity of regional transfer.

3) Impact of Different Image Encoder Backbones on Model Prediction Performance: In Section IV-C.1, ResNet-18 [45] was selected as the backbone to construct the Image Encoder for extracting satellite imagery features, as it is one of the most classic and widely-used backbones in computer vision field with excellent performance in image feature extraction. To analyze the impact of different backbones on SERT's performance and validate SERT's stability, we replaces ResNet-18 in (5) with VGG11 [52] and ViT-B-16 [53] respectively, yielding two variants SERT-VGG and SERT-ViT for comparison with the original SERT, while keeping all other model configurations unchanged.

The traffic flow prediction results of SERT and its variants are shown in Table VI. SERT based on ResNet achieves the best performance. SERT-ViT performs better in a few tasks but is slightly inferior to SERT overall, mainly because ViT's unique operation of splitting input images into patches makes it difficult to capture pixel-level details, especially for high-resolution images such as satellite imagery [54]. SERT-VGG exhibits the worst performance due to the absence of residual connections found in ResNet, which increases the risk of gradient explosion or vanishing and makes the model prone to overfitting. Therefore, this paper adopts ResNet as the default backbone for the Image Encoder.

D. Sensitivity Analysis

We analyze the influence of critical hyperparameters on the performance of SERT, including: the transfer loss weight ω as defined in (11); the temperature parameter introduced in Algorithm 1; the volume of training data available for both source and target cities; the methodology employed for computing regional feature similarity as described in (8); and the size of the region.

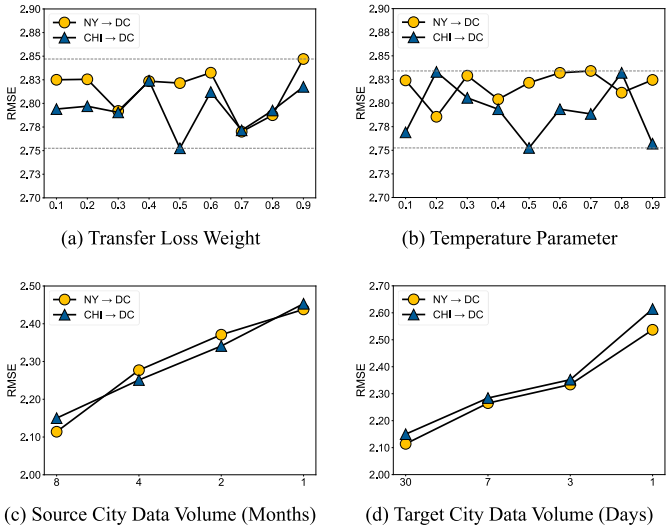
1) Transfer Loss Weight and Temperature Parameter: Figs. 7(a)-(b) demonstrate the performance of SERT on the validation set of the target city concerning variations in the transfer loss weight and the temperature parameter, utilizing the *Bike-7days* datasets. The transfer loss weight balances the traffic flow prediction loss against the knowledge transfer loss in SERT's objective function; higher values increase the influence of the latter. The temperature parameter modulates the distribution of positive and negative samples within the

TABLE VI
PERFORMANCE COMPARISON OF SERT WITH DIFFERENT IMAGE ENCODER BACKBONES FOR TRAFFIC FLOW PREDICTION

Models		SERT		SERT-VGG		SERT-ViT	
Metrics		RMSE	MAE	RMSE	MAE	RMSE	MAE
NY → DC Taxi	30	**3.781**	**1.353**	3.923	1.365	3.842	1.368
	7	**3.976**	1.445	4.129	1.474	4.000	**1.440**
	3	**4.150**	1.493	4.276	1.527	4.232	**1.473**
NY → DC Bike	30	**2.114**	0.954	2.199	1.002	2.127	**0.935**
	7	2.245	1.000	2.346	1.051	**2.233**	**0.970**
	3	**2.341**	**1.033**	2.419	1.079	2.421	1.073
CHI → DC Taxi	30	**3.792**	**1.355**	3.919	1.450	3.886	1.411
	7	**3.971**	**1.408**	4.070	1.436	4.026	1.450
	3	**4.125**	**1.492**	4.192	1.557	4.157	1.519
CHI → DC Bike	30	**2.150**	0.949	2.196	0.959	2.152	**0.934**
	7	**2.269**	**0.986**	2.404	1.034	2.293	0.988
	3	**2.346**	**1.005**	2.424	1.068	2.452	1.050

TABLE VII
THE IMPACT OF SIMILARITY METRICS (ED: EUCLIDEAN DISTANCE, CS: COSINE SIMILARITY) ON MODEL PERFORMANCE

Metrics	NY → DC Taxi						CHI → DC Bike					
	30		7		3		30		7		3	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
ED	3.819	1.368	3.992	**1.422**	4.152	**1.458**	2.173	**0.937**	2.285	**0.972**	2.351	1.007
CS	**3.781**	**1.353**	**3.976**	1.445	**4.150**	1.493	**2.150**	0.949	**2.269**	0.986	**2.346**	**1.005**



(a) Transfer Loss Weight (b) Temperature Parameter

(c) Source City Data Volume (Months) (d) Target City Data Volume (Days)

Fig. 7. Evaluation of SERT's sensitivity to various hyperparameter settings and available data amount.

feature space during contrastive domain adaptation, with lower values resulting in more dispersed sample distributions and enhancing contrast. The results indicate that SERT's performance remains robust within the parameter range (0, 1), with RMSE differences confined to 0.1.

2) Training Data Volume: Figs. 7(c)-(d) depict the impact of varying training data volume for either the source or target city on SERT's performance on the target city's test set, utilizing the *Bike* datasets. In Fig. 7(c), with the target city training set fixed at 30 days, a decrease in source city data results in a performance decline, suggesting that reduced knowledge from the source city adversely affects outcomes. In Fig. 7(d), with the source city training set fixed at 8 months, reducing the target city training set from 30 days to 3 days results in a more gradual performance decline, as the model effectively compensates with source city knowledge. However, when the target city data is limited to just 1 day, performance significantly deteriorates due to the insufficient data (only 24 time steps) for accurately capturing the target city's data distribution, resulting in substantial interference from the source city's data distribution. This observation highlights the distribution discrepancies between domains and emphasizes the necessity of contrastive domain adaptation to align feature distributions across different cities.

3) Similarity Calculation Method: After extracting the auxiliary features of the regions, cosine similarity is used by default to calculate regional similarity for matching region pairs. To evaluate the robustness of SERT, we substituted the cosine similarity metric with Euclidean distance and conducted experiments. The results presented in Table VII demonstrate that using either Euclidean distance (ED) or cosine similarity (CS) has minimal impact on SERT's final predictive performance, indicating the model's resilience to different similarity metrics for assessing regional feature similarity.

E. Case Study

1) Effectiveness of Regional Feature Alignment: To evaluate the effectiveness of contrastive domain adaptation in aligning regional features, we employed t-SNE [55] to project the 256-dimensional features of each region from both source and target cities into a 2D space for visualization. This experiment

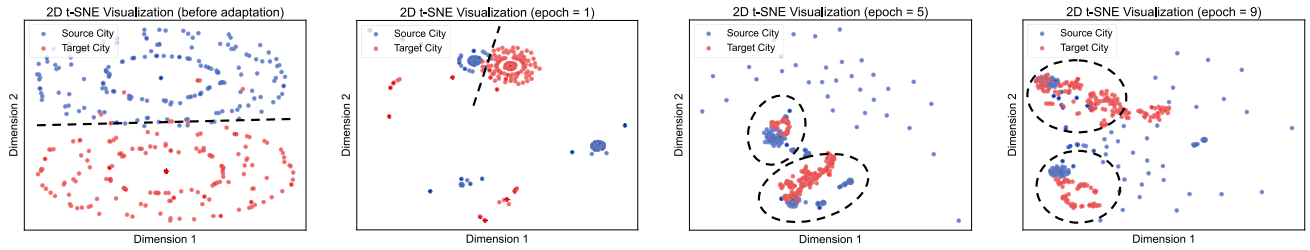


Fig. 8. Evolution of regional feature distributions during the 10-epoch training process of SERT.

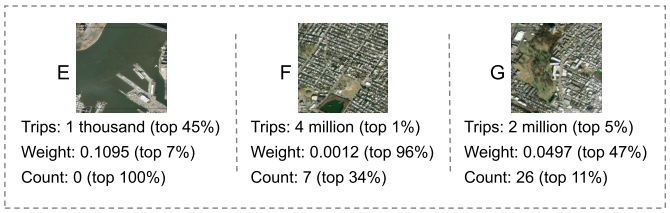


Fig. 9. Comparison on regional knowledge utilization of CrossTReS and SERT for three regions in the source city.

was conducted on the *NY-Bike-7days-pickup* datasets, with the model trained for 10 epochs.

Fig. 8 illustrates the evolution of regional feature distributions throughout the training process. Initially, before domain adaptation, a significant discrepancy exists between the feature distributions of the two cities, attributed to factors such as varying levels of urban development. As knowledge transfer progresses, the features of the source and target regions gradually converge, visually demonstrating the effectiveness of the proposed contrastive domain adaptation in achieving feature alignment between the regions. Notably, at epochs 5 and 9, many source regions are sparsely distributed across the feature space, without overlapping or forming dense clusters. This phenomenon arises from treating dissimilar source regions as negative samples in the contrastive domain adaptation process, thereby increasing their distance from positive samples through the $\mathcal{L}_{cda}$. Consequently, the features of all regions in the feature space are neither overly dense nor excessively dispersed, effectively preventing model collapse.

2) Knowledge Selection Strategy: We compared SERT and CrossTReS using real regional examples to highlight differences in their knowledge selection strategies. CrossTReS [14], as the pioneering approach proposing *selective transfer*, utilizes meta-learning to assign weights to each source region, reflecting its relevance to the target city. In contrast, SERT filters out irrelevant source regions by extracting region-specific features and establishing inter-region relationships.

The two strategies were evaluated using the *NY-Taxi-30days-pickup* datasets. Fig. 9 presents the satellite images of three specific regions in New York, annotated with: (i) Trips: total annual taxi flow for the region; (ii) Weight: influence weight attributed to the region by CrossTReS (a larger value indicates higher regional importance); and (iii) Count: number of target regions this region matches with in SERT. Additionally, the rank of each value among all 460 regions in New York is included (e.g., "top 0%" indicates the highest value). Based on this comparison, the following observations can be made:

- CrossTReS tends to assign relatively high weights to regions with limited traffic flow volumes (e.g., Region E), while assigning lower weights to regions with substantial data (e.g., Regions F and G). Intuitively, regions with more data should encompass richer spatial-temporal knowledge and should and hold greater importance for the target city, yet the actual weighting results from CrossTReS diverge from this principle.
- In contrast, SERT leverages Regions F and G multiple times, ensuring that knowledge from these regions is extensively applied to the target city, while Region E is excluded during the source region selection process due to its limited knowledge.

We further conducted a statistical analysis across all regions in New York to demonstrate that the aforementioned findings are not isolated cases. On average, CrossTReS assigns a weight of 0.092 to all source regions. For regions filtered out by SERT, the average weight assigned by CrossTReS is 0.089, with 49.3% of these regions receiving a weight greater than the overall average of 0.092. In contrast, SERT effectively excludes these regions, with 65.8% of them exhibiting significantly low data volumes.

The above analyses highlight SERT's superiority over CrossTReS in knowledge selection strategy. By eliminating irrelevant source regions, SERT not only ensures higher-quality knowledge transfer but also reduces the computational cost associated with transferring extraneous knowledge. Furthermore, SERT's region-to-region transfer mechanism facilitates the application of selected knowledge to appropriate locations in the target city, whereas CrossTReS lacks a detailed analysis of this process.

VI. Conclusion

In this paper, we propose SERT, a novel transfer learning method designed to enhance traffic flow prediction in data-scarce scenarios. By implementing fine-grained regional transfer, SERT facilitates the transfer of relevant knowledge from the source city to appropriate locations within the target city. To establish inter-regional relationships, we incorporate satellite imagery as auxiliary data to extract region-specific features. We also propose a contrastive domain adaptation approach that aligns features between matched regions and mitigates the negative impact of unmatched regions. Experiments on real-world datasets demonstrate that SERT

outperforms state-of-the-art baselines in prediction accuracy while maintaining exceptional computational efficiency.

REFERENCES

[1] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: Concepts, methodologies, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, pp. 1–55, 2014.

[2] Z. Liu, Z. Li, K. Wu, and M. Li, "Urban traffic prediction from mobility data using deep learning," *IEEE Netw.*, vol. 32, no. 4, pp. 40–46, Jul. 2018.

[3] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2017, vol. 31, no. 1, pp. 1655–1661.

[4] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, "Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning," *Transportmetrica A, Transp. Sci.*, vol. 15, no. 2, pp. 1688–1711, Nov. 2019.

[5] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Acad. Annu. Conf. Chin. Assoc. Autom. (YAC)*, Nov. 2016, pp. 324–328.

[6] S. Lu, Q. Zhang, G. Chen, and D. Seng, "A combined method for short-term traffic flow prediction based on recurrent neural network," *Alexandria Eng. J.*, vol. 60, no. 1, pp. 87–94, Feb. 2021.

[7] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell., (IJCAI)*, 2018, pp. 3634–3640.

[8] Y. Jin, K. Chen, and Q. Yang, "Transferable graph structure learning for graph-based traffic forecasting across cities," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2023, pp. 1032–1043.

[9] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[10] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2009.

[11] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang, "Cross-city transfer learning for deep spatio-temporal prediction," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1893–1899.

[12] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities: A meta-learning approach for spatial–temporal prediction," in *Proc. World Wide Web Conf.*, May 2019, pp. 2181–2191.

[13] S. Wang, H. Miao, J. Li, and J. Cao, "Spatio-temporal knowledge transfer for urban crowd flow prediction via deep attentive adaptation networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 4695–4705, May 2022.

[14] Y. Jin, K. Chen, and Q. Yang, "Selective cross-city transfer learning for traffic prediction via source city region re-weighting," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 731–741.

[15] Y. Li, W. Huang, G. Cong, H. Wang, and Z. Wang, "Urban region representation learning with OpenStreetMap building footprints," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2023, pp. 1363–1373.

[16] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, "Fine-tuning can distort pretrained features and underperform out-of-distribution," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2022, pp. 1–42.

[17] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Sep. 2019, pp. 4893–4902.

[18] S. Herath, M. Harandi, B. Fernando, and R. Nock, "Min-max statistical alignment for transfer learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9280–9289.

[19] C. Yeh et al., "Using publicly available satellite imagery and deep learning to understand economic well-being in Africa," *Nature Commun.*, vol. 11, no. 1, p. 2583, May 2020.

[20] Y. Xi, T. Li, H. Wang, Y. Li, S. Tarkoma, and P. Hui, "Beyond the first law of geography: Learning representations of satellite imagery by leveraging point-of-interests," in *Proc. ACM Web Conf.*, Apr. 2022, pp. 3308–3316.

[21] Y. Liu, X. Zhang, J. Ding, Y. Xi, and Y. Li, "Knowledge-infused contrastive learning for urban imagery-based socioeconomic prediction," in *Proc. ACM Web Conf.*, Apr. 2023, pp. 4150–4160.

[22] M. Burke, A. Driscoll, D. B. Lobell, and S. Ermon, "Using satellite imagery to understand and promote sustainable development," *Science*, vol. 371, no. 6535, 2021, Art. no. eabe8628.

[23] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon, "Combining satellite imagery and machine learning to predict poverty," *Science*, vol. 353, no. 6301, pp. 790–794, Aug. 2016.

[24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, H. D. III and A. Singh, Eds., Jul. 2020, pp. 1597–1607.

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[26] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with Arima time series models to forecast traffic flow," *Transp. Res. C, Emerg. Technol.*, vol. 4, no. 5, pp. 307–318, Oct. 1996.

[27] C. Chatfield, "The holt-winters forecasting procedure," *J. Roy. Stat. Soc., Ser. C (Appl. Statist.)*, vol. 27, no. 3, pp. 264–279, 1978.

[28] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Oct. 2016, pp. 1–4.

[29] H. Yao et al., "Deep multi-view spatial–temporal network for taxi demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, vol. 32, no. 1, pp. 2588–2595.

[30] X. Geng et al., "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3656–3663.

[31] X. Shi, Z. Chen, H. Wang, D. Yeung, W. K. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2015, pp. 1–9.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[33] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 4365–4373.

[34] Y. Wei, Y. Zheng, and Q. Yang, "Transfer knowledge between cities," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1905–1914.

[35] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 1126–1135.

[36] C. Wang, Y. Wu, S. Liu, Z. Yang, and M. Zhou, "Bridging the gap between pre-training and fine-tuning for end-to-end speech translation," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 5, pp. 9161–9168.

[37] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[38] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, vol. 2, Jul. 2015, pp. 1180–1189.

[39] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2015, pp. 97–105.

[40] A. Gretton et al., "Optimal kernel choice for large-scale two-sample tests," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, Dec. 2012, pp. 1205–1213.

[41] B. Lu, X. Gan, W. Zhang, H. Yao, L. Fu, and X. Wang, "Spatio-temporal graph few-shot learning with cross-city knowledge transfer," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 1162–1172.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 60, May 2017, pp. 84–90.

[43] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[44] C. C. Robusto, "The cosine-haversine formula," *Amer. Math. Monthly*, vol. 64, no. 1, pp. 38–40, Jan. 1957.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[46] B. Guo, J. Li, V. W. Zheng, Z. Wang, and Z. Yu, "CityTransfer: Transferring inter- and intra-city knowledge for chain store site recommendation based on multi-source urban data," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 1–23, Jan. 2018.

[47] H. Yan, H. Wang, D. Zhang, and Y. Yang, "Identifying regional driving risks via transductive cross-city transfer learning under negative transfer," in *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2023, pp. 2877–2886.
[48] F. Wang and H. Liu, "Understanding the behaviour of contrastive loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2495–2504.
[49] S. Mannor, D. Peleg, and R. Rubinstein, "The cross entropy method for classification," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 561–568.
[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[51] W. Zhang, L. Deng, L. Zhang, and D. Wu, "A survey on negative transfer," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 2, pp. 305–329, Feb. 2023.
[52] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.
[53] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2020, pp. 1–21.
[54] N. Park and S. Kim, "How do vision transformers work?" in *Proc. 10th Int. Conf. Learn. Represent.*, Jan. 2022, pp. pp. 1–26.
[55] L. Van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, Jan. 2008.

Zhidan Liu (Senior Member, IEEE) received the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2014. After that, he was a Research Fellow with Nanyang Technological University, Singapore, and a Faculty Member with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He is currently an Assistant Professor with the Intelligent Transportation Thrust, System Hub, The Hong Kong University of Science and Technology (Guangzhou). His research interests include the artificial Internet of Things, mobile computing, urban computing, and big data analytic. He is a Senior Member of CCF.

Zhengze Sun received the B.E. degree in data science and big data technology from Wuhan University of Technology, Wuhan, China, in 2022. He is currently pursuing the master's degree with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, under the supervision of Dr. Zhidan Liu. His research interests include spatial–temporal data mining and transfer learning.

Junru Zhang received the B.S. degree from the Department of Computer Science and Technology, Henan Normal University, in 2021. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Zhejiang University. Her research interests include time series data mining, with a specific focus on the development of self-supervised learning and transfer learning.

Bolin Zhang received the B.S. degree in software engineering from Shenzhen University, Shenzhen, China, in 2023, where he is currently pursuing the master's degree with the College of Computer Science and Software Engineering under the supervision of Dr. Zhidan Liu. His research interests include trajectory data analysis and urban computing.

Panrong Tong received the B.E. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2015, and the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore, in 2020. He is currently a Researcher with Alibaba Cloud Computing, China. His research interests include machine learning, intelligent transportation systems, smart city, and urban computing.