Traffic Prediction with Transfer Learning: A Mutual Information-based Approach

Yunjie Huang, Xiaozhuang Song, Yuanshao Zhu, Shiyao Zhang, *Member, IEEE*, and
James J.Q. Yu, *Senior Member, IEEE*,

Abstract—In modern traffic management, one of the most essential yet challenging tasks is accurately and timely predicting traffic. It has been well investigated and examined that deep learning-based Spatio-temporal models have an edge when exploiting Spatio-temporal relationships in traffic data. Typically, data-driven models require vast volumes of data, but gathering data in small cities can be difficult owing to constraints such as equipment deployment and maintenance costs. To resolve this problem, we propose TrafficTL, a cross-city traffic prediction approach that uses big data from other cities to aid data-scarce cities in traffic prediction. Utilizing a periodicity-based transfer paradigm, it identifies data similarity and reduces negative transfer caused by the disparity between two data distributions from distant cities. In addition, the suggested method employs graph reconstruction techniques to rectify defects in data from small data cities. TrafficTL is evaluated by comprehensive case studies on three real-world datasets and outperforms the state-of-the-art baseline by around 8 to 25 percent.

Index Terms—Traffic Prediction, Transfer Learning, Graph Neural Network, Time-series Cluster, Mutual Information.

I. INTRODUCTION

RELIABLE traffic forecasting is considered an indispensable component in modern smart city construction, which can benefit tasks such as route planning, traffic control, and traffic network management [1]–[3] in the Intelligent Transportation Systems (ITSs). A number of studies have investigated traffic prediction for decades. Autoregressive integrated moving average (ARIMA) [4] and Kalman filter have been widely used for time-series data in traffic prediction problems. However, these methods use only historical observations from a single site to make predictions of future speed at that location. As technology advances, data collection devices' geographical information, i.e., longitude and latitude, are easily accessible for constructing spatial relationships among the raw data. Convolutional Neural Networks (CNNs) are applied to capture spatial features, considering that traffic states in nearby areas are influencing each other. In addition, taking the historical observation value of time-series into account, Recurrent Neural Networks (RNN) are popular in capturing temporal features within traffic data.

In recent years, traffic prediction methods, especially graph-based methods, have attracted remarkable attention from the intelligent transportation community [5]. Benefiting from the fact that road networks inherently resemble graph structures, i.e., data collection devices on roads are represented as nodes, and the connection between the roads are represented as edges, Graph Neural Networks (GNNs) are intuitively more suitable than CNNs to capture the non-Euclidean spatial features from the road network [6]. Representative methods including Temporal Graph Convolutional Network (T-GCN) model [5], Spatio-Temporal Graph Convolutional Networks (STGCN) [6], Diffusion Convolutional Recurrent Neural Network (DCRNN) [7] exploit GNNs to model the spatial information of road network and furthermore embed it by the fine-grained high-dimensional features, and achieving satisfactory results.

Although graph-based methods have brought strides in traffic prediction, these methods ignore a practical problem of this ITS application: the difficulty in data acquisition. Despite the emergence of the industrial internet of things and embedded computing devices, massive amounts of traffic data are collected by interconnected stationery or dynamic traffic devices (e.g., loop detectors [8], [9], radio frequency identification detectors [10], and traffic cameras [11]). However, given the high maintenance costs, medium and small cities are unable to deploy or maintain a large number of traffic sensors in the long term to collect sufficient available traffic data. Consequently, these cities may lack the methods of accessing data or the correct prior knowledge to guide the construction of smart cities [12], in particular ITS. Simultaneously, a lack of smart city construction precludes the acquisition of more data. This phenomenon is also known as the "cold start" problem. It also leads to a data imbalance in traffic data collection in different cities. For instance, smaller cities are more challenging to acquire data than larger cities, and they have relatively rare knowledge of their historical patterns. On the other hand, megacities with improving infrastructure can conveniently collect traffic data from more connected vehicles.

Yunjie Huang, Yuanshao Zhu, and James J.Q. Yu are with Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China and Shenzhen Key Laboratory of Safety and Security for Next Generation of Industrial Internet, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: hyjocean132@gmail.com, yasozhu@gmail.com, yujq3@sustech.edu.cn).

Xiaozhuang Song is with the Schools of Data Science at the Chinese University of Hong Kong, Shenzhen, China (e-mail: shawnsxz97@gmail.com).

Shiyao Zhang is with the Research Institute for Trustworthy Autonomous Systems, Southern University of Science and Technology, Shenzhen, China (e-mail: zhangsy@sustech.edu.cn).

Consequently, devising a reliable traffic forecasting algorithm upon scarce and imbalanced data is essential for small and medium cities.

The recent transfer learning technique can be a feasible solution to achieve this objective. The core idea of transfer learning for traffic prediction is to use large amounts of data from other cities to provide intelligent predictions for cities with scarce historical data [13]. It has preliminary applications in other data analytic services, e.g., air quality detection [14], ride-sharing detection [15], and cross-city chain-store site-selection recommendation [16].

In the context of transfer learning, we regard the city that provides a significant volume of data as the *source* city. In the meantime, the *target* city refers to a city where data is scarce. In transferable traffic forecasting tasks, [2] trains a model directly with the large amount of source data collected by the source city and then fine-tunes it with a small amount of target data from the target city in the model's last few layers. [17] used support vector machine, and [18] applied dynamic time wrapping to assist cross-city speed prediction. Others introduced graph clustering to learn data transfer methods [19]. However, the research above faces notable challenges: 1) Uneven data distribution in different cities may easily cause negative transfer of models in transfer learning, which refers to the phenomenon that knowledge learned on the source domain[1] has a negative effect on target domain feature extraction [20]; 2) Although the core discipline of transfer learning is to share the same knowledge in two similar domains, sharing incorrect prior knowledge may lead to significant error accumulation [21]. To summarize, existing transfer learning-based traffic speed forecasting approaches mainly share the following drawbacks:

- **Data distribution differences:** While transfer learning is to learn from the source domain and transfer knowledge to the target domain, in almost all cases, the source and target domains follow two different distributions. The difference in data distribution has a non-negligible adverse impact on the effectiveness of transfer (negative transfer). In addition, [22] points out that even time-series from the same data source have uneven data distribution in different time slots. Therefore, avoiding negative transfer is a significant challenge in this problem.
- **prior knowledge error:** Existing traffic data often record the GPS geographic location of detectors, allowing GNN-based methods to capture features on the non-Euclidean spatial structure constructed by location information of traffic data. This approach requires predefined interconnection information developed by similarity or metric distance, which requires significant domain knowledge [23]. Furthermore, graphs generated in this way are usually counter-intuitive and incomplete, contributing irrelevantly to the target task [21]. In addition, the data collection in the target cities is limited for the high costs of deploying and maintaining facilities. Consequently, there

[1]"Domain" in transfer learning consists of two parts: the feature space of the city's data and its marginal probability distribution. Typically, one city is considered to be one domain.
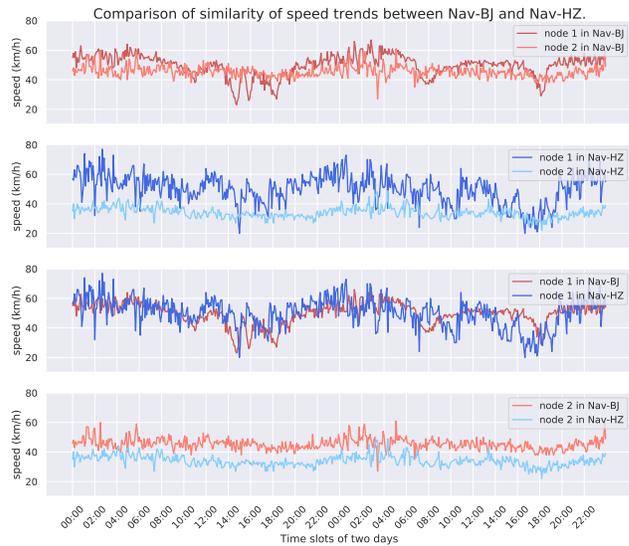

Fig. 1. Red lines represents two Nav-BJ roads and blue lines are two Nav-HZ roads, which come from different datasets. The figure shows a case where roads in the same city can have different daily traffic trend, and ones from different cities can be similar.

are concerns such as missing data from traffic sensors and incorrectly designed adjacency matrix, indicating that prior knowledge is not always reliable.

- **Indistinguishable parameter sharing**: Although deep learning-based traffic prediction algorithms have shown promising results, they tend to accentuate prominent properties in the dataset. The data from the dataset follow the same distribution, making traffic prediction possible within one dataset. However, the shared parameters make current methods perform poorly in capturing fine-grained data features if transferred to another city [21]. As mentioned above, considering the different distributions of two domains, directly transferring the model from the source city to the target may inevitably result in inaccurate predictions. Furthermore, the error caused by such knowledge gaps accumulates during training.

To address these challenges, we design a graph-based traffic prediction framework with transfer learning. In particular, the main highlights of the proposed approach are summarized as follows:

- **Temporal Cluster Block:** Temporal periodicity has been proposed early on traffic tasks but is rarely applied to transfer learning [24]. The graph structure's properties may differ for two data with divergent distributions. However, traffic speed trends over time can be similar in different cities, as shown in Fig. 1. Such common features give us the idea of using transfer learning methods for traffic prediction for small city governors. However, clustering of two differently distributed data in transfer learning with conventional methods may produce degenerate solutions. Therefore, we propose a temporal clustering block (TCB) that uses mutual information to better cluster road segments with similar temporal properties.

- **Graph Reconstruction Block:** For the second challenge, we propose a graph reconstruction block (GRB) that generates a new graph of the clustered data produced by TCB. The clustered data includes a subset of the source and target data. This block uses the clustered data without raw graph structure information to generate a new adjacency matrix. The newly generated graph better focuses on trend-categorically consistent data, i.e., data with stronger speed-time trend relevance, which is helpful to retain similar trend information during the transfer.
- **Ensemble model:** To address the third challenge (indistinguishable parameter sharing), we propose an ensemble model particularly focusing on local parameter sharing. We feed multiple batches of clustered data into different sub-models for training, reducing the data divergence among the sub-models. Consequently, the more data similarity, the better the transfer efficiency, notably reducing negative transfer influence.

The rest of this paper is organized as follows. We first review the background of traffic prediction, time-series cluster, and transfer learning in Sec. II. Then, we present the preliminaries in Sec. III. Sec. IV elaborates on the proposed system. We conduct case studies and provide analytical discussions in Sec. V. Finally, we conclude this paper in Sec. VII.

II. RELATED WORK

In this section, we review previous efforts on traffic prediction. Transfer learning and time-series unsupervised classification-related works are also presented.

A. Traffic Prediction

Traffic prediction has been extensively studied in past decades. Some statistical methods, such as HA [25] and ARIMA [4], are broadly adopted in the time-series community. But these methods only consider temporal information, neglecting the impact of adjoining roadways on each other. Deep learning approaches show superior performance in capturing spatial-temporal correlations and non-linear relations, compared to the statistical methods above and machine learning techniques such as support vector regression (SVR) [26] and K-nearest neighbor (KNN). Recurrent Neural Network (RNN) is well-recognized for modeling time-series by extracting information from past instances to capture temporal features. Yet it suffers from the gradient vanishing and exploding problems [27], [28]. The development of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) [29]–[31] has advanced the capabilities of Recurrent Neural Networks (RNNs) for processing time-series data. These techniques have evolved RNNs into a new generation of models for handling time-dependent data. They address the shortcomings of RNN and GRU, which further significantly alleviates the computation burden by reducing the total number of parameters learned from raw data.

In addition to RNNs and their variations, CNNs are another type of deep learning approach that is frequently utilized to incorporate spatial contextual information for traffic prediction tasks [31]–[33]. In vanilla CNN-based traffic predictors, the geographic map is intuitively divided into square cells to form a grid. Each cell's incoming and outgoing traffic flow data are aggregated as the ground truth for prediction [34]. However, this approach destroys the non-Euclidean structure of the road network. As a recent alternative, graph-based neural networks provide a better choice to preserve the connectivity and adjacency information of traffic networks [5], [35]–[38]. Specifically, Graph Convolutional Networks (GCNs) utilize principles of spectral graph theory to extend the concept of convolutional operations to non-Euclidean domains, as demonstrated in [39], [40]. These networks have been shown to be effective in a variety of tasks, including those involving graph-structured data. For example, [5] presented T-GCN and applied GCN to learn the topology of road networks and GRU to learn the dynamical changes of traffic flow. [38] proposed an Attention-based Spatial-Temporal Graph Convolutional Network (ASTGCN), which incorporates an attention mechanism to better model spatial-temporal correlation. Furthermore, [7] extends the graph convolution operator by introducing diffusion convolution and proposes a deep learning framework named DCRNN for traffic flow prediction. It uses bi-directional random-walk on graphs to capture spatial correlation and encoder-decoder architecture with GRU to capture temporal correlation. [41] uses a dynamic graph to improve traffic flow prediction. [42] proposes a new graph neural network layer with an attention mechanism to better aggregate traffic flows from neighboring roads.

The aforementioned methods provide extra spatial information to traffic prediction while improving the prediction's overall performance. However, the road structures captured by these methods are all predicated based on acquiring complete, high-quality, and large volumes of traffic data. For cities that lack accurate or sufficient historical data, typical traffic predictors are insufficient to generate high-quality forecasts. Alternative solutions are required in this context.

B. Transfer Learning for Traffic Forecasting Problems

Research on building traffic forecasting models has significantly progressed and been applied in practical applications. However, these outstanding results are mainly produced on large-scale datasets, making it difficult to train a model with limited data. Concerning this fact, methods like RegionTrans [43], and MetaST [44] try to tackle the data scarcity problems in traffic forecasting by applying transfer learning techniques. These two works focus on transferring knowledge by building a region-matching and co-training principle between cities. RegionTrans [43] transfers knowledge between cities via regional similarity regularization. MetaST [44] uses a vector array named ST-mem to store the meta-information of source cites' region clusters. It subsequently transfers knowledge between domains by matching the target city's region to ST-mem with an attention mechanism-based approach. Nonetheless, either approach relies heavily on the quality of initial road clustering: anomalous ones may lead to negative transferability. Both methods are built based on parameter sharing, which shares network parameters during model training and transferring. They optimize the model by calculating the losses between the

prediction and ground truth on different matching regions and combining them with pre-defined coefficients. However, they ignore the inductive sharing bias between the source and target domain, i.e., the gradients computed on different sub-datasets clustered by region matching are likely to cause conflicts in the gradient descent direction, thereby undermining the overall performance of the model.

Considering the defect of the aforementioned "global" parameter sharing scheme, local parameter sharing is a worthy strategy to reduce the negative transfer between models. TL_DCRNN [19] reduce information sharing on graphs with fewer correlations by graph partitioning. However, this graph partitioning approach requires that the sub-graph sizes be aligned. Otherwise, the miss-aligned parts will be filled with zero values, adversely affecting learning performance. Moreover, as the graph partitioning operation is required on both the source and target datasets, the accuracy of traffic graphs becomes a crucial basis that influences the transfer efficiency.

In this work, we follow this idea of sharing parameters in local regions. This requires clustering methods to relax the strong dependence on graph structure. Following, we briefly introduce the major existing time-series clustering methods.

C. Time-series Clustering

Time-series clustering methods are typically classified into three categories: *Whole time-series clustering*, *Subsequence time-series clustering*, and *Time points clustering*. We focus on the *Whole time-series clustering* approach for its clear correlation to this work. Interested readers are referred to [45] for a more comprehensive overview of the clustering method.

Whole time-series clustering refers to clustering approaches on discrete objects, where the objects in our work are the discrete nodes with temporal properties. Generally, these clustering methods can be further divided into the following groups: partitioning, hierarchical, grid-based, density-based, model-based, and multi-step clustering [45].

The most common partitioning method divides the n unlabeled nodes into k groups, each containing at least one node from the dataset. K-means [46] and K-Medoids [47] are two widely-recognized representatives. However, they face the challenge of determining the number of clusters k in advance, which is difficult in practice [47]. While the hierarchical analysis does not have this trouble, it is restricted to small datasets due to the quadratic computing complexity [48]. At the same time, the grid-based method is either too slow or too imprecise [49]. And the density-based way is not widely employed in time-series data clustering due to its high complexity [45]. Model-based methods attempt to recover the original model from a set of data. Representative models are Markov chain [50], Neural Network [51], and Gaussian mixture class models [52], etc. Previously, neural networks were considered a less favored technique due to their prolonged execution time and small data volume capacity. Nevertheless, the recent development of computing facilities and deep learning techniques assist neural networks in analyzing larger amounts of data with improved time and accuracy performance. This method is
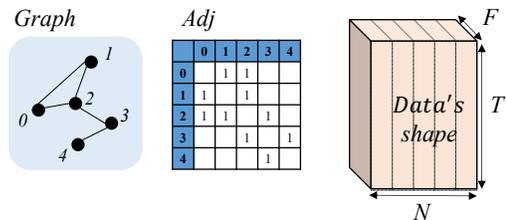

Fig. 2. Graph construction process and the resulting feature tensor.
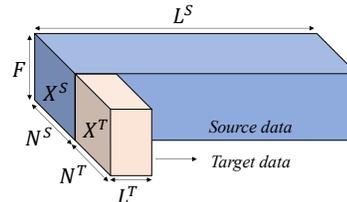

Fig. 3. Data of the source and target cities in transfer learning.

becoming more prevalent than ever in time-series clustering. Finally, the multi-step method integrates multiple methods introduced above, aiming at their respective advantages. In addition, [53] proposes calculating the temporal similarity within a fixed time interval and treating each road section as a single cluster merging the most similar sections bottom-up to achieve clustering. [54] uses a hierarchical clustering algorithm with Pearson correlation coefficients, again by continuously merging similar road sections for clustering. [55] uses Gaussian mixture model clustering. The first two, taken together, use a hierarchical clustering approach to merge bottom-up sections with similar distance similarity functions. Still, this approach has vague termination conditions, making the cluster formation difficult to change. In contrast, the third uses a GMM approach to clustering using probability, which may result in a single data point belonging to multiple clusters, not the effect we want to achieve in our task.

III. TRANSFERABLE TRAFFIC PREDICTION

This section introduces the preliminaries involved in transfer prediction, including the definition of the traffic network, transfer learning in traffic data, and the definition of traffic prediction with transfer learning.

A. Traffic Network

We define each traffic road network as a graph $G = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$, where $\mathcal{V}$ is a set of N nodes (i.e., traffic sensors in the city's roads.); $\mathcal{E}$ is a set of edges, presenting the connectivity among $\mathcal{V}$; $\boldsymbol{A} \in \mathbb{B}^{N \times N}$ is the adjacent matrix of G. Fig. 2 gives a typical example of this graph construction scheme and the resulting tensor formulation. If node 1 has a connection to node 2, $A_{12} = A_{21} = 1$. Note that we reconstruct the adjacency matrix in Sec. IV-B for the proposed approach to capture the data correlation when historical data is insufficient to build the nodal connectivity.
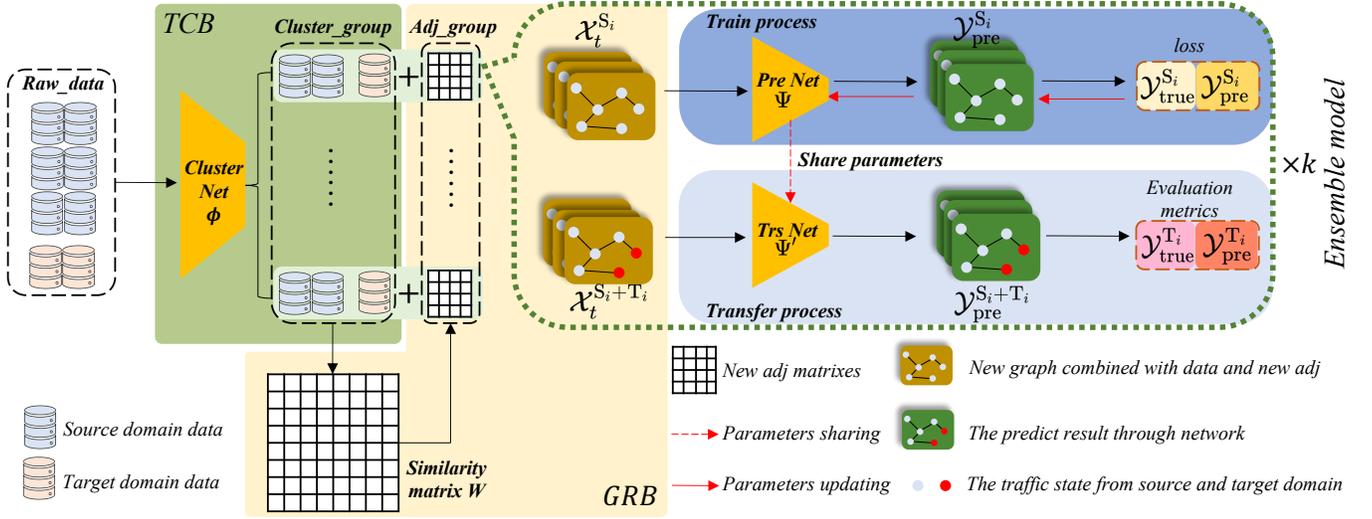
Fig. 4. Overview of the proposed traffic prediction framework with transfer learning. The TCB module clusters data with similar periodicity in the source and target city. GRB reconstructs the traffic graph with the similarity matrix W. The training process uses the source data for Spatio-temporal information extraction, and the transfer process replaces part of the source data with target data to get the prediction results.

B. Transfer Learning in Traffic Network

Transfer learning is the knowledge transfer between the source and the target domain. Let superscripts $\mathcal{S}$ and $\mathcal{T}$ represent the data of the source and target cities, respectively. Subsequently, we denote the road network graph of the source and target cities as $G^{\mathcal{S}} = (\mathcal{V}^{\mathcal{S}}, \mathcal{E}^{\mathcal{S}}, \boldsymbol{A}^{\mathcal{S}})$, $G^{\mathcal{T}} = (\mathcal{V}^{\mathcal{T}}, \mathcal{E}^{\mathcal{T}}, \boldsymbol{A}^{\mathcal{T}})$ respectively. Note that the number of nodes in the two cities can be different. Fig. 3 gives an illustration to the shape of source and target city data, namely, raw source data $X^{\mathcal{S}} \in \mathbb{R}^{L^{\mathcal{S}} \times N^{\mathcal{S}} \times F}$ and raw target data $X^{\mathcal{T}} \in \mathbb{R}^{L^{\mathcal{T}} \times N^{\mathcal{T}} \times F}$, where $N^{\mathcal{S}} \neq N^{\mathcal{T}}$ and the number of nodes in both cities, $L^{\mathcal{S}} \gg L^{\mathcal{T}}$ are the length of time in the raw data from both cities, and F is the number of traffic features. In our work, we primarily investigate the traffic speed data transferability, i.e., $F = 1$.

C. Problem Definition

The primary objective of transferable traffic prediction is to predict future traffic data in data-scarce cities with the transferred traffic knowledge from data-rich cities. In this work, we integrate the data from the target city $X^{\mathcal{T}}$ and the source city $X^{\mathcal{S}}$ in order to obtain *trend-similar* node clusters $\mathcal{C} = (C_1, C_2, \cdots, C_c)$ to facilitate the transfer, where c is the number of clusters. Each cluster C_i includes a subset $X^{\mathcal{S}} \in \mathbb{R}^{L^{\mathcal{S}} \times N_i^{\mathcal{S}} \times F}$, and $X^{\mathcal{T}} \in \mathbb{R}^{L^{\mathcal{T}} \times N_i^{\mathcal{T}} \times F}$ in the source and the target cities, respectively, where $\sum_{i=1}^{c} N_i^{\mathcal{S}} = N^{\mathcal{S}}$ and $\sum_{i=1}^{c} N_i^{\mathcal{T}} = N^{\mathcal{T}}$ are the respective nodes in all clusters.

For each cluster C_i, the H historical traffic observations on graph $G^{\mathcal{S}_i}$ of the source city at time t is denoted as $\mathcal{X}_t^{\mathcal{S}_i} = (x_{t-H}^{\mathcal{S}_i}, \cdots, x_{t-2}^{\mathcal{S}_i}, x_{t-1}^{\mathcal{S}_i}) \in \mathbb{R}^{H \times N_i^{\mathcal{S}} \times F}$. Each $x_t^{\mathcal{S}_i} \in \mathbb{R}^{1 \times N_i^{\mathcal{S}} \times F}$ represents a traffic state on sub-graph $G^{\mathcal{S}_i}$ at time t. The aggregated $\mathcal{X}_t^{\mathcal{S}_i} \in \mathbb{R}^{H \times N_i^{\mathcal{S}} \times F}$ is a set of traffic states from time $t - H$ to time $t - 1$. We apply $\mathcal{X}_t^{\mathcal{S}_i}$ to train a model for predicting the traffic speed of the next Q time instances

$\mathcal{Y}_t^{\mathcal{S}_i} = (x_t^{\mathcal{S}_i}, x_{t+1}^{\mathcal{S}_i}, \cdots, x_{t+Q-1}^{\mathcal{S}_i}) \in \mathbb{R}^{(Q) \times N_i^{\mathcal{S}} \times F}$. In the transferring process, we combine $\mathcal{X}_t^{\mathcal{S}_i}$ and $\mathcal{X}_t^{\mathcal{T}_i}$ to predict the next Q values of the target city $\mathcal{Y}_t^{\mathcal{T}_i} = (x_t^{\mathcal{T}_i}, x_{t+1}^{\mathcal{T}_i}, \cdots, x_{t+Q-1}^{\mathcal{T}_i}) \in \mathbb{R}^{Q \times N_i^{\mathcal{T}} \times F}$. Finally, the following Q-step predictions on all nodes of the target city at time t are derived by the results from all clusters, i.e., $\mathcal{Y}_t^{\mathcal{T}} = \{\mathcal{Y}_t^{\mathcal{T}_i} \in \mathbb{R}^{Q \times N_i^{\mathcal{T}} \times F}\}_i$.

IV. TRAFFICTL

This section proposes a novel transferable traffic prediction model named TrafficTL to address the aforementioned challenges in Sec. I. As depicted in Fig. 4, we first input the integrated source and target city data into the proposed clustering block (TCB), which is used to aggregate nodes with similar temporal trends. With TCB, a group of clusters can be obtained, each containing a portion of nodes with a similar speed-trend in the source and target city. Subsequently, we use a novel graph relation mining (GRB) module for each cluster to calculate the similarities among nodes in the cluster and obtain a similarity matrix $\boldsymbol{W}$, which is used as the adjacency matrix $\boldsymbol{A}$. Finally, the adjacency matrix calculated by GRB and the source city data are adopted in an ensemble module for each cluster to train the network Ψ. During the transfer, We replace a percentage β of the input data in the training network with nodes from the target city. The training and transferable network share parameters to obtain the prediction results from the mixed cities' data. We elaborate on the three key building blocks in the following sub-sections.

A. Temporal Cluster Block

With regard to the first challenge (data distribution difference), we propose Temporal Cluster Block (TCB) to cluster all potential shared factors based on a hidden feature generated by a pre-trained GRU as shown in Fig. 5. We next add a layer FC as a classification layer, which will be retrained with a new
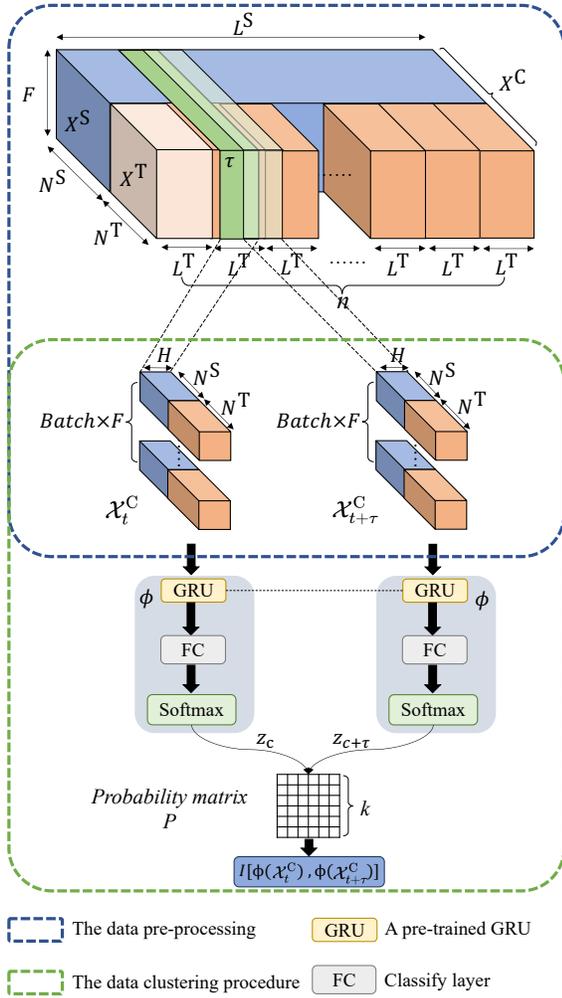
Fig. 5. Data pre-processing and node aggregation in TCB.

number of clusters. The training optimization objective is to maximize the mutual information between the groups obtained after the left and right parts GRU-FC-Softmax models. A comprehensive explanation of the method will be displayed in the following paragraphs. In addition, the process of clustering nodes is referred to as *Node Aggregation* in the sequel.

It is worth noting that the traffic data are naturally unlabeled, limiting the possibility of aggregating nodes in the source and target cities with similar trends only based on the feature data. Therefore, the problem becomes how to use unsupervised learning to cluster similar nodes belonging to two different distributed datasets. To that end, we propose creating a comprehensive dataset that includes both source and target data to eliminate the distribution gap. The samples from the comprehensive dataset contain all nodes of the source and target cities. Following that, we employ a contrastive learning approach to classify all nodes in this comprehensive dataset. Contrastive learning is an unsupervised learning method based on mutual information. The basic idea is to cross-reflect the object's hidden features using different forms of the same object. See refs [56]–[58] for more detailed introduction.

During the process of node aggregation, we keep the set of classification results with the largest mutual information and regard it as inputs for the following blocks. In the sequel, we first give the original definition of mutual information. Then we describe the data pre-processing before node aggregation and, finally, the mutual information computation in this process. Fig. 5 illustrates the overall framework of TCB.

Mutual information refers to the infomax principle proposed by Linsker [59], which plays a vital role in the quality of the representations constructed and learned by generative models [60]. The detailed definition of mutual information is as follows:

Definition 1 (Mutual information): For two discrete variables X and Y whose joint probability distribution is $\mathbb{P}_{XY}(x, y)$, the mutual information between them, denoted by $I(X; Y)$, is defined by

$$I(X; Y) = \sum_{x,y} \mathbb{P}_{XY}(x, y) log \frac{\mathbb{P}_{XY}(x, y)}{\mathbb{P}_X(x, y)\mathbb{P}_Y(x, y)}. \quad (1)$$

Mutual information quantifies the probability that these two variables share information. In the TCB module, we use it as the loss function for the clustering network Φ to select the best number of clusters k and calculate each node belonging cluster.

1) Data Pre-processing: Before aggregating nodes from multiple cities, we pre-process and align the raw data for easier subsequent computation. Without any data pre-processing, traffic speed data of source and target cities are shaped as shown in Fig. 3. We assume the source data to be n times the length of the target data along the time dimension, i.e., n is the ratio of source-target data quantities. To make them of the same lengths, we employ a naïve resampling method and replicate $(n - 1) \times X^{\mathcal{T}}$ along the time dimension to increase the length, i.e., $n \cdot L^{\mathcal{T}} = L^{\mathcal{S}}$ as shown in Fig. 5. (The light orange block is copied to dark orange ones and stacked to reach the same length as the blue one, where n is the number of orange blocks.) Subsequently, we concatenate the source and target city data (the blue and orange blocks) to make $X^{\mathrm{C}} = [X^{\mathcal{S}}, X_1^{\mathcal{T}} \| X_2^{\mathcal{T}} \| \ldots \| X_n^{\mathcal{T}}] \in \mathbb{R}^{L^{\mathcal{S}} \times (N^{\mathcal{S}} + N^{\mathcal{T}}) \times F}$, where $X_i^{\mathcal{T}}$ is the i-th replication of the original $X^{\mathcal{T}}$. The new raw dataset X^{C} contains a total of $N^{\mathcal{S}} + N^{\mathcal{T}}$ nodes. Following the idea of contrastive learning that applies different forms (views) of the same object to cross-reflect the object's hidden features, we sample two subsets from the comprehensive dataset X^{C} with time difference τ, where $0 < \tau < H$. In traffic prediction, we predict the traffic states at future times with historical data containing latent information about the future data. Whereas the two subsets with a time gap of τ have overlapping parts, i.e., they share potential future information. Consequently, their predicted classification results should be intuitively similar. In this work, the comprehensive dataset X^{C} is an object whose two subsets with gap τ are different forms of itself. We expect the classification probabilities of each node learned from these two subsets to be as consistent as possible and to maintain maximum mutual information.

We define these two subsets which sample from the comprehensive dataset as follows: The immediate past H historical observations at time step t is the first subset $\mathcal{X}_t^{\mathrm{C}} =$

$(x^{\mathrm{C}}_{t-H}, \cdots, x^{\mathrm{C}}_{t-2}, x^{\mathrm{C}}_{t-1}) \in \mathbb{R}^{H \times (N^{\mathcal{S}} + N^{\mathcal{T}}) \times F}$. Take another H historical observation traffic states at time step $t + \tau$ as the second subset $\mathcal{X}^{\mathrm{C}}_{t+\tau} = (x^{\mathrm{C}}_{t+\tau-H}, \cdots, x^{\mathrm{C}}_{t+\tau-2}, x^{\mathrm{C}}_{t+\tau-1}) \in \mathbb{R}^{H \times (N^{\mathcal{S}} + N^{\mathcal{T}}) \times F}$ after τ. Then, we combine the two subsets $\mathcal{X}^{\mathrm{C}}_t$ and $\mathcal{X}^{\mathrm{C}}_{t+\tau}$ into a pair of input data $(\mathcal{X}^{\mathrm{C}}_t, \mathcal{X}^{\mathrm{C}}_{t+\tau})$.

2) Node Aggregation: $(\mathcal{X}^{\mathrm{C}}_t, \mathcal{X}^{\mathrm{C}}_{t+\tau}) \in X^{\mathrm{C}}$ is a paired data sampled from a joint probability distribution $P(\mathcal{X}^{\mathrm{C}}_t, \mathcal{X}^{\mathrm{C}}_{t+\tau})$. We devise a neural network called Cluster Net Φ to extract features from the pair of data, making probabilistic predictions about classifying a couple of data with the same distribution and reinforcing the mutual information collected in the classification results. The network comprises a pre-trained GRU, a Fully Connected (FC) layer as a classification layer, and a softmax activation layer. Specifically, GRU aims to extract hidden temporal information. FC and the softmax active layers generate the probability of which clusters the nodes belong to. FC layer is retrained every time with a different number of clusters c. We further acquire probabilistic information of classifications when the inputs are $\mathcal{X}^{\mathrm{C}}_t$ and $\mathcal{X}^{\mathrm{C}}_{t+\tau}$, respectively by a cluster network Φ. Following that, we calculate the likelihood (mutual information) of classification possibility obtained by network Φ, namely, $\mathcal{I}[\Phi(\mathcal{X}^{\mathrm{C}}_t, \mathcal{X}^{\mathrm{C}}_{t+\tau})]]$ depicted in Fig. 5. The output classification probabilities $\Phi_c(\mathcal{X}) \in [0, 1]^c$ over c clusters can be represented by a distribution of a discrete random variable z, namely, $\mathbb{P}(z = c | \mathcal{X}) = \Phi_c(\mathcal{X})$, where c is determined by mutual information, and z is the representative feature of the input data. For $\mathcal{X}^{\mathrm{C}}_t$ and $\mathcal{X}^{\mathrm{C}}_{t+\tau}$, z_{c} and $z_{\mathrm{c}+\tau}$ are their discrete representations, respectively. With the pair of input data $(\mathcal{X}^{\mathrm{C}}_t, \mathcal{X}^{\mathrm{C}}_{t+\tau})$, the conditional joint distributions $\mathbb{P}(z_{\mathrm{c}} = c, z_{\mathrm{c}+\tau} = c' | \mathcal{X}^{\mathrm{C}}_t, \mathcal{X}^{\mathrm{C}}_{t+\tau}) = \Phi_c(\mathcal{X}^{\mathrm{C}}_t) \cdot \Phi_c(\mathcal{X}^{\mathrm{C}}_{t+\tau})$ are specified. This formula suggests that z_{c} and $z_{\mathrm{c}+\tau}$ should be independent. However, for the network Φ, the inputs $\mathcal{X}^{\mathrm{C}}_t$ and $\mathcal{X}^{\mathrm{C}}_{t+\tau}$ obey the same distribution since both of them are taken from the dataset X^{C}. Hence, z_{c} and $z_{\mathrm{c}+\tau}$ are not entirely independent. This accords with the intuition that we can learn from historical data to predict future data since both contain common latent features. Accordingly, the classification probability obtained by the same network Φ should be similar when the two historical data are developed with a time difference τ. In other words, the predicted classification probability for z_{c} can alternatively be indicated by $z_{\mathrm{c}+\tau}$. The joint probability distribution $\mathbb{P}(z_{\mathrm{c}} = c, z_{\mathrm{c}+\tau} = c')$ is given by the $c \times c$ matrix $\boldsymbol{P}_{cc'}$,

$$\boldsymbol{P}_{cc'} = \frac{1}{m} \sum_{i=1}^{m} \Phi_c(\mathcal{X}^{\mathrm{C}}_t) \cdot \Phi_c(\mathcal{X}^{\mathrm{C}}_{t+\tau})^{\top}, \qquad (2)$$

where m is the number of samples. Summing over the rows and columns of this matrix yields marginal probabilities $\boldsymbol{P}_c = \mathbb{P}(z_{\mathrm{c}} = c)$ and $\boldsymbol{P}_{c'} = \mathbb{P}(z_{\mathrm{c}+\tau} = c')$. For convenient calculation, $\boldsymbol{P}_{cc'}$ is symmetrized by

$$\hat{\boldsymbol{P}}_{cc',jk} = \hat{\boldsymbol{P}}_{cc',kj} = \frac{\boldsymbol{P}_{cc',jk} + \boldsymbol{P}_{cc',kj}}{2}, \qquad (3)$$

where j, k are the indices of rows and columns of $\boldsymbol{P}_{cc'}$. The objective function can be correspondingly derived as follows:

$$\mathcal{I}(z_{\mathrm{c}}, z_{\mathrm{c}+\tau}) = \mathcal{I}(\hat{\boldsymbol{P}}_{cc'}) = \sum_{1}^{c} \sum_{1}^{c'} \hat{\boldsymbol{P}}_{cc'} \ln \frac{\hat{\boldsymbol{P}}_{cc'}}{\boldsymbol{P}_c \cdot \boldsymbol{P}_{c'}}, \qquad (4)$$

Algorithm 1: Temporal Cluster Block (TCB)

Input:
 Data $\mathcal{X}^{\mathrm{C}}_t$, $\mathcal{X}^{\mathrm{C}}_{t+\tau}$,
 the number of clusters $c \in \{c_1, c_2, \ldots, c_m\}$
Output:
 Clustering result $\mathcal{C} = \{C_1, C_2, \ldots C_k\}$.
 the optimal number of cluster c
1: Initialize $\mathcal{I}_{\mathrm{bst}} = \text{None}$, $\mathcal{C} = \text{None}$ and $c_{bst} = \text{None}$
2: **for** each c **do**
3: **repeat**
4:   Calculate the mutual information $\mathcal{I}$
5:   **if** $\mathcal{I}_{\mathrm{bst}}$ is $None$ **or** $\mathcal{I}_{\mathrm{bst}} < \mathcal{I}$ **then**
6:    $\mathcal{C}_{\mathrm{bst}} \leftarrow \mathcal{C}$
7:    $c_{bst} \leftarrow c$
8:   **end if**
9: **until** convergence
10: **end for**
11: **return** c_{bst} and $\mathcal{C}_{bst}$.

The goal of TCB is to maximize (4) for the best clustering results and the number of clusters with the largest mutual information. The algorithm is shown in Alg. 1.

B. Graph Reconstruction Block

With respect to the second challenge (prior knowledge error), we propose a new scheme for cluster correction on the distance to compensate for the error. Besides, the scheme also corrects accumulated errors in TCB. In Sec IV-A, we presented employing mutual information for node aggregation of multiple cities' mixed data. The clustering relies entirely on validating the mutual information without using the cluster label of the raw data. While it is true that no label exists in practice to verify the clustering results, the minor clustering error may accumulate and lead to notable performance degradation if not properly counteracted. This issue needs to be resolved for better prediction transferability, and we rely on the proposed GRB to address the two issues.

Distance representation is commonly adopted in various clustering and classification tasks. Among the representations, Euclidean and Manhattan distances are more successful than others for Euclidean-structured data like images. A variety of improved versions have been developed to aid image processing. However, time-series data is not naturally fit for measuring Euclidean distance. This is because time-series are more susceptible to temporal transformations, such as shifts and practical scaling than others, which are inevitable during data collection. If the error between data is established based on Euclidean distance, it may not accurately reflect their similarity. In contrast, DTW is more frequently used to determine the time-series distance.

In the previous section, we employed H historical traffic states for learning data periodicity. For most studies, H historical traffic states tend only to contain information for a limited duration, e.g., 1 hour. On the other hand, the periodicity is statistically observed in other time spans, e.g., days, weeks, and even years. As a result, we apply the DTW approach in

GRB to reconstruct the graph structure among nodes based on long-term internode correlation and strengthen the connection of their periodical similarities.

For cluster $C_i, i \in [1, 2, \cdots, c]$ that contains $N_i^C = N_i^S + N_i^T$ nodes (includes source and target), GRB first randomly selects one day of data on each node. Let D be the length of one day's data[2], and $D \leqslant L^T$. For any two nodes $N_1, N_2 \in \mathbb{R}^{D \times 1}$ in $C_i, i \in [1, 2, \cdots, c]$, we compute their distance matrix $M \in \mathbb{R}^{D \times D}$, $M_{jk} = |N_{1,j} - N_{2,k}|, j, k \in [1, 2, \cdots, D]$. $N_{1,j}$ is the j-th speed of node N_1 and $N_{2,k}$ is the k-th speed of node N_2. The next step is to find the shortest path from M_{11} to M_{DD} as follows:

$$
\begin{aligned}
L_{\min}(1,1) =& M_{11}; \\
L_{\min}(j,k) =& \min\{L_{\min}(j-1,k), L_{\min}(j,k-1), \quad (5) \\
& L_{\min}(j-1,k-1)\} + M_{jk},
\end{aligned}
$$

where $L_{\min}(j, k)$ represents the minimized distance from M_{11} to M_{jk}, and $L_{\min}(D, D)$ is the minimum distance between nodes N_1 and N_2.

The shortest distance $L_{\min}(D, D)$ of every two nodes in the cluster C_i is calculated to form a similarity matrix $W = \{W_{mn}\} \in \mathbb{R}^{N_i^C \times N_i^C}$, where W_{mn} represents the minimum distance between node N_m and $N_n, m, n \in [1, 2, \cdots, N_i^C]$. The adjacency matrix A can be obtained by distance calculation from geographic location information and can also be regarded as a distance similarity matrix. In TrafficTL, to attenuate the adverse influence of incorrect prior adjacency matrix and to bolster the temporal trend connectedness between nodes, we use the similarity matrix W in place of A for convolution operation on graphs.

C. Ensemble Model

In response to the third challenge (indistinguishable parameters), we propose an ensemble scheme to minimize the differences between data of multiple cities by training various models with different clustered subsets to reduce the negative impact of indistinguishable shared parameters in one model. Each cluster in this design (depicted in Fig. 4) has its network Ψ for training and sharing parameters with the transferable network Ψ'. The whole complete model is stacked by c Ψs and Ψ's. The architecture of networks Ψ and Ψ' in TrafficTL is shown in Fig. 6. It is composed of an encoder and a decoder. The encoder includes H GCN-GRU connection layers, each receiving an observation state $x_t, t \in (1, 2, \cdots, H)$, and then saves and delivers the encoded information to the following layer. The decoder comprises Q GCN-GRU layers to recover encoded information and convert it to the following Q-steps traffic states. Among the layers, GCN is a type of neural network that captures spatial information on topological graph structures. The spatial information of the topological graph is represented by the adjacency matrix A, which records the node connections. Additionally, GRU is a variant structure of LSTM and RNN which alleviates the gradient disappearance and explosion problem and has better results in solving the long-term dependence problem over canonical RNNs.

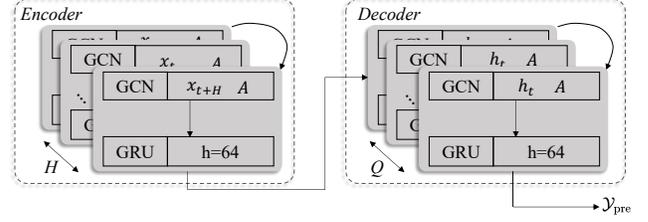[2]We use one day instead of weeks or years due to heavy computation and scarce target city data concerns.

Fig. 6. The architecture of network Ψ's and Ψ''s in ensemble models. H historical traffic states with their adjacency matrix A are fed into the connective GCN-GRU layers. h is the hidden size of the GRU layer. The final output of encoder h_t contains the past H temporal and spatial information. And we get the prediction $\mathcal{Y}_{\text{pre}}$ through the decoder to decode the hidden state h_t.

1) GCN layer: In GCN layers, we denote A as the adjacency matrix, and $\tilde{A} = A + I_N$ is the adjacency matrix with added self-connections, where I_N is the identity matrix. Thereby, the graph Laplacian matrix can be computed as:

$$
L = D^{-\frac{1}{2}} \tilde{A} D^{-\frac{1}{2}}, \quad (6)
$$

where $D = diag(\sum_j A_{ij})$ is the diagonal degree matrix.

For each graph convolution operation, we have

$$
H^{(l+1)} = \sigma(L H^{(l)} \theta^{(l)}), \quad (7)
$$

where l is the layer index, $H^{(l)}$ stands for the output of the l-th layers, $\theta^{(l)}$ represents the l-th layer's parameters, and $\sigma(\cdot)$ is the sigmoid function. Combined with (7), the output is calculated as follows:

$$
f(X_t, A) = \sigma(L \, \text{ReLU}(L X W_0) W_1), \quad (8)
$$

where $f(X_t, A)$ is the output, X_t is the feature matrix of the set $\mathcal{X}_t^{S_i}$ and $\mathcal{X}_t^{T_i}$, W_0 stands for the weight matrix mapping input to the hidden unit, and W_1 represents the weight of the next layer. $\text{ReLU}(\cdot)$ is the Rectified Linear Unit [61]. It is worthwhile to note that the involved adjacency matrix A is reconstructed as previously introduced in Sec IV-B considering the lack of target city data.

2) GRU layer: GRU contains a reset gate r and an update gate z. The former aims to combine new information with previous information, and the latter aims to choose the information that needs to be remembered. The two gates can be expressed as

$$
r_t = \sigma(W_r \cdot [h_{t-1}, X_t]), \quad (9)
$$

$$
z_t = \sigma(W_z \cdot [h_{t-1}, X_t]), \quad (10)
$$

where σ represents the sigmoid activate function, h_{t-1} is the previous state at time step $t-1$, and X_t is the input time-series feature matrix at time t. Through the reset gate r_t, we can get the next state $\hat{h}_t$ which contains part of previous information determined by r_t

$$
\hat{h}_t = \tanh([r_t \odot h_{t-1}, X_t]), \quad (11)
$$

where $\odot$ represents the Hadamard product, the $\tanh$ function can alleviate the gradient explosion and reduce the computation. The update state at the current step is as follows:

$$
h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t. \quad (12)
$$

3) Transfer process: As shown in Fig. 4, we take nodes belonging to the source city in cluster C_i and their adjacency matrix $\boldsymbol{A}^{\mathcal{S}_i}$ to make up the input data $\mathcal{X}_t^{\mathcal{S}_i}$ for training network Ψ_i. $\mathcal{Y}_{\text{pre}}^{\mathcal{S}_i}$ is a set of the following Q traffic states' predictions at time t. Network Ψ_i is optimized by minimizing the L_1 loss between $\mathcal{Y}_{\text{pre}}^{\mathcal{S}_i}$ and $\mathcal{Y}_{\text{true}}^{\mathcal{S}_i}$, where $\mathcal{Y}_{\text{true}}^{\mathcal{S}_i}$ is the Q traffic states' ground truth. Subsequently, we replace a portion (defined by a control parameter β) of the source city nodes $\mathcal{X}_t^{\mathcal{S}_i}$ with higher similarity to the target city nodes (or randomly replace some nodes in the source city), shown as red dots in Fig. 4. The predictive network Ψ_i shares parameters with the transferable network Ψ_i'. $\mathcal{Y}_{\text{pre}}^{\mathcal{S}_i+\mathcal{T}_i}$ is the prediction of the mix-cities input data. Evaluation metrics are calculated by the forecasting results of the target city $\mathcal{Y}_{\text{pre}}^{\mathcal{T}_i}$ with their ground truth $\mathcal{Y}_{\text{true}}^{\mathcal{T}_i}$. The whole process can be described as follows:

$$\mathcal{X}_t^{\mathcal{S}_i+\mathcal{T}_i} \leftarrow \text{Rep}(\beta \cdot \mathcal{X}_t^{\mathcal{S}_i}), \tag{13}$$

$$\Psi_i \overset{\theta_i}{\leftrightarrow} \Psi_i', \tag{14}$$

$$\mathcal{Y}_{\text{pre}}^{\mathcal{T}_i} = \Psi_i'(\mathcal{X}_t^{\mathcal{S}_i+\mathcal{T}_i}), \tag{15}$$

$$\mathcal{M}_i = \text{Evaluation}(\mathcal{Y}_{\text{pre}}^{\mathcal{T}_i}, \mathcal{Y}_{\text{true}}^{\mathcal{T}_i}), \tag{16}$$

where $\text{Rep}(\cdot)$ means replacing β of $\mathcal{X}_t^{\mathcal{S}_i}$ with target nodes, θ_i is the shared parameters in predictive network Ψ_i and transferable network Ψ_i', $\mathcal{M}_i$ is a set of metrics of cluster C_i and $\text{Evaluation}(\cdot, \cdot)$ stands for the performance evaluation metric. At last, we summarize the overall metric $\mathcal{M}$ by,

$$\mathcal{M} = \frac{1}{c}\sum_{i=1}^{c}\mathcal{M}_i, \tag{17}$$

where c is the number of clusters. We also provide the algorithm of Ensemble models in Alg. 2.

V. CASE STUDIES

This section compares the proposed TrafficTL with current state-of-the-art over three widely-recognized datasets. Next, we evaluate the effectiveness of TCB and GRB modules. Finally, we investigate the model sensitivity to hyper-parameters.

A. Dataset and Configurations

In the case studies, we adopt three real-world traffic speed datasets as follows.

- **Nav-BJ:** The dataset collects the average vehicle speeds of $1,362$ on major roads in Beijing, China, from January 1st to July 1st, 2019. Specifically, there are two sets of available data, the adjacency matrix and the feature matrix. The former is binary and consists of $1,362$ roads. The $52,128 \times 1,362$ feature matrix contains $8,990$ time steps, and row data exhibits the speed of all roads at the corresponding time.
- **Nav-SH:** The dataset includes the average vehicle speeds of $1,401$ on major roads in Shanghai, China, from January 1st to May 1st, 2019. The feature matrix's shape in Nav-SH is $34,560 \times 1,401$, and the adjacency matrix shape is $1,401 \times 1,401$. We employ the feature matrix as training data.

Algorithm 2: Ensemble models algorithm

Input:
  Clustering set $\mathcal{C} = \{C_1, C_2, \ldots C_c\}$
Output:
  metrics $\mathcal{E}$
1: Initialize sub-model parameters θ_c
2: **for** each $C \in \mathcal{C}$ **do**
3:   **repeat**
4:     Train model with data $\mathcal{X}^{\mathcal{S}} \in C$
5:     Calculate the loss
6:     Update model parameters θ_k
7:   **until** convergence
8:   Save sub-model M_c
9: **end for**
10: **for** each $C \in \mathcal{C}$ **do**
11:   **for** round $1, 2, \ldots$ **do**
12:     Replace β of all nodes in $\mathcal{X}^{\mathcal{S}}$ with $\mathcal{X}^{\mathcal{T}}$
13:     Get prediction $\mathcal{Y}_{\text{pre}}^{\mathcal{S}+\mathcal{T}}$ with sub-model M_c
14:     Calculate metric between $(\mathcal{Y}_{\text{pre}}^{\mathcal{T}}, \mathcal{Y}_{\text{true}}^{\mathcal{T}})$
15:   **end for**
16:   Calculate average cluster metric $\mathcal{M}_i$
17: **end for**
18: Calculate average total cluster metric $\mathcal{M} = \frac{1}{c}\sum_{i=1}^{c}\mathcal{M}_i$.
19: **return** $\mathcal{M}$

- **Nav-HZ:** The dataset includes the average speed in Hangzhou from January 1st to July 1st, 2019. The survey data includes 413 major roads in Hangzhou, China. It includes a $52,128 \times 413$ feature matrix and a 413×413 adjacency matrix.

Fig. 7. Overview of the three investigated cities with road segments colored by black.

All three datasets were collected with a constant $5\,\text{min}$ interval. The main collection roads of the three datasets are visualized in Fig. 7. We investigate the effect of transfer learning on traffic prediction for cities with scarce data. In particular, we use Nav-SH, which contains fewer data in time horizon, and Nav-HZ, with fewer roads (413) as the target cities to be transferred to. We use only one day of data (i.e., 288 time steps) from the target city as the validation dataset and ten days of data as the test dataset. In order to obtain the correlation on the time trend, max-min normalization is applied to the data.

All experiments are conducted on a Linux server with Intel E5-2620v4 CPUs and GeForce RTX 2080Ti GPUs. All baselines and the proposed model are built with Pytorch 1.7.0 and Python 3.8.3. When making predictions, we use

historical data of the immediate past hour to forecast the next $15\,\mathrm{min}/30\,\mathrm{min}/60\,\mathrm{min}$ traffic speed, namely, $H = 12$, $Q = 3/6/12$, respectively. We train our model with the Adam optimizer. The initial learning rate is set to 0.05, the batch size is 64, the maximum training epoch for proposed models is 50, and the early stopping strategy is adopted where the training is terminated if the best result is not updated for consecutive 5 epochs. The cluster number is 10. The percentage β of the replaced nodes is 0.2.

B. Evaluation Metrics

We adopt the following three metrics to evaluate the performance of models:

1) Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{1}{m} \sum_{i=1}^{m} \left| (Y_i - \hat{Y}_i) \right| \tag{18}$$

2) Root Mean Squared Error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (Y_i - \hat{Y}_i)^2} \tag{19}$$

3) Mean Absolute Percentage Error (MAPE):

$$\mathrm{MAPE} = \frac{100\%}{m} \sum_{i=1}^{m} \left| \frac{\hat{Y}_i - Y_i}{Y_i} \right| \tag{20}$$

where Y_i and $\hat{Y}_i$ are ground truth and predicted speed, and m is the number of samples.

C. Baseline Models

We compare the proposed model with two categories of methods: non-transfer learning and transfer learning methods. For non-transfer learning methods, we employ T-GCN, ASTGCN, DCRNN and XGBoost as baselines. For transfer learning models, we choose RegionTrans, MetaST, and TL-DCRNN:

- T-GCN [5]: A temporal graph convolutional network model combining GCN with GRU.
- ASTGCN [38]: A Spatial-Temporal Graph Convolution Network model based on a self-attention mechanism.
- DCRNN [7]: A traffic predicting model capturing the spatial dependency with bidirectional random walks on the graph.
- XGBoost [62]: An efficient gradient-boosting decision tree algorithm that uses multiple trees to make decisions together.
- RegionTrans [43]: A transferable model with knowledge transfer that matches source cities with similar regions in target cities by learning city-region matching functions.
- MetaST [44]: A spatial-temporal network with a meta-learning paradigm learns the generic initialization of a spatio-temporal network to adapt to the target city and achieve knowledge transfer.
- TL_DCRNN [19]: A model for transferable traffic prediction based on DCRNN model, and transfer the knowledge between source and target cities with graph partitioning and clustering.

D. Model Performance Comparison

Table. I presents the simulation results of the proposed and baseline approaches. Underlines highlight the best-performing results. It is obvious that the proposed TrafficTL outperforms all competing baselines by achieving the lowest MAPE and RMSE. MAE is also the lowest in most cases and is minusculely inferior to the best non-transfer model DCRNN in the two cases.

Compared with the non-transfer models without $\dagger$, TrafficTL adds additional information from the source city data for the traffic prediction task in the target city. As a non-transfer approach, T_GCN, ASTGCN, and DCRNN are trained using one day of the target data training for prediction. TrafficTL, on the other hand, employs the data from the source city instead of the target city in training, which decreases the inaccuracy caused by defective prior knowledge in the target city's data in comparison. It effectively transfers potential knowledge from the source city to the target city, reducing negative transfer. Additionally, TrafficTL is superior to XGBoost in that it can simultaneously learn the data of large cities, allowing it to understand the later state of the target city. XGBoost is capable of learning the temporal features of individual roads from corresponding single nodes, leading to valuable predictions. Specifically, the data used for these predictions is more closely aligned with the distribution of data collected for the target road, allowing for a more accurate representation of traffic conditions on the concrete road. However, the potential for missing data and long intervals between training and test datasets can ultimately hinder the ability of XGBoost to forecast future traffic conditions in a target city accurately. Compared to the model with $\dagger$ and without, incorrect prior knowledge negatively impacts the whole model. Nevertheless, TrafficTL avoids the risk of negativization initially because it does not rely on information from unreliable data.

In terms of the transfer models, Transfer models are overall better than non-transfer models, except for TL_DCRNN. The reason for this is that the model's performance on the target task may be hampered by the generalization gained from the source domain. TL_DCRNN may result in different types of roads being classified together. Then features from other roads may interfere with learning road features in one class, causing negative effects. Therefore, TrafficTL is taken to learn the respective road section situations, even if the same type of intersection is divided into multiple classes, to minimize the occurrence of misclassification and to try to avoid the interference of negative transfer.

TrafficTL outperforms all three baselines, namely, TL_DCRNN, MetaST, and RegionTrans. Compared with TL_DCRNN, TrafficTL uses a more flexible node clustering method (TCB) and generates a comprehensive dataset to alleviate the source and target dataset gap. In addition, an error rectified module (GRB) is applied to eliminate the error between the prior knowledge and the clustered data. Compared with MetaST, RegionTrans, we apply ensemble models to distinguish parameters for reducing the conflicts caused by combining coefficients in the direction of gradient descent. Furthermore, our approach surpasses the non-transfer

TABLE I
THE PERFORMANCE OF TRAFFICTL AND BASELINE APPROACHES

| | | Nav-BJ to Nav-SH | | | | | | | | |
| | | 15 min | | | 30 min | | | 60 min | | |
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Non-transfer	TGCN	6.46	8.69	23.49	6.48	8.73	23.52	6.56	8.78	24.02
	ASTGCN	6.48	8.69	23.74	6.49	8.72	23.65	6.57	8.78	24.16
	DCRNN	3.79	6.20	15.67	4.00	6.33	16.25	4.76	6.96	19.16
	TGCN†	7.34	9.86	26.78	7.56	9.98	27.86	8.01	10.45	28.96
	ASTGCN†	7.86	10.23	28.78	7.94	10.37	29.15	8.23	10.43	29.55
	DCRNN†	4.38	6.54	17.12	6.23	8.02	18.84	6.53	8.64	20.13
	XGBoost	4.33	6.51	14.53	4.45	6.63	14.93	4.59	6.78	15.35
Transfer	TL_DCRNN	6.90	9.76	22.92	7.61	10.85	25.21	7.90	10.76	28.21
	MetaST	4.29	6.17	13.20	4.37	6.27	13.47	4.48	6.40	13.86
	RegionTrans	4.24	6.11	13.34	4.79	6.71	13.64	5.07	7.18	14.03
	TrafficTL (Ours)	4.09	5.87	12.63	4.03	5.69	12.65	4.06	5.71	12.78
	TrafficTL (w/o TCB)	4.49	5.97	13.49	4.55	5.88	13.86	4.59	5.97	14.18
	TrafficTL (w/o GRB)	4.45	5.85	13.42	4.43	5.73	13.48	4.49	5.80	13.82
		Nav-BJ to Nav-HZ								
		15 min			30 min			60 min		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Non-transfer	TGCN	6.20	9.29	19.86	6.25	9.33	20.04	6.36	9.44	20.29
	ASTGCN	6.30	9.29	20.29	6.31	9.33	20.46	6.43	9.39	20.76
	DCRNN	3.74	6.49	13.90	4.03	6.72	14.69	6.15	9.10	20.77
	TGCN†	7.23	10.39	20.18	8.74	12.07	21.28	7.55	10.58	22.31
	ASTGCN†	7.14	10.83	20.98	7.92	11.56	22.15	8.80	13.13	24.46
	DCRNN†	4.70	6.80	14.06	5.18	7.46	15.10	7.85	11.73	22.08
	XGBoost	3.95	5.90	12.27	4.08	6.05	12.65	4.23	6.21	13.07
Transfer	TL_DCRNN	6.88	9.41	18.26	8.14	10.82	24.66	11.56	16.59	30.62
	MetaST	3.42	5.01	11.83	3.52	5.13	12.17	3.65	5.28	12.60
	RegionTrans	4.20	5.55	13.87	4.45	5.89	14.23	4.69	6.26	14.94
	TrafficTL (Ours)	3.07	4.37	9.34	3.02	4.25	9.38	3.09	4.33	9.63
	TrafficTL (w/o TCB)	4.20	5.54	11.77	4.19	5.43	11.99	4.29	5.55	12.37
	TrafficTL (w/o GRB)	4.39	5.90	12.10	3.95	5.72	12.19	4.52	5.77	12.91

† The adjacency matrix of these models was acquired by randomly removing edges on the original adjacency matrix following [56].

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT AREAS. METRICS ARE SHOWN IN MAE/RMSE/MAPE. MAPE IS REPORTED IN PERCENTAGE (%).

| Approach | Nav-BJ to Hongkong | | |
	15 min	30 min	60 min
DCRNN	2.82/4.99/6.72	3.18/5.47/7.80	3.76/6.48/9.50
TrafficTL	2.27/4.57/6.54	2.87/4.80/7.16	3.56/5.98/9.06
	Nav-BJ to PEMSD7		
	15 min	30 min	60 min
DCRNN	2.75/4.48/6.03	2.97/5.15/6.78	4.20/6.91/9.56
TrafficTL	2.05/3.90/4.17	2.34/4.45/4.93	2.72/5.24/5.86

learning from source datasets since the dataset in the source city provides sufficient auxiliary samples for the target with fewer nodes to learn. Fig. 8 visualizes the predicted values of several non-transfer methods for four road sections in Nav-HZ.[3] In order to compare the transfer effects among different countries and datasets, we also used the Hong Kong dataset and the PEMSD7 dataset to compare different types of regions. We use the second-best performing DCRNN in Table I as the comparison model, and Table II shows the performance of TrafficTL on the two datasets. From Table II, we can see that the transfer effect of TrafficTL is improved on both datasets. The improvements are greater on dataset PEMSD7 than Hongkong, which may be due to the difference in the similarity between the datasets.

E. Ablation Study

Three components comprise TrafficTL, where TCB enables unsupervised clustering of traffic nodes across multiple cities, and GRB corrects prior and cumulative errors. Finally, the ensemble calculates the results across different clusters. To

model in terms of long-term series prediction stability. When predicting the long-term traffic speed in the target city, a much larger amount of historical data is required to extract the long-term data dependency. However, such historical data are scarce in the target city. Incorporating source data in this scenario adds value to long-term prediction, which cannot be utilized in other baselines. Finally, the dataset with fewer nodes (Nav-HZ) produces superior results when

[3] DCRNN results are adopted from DL-Traff-Graph [63]. We employ source codes published by the authors for T-GCN [5] and ASTGCN [38].

evaluate the performance of TCB and GRB on TrafficTL, we remove a block each time while keeping the others. Note that the ensemble aggregates the results of each cluster, which change according to the clustering module TCB. Hence, it is coupled with TCB in the ablation study. We define two new variants of TrafficTL as follows:

- Traffic(w/o TCB): The traffic nodes of the two cities are not clustered by TCB.
- Traffic(w/o GRB): The adjacency matrix is the typical $\mathrm{diag}(\boldsymbol{A}^{\mathcal{S}_i}, \boldsymbol{A}^{\mathcal{T}_i}) \in \mathbb{R}^{N_i^C \times N_i^C}$ instead of the similarity matrix $\boldsymbol{W}$ generated by GRB.

The results of the ablation study are shown in Table I. Without the TCB or GRB block, the models' performance falls short of TrafficTL. Nonetheless, TrafficTL(w/o TCB) and TrafficTL(w/o GRB) still perform better in the transfer process from Nav-BJ to Nav-SH and Nav-HZ than the non-transfer models. This result is caused by the fact that there are still data from the source city that can be used as learning samples for the target city's traffic prediction. In addition, a comparison of the results obtained from the transfer of traffic data from Nav-BJ to Nav-SH versus the transfer from Nav-BJ to Nav-HZ revealed that the latter process resulted in a more successful transfer when the target city (Nav-HZ) had a smaller number of road nodes (413). Specifically, the transfer from a source city with a larger number of road nodes (e.g., Nav-BJ) to a target city with a smaller number of road nodes (e.g., Nav-HZ) was found to be more successful than the transfer from the same source city to a target city with a larger number of road nodes (e.g., Nav-SH). This suggests that the presence of fewer road nodes in the target city may facilitate the process of finding similar nodes in the source city in terms of traffic states, leading to a more effective transfer. Moreover, it also indicates that better results can be obtained when performing transfer learning in truly small cities. While it is unfortunate that datasets from three large cities were used as ground truth to validate the model's effectiveness, this finding suggests that better results may be obtained when performing transfer learning on truly small cities that have fewer roads.

F. Transfer ability analyze

In this sub-section, we try to give an example to clarify transferable knowledge. Fig. 9 shows the data collected on a particular road. It can be seen that there are missing values for a significant portion at the beginning of the section. In this experiment, only a tiny portion of the data prior to the green line is utilized as training data, while the portion after the green line is used as test data to evaluate the model's performance.

Fig. 10 shows the predictions of each model for the ground truth between the two red lines in Fig. 9. As demonstrated in Section V-D, XGBoost's performance is largely dependent on the historical data distribution for a specific road. While this focus on individual road data can be beneficial, it also limits XGBoost's ability to make accurate predictions in situations where no available data for learning. As seen in Fig. 10, XGBoost cannot predict the future well from historical values with missing data because it uses historical data for each road segment, i.e., only temporal attributes are involved. Several

other models can learn some traffic states of the road node from the normal recorded values in other adjacent road nodes through the adjacency matrix. However, they are limited by the size of the available data, and the performances of these models are not better than the predictions learned by TrafficTL through a large amount of reliable source city data.

G. Parameter Sensitivity

In this subsection, we investigate the influence of cluster number c and source data quantity n on the transfer prediction performance. Given the small data size (fewer nodes) of Nav-HZ, it is more representative of the task of providing traffic forecasts for small cities by transferring data from large cities. The results presented in this subsection are based on the transfer from Nav-BJ to Nav-HZ.

1) The optimal number of clusters: We explore the ideal number of clusters in this test. Selecting the optimal number of clusters c is a well-known challenging problem in unsupervised clustering [64] In TrafficTL, we explore the optimal cluster number according to mutual information and model performance, and the candidate c values are from 2 to 20. Note that mutual information is calculated according to the likelihood that the possibility of the source and target city's nodes (with dataset X^C) are classified, which is described in detail in Sec. IV-A.

The Mutual Information clustering scores (MI scores) of TCB are shown in Fig. 11. When cluster number c gets large, there is a tendency for the clusters to share more mutual information, which is consistent with the intuition. The larger the number of clusters, the more nodes each cluster contains, and the more fine-grained the same features. However, increasing the number of clusters increases training time and model storage requirements. From Fig. 11, we can observe that when cluster number c reaches 9, the trend of MI increase is slowed down. Besides, the figure also shows the performance (MAPE) of different c in Fig. 11. A negative correlation is clearly reflected between the MI score and MAPE. Furthermore, the cluster block TCB based on mutual information provides an unsupervised node aggregation method. The correlation between the MI score and MAPE demonstrates its effectiveness.

2) Transferability of source data quantity: The concept of transfer learning is to use a large amount of data from other cities to assist cities where data is scarce. Therefore, a natural question arises: how much source information is required to help the target city learn? Finding an appropriate amount of data minimizes the computational effort of the model and improves its efficiency. We investigate the effect of source-target data quantity size ratio n used on the transfer performance.

As illustrated in Fig. 12, MAE and RMSE decrease before $n = 120$, where the degradation is slowed down thereafter. When the source data "richness" is saturated, more source data do not further improve the transferability, thereby, the performance metrics. In addition, increasing n involves more training data. However, the figure illustrates that increasing n does not lead to a consistent linear improvement in performance, resulting in unnecessary computation.

Fig. 8. The predictions of T-GCN, DCRNN, ASTGCN, and TrafficTL for the selected roads in Nav-HZ.
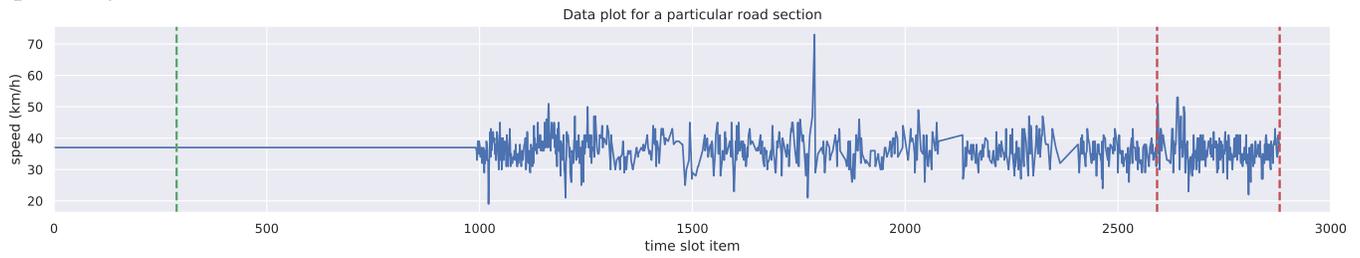
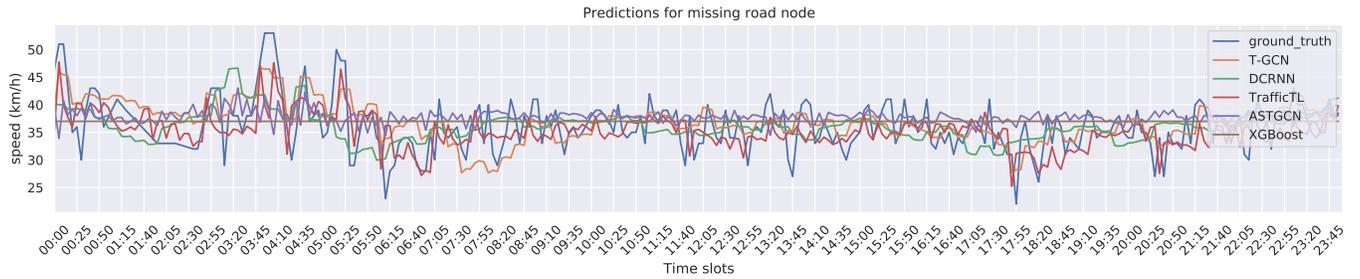

Fig. 9. An illustration of one road in dataset.

Fig. 10. Predictions with several methods of the data between red lines.
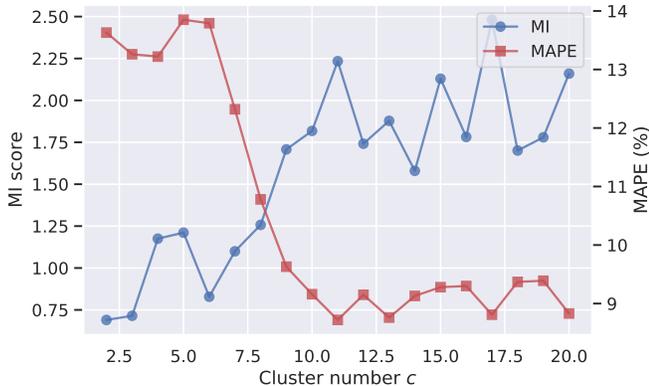
Fig. 11. The mutual information scores and model prediction performance shown with MAPE under different number of clusters.
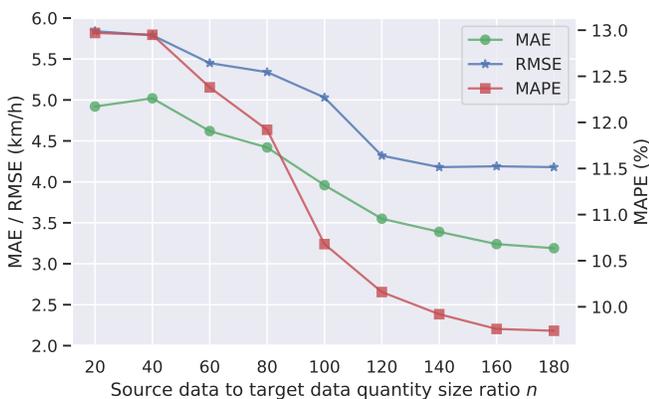

Fig. 12. The performance metrics under various ratio of source-target data quantity size.

VI. DISCUSSION

In this section, we aim to explore potential concerns related to the applications of our work. As mentioned earlier, during the initial stages of urban development, the deployment of equipment can be quite expensive. Our previous sections have highlighted that transfer learning can potentially reduce these costs by offering cost-effective city-wide traffic forecasts. However, two questions naturally arise with this model: "Is it necessary to build a data collection network of fixed-point devices for cities?" and "Why do we need transfer learning?"

Regarding the first problem, Floating Car Data (FCD) has been widely utilized in transportation research for some time. According to a 2008 report [65], FCD offers low cost and ease of access, and with improvements in GPS accuracy, it has become a potentially viable data source for traffic analysis. However, several academic papers have noted challenges in utilizing FCD as a source of information for traffic analysis. These challenges include privacy concerns when working with companies that collect FCD, and difficulties in effectively integrating and processing FCD data collected by the government.

- One such challenge is privacy concerns when working with companies that collect FCD. As FCD collection becomes more feasible with advancements in GPS tech-
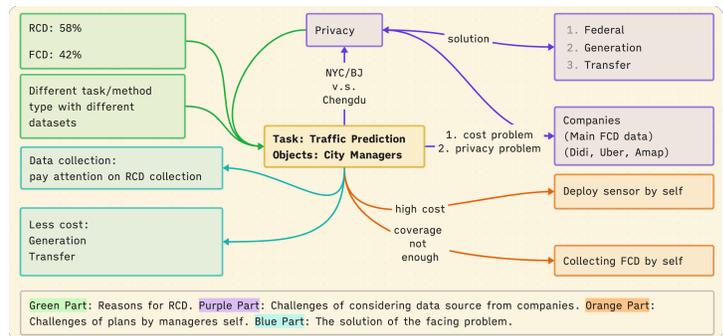

Fig. 13. (best viewed in color) Green Part: Reasons for RCD. Purple Part: Challenges of considering data source from companies. Orange Part: Challenges of plans by managers self. Blue Part: The solution of the facing problem.

nology [66], there is growing concern about protecting personal privacy [67], [68]. When working with major providers of FCD collection sources such as Didi and Uber, governments may become embroiled in commercial transactions and privacy-related issues [69].

- Another challenge is related to data collection by the city itself. There may be difficulties in effectively integrating and processing data when the government collects FCD. Several challenges arise when collecting FCD data, such as inaccurate data resulting from drivers forgetting to end their trips, the complexity of integrating multiple tracks of varying lengths for a particular road [66], [69]–[71], and the limitation of data coverage, as the government can only mandate the installation of FCD collection devices for its controlled departments such as taxi groups, rather than requiring all residents to upload their data [67].

Further, we figured that the form of data used has a high correlation with the task and the proposed method according to [63]. Methods based on graph structures have become more prevalent since the publication of STGCN in 2018, as fixed-point data in the Roadside Collected Data (RCD) style makes it easier to turn traffic networks into graphs that capture spatial information. While 58% of the works still relied on RCD as their data source, 42% utilized FCD as their data source [72]. City managers aiming for growth should consider utilizing both types of data [73] and give greater attention to establishing their own intelligent transportation data collection networks.

Regarding the second problem, data collection can often take several months [66]. Utilizing transfer learning as a means to expedite the implementation of urban prediction models is a viable approach that only entails a modicum of additional expenses [74]. The deployment of detectors by the government is not a one-time affair, and implementing it in phases would prolong the data collection period and extend the time required to build and deploy models. Privacy concerns restrict the public release of collected data, making transfer learning a promising approach to overcome these limitations. Models trained on large datasets from cities may perform poorly on smaller datasets [74], leading to the exploration of alternative methods such as federal learning and data

generation techniques. Transfer learning, which utilizes large amounts of publicly available data to optimize models, is a promising approach that reduces the time and financial burden associated with constructing models during the early stages of urban development.

In conclusion, transfer learning is a cost-effective and beneficial approach for city managers, particularly for those cities in their early stages of development. We have distilled the potential risks, challenges, and questions into Figure 13, offering a visual illustration of the concerns.

VII. CONCLUSION

In this paper, we propose a cross-city traffic prediction method named TrafficTL. This approach can extract common knowledge in datasets from data-rich cities to provide traffic forecasts for data-scarce cities and help traffic management agencies make decisions accordingly. TrafficTL devises a node aggregation scheme (TCB) to reduce the negative impact of the data distribution difference between source and target cities. A graph reconstruction method (GRB) is proposed to rectify the negative impact of prior knowledge inaccuracies from data-scarce cities and the accumulated errors that come from unsupervised clustering in TCB. Additionally, it integrates ensemble learning to distinguish model parameters to lessen the adverse effects of parameter sharing across node clusters.

To evaluate the performance of TrafficTL, we conduct comprehensive experiments based on three real-world datasets. Comparisons with the state-of-the-art baselines show the superiority of TrafficTL over existing solutions. In addition, we test the model performance over a variety of cluster numbers to reveal the influence of clustering on the transferability of TrafficTL. Furthermore, we investigate the ratio of the source data provision in transfer to reduce the computational burden and trial-and-error cost.

In the future, we plan to explore other transfer learning mechanisms from perspectives such as traffic graph construction and graph comparison. We look forward to follow-up research on developing general schemes for traffic transfer prediction based on graph learning. In addition, the findings of our analysis in Section V-E indicate that transfer learning may be effective for small cities with fewer urban roads. However, the data used in our study were drawn from three large cities in order to provide ground-truth for our experiments, which may not be representative of the transfer effects between large and small cities. We look forward to obtaining more consistent data in order to validate our results in the future.

REFERENCES

[1] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2021.

[2] J. Li, F. Guo, Y. Wang, L. Zhang, X. Na, and S. Hu, "Short-term traffic prediction with deep neural networks and adaptive transfer learning," in *IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, 2020, pp. 1–6.

[3] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis, and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4927–4943, 2022.

[4] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using box-jenkins techniques," *Transp. Res. Rec.*, 1979.

[5] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, pp. 3848–3858, 2020.

[6] 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.

[7] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=SJiHXGWAZ

[8] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, pp. 2191–2201, 2014.

[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, pp. 865–873, 2015.

[10] K.-H. Chao and P.-Y. Chen, "An intelligent traffic flow control system based on radio frequency identification and wireless sensor networks," *Int. J. Distrib. Sens. Netw.*, vol. 10, no. 5, p. 694545, 2014. [Online]. Available: https://doi.org/10.1155/2014/694545

[11] Y. Xia, X. Shi, G. Song, Q. Geng, and Y. Liu, "Towards improving quality of video-based vehicle counting method for traffic flow estimation," *Signal Process.*, vol. 120, pp. 672–681, 2016.

[12] P. Wang, J. Lai, Z. Huang, Q. Tan, and T. Lin, "Estimating traffic flow in large road networks based on multi-source traffic data," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, pp. 5672–5683, 2021.

[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, pp. 1345–1359, 2010.

[14] Y. Wei, Y. Zheng, and Q. Yang, "Transfer knowledge between cities," *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016.

[15] L. Wang, X. Geng, X. Ma, D. Zhang, and Q. Yang, "Ridesharing car detection by transfer learning," *Artif. Intell.*, vol. 273, pp. 1–18, 2019.

[16] 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. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 4, jan 2018. [Online]. Available: https://doi.org/10.1145/3161411

[17] F. F. Xu, B. Y. Lin, Q. Lu, Y. Huang, and K. Q. Zhu, "Cross-region traffic prediction for china on openstreetmap," in *Proc. 9th ACM SIGSPATIAL Int. Workshop on Comput. Transp. Sci. (IWCT)*, 2016, p. 37–42. [Online]. Available: https://doi.org/10.1145/3003965.3003972

[18] B. Y. Lin, F. F. Xu, E. Q. Liao, and K. Q. Zhu, "Transfer learning for traffic speed prediction: A preliminary study," in *The Workshops of the The Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, ser. AAAI Technical Report, vol. WS-18. AAAI Press, 2018, pp. 174–177. [Online]. Available: https://aaai.org/ocs/index.php/WS/AAAIW18/paper/view/16708

[19] T. Mallick, P. Balaprakash, E. Rask, and J. MacFarlane, "Transfer learning with graph neural networks for short-term highway traffic forecasting," in *25th Int. Conf. Pattern Recognit. (ICPR)*, 2021, pp. 10 367–10 374.

[20] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, pp. 1345–1359, 2010.

[21] L. BAI, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Advances in Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 17 804–17 815. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/ce1aad92b939420fc17005e5461e6f48-Paper.pdf

[22] Y. Du, J. Wang, W. Feng, S. Pan, T. Qin, R. Xu, and C.-J. Wang, "Adarnn: Adaptive learning and forecasting of time series," *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2021.

[23] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Assoc. Advancement Artif. Intell. (AAAI)*, 2019.

[24] Y. Zou, X. Hua, Y. Zhang, and Y. Wang, "Hybrid short-term freeway speed prediction methods based on periodic analysis," *Can. J. Civ. Eng.*, vol. 42, pp. 570–582, 2015.

[25] D. J. Jeffery, K. Russam, and D. I. Robertson, "Electronic route guidance by autoguide: The research background," *Traffic Eng. Control.*, vol. 28, pp. 525–529, 1987.

[26] A. Ding, X. Zhao, and L. Jiao, "Traffic flow time series prediction based on statistics learning theory," *Proc. IEEE 5th Int. Conf. Int. Transp. Syst. (ITSC)*, pp. 727–730, 2002.

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

[28] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. 30th Int. Conf. Mach. Learn.*, vol. 28, no. 3, 17–19 Jun 2013, pp. 1310–1318. [Online]. Available: https://proceedings.mlr.press/v28/pascanu13.html

[29] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transp. Res. Part C Emerging Technol.*, vol. 90, pp. 166–180, 2018.

[30] Y. Gu, W. Lu, L. Qin, M. Li, and Z. Shao, "Short-term prediction of lane-level traffic speeds: A fusion deep learning model," *Transp. Res. Part C Emerging Technol.*, vol. 106, pp. 1–16, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X18315687

[31] V. Mittal, S. Sasetty, R. Choudhary, and A. Agarwal, "Deep-learning spatiotemporal prediction framework for particulate matter under dynamic monitoring," *Transp. Res. Rec.*, vol. 2676, no. 8, pp. 56–73, 2022. [Online]. Available: https://doi.org/10.1177/03611981221082589

[32] 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, pp. 865–873, 2015.

[33] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Assoc. Advancement Artif. Intell. (AAAI)*, 2017.

[34] X. Ma, H. Yu, Y. Wang, and Y. Wang, "Large-scale transportation network congestion evolution prediction using deep learning theory," *PLoS ONE*, vol. 10, no. 3, p. e0119044, 2015. [Online]. Available: https://app.dimensions.ai/details/publication/pub.1046447126

[35] J. J. Q. Yu, C. Markos, and S. Zhang, "Long-term urban traffic speed prediction with deep learning on graphs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7359–7370, 2022.

[36] C. Zhang, S. Zhang, J. J. Q. Yu, and S. Yu, "Fastgnn: A topological information protected federated learning approach for traffic speed forecasting," *IEEE Trans. Ind. Inf.*, vol. 17, no. 12, pp. 8464–8474, 2021.

[37] J. J. Q. Yu, "Graph construction for traffic prediction: A data-driven approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15 015–15 027, 2022.

[38] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. Assoc. Advancement Artif. Intell. (AAAI)*, vol. 33, no. 01, 2019, pp. 922–929.

[39] S. Yang, W. Ma, X. Pi, and S. Qian, "A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources," *Transp. Res. Part C Emerging Technol.*, vol. 107, pp. 248–265, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X18313780

[40] X. Xiong, K. Ozbay, L. Jin, and C. Feng, "Dynamic origin–destination matrix prediction with line graph neural networks and kalman filter," *Transp. Res. Rec.*, vol. 2674, no. 8, pp. 491–503, 2020. [Online]. Available: https://doi.org/10.1177/0361198120919399

[41] H. Peng, B. Du, M. Liu, M. Liu, S. Ji, S. Wang, X. Zhang, and L. He, "Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning," *Inform sciences*, vol. 578, pp. 401–416, 2021.

[42] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, "Traffic flow prediction via spatial temporal graph neural network," in *Proc. Web Conf. (WWW)*, 2020, pp. 1082–1092.

[43] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang, "Cross-city transfer learning for deep spatio-temporal prediction," in *Int. Joint Conf. Artif. Intell. (IJCAI)*, 2019.

[44] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. J. Li, "Learning from multiple cities: A meta-learning approach for spatial-temporal prediction," *World Wide Web Conf. (WWW)*, 2019.

[45] S. R. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering - a decade review," *Inf. Syst.*, vol. 53, pp. 16–38, 2015.

[46] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probability*, vol. 1, no. 14, 1967, pp. 281–297.

[47] A. S. Hadi, "Finding groups in data: An introduction to chster analysis," *Technometrics*, vol. 34, pp. 111–112, 1991.

[48] X. Wang, K. Smith, and R. Hyndman, "Characteristic-based clustering for time series data," *Data Min. Knowl. Discov.*, vol. 13, no. 3, pp. 335–364, 2006.

[49] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *ACM SIGKDD Conf. Knowl. Discov. and Data Min. (KDD)*, 1996.

[50] M. Ramoni, P. Sebastiani, and P. Cohen, "Multivariate clustering by dynamics," in *Assoc. Advancement Artif. Intell. / Innovative Appl. Artif. Intell. (AAAI/IAAI)*, 2000, pp. 633–638.

[51] X. Wang, K. A. Smith, and R. J. Hyndman, "Dimension reduction for clustering time series using global characteristics," in *Computational Science–ICCS 2005: 5th International Conference, Atlanta, GA, USA, May 22-25, 2005, Proceedings, Part III 5*. Springer, 2005, pp. 792–795.

[52] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 7, pp. 719–725, 2000.

[53] Z. Zhang, F. He, X. Lin, Y. Wang, and M. Li, "High-performance traffic speed forecasting based on spatiotemporal clustering of road segments," *IET Intel. Transport Syst.*, vol. 15, no. 2, pp. 225–234, 2021.

[54] G. Shen, C. Chen, Q. Pan, S. Shen, and Z. Liu, "Research on traffic speed prediction by temporal clustering analysis and convolutional neural network with deformable kernels (may, 2018)," *IEEE Access*, vol. 6, pp. 51 756–51 765, 2018.

[55] Y.-j. Kim, J.-s. Hong *et al.*, "Urban traffic flow prediction system using a multifactor pattern recognition model," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2744–2755, 2015.

[56] Y. Zhu, Y. Xu, Q. Liu, and S. Wu, "An empirical study of graph contrastive learning," in *Proc. Neural Inf. Process. Syst. Track Datasets Benchmarks*, vol. 1, 2021. [Online]. Available: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/0e01938fc48a2cfb5f2217fbfb00722d-Paper-round2.pdf

[57] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, 13–18 Jul 2020, pp. 4116–4126.

[58] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 5812–5823, 2020.

[59] R. Linsker, "Self-organization in a perceptual network," *Comput. J.*, vol. 21, pp. 105–117, 1988.

[60] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *Int. Conf. Learn. Represent. (ICLR)*, 2019. [Online]. Available: https://openreview.net/forum?id=Bklr3j0cKX

[61] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Int. Conf. Mach. Learn. (ICML)*, 2010.

[62] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD)*, 2016, pp. 785–794.

[63] R. Jiang, D. Yin, Z. Wang, Y. Wang, J. Deng, H. Liu, Z. Cai, J. Deng, X. Song, and R. Shibasaki, "Dl-traff: Survey and benchmark of deep learning models for urban traffic prediction," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2021, pp. 4515–4525.

[64] X. Wang, K. Smith, and R. Hyndman, "Characteristic-based clustering for time series data," *Data Min Knowl Discov*, vol. 13, no. 3, pp. 335–364, 2006.

[65] G. Leduc *et al.*, "Road traffic data: Collection methods and applications," *Work. Papers Energy, Transp. Climate Change*, vol. 1, no. 55, pp. 1–55, 2008.

[66] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. M. Choudhury, and A. K. Qin, "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1544–1561, 2020.

[67] C. Costa, G. Chatzimilioudis, D. Zeinalipour-Yazti, and M. F. Mokbel, "Towards real-time road traffic analytics using telco big data," in *Proc. Int. workshop on real-time business intelligence and analytics*, 2017, pp. 1–5.

[68] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang, "Cross-city transfer learning for deep spatio-temporal prediction," *Proc. 22nd Int. Joint Conf. Artif. Intell (IJCAI)*, 2019.

[69] A. Neilson, B. Daniel, S. Tjandra *et al.*, "Systematic review of the literature on big data in the transportation domain: Concepts and applications," *Big Data Res.*, vol. 17, pp. 35–44, 2019.

[70] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *ACM T. Intel. Syst. Tec.*, vol. 5, no. 3, pp. 1–55, 2014.

[71] K. Zhao, S. Tarkoma, S. Liu, and H. Vo, "Urban human mobility data mining: An overview," in *2016 IEEE Int. Conf. Big Data*. IEEE, 2016, pp. 1911–1920.

[72] E. L. Manibardo, I. Laña, and J. Del Ser, "Deep learning for road traffic forecasting: Does it make a difference?" *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6164–6188, 2021.
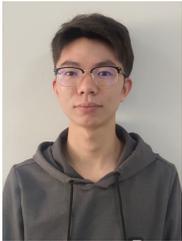
[73] J. Holmgren, H. Fredriksson, and M. Dahl, "Traffic data collection using active mobile and stationary devices," *Procedia Comput. Sci.*, vol. 177, pp. 49–56, 2020.

[74] L. Wang, B. Guo, and Q. Yang, "Smart city development with urban transfer learning," *Comput.*, vol. 51, no. 12, pp. 32–41, 2018.

James J. Q. Yu (S'11–M'15–SM'20) is an assistant professor at the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China, and an honorary assistant professor at the Department of Electrical and Electronic Engineering, the University of Hong Kong. He received the B.Eng. and Ph.D. degree in electrical and electronic engineering from the University of Hong Kong, Pokfulam, Hong Kong, in 2011 and 2015, respectively. He was a post-doctoral fellow at the University of Hong Kong from 2015 to 2018. His general research interests are in smart city and privacy computing, deep learning, intelligent transportation systems, and smart energy systems. His work is now mainly on forecasting and decision making of future transportation systems and artificial intelligence techniques for industrial applications. He was listed in the World's Top 2% Scientists of from 2020 to 2022 by Stanford University, ranked at top 0.32% of all Artificial Intelligence scholars. He is an Editor of the IET SMART CITIES journal and a Senior Member of IEEE.

Yunjie Huang received the B.Eng degree from the Northeastern University at Qinhuangdao, China, in 2020. She is currently a Master Student with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. Her research interests include transfer learning in smart city and graph representation, deep learning and intelligent transportation systems.

Xiaozhuang Song (S'20) received the B.Eng. degrees in digital media technology from Beijing University of Posts and Telecommunications, Beijing, China in 2015, and the M.S degree in computer science and technology from the Southern University of Science and Technology, Shenzhen, China in 2022. He is currently a research assistant at The Chinese University of Hong Kong, Shenzhen. His research interests include multitask learning, graph neural networks, and intelligent transportation systems.
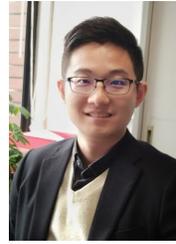

Yuanshao Zhu (Student Member, IEEE) received the B.Eng. degree in communication engineering from the Shandong University, Weihai, China, in 2019 and the M.S. degree in electrical science and technology from the Southern University of Science and Technology, Shenzhen, China, in 2022. He is currently pursuing the Ph.D. degree with the Southern University of Science and Technology and the City University of Hong Kong joint program. His research interests include deep learning in smart city, intelligent transportation systems, and federated learning.

Shiyao Zhang (S'18-M'20) received the B.S. degree (Hons.) in Electrical and Computer Engineering from Purdue University, West Lafayette, IN, USA, in 2014, the M. S. degree in Electrical Engineering from University of Southern California, Los Angeles, CA, USA, in 2016, and the Ph.D. degree from the University of Hong Kong, Hong Kong. He was a Post-Doctoral Research Fellow with the Academy for Advanced Interdisciplinary Studies, Southern University of Science and Technology from 2020 to 2022. He is currently a Research Assistant Professor with the Research Institute for Trustworthy Autonomous Systems, Southern University of Science and Technology. His research interests include smart cities, smart energy systems, intelligent transportation systems, optimization theory and algorithms, and deep learning applications.