Coupling Makes Better: An Intertwined Neural Network for Taxi and Ridesourcing Demand Co-Prediction

Jie Zhao, *Graduate Student Member, IEEE*, Chao Chen, *Senior Member, IEEE*, Wanyi Zhang, Ruiyuan Li, Fuqiang Gu, *Member, IEEE*, Songtao Guo, *Senior Member, IEEE*, Jun Luo, and Yu Zheng, *Fellow, IEEE*

***Abstract*— While a variety of innovative travel modes, such as taxi service and ridesourcing service, have been launched to improve the transportation efficiency, people still encounter travel problems in real life. The major cause is the imbalance between transportation supply and demand. To strike a balance, it is well-recognized that an accurate and timely passenger demand prediction model is the foundation to enable high-level human intelligence (i.e., taxi drivers) or machine intelligence (i.e., ride-hailing platforms) to allocate resources in advance. Although quite a lot of deep models have been designed to model the complicated spatial and temporal dependencies in a data-driven way, they focus on the demand prediction of a single mode and ignore the fact that passengers may shift between different modes, especially between taxis and ridesourcing cars. In this paper, we target a co-prediction problem that considers the prediction of taxi and ridesourcing as two coupled and associated tasks, and propose a novel Temporal and Spatial Intertwined Network (TSIN) that consists of two twin components and an intertwined component. Each twin in the TSIN model is able to extract spatial and temporal dependencies from its corresponding travel mode separately (i.e., intra-mode features), and the in-between intertwined component is designed to bridge the twins and allow them to exchange information (i.e., inter-mode features), thus enabling better prediction. We first evaluate our model on four real-world datasets. Results demonstrate the outstanding performance of our model and the necessity to take into account the influence between modes. Based on an additional demand data from bike in NYC, we then discuss the generalizability in coupling more transportation modes. Further results demonstrate that our proposed intertwined neural network is highly flexible and extendable, and can yield better prediction performance.**

***Index Terms*— Demand prediction, ridesourcing, temporal and spatial dependencies, intertwined neural network, deep models.**

Manuscript received 12 November 2022; revised 6 May 2023 and 21 August 2023; accepted 1 September 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62322601, Grant 62172066, Grant 42174050, and Grant 62202070; in part by the Excellent Youth Foundation of Chongqing under Grant CSTB2023NSCQJQX0025; and in part by the China Post-Doctoral Science Foundation under Grant 2022M720567. The Associate Editor for this article was Y. Tian. *(Jie Zhao and Chao Chen are co-first authors.) (Corresponding author: Wanyi Zhang.)*

Jie Zhao, Wanyi Zhang, Ruiyuan Li, Fuqiang Gu, and Songtao Guo are with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: wanyi.zhang@cqu.edu.cn).

Chao Chen and Jun Luo are with the State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing 400044, China.

Yu Zheng is with JD Intelligent Cities Research and JD Intelligent Cities Business Unit, Beijing 100176, China.

This article has supplementary downloadable material available at https://doi.org/10.1109/TITS.2023.3312224, provided by the authors.

Digital Object Identifier 10.1109/TITS.2023.3312224

I. Introduction

IN RECENT decades, the rising travel demands coming after the overcrowded population imposes huge stress and serious challenges on the urban transportation systems [5], [8], [9], [31], [44]. To address the increasing difficulty in travelling, a variety of innovative travel modes have been launched and become dominant. As the most representative, the Mobility-on-Demand (MoD) services provided by street-hailing taxi companies and Transportation Network Companies (TNCs) such Uber, Lyft, and DiDi, offer passengers flexible and convenient travel choices. Nevertheless, the travel difficulty problem remains unsolved or even worse, mainly due to the imbalance between transportation supply and demand.

To alleviate such imbalance issue, it is well-recognized that the short-term passenger demand prediction is the most fundamental and essential for no matter human beings (i.e., taxi drivers) or machines (i.e., ride-hailing platforms) to make the best online decisions [20]. On the one hand, the future predictions can help taxi drivers seek out potential passengers as quickly as possible, thus reducing their cruising time around the streets. Such an effective passenger-finding strategy, supported by timely predictions, enhances not only the operational efficiency of traditional taxi companies, but also the satisfaction of passengers. On the other hand, the ride-hailing platforms can also benefit from the predicted demands. For instance, the demand predictor is usually built into the vehicle dispatching system [51], and provide a practical understanding of the relationship between supply and demand. In this way, TNCs are able to better match potential demands with their current vehicle supply, and conduct reasonable rider assignment and vehicle scheduling. Moreover, simultaneous prediction of multi-modal demands in the transportation system can also provide more comprehensive information for traffic management.

It is worth noting that, the taxi service and ride-hailing service are quite *different but related* transportation modes [25]. However, most of the existing studies focus on the prediction of demands for a single mode [33], [36], [40], [46], [48]. These prediction models are based on a premise that single-transportation-mode service is a rather independent system, and the future demands are only related to the historically observed demands. In fact, passengers usually do not stick to one transportation mode. They may shift among different

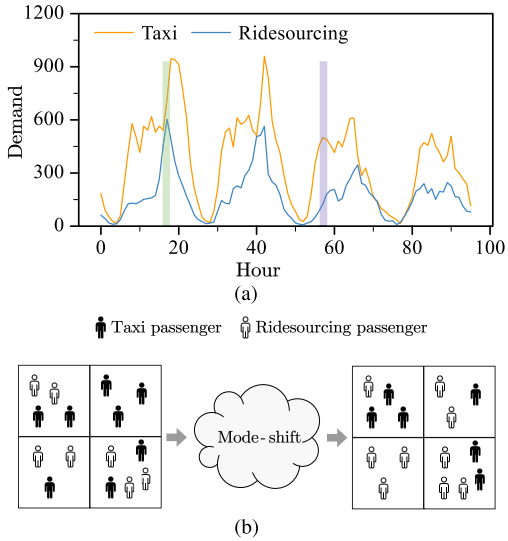


Fig. 1. An illustrative example showing the temporal dependency (a) and the mode-shift effect (b). Curves are plotted based on the real demand data of the taxi mode and the ridesourcing mode.

modes as needed. *While mode-shift behaviours exist, they happen more intensively and frequently between taxis and ridesourcing cars* [12], [13], [25]. For example, when a passenger has waited too long for a taxi, he/she may turn to the ride-hailing service, but seldom seeks for buses/metros around. Inversely, due to the surge pricing strategy of the platform, a ride-hailing passenger may choose to get a taxi to save money if the dynamic price is too high. These mode-shift behaviours indicate that the passenger demands for taxi and ride-hailing are different but interact with each other. In other words, the aforementioned premise does not always hold. Moreover, these influences dynamically exist in both spatial and temporal dimensions, and form a complicated relationship between each mode demand.

Inspired by the above observations, we believe that the demand prediction for one mode should take into account not only its historical demands but also the dynamic changes of demands from other modes. The prediction of demands for taxi mode and ride-hailing mode should be coupled and viewed as two associated tasks. In this regard, it is natural to aggregate them into a multi-task learning framework to make co-predictions. Overall, it is an approach to improve the transportation system efficiencies and it is worth to investigate. It is also necessary to point out that although the co-prediction can theoretically take advantage of multi-source data, it still faces the two main challenges:

Challenge 1: It is challenging to model the dynamic temporal dependencies from both intra- and inter-mode perspectives. Intuitively, the future demand for one mode is influenced by its historical variations, but could also involve potential dependencies with the demand sequence of another mode. As shown in Fig. 1 (a), taxi and ridesourcing demand sequences have their individual temporal patterns, and also correlate with each other at some time periods. For example, the peak of taxi demand comes after the peak of ridesourcing demand (the green bar in the figure), and sometimes the situation is reversed (see the purple bar). Worse still, such temporal interactions between modes are evolving over time and are intertwined with temporal dependencies of individual demand sequences.

Challenge 2: It is difficult to model the complex spatial dependencies in the case of interaction between two modes. Generally, the demand of one mode is self-correlated in space due to geographical proximity [40], [43]. However, in our case, the influence of another demand in the same space cannot be overlooked. For instance, in Fig. 1 (b), the spatial distribution of demands for both modes would be changed due to the passengers' shift behaviour. More notably, such spatial interactions between the demands of different modes are associated with human mobility and may occur in any region. As a result, the intra- and inter-mode dependencies are also widespread and hybrid in the spatial dimension.

To address the above challenges, we propose a co-prediction model named **T**emporal **S**patial **I**ntertwined **N**etwork (**TSIN** for short hereafter). It contains a *taxi twin* component and *ridesourcing twin* component to perform the prediction of taxi demand and ridesourcing demand, respectively. In each of them, a spatial convolutional layer and a temporal convolutional layer are applied to capture the *intra-mode* dependencies. Furthermore, an intertwined component is designed between twins. The main idea is to build a bridge for each side of mode to learn the *inter-mode* dependencies.

In summary, the main contributions of our work include:

- Instead of improving the demand prediction for single mode using more advanced deep models, this work targets a co-prediction problem of taxi and ridesourcing demands *as a whole* since they are not independent. We conduct extensive experiments based on four real-life datasets from two representative cities in US and compare our method with 11 baselines. Results demonstrate that our proposed **TSIN** outperforms the state-of-the-art methods and its generalizability in different cities.
- In addition to modelling the temporal and spatial dependencies of each mode within each twin component separately, we further propose an intertwined component to enable the spatial and temporal information transfer between different modes to increase the demand prediction accuracy for both taxi and ridesourcing. Moreover, our co-prediction framework is highly flexible and extendable, it is very easy to plugin more twins of other transportation modes (e.g., bike, e-scooter) to achieve a higher accuracy than predicting individually.

The remainder of the paper is organized as follows. In Section II, we summarize the research related to the demand prediction. The problem of demand co-prediction is formulated in Section III. In Section IV, we introduce the details of **TSIN** model. Section V presents the experimental settings and results. We discuss the flexibility and extendability in coupling more transportation modes in Section VI. This paper is concluded in Section VII.

II. Related Work

In this section, we briefly review the related work which can be grouped into two categories, i.e., passenger demand prediction for single mode and passenger demand co-prediction for multi-modes.

A. Passenger Demand Prediction for Single Mode

As a branch of generic traffic prediction, passenger demand prediction has attracted extensive attention in the past decades. Most early studies treat the passenger demand as time series data and use statistic based methods for prediction, such as Historical Average and ARIMA [24]. Meanwhile, many machine learning techniques have also been applied to this field due to their ability to handle complex data, such as the Support Vector Machine (SVM) [6], K-Nearest Neighbor (KNN) [4], Linear Regression (LR) [30], Bayesian network [27] and Gaussian process [10]. Among them, [30] proposes a linear regression model with more than 200 million dimensions of features to predict taxi demands for large-scale online taxicab platforms. In the work of [10], the authors design a censored Gaussian process model to estimate the demand for the bike-sharing system.

In recent years, deep learning models have become the most popular approaches for demand prediction owing to their ability to capture nonlinear and complex features from data [3], [39]. To name a few, [38] proposes a sequence learning model based on Long Short Term Memory (LSTM) network and mixture density networks to predict the taxi demand. Reference [45] proposes a multi-task learning temporal convolutional neural network to predict passenger demands in multiple regions. However, these methods consider only the temporal dynamics and do not explicitly model the spatial dependence of passenger demands. To tackle this issue, some studies such as [15] and [50] incorporate Convolutional Neural Networks (CNN) into LSTM to simultaneously capture the spatial-temporal dependencies of passenger demands. For instance, [23] designs a convolutional recurrent neural networks to predict taxi demand. In the work of [40], the authors integrate spatial, temporal and semantic view together to model spatial-temporal relations for predicting taxi demand.

Despite the excellent capability of CNN in spatial modeling, it cannot capture non-Euclidean spatial dependencies, especially for irregular regions [7]. For this reason, many researches have resorted to Graph Convolutional Networks (GCN) to model the spatial relationship for spatial-temporal prediction tasks [1], [35]. For example, [1] designs a hierarchical graph convolutional structure to capture both spatial and temporal correlations simultaneously for passenger demand prediction. Many of these graph-based methods usually require a predefined adjacency matrix to specify the spatial relationship between nodes. However, that may hinder the model for capturing complex and hidden spatial dependencies. To address this problem, some studies characterize the diverse relationships between regions by constructing multi-graphs, such as geographic proximity, functional similarity, and transportation connectivity [7]. Another solution is to learn an adaptive adjacency matrix to discover the spatial dependency of traffic data [2], [42], [47]. Readers can refer to the survey paper [29] for a more completed view.

B. Passenger Demand Co-Prediction for Multi-Modes

The aforementioned extensive approaches, especially the deep models, greatly improve the accuracy of demand prediction, but they focus on the demand for single mode. In fact, there are multiple transportation modes in a city, and foreseeing the demands for multiple modes at the same time can provide more support for a smart transportation system. Thus, some researchers have paid attention to the simultaneous prediction of multi-mode demands. To name a few, [41] integrates a convolutional auto-encoder and a heterogeneous LSTM to jointly predict the pick-up and drop-off demands for taxis and shared bikes. Reference [14] proposes a novel multi-task multi-graph learning approach to enable the joint prediction of solo and shared ride-hailing demands. In [22], the authors integrate stations of different transportation modes into a heterogeneous graph, and design a self-learning approach to capture the spatial dependencies of both homogeneous and heterogeneous stations. Reference [32] presents a framework to co-predict travel demands for taxi and bike sharing, which contains a shared component and the unique component to extract the shared knowledge and the unique knowledge, respectively. Reference [19] proposes a graph learning based approach to predict demands for multi-modal systems with heterogeneous spatial units. Reference [37] designs a multi-mode traffic prediction framework based on attention mechanism, and uncovers the impact of traffic mode interactions on traffic demand.

TABLE I
COMPARISON OF CO-PREDICTION WORKS. S/T IS THE ABBREVIATION FOR SPATIAL/TEMPORAL

Model	S/T Interaction	Transportation Modes
CoST-Net [41]	✓/ ✗	Taxi & Bike
RTC/MLR-MGC [14]	✗/ ✓	Solo- & Shared-ridesourcing
CoGNN [22]	✗/ ✓	Taxi & Bike
MultiST [32]	✓/ ✗	Taxi & Bike
ST-MRGNN [19]	✗/ ✓	Metro & Ridesourcing
CMGAT [37]	✗/ ✓	Taxi & Bike
TSIN [ours]	✓/ ✓	Taxi & Ridesourcing & ···

To facilitate comparison of these studies, we summarize these works in Table I. It can be seen that these works model the interaction of different demands in only one dimension (i.e., temporal or spatial). To be specific, the temporal interaction between different demands is performed by integrating multiple demand sequences into a heterogeneous LSTM [41], or by designing a shared component to extract shared knowledge from different sequences [32]. On the other hand, the spatial interaction is modeled by constructing a heterogeneous graph with multi-mode transportation stations [22], [37], or by learning cross-mode relations between different graphs [14], [19]. Nevertheless, the demands for different modes may interact with each other in both time and space, thus it is not enough to capture the temporal or spatial interaction alone. It is also worth mentioning that when introducing more transportation modes to the prediction framework, constructing a large heterogeneous graph is complicated although it is intuitive.

In addition, we find that the combination of taxi and bike-sharing is the most common one in co-prediction tasks, since bike-sharing systems are often used for a last/first mile

of transport. However, little effort has been devoted to the demand co-prediction for taxi and ridesourcing, which is the most common MoD services with inherent correlations. In our work, we aim to develop a multi-task framework for demand co-prediction of two modes, and also investigate the extension of other more transportation modes for the co-prediction.

III. Preliminaries

In this section, we first introduce several important notions, followed by the problem statement of co-prediction for taxi and ridesourcing demands.

A. Notions

Definition 1 (Region): A city can be segmented into a number of non-overlapping regions, i.e., $V = \{r_1, r_2, \cdots, r_N\}$.

Definition 2 (Region Demand): The region demand refers to the number of passengers picked up by taxi or ridesourcing car in r_i *at* j*-th time interval. Specifically, we denote these two types of demands as* $x^j_{TA,i}$ *and* $x^j_{RS,i}$*, where* i *and* j *refer to the indices of the region and the time interval, respectively. Further, the demands of* N *regions during the historical* L *time intervals can be organized using a demand matrix. Here, we use two symbols, i.e.,* $\mathbf{X}_{TA} \in \mathbb{R}^{N\times L}$ *and* $\mathbf{X}_{RS} \in \mathbb{R}^{N\times L}$*, to differentiate the demand matrix of taxi and ridesourcing.*

Definition 3 (Region Graph): The region graph is defined as $\mathcal{G} = (V, A)$*, where* V *is the set of nodes (i.e.,* N *regions);* $A \in \mathbb{R}^{N\times N}$ *is the region adjacency matrix. In particular, we will construct multiple adjacency matrices in a parameterized manner to characterize the spatial relationships for different types of demands.*

B. Problem Statement

Given the region graph $\mathcal{G}$ initialized by adaptive adjacency matrices, and the historical demands of taxi and ridesourcing (i.e., $\mathbf{X}_{TA}$ and $\mathbf{X}_{RS}$), our goal is to simultaneously predict the future demands $\mathbf{Y}_{TA}$ and $\mathbf{Y}_{RS} \in \mathbb{R}^N$ of all regions at the forthcoming time interval. Formally, the co-prediction problem can be formulated as:

$$(\hat{\mathbf{Y}}_{TA}, \hat{\mathbf{Y}}_{RS}) = \mathcal{F}_\Theta\left(\mathbf{X}_{TA}, \mathbf{X}_{RS}; \mathcal{G}\right), \tag{1}$$

where $\mathcal{F}_\Theta(\cdot)$ is a function implemented by the neural network model. Note that the graph $\mathcal{G}$ does not contain any predefined adjacency matrix, instead, our model automatically learns the adjacency matrix to guide the spatial convolution. Details on how to derive the adaptive adjacency matrix would be given in Section IV-B.2. In summary, the objective is to determine the optimal function parameters Θ^* by minimizing the error between the estimated and true values:

$$\Theta^* = \arg\min_\Theta \left(\mathcal{L}\left(\mathbf{Y}_{TA}, \hat{\mathbf{Y}}_{TA}\right) + \mathcal{L}\left(\mathbf{Y}_{RS}, \hat{\mathbf{Y}}_{RS}\right)\right), \tag{2}$$

where $\mathcal{L}$ represents the loss function.

IV. The **TSIN** Model

In this section, we introduce our proposed coupling model named **TSIN** in detail, and describe how it works for the co-prediction task of taxi and ridesourcing demands.

A. Overview of **TSIN**

The architecture of **TSIN** is shown in Fig. 2, which consists of *two twin components, and an in-between temporal and spatial intertwined component.* Each twin component (i.e., the taxi twin or the ridesourcing twin) in the couple contains a number of stacked spatial-temporal blocks (i.e., ST Block) and an output layer. One ST Block is constructed by a temporal convolutional layer and a spatial convolutional layer, which are used to extract temporal and spatial features from the historical demands of taxi or ridesourcing, respectively. By stacking a number of ST Blocks, each twin can capture demand patterns at different temporal and spatial scales *separately*. In addition, the in-between temporal and spatial intertwined component is designed to *bridge* the twins and allow them to exchange information while performing feature extraction, that is, each twin could *collect the relevant information and share what is necessary.* The output layer of each twin collects the extracted features in multiple ST Blocks by skip connections, and fuse them for the specific prediction task.

B. The Taxi/Ridesourcing Couple

Here, we just focus on the taxi twin in the taxi/ridesourcing couple as an example to introduce how to extract spatial-temporal features from the historical taxi demands, since the ridesourcing twin has the exactly same structure and function. The taxi twin is consisted of two important kinds of layers, i.e., Temporal Convolutional Layer (TCL) and Spatial Convolutional Layer (SCL), which are mainly responsible of modelling temporal dependencies and spatial dependencies respectively, with the details as follows.

1) Temporal Convolutional Layer: Generally, the historical demands of each region can be viewed as a time sequence. Owing to the evolution of demand over time, its future value is inevitably affected by the fluctuation during the recent time period. Thus, an important aspect of the demand prediction is to accurately capture temporal dependencies from previous observations. To this end, we design a TCL in each ST Block to capture *intra-mode* temporal dependencies by just focusing on single transportation mode.

In more detail, the TCL is constructed based on the dilated causal convolution [17], which is capable of extracting information from the long sequence. Compared to the standard 1D convolution, the dilate causal convolution makes two major improvements: (1) It allows the convolution operation to skip values at a certain distance in the sequence by introducing a dilation factor, thus enabling a larger receptive field; and (2) It only uses the information at its preceding positions by constraining the convolution operation at the j-th position in the sequence, thus preserving the causality of the time sequence.

For a region r_i, given its taxi demand sequence $\mathbf{X}_{TA}(i,:) \in \mathbb{R}^L$, and a kernel $\mathbf{f}_{TA} = \left[w_0, w_1, \ldots, w_{K-1}\right]$, the dilate causal convolution applied on the j-th ($1 \le j \le L$) position of $\mathbf{X}_{TA}(i,:)$ can be expressed as:

$$\mathbf{X}_{TA}(i,j) \star \mathbf{f}_{TA} = \sum_{s=0}^{K-1} w_s \cdot \mathbf{X}_{TA}(i, j - d \times s), \tag{3}$$

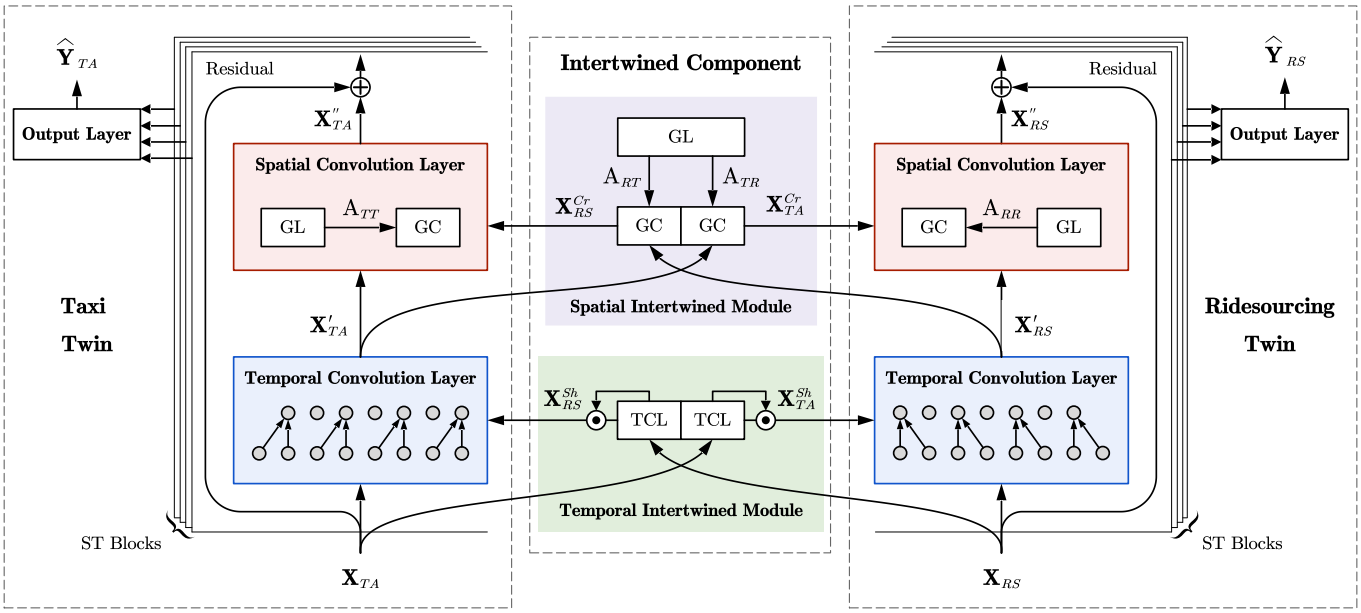


Fig. 2. The architecture of **TSIN**. Some important notions are also marked in the figure for the reference.

where K is the kernel size; d is the dilation factor that controls the skipping distance.

To capture diverse patterns from the demand sequence, we perform the convolution operation at L positions of $\mathbf{X}_{TA}(i,:)$ using F_t different kernels, which constitutes our TCL:

$$\mathbf{X}'_{TA}(i,:) = \mathrm{TCL}_{TA}\left(\mathbf{X}_{TA}(i,:), \mathbf{f}^{c}_{TA}\,|^{F_t}_{c=1}\right) \in \mathbb{R}^{L' \times F_t}, \quad (4)$$

where $\mathbf{X}'_{TA}(i,:)$ is a new feature sequence obtained after the temporal convolution, with a new length of $L' = L - d \times (K-1)$; F_t is the number of channels (also known as the feature dimension in TCL).

Furthermore, the TCL can be conducted simultaneously over the demand sequence of N regions:

$$\mathbf{X}'_{TA} = \mathrm{TCL}_{TA}\left(\mathbf{X}_{TA}\right) \in \mathbb{R}^{N \times L' \times F_t}, \quad (5)$$

where $\mathbf{X}'_{TA}$ is the new demand matrix of the taxi, as well as the final output of the TCL.

Similarly, we also design a TCL with the exactly same structure, to extract temporal features from the ridesourcing demand matrix $\mathbf{X}_{RS}$:

$$\mathbf{X}'_{RS} = \mathrm{TCL}_{RS}\left(\mathbf{X}_{RS}\right) \in \mathbb{R}^{N \times L' \times F_t}. \quad (6)$$

2) Spatial Convolutional Layer: In the real world, passenger demands of different regions are often correlated to each other. For example, two regions in close proximity or with identical functionality may show similar changes in demand during the same time period [40]. How to capture these complicated spatial dependencies effectively is another issue to be tackled in the demand prediction. Here, we first design a graph learning sub-layer to discover the dependencies among different regions automatically, then derive an adaptive adjacency matrix. Afterwards, we propose a graph convolution sub-layer to aggregate demand features from relevant regions for each region, where the "relevant regions" are specified by the learned adaptive adjacency matrix. In short, there are two major tasks, i.e., graph learning and graph convolution in SCL, detailed as follows.

Graph Learning (GL): We initialize two region embeddings $\mathbf{E}_{Tq}, \mathbf{E}_{Tk} \in \mathbb{R}^{N \times d_e}$ for taxi demands, where d_e is the dimension of embeddings. The adaptive adjacency matrix of taxi demands among N regions is computed as:

$$A_{TT} = \mathrm{Softmax}(\mathrm{ReLU}(\mathbf{E}_{Tq} \cdot \mathbf{E}^{\top}_{Tk})) \in \mathbb{R}^{N \times N}, \quad (7)$$

where $\mathbf{E}_{Tq}, \mathbf{E}_{Tk}$ are composed of learnable parameters. The operation $\mathbf{E}_{Tq} \cdot \mathbf{E}^{\top}_{Tk}$ calculates the similarities between the embedding vectors among different regions, which can be regarded as the weights of spatial dependency. The activation function ReLU is used to remove small weights, and the Softmax function is applied to normalize the matrix.

It is crucial to emphasize that A_{TT} can be updated automatically during the training phase according to the error feedback of the prediction task. For example, if the feature aggregation of taxi demands between region r_1 and r_2 makes their predictions worse, the model could weaken their dependence by reduce the value of $A_{TT}(1,2)$.

Graph Convolution (GC): After obtaining the adaptive adjacency matrix A_{TT}, we employ the graph convolution to model the *intra-mode* spatial dependencies among N regions, similarly, by just focusing on single transportation mode. Specifically, the graph convolutional layer takes A_{TT} and $\mathbf{X}'_{TA}$ as inputs, to generate a new demand feature for each region by performing aggregation and transformation operations. $\mathbf{X}''_{TA}$ is returned by SCL. Formally, we have:

$$\mathbf{X}''_{TA} = \sigma\left(A_{TT}\mathbf{X}'_{TA}W_{TT}\right) \in \mathbb{R}^{N \times L' \times F_s}, \quad (8)$$

where $W_{TT} \in \mathbb{R}^{F_t \times F_s}$ is a projection for feature transformation; $\sigma(\cdot)$ represents the GLU activation function, and F_s is the output dimension of the graph convolutional layer.

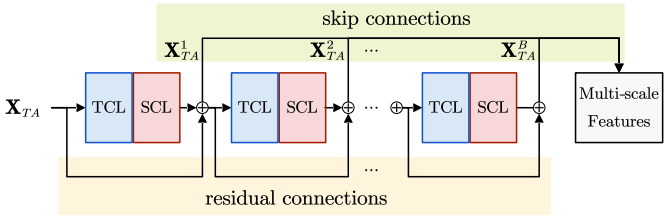


Fig. 3. Multiple stacked ST Blocks with residual connections and skip connections.

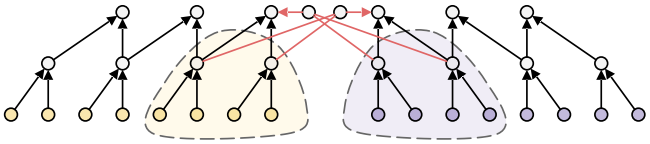


Fig. 4. Diagram of the temporal intertwined module. For simplicity, the self-gating mechanism is omitted from the figure.

Likewise, with the exactly same network structure to the taxi demands, we also first construct two region embeddings $\mathbf{E}_{Rq}, \mathbf{E}_{Rk} \in \mathbb{R}^{N \times d_e}$ for the ridesourcing demands, and represent the adaptive adjacency matrix by:

$$A_{RR} = \text{Softmax}(\text{ReLU}(\mathbf{E}_{Rq} \cdot \mathbf{E}_{Rk}^{\top})) \in \mathbb{R}^{N \times N}. \tag{9}$$

Similarly again, the corresponding graph convolutional layer is defined as:

$$\mathbf{X}''_{RS} = \sigma\left(A_{RR}\mathbf{X}'_{RS}W_{RR}\right) \in \mathbb{R}^{N \times L' \times F_s}, \tag{10}$$

where $W_{RR} \in \mathbb{R}^{F_t \times F_s}$ is a matrix that is used for the linear transformation.

Overall, the TCL together with the SCL form a ST Block to capture spatial-temporal dependencies simultaneously. To avoid the problem of gradient vanishing when stacking blocks [34], a residual connection is added to each block. Then, the operations of b-th block in the taxi and ridesourcing twins can be concisely represented by:

$$\begin{aligned}\mathbf{X}_{TA}^{b+1} &= \text{SCL}_{TA}^{b}\left(\text{TCL}_{TA}^{b}\left(\mathbf{X}_{TA}^{b}\right)\right) + \mathbf{X}_{TA}^{b},\\ \mathbf{X}_{RS}^{b+1} &= \text{SCL}_{RS}^{b}\left(\text{TCL}_{RS}^{b}\left(\mathbf{X}_{RS}^{b}\right)\right) + \mathbf{X}_{RS}^{b},\end{aligned} \tag{11}$$

where the second term of each equation represents a residual connection.

Figure 3 illustrates the stacking of multiple blocks with residual connections between the input and output of each block. In addition, the skip connections are designed to readout the learned features by each block. In this way, the model can collect multi-scale spatial-temporal features as the blocks are stacked. To be more specific, the multiple temporal convolutions in different blocks enable the model to capture *local and global* variations in the demand sequence, and the execution of multiple spatial convolutions allows the model to aggregate the spatial information of multi-hop neighbours *hierarchically*.

C. Temporal and Spatial Intertwined Component

1) Temporal Intertwined Module: Although the temporal convolutional layers TCL_{TA} and TCL_{RS} can capture the temporal dependencies from the two demand sequences separately, the *inter-mode* temporal dependencies between the taxi and ridesourcing demands are overlooked. To solve the issue, we propose a temporal intertwined module to bridge the two separated TCLs, so that they can exchange valuable information during the temporal dependency modelling.

First, we set up two additional convolutional layers TCL_1 and TCL_2 to extract temporal features as information to be shared on the taxi and ridesourcing demand sequences, respectively.

$$\begin{aligned}\mathbf{X}_{TA}^{Sh} &= \text{TCL}_1\left(\mathbf{X}_{TA}\right),\\ \mathbf{X}_{RS}^{Sh} &= \text{TCL}_2\left(\mathbf{X}_{RS}\right).\end{aligned} \tag{12}$$

Next, to avoid passing irrelevant information between different sequences, a self-gating mechanism is designed to filter the shared information:

$$\begin{aligned}\mathbf{X}_{TA}^{Sh} &= \sigma\left(\mathbf{X}_{TA}^{Sh}\right) \odot \mathbf{X}_{TA}^{Sh},\\ \mathbf{X}_{RS}^{Sh} &= \sigma\left(\mathbf{X}_{RS}^{Sh}\right) \odot \mathbf{X}_{RS}^{Sh},\end{aligned} \tag{13}$$

where σ represent the Sigmoid function.

Finally, we fuse the inter-mode feature (e.g., $\mathbf{X}_{RS}^{Sh}$) with the intra-mode feature (e.g., $\mathbf{X}'_{TA}$) at the last channel (i.e., L') of the time dimension:

$$\begin{aligned}\mathbf{X}'_{TA}(:, L', :) &= \mathbf{X}'_{TA}(:, L', :) + \mathbf{X}_{RS}^{Sh}(:, L', :),\\ \mathbf{X}'_{RS}(:, L', :) &= \mathbf{X}'_{RS}(:, L', :) + \mathbf{X}_{TA}^{Sh}(:, L', :).\end{aligned} \tag{14}$$

Figure 4 gives a simplified illustration of the temporal intertwined module, in which the red lines indicate the interaction of the temporal features for the two types of demands, i.e., the exchange of information through TCL_1 and TCL_2. By modeling the *intra- and inter-mode* dependencies, the final temporal features $\mathbf{X}'_{TA}$ and $\mathbf{X}'_{RS}$ are capable of incorporating both the demand patterns of taxi and ridesourcing.

2) Spatial Intertwined Module: As mentioned before, the spatial dependencies in this work are also two-fold, i.e., the intra- and inter-mode dependencies. For example, the taxi demand in a region is not only related to the taxi demands of other regions, but may also be influenced by the demands for ridesourcing. However, the SCL in the ST Block is designed only for modelling intra-mode spatial dependencies of one type of demand, failing to capture the spatial interactions between two modes of demands. Thus, we design an additionally spatial intertwined module to couple the two SCLs in the twin components, detailed as follows.

First, we reuse the region embeddings obtained in spatial dependency modelling to further construct the adjacency matrices between taxi and ridesourcing demands:

$$\begin{aligned}A_{RT} &= \text{Softmax}(\text{ReLU}(\mathbf{E}_{Rq}\mathbf{E}_{Tk}^{\top})) \in \mathbb{R}^{N \times N},\\ A_{TR} &= \text{Softmax}(\text{ReLU}(\mathbf{E}_{Tq}\mathbf{E}_{Rk}^{\top})) \in \mathbb{R}^{N \times N},\end{aligned} \tag{15}$$

where a row in A_{RT} indicates the relation between the taxi demand of one region and the ridesourcing demand of other regions, while A_{TR} represents the opposite relation. It is worth noting that all adaptive adjacency matrices are derived from low-dimensional region embeddings, which requires only

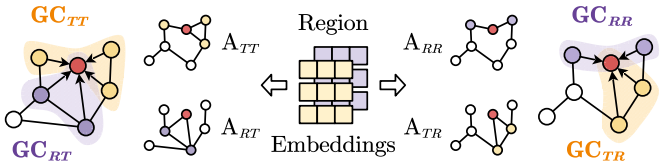


Fig. 5. Diagram of the spatial intertwined module.

a small number of parameters and makes the intra- and inter-mode adjacency matrices associate with each other at the very underlying level.

Then, we build two graph convolutional sub-layer GC$_{RT}$ and GC$_{TR}$ to extract cross-mode spatial features with the support of A_{RT} and A_{TR}, respectively:

$$\begin{aligned}\mathbf{X}_{RS}^{Cr} &= \sigma\left(A_{RT}\mathbf{X}'_{RS}W_{RT}\right),\\ \mathbf{X}_{TA}^{Cr} &= \sigma\left(A_{TR}\mathbf{X}'_{TA}W_{TR}\right),\end{aligned} \tag{16}$$

where $W_{RT}, W_{TR} \in \mathbb{R}^{F_t \times F_s}$ are two feature transformation matrices.

Finally, we incorporate the inter-mode feature (e.g., $\mathbf{X}_{RS}^{Cr}$) into the intra-mode feature (e.g., $\mathbf{X}''_{TA}$). Formally, we have:

$$\begin{aligned}\mathbf{X}''_{TA} &= \mathbf{X}''_{TA} + \mathbf{X}_{RS}^{Cr},\\ \mathbf{X}''_{RS} &= \mathbf{X}''_{RS} + \mathbf{X}_{TA}^{Cr}.\end{aligned} \tag{17}$$

The above process is demonstrated in Fig. 5. In a nutshell, the spatial intertwined module achieves the cross-mode fusion of spatial features through two graph convolution operations based on matrices A_{RT} and A_{TR}, respectively. Essentially, the model equips each SCL with two receptive fields, thus enabling one region to aggregate both taxi and ridesourcing demand features from the other regions.

D. Output Layer

As mentioned before, the extracted spatial-temporal features in stacked blocks are at different temporal and spatial scales. To take advantage of these rich features, we collect the output of all blocks through skip connections, then concatenate them together and then feed into a two-layer fully connected network to make the final predictions. The output layers of the taxi/ridesourcing couple can be expressed as:

$$\begin{aligned}\hat{\mathbf{Y}}_{TA} &= \sigma\left(\text{Concat}\left[\mathbf{X}_{TA}^{b}\,|_{b=1}^{B}\right]W_1\right)W_2,\\ \hat{\mathbf{Y}}_{RS} &= \sigma\left(\text{Concat}\left[\mathbf{X}_{RS}^{b}\,|_{b=1}^{B}\right]W_3\right)W_4,\end{aligned} \tag{18}$$

where B is the number of ST Blocks; $W_1, W_3 \in \mathbb{R}^{BF_s \times F_h}$ and $W_2, W_4 \in \mathbb{R}^{F_h \times 1}$ are learnable weights; and $\sigma(\cdot)$ is the ReLU activation function.

E. Learning Optimization

Here, we use the Mean Squared Error (MSE) as the loss function to evaluate the error of taxi and ridesourcing demand predictions as follows:

$$\begin{aligned}\mathcal{L}_1 &= \text{MSE}\left(\mathbf{Y}_{TA}, \hat{\mathbf{Y}}_{TA}\right),\\ \mathcal{L}_2 &= \text{MSE}\left(\mathbf{Y}_{RS}, \hat{\mathbf{Y}}_{RS}\right).\end{aligned} \tag{19}$$

In addition, we employ the uncertainty weighting mechanism [16] to balance $\mathcal{L}_1$ and $\mathcal{L}_2$, thus the final loss function is defined as:

$$\mathcal{L}(\sigma_1, \sigma_2) = \frac{1}{2\sigma_1^2}\mathcal{L}_1 + \frac{1}{2\sigma_2^2}\mathcal{L}_2 + \log\sigma_1\sigma_2, \tag{20}$$

where σ_1 and σ_2 are two noise parameters, which are used to balance the task-specific losses during training and can be updated through back-propagation.

Algorithm 1 The Learning Process of **TSIN**

Input: The historical demands $\mathbf{X}_{TA}, \mathbf{X}_{RS}$;
Output: The predicted demands $\hat{\mathbf{Y}}_{TA}, \hat{\mathbf{Y}}_{RS}$

1: **Init**: $\mathbf{E}_{Tq}, \mathbf{E}_{Tk}, \mathbf{E}_{Rq}, \mathbf{E}_{Rk}$
2: $\mathbf{X}_{TA}^{1:B} = \varnothing$, $\mathbf{X}_{RS}^{1:B} = \varnothing$
3: $\mathbf{X}_{TA}^{1} = \mathbf{X}_{TA}$, $\mathbf{X}_{RS}^{1} = \mathbf{X}_{RS}$
4: **for** $b \in [1, B]$ **do**
5: $\quad\mathbf{X}_{TA}^{b,\prime}, \mathbf{X}_{RS}^{b,\prime} \leftarrow$ perform TCL by Eqns. 5 and 6
6: $\quad$**Update** $\mathbf{X}_{TA}^{b,\prime}, \mathbf{X}_{RS}^{b,\prime}$ by Eq. 14
7: $\quad A_{TT}, A_{RR} \leftarrow$ graph learning by Eqns. 7 and 9
8: $\quad\mathbf{X}_{TA}^{b,\prime\prime}, \mathbf{X}_{RS}^{b,\prime\prime} \leftarrow$ perform SCL by Eqns. 8 and 10
9: $\quad A_{RT}, A_{TR} \leftarrow$ graph learning by Eq. 15
10: $\quad$**Update** $\mathbf{X}_{TA}^{b,\prime\prime}, \mathbf{X}_{RS}^{b,\prime\prime}$ by Eq. 17
11: $\quad\mathbf{X}_{TA}^{b+1} \leftarrow \mathbf{X}_{TA}^{b,\prime\prime} + \mathbf{X}_{TA}^{b}$, $\mathbf{X}_{RS}^{b+1} \leftarrow \mathbf{X}_{RS}^{b,\prime\prime} + \mathbf{X}_{RS}^{b}$ *// residual connection*
12: $\quad\mathbf{X}_{TA}^{1:B} \cup \{\mathbf{X}_{TA}^{b+1}\}$, $\mathbf{X}_{RS}^{1:B} \cup \{\mathbf{X}_{RS}^{b+1}\}$ *// skip connection*
13: **end for**
14: $\hat{\mathbf{Y}}_{TA}, \hat{\mathbf{Y}}_{RS} \leftarrow$ perform output layers by Eq. 18
15: $\mathcal{L} \leftarrow$ calculate losses by Eqns. 19 and 20
16: Back-propagate $\mathcal{L}$ and optimize the model

Algorithm 1 overviews the pseudo-code for the learning process of **TSIN**. First, the model takes the demand matrices $\mathbf{X}_{TA}$ and $\mathbf{X}_{RS}$ as the input, and randomly initializes the region embeddings for graph learning (Lines 1∼3). Then, the inputs pass through B stacked blocks in twins and the intertwined component to model intra- and inter-mode dependencies systematically. As shown in Line 4∼13, we can extract the spatial-temporal features of different scale at each block. Next, the future demands for taxi and ridesourcing can be predicted using the multi-scale features by Eq. 18. Finally, the loss function is calculated and optimized in Lines 15∼16.

V. Experiments

In this section, we firstly introduce the details of the experiments, including real-world datasets, the state-of-the-art baselines, and the hyper-parameter settings. Then, we present the experiment results, including the performance at different time and on different regions, the interpretability of our model, and the ablation study. Finally, we present the comparison results in terms of prediction accuracy and computational efficiency with the 11 baselines to highlight the superiority of the model. The source code for **TSIN** is available at https://github.com/csjiezhao/TSIN.

TABLE II
DATASET DESCRIPTION

Dataset	Regions	Orders	Time Span
NYC-Taxi	63	88,894,262	Jan.1 $\sim$ Dec.31, 2018
NYC-FHV	63	74,382,304	Jan.1 $\sim$ Dec.31, 2018
CHI-Taxi	77	14,629,392	Jan.1 $\sim$ Dec.31, 2019
CHI-TNP	77	96,875,357	Jan.1 $\sim$ Dec.31, 2019

A. Datasets

We conduct experiments on four real-world trip datasets generated by taxi and ridesourcing cars in two representative cities in US, i.e., New York (NYC)[1] and Chicago (CHI)[2]:

- *NYC-Taxi*: It contains trip records of green and yellow taxi in Manhattan, NYC. Each record mainly includes pickup date time, pickup region (i.e., the taxi zone in NYC), dropoff date time, and dropoff region.
- *NYC-FHV*: It consists of trip records reported by for-hire vehicles (FHV) such as Uber and Lyft, including trip-specific information such as pickup and dropoff time, origin and destination region. It provides the information about ridesoucing demand in NYC.
- *CHI-Taxi*: It collected trip records of Chicago Taxi, each record includes the trip start/end timestamp and the pickup/dropoff region (i.e., the community area).
- *CHI-TNP*: It contains trip records reported by Transportation Network Providers (TNP) in Chicago. It provides the information about ridesoucing demand in CHI.

More detailed information for the datasets is summarized in Table II. For each dataset, we set the time interval as 30 minutes, and count the number of pickups in all regions at each time interval to construct the demand matrix. Training set, validation set and test set are partitioned according to the ratio of 7:1:2. Then we use Z-Score normalization to scale the input features. Historical demands of the most recent 12 time intervals are used to predict the demands at the next one.

In addition, we use one more dataset (i.e., *NYC-POI*) to validate the interpretability of our model. The dataset contains 5,343 points of interests in 63 regions of Manhattan. Each POI is with a category tag clearly showing its functionality, such as "education facility" and "transportation facility". We count the number of POIs in each category for each region, and then obtain the POI distributions of all regions.

B. Baseline Methods and Evaluation Metrics

- **HA**: Historical average is a very basic statistical method for prediction. It predicts the target demand by simply averaging all the historical values.
- **FC-LSTM** [28]: LSTM with a Fully Connected (FC) layer is widely used for time sequence modelling. The size of the hidden state is set to 256.
- **DCRNN** [18]: It proposes a diffusion graph convolution module to model the traffic flow as a diffusion process, and integrates the module into a seq2seq framework to predict traffic series data.
- **T-GCN** [49]: Temporal GCN combines general graph convolution and GRU to extract spatio-temporal correlations in traffic data for prediction.
- **ASTGCN** [11]: An attention-based network, which designs spatial and temporal attention mechanisms to capture dynamic patterns in traffic data.
- **STSGCN** [26]: This model designs a spatial-temporal synchronous modeling mechanism to capture the localized correlations for traffic prediction.
- **AGCRN** [2]: It proposes an adaptive graph convolutional recurrent network to capture spatial-temporal correlation from traffic data in an automatic manner.
- **MTGNN** [34]: This is a framework for modeling multivariate time series data and learning graph structures.
- **CCRNN** [42]: This is an adaptive graph based model with a layer-wise coupling mechanism.
- **GMSDR** [21]: This model is a novel recurrent neural network for capturing multi-step dependency relation, and also considers spatial information to support general spatial temporal prediction.
- **CoGNN** [22]: It proposes a framework for co-prediction of station-based multi-modal transportation demands.

For a fair comparison, the input to all method on two datasets are the same, that is, the historical demands of past $L = 12$ time intervals. For the hyper-parameters, we choose the default values according to their origin proposals.

To evaluate the performance of all methods, we employ three metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE), which are defined as follows:

$$\begin{aligned}
\mathrm{MAE}(y,\hat{y}) &= \frac{1}{n}\sum_{i=1}^{n}\left|y_i-\hat{y}_i\right| \\
\mathrm{MAPE}(y,\hat{y}) &= \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right|, \\
\mathrm{RMSE}(y,\hat{y}) &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2},
\end{aligned} \tag{21}$$

where y and $\hat{y}$ the true and predicted values, respectively.

Notice that MAE is a widely used measure for absolute error; MAPE considers the ratio of the absolute error with respect to the ground-truth, but it receives more punishments for smaller true values; RMSE is more sensitive to outliers. Therefore, the combination of three metrics evaluates the performance of inference methods more comprehensively.

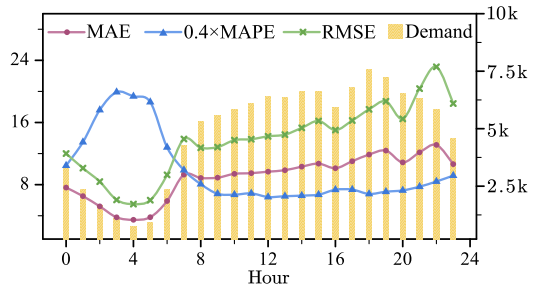
C. Experimental Settings

We implement our **TSIN** model using Pytorch on a workstation running Ubuntu OS (GPU: GeForce RTX 2080 Ti). The maximum training epoch is set as 500. The learning rate is set as 0.001 and the batch size is 64. We employ the Adam optimizer to minimize the loss function. The hyper-parameters of our model are determined by comparing the performance on the validation set, which are given in Table III.

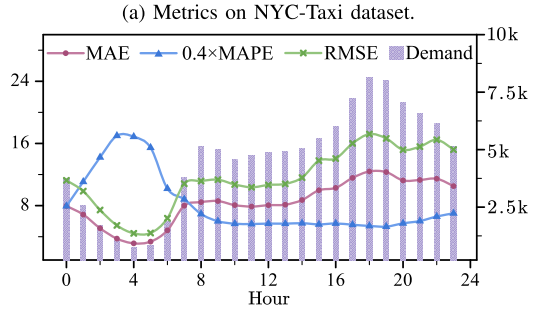
[1]https://opendata.cityofnewyork.us/
[2]https://data.cityofchicago.org/

TABLE III
HYPER-PARAMETER SETTING

Parameter	Meaning	Value
B	Number of blocks	4
D	dilation factor in blocks	[1, 2, 3, 4]
K	Kernel size in TCL of blocks	[3, 2, 2, 2]
d_e	Size of node embedding	10
F_t	Number of channels in TCL	32
F_s	Number of channels in SCL	32
F_h	Number of hidden units in the output layer	256

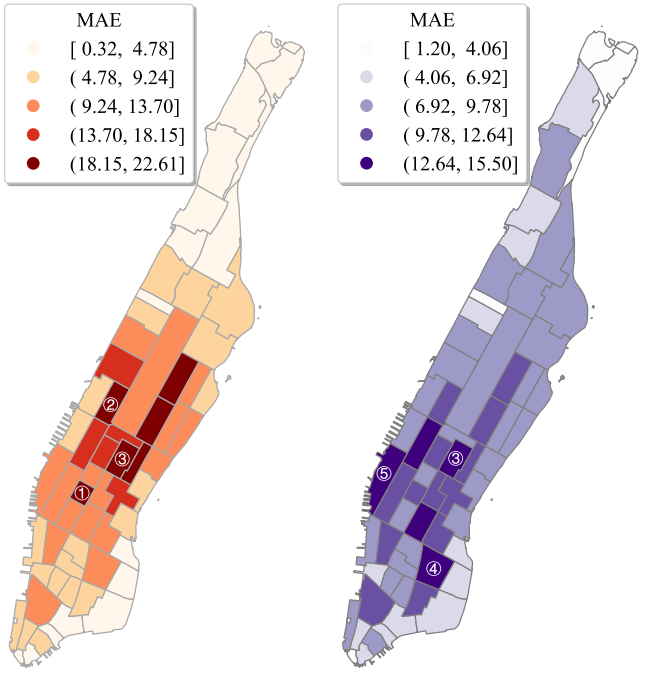


(a) Metrics on NYC-Taxi dataset.



(b) Metrics on NYC-FHV dataset.

Fig. 6. The average error values and actual demand values at different hours over a day.



(a) Taxi demand. (b) Ridesourcing demand.

Fig. 7. MAE over different regions for demand predictions. Region ①: Penn. Station; ②: Lincoln Square East; ③: Midtown Center; ④: East Village; ⑤: West Chelsea.

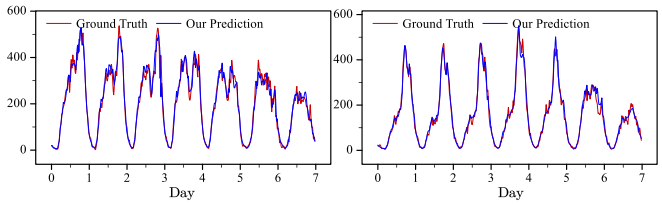
D. Results and Analysis

1) Performance at Different Time: In this part, we investigate the performance of our model from the temporal dimension. Since the human mobility behaviour varies within a day, the performance at different hours shows the impact of human activity on the demand prediction. Figure 6 reports the results on taxi dataset and ridesourcing dataset in NYC, respectively. In each of the figures, we plot the average error values with curves and the actual demands with bars at different hours over a day. Note that, to keep the different metrics studied at the same scale, we multiply RMSE value by 0.4. We can observe that:

- The curves of MAE (purple) and RMSE (green) are affected by the changes of the actual demand, i.e., the larger the actual demand value, the larger the absolute error. Larger demands reveal that more complicated human activity and mobility are taken place at that moment. It increases the difficulty in making accurate predictions, thus the absolute errors and the root mean squared of absolute errors raise with the demands increasing.
- MAPE changes, however, in the opposite way with the other two. This is because that the percentage absolute error is sensitive to the small actual demand. For example, the actual demands are the lowest at 4 am when the human traveling is the most inactive. The slight difference between the predicted demand and the actual demand would be magnified as divided by a small demand value. That explains why we observe the peak of MAPE and valleys of MAE and RMSE at the same hours.
- The performance on both datasets are similar. When the actual demand is low during 0 am to 6 am, the absolute error MAE is the lowest and MAPE is the highest. In the rest of the day, MAE goes higher and MAPE goes lower. Overall, RMSE and MAE have similar trend except that RMSE is magnified comparing with MAE.

2) Performance on Different Regions: Similarly, we investigate the model's performance on different regions of Manhattan, and Fig. 7 (a) and (b) display the distribution of MAE over regions for the taxi dataset and ridesourcing dataset, respectively. Here only MAE values are used to indicate the error levels, since RMSE has very similar distribution and MAPE is sensitive to the small actual demand values.

In the maps, the regions with darker color are the regions with higher prediction errors. It clearly shows that the spatial distribution of the errors is non-uniform, and the regions with larger errors are clustered mainly in Midtown Manhattan and Lower Manhattan. The top three regions with the largest prediction errors in each map are marked, i.e., region ①∼③ in Fig. 7 (a), and region ③∼⑤ in Fig. 7 (b), respectively. We notice that these regions are the busy commercial or residential districts, and often with lots of travel demands. The reason for the larger error on these regions could be twofold. First, predicting large demand values is prone to have large absolute errors. Second, plenty of crowd movements in popular



(a) Taxi demand prediction. (b) Ridesourcing demand prediction.

Fig. 8. Prediction results on Region ③.

regions contain diverse and complicated travel patterns, which brings difficulties in making accurate demand prediction.

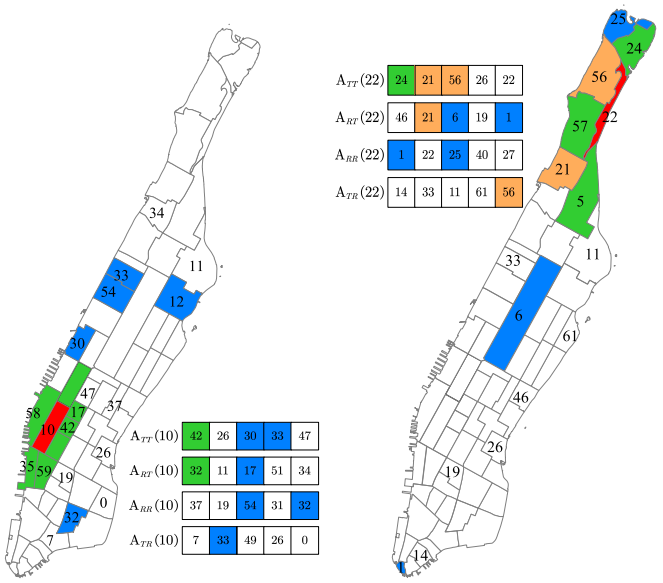
Nevertheless, in the real world, it is important for traffic management to provide reliable demand prediction in popular regions. Thus, we focus on region ③, the Midtown Center area, which has relatively high error level and is marked in both maps, and we look deeper into the predicted and the actual demand values. The results of comparison are shown in Fig. 8. The red curves indicate the ground truth values of the demand, and the blue curves are the prediction from our model. It shows that our model can capture the demand pattern over time, and for most of the time our model is very accurate in tracing the ground truth curves even at peaks and valleys where value changes at a sudden. The result expresses that our model is still effective in the popular regions with heavy traffic and can provide accurate predictions.

3) Interpretability: Furthermore, we study the interpretability of our model. In **TSIN** model, one important step is to construct multiple adaptive adjacency matrices to guide the modelling of intra- and inter- mode spatial dependencies. Essentially, the model discovers and constructs the correlations among regions *in a data-driven way*. Here, we go deeper into these learned matrices, and compare them with some prior knowledge (i.e., geographical proximity and POI distribution). By analyzing the results, we hope to shed some light on opening the black-box of our deep model.

We firstly define two types of neighbours of a center region by using the geographical proximity and POI distribution information respectively: (1) Geographical neighbours: all the regions that geographically connected to the center region; and (2) Functional neighbours: the top-5 regions that have the most similar POI distribution with the center region (the *cosine similarity* is adopted to calculate the similarity). Then we take regions r_{10} and r_{22} in Manhattan as examples and compare their geographical neighbours, functional neighbours, and top-5 learned neighbours that come from the adaptive adjacency matrices with the top-5 highest weights.

The visualization of the result is shown in Fig. 9. In these maps, we mark the center regions in red, their geographical neighbours in green, their functional neighbours in blue, and the neighbours that are both geographical and functional in orange. Their learned neighbours from different adjacency matrices are also displayed by the maps. We can draw the following conclusions:

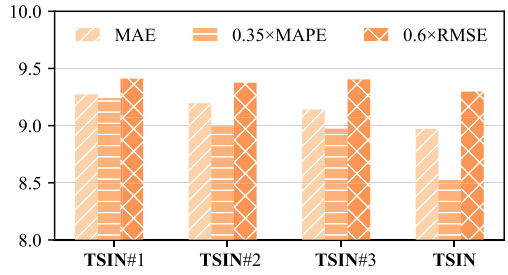
- Taking r_{10} as an example, when predicting its taxi demands, it will simultaneously learn the taxi-impact-on-taxi-demand neighbours ($A_{TT}(10)$) and the ridesourcing-impact-on-taxi-demand neighbours ($A_{RT}(10)$), respectively. It shows that both sets of learned neighbours overlap with r_{10}'s geographical neighbours and its functional neighbours, even though this prior knowledge is not input to our model. This demonstrates that our method can discover knowledge in a data-driven way as expected.
- Similarly, when predicting r_{10}'s ridesourcing demands, our model will learn its ride-impact-on-ride-demand neighbors $A_{RR}(10)$ and taxi-impact-on-ride-demand neighbors $A_{TR}(10)$. The difference between $A_{RR}(10)$ and $A_{TR}(10)$ shows the overall diversity of spatial dependencies. Moreover, it also implies the distinct difference between intra-mode and inter-mode.
- As for r_{22}, we can have observations similar to the above two. In addition, combining the results of r_{10} and r_{22}, we have the following findings. First, the geographic proximity is indeed the efficient prior knowledge. Especially for r_{22}, the model aggregates information from most of its geographically neighbours. Thus, the geographic proximity is a choice worth to consider when modeling the spatial dependencies. Second, we can observe that, one region's neighbours, which have spatial dependencies with it, are widely distributed in the map. They could be the regions that actually connect to the center region, but also the regions far away with similar POI distribution.



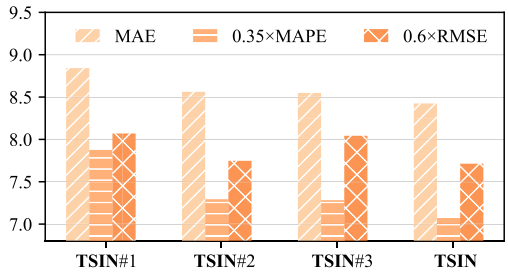
(a) Region r_{10}. (b) Region r_{22}.

Fig. 9. Learned neighbours of region r_{10} and r_{22}.

Therefore, it is not sufficient to model the spatial dependencies by using adjacency matrices based on the geographical proximity only. This type of adjacency matrix can only describe the relations between local regions, and can hardly capture the extensive spatial dependencies as aforementioned. This is also the motivation of most works on capturing spatial dependencies by constructing in multi-graph. Worse still, quite a big number of learned neighbours are neither geographical nor functional ones of the center regions. Such result implies that the spatial dependency of urban transportation systems is quite *complex* and *heterogeneous* and cannot be predefined.



(a) NYC-Taxi dataset.



(b) NYC-FHV dataset.

Fig. 10. Ablation study on NYC-Taxi and NYC-FHV datasets.

4) Ablation Study: To verify the effectiveness of our model, we conduct the ablation study on NYC-Taxi and NYC-FHV datasets. We compare our model with the following variations:

- **TSIN** #1: It removes both spatial and temporal intertwined modules;
- **TSIN** #2: It removes the spatial intertwined module;
- **TSIN** #3: It removes the temporal intertwined module.

Despite the removed modules, all variations have the same framework and parameter settings. We repeat each experiment 3 times and take the average as the results. The corresponding performance comparison are shown in Fig. 10. The height of the bars indicates the average errors of different variations with different metrics. We can observe that:

- Removing spatial or temporal intertwined modules, or both, from original **TSIN**, the prediction error will increase in varying degrees, which fully validates the effectiveness of modeling the inter-mode dependencies. The spatial/temporal intertwined modules in our model are able to improve the prediction accuracy for both mode.
- Mostly, the improvements of adding only spatial or only temporal intertwined module are quite limited and nearly equivalent, while adding both of them could achieve significant improvement. It demonstrates that this information exchange between different modes helps to reduce the prediction difficulty, and this interaction in spacial and temporal dimension are equally important.
- The change of the error on the ridesourcing dataset is more significant than that on the taxi dataset. That indicates the ridesourcing twin can benefit relatively more from the taxi twin. One of the reasons could be that, the passenger load for taxis in Manhattan is over around 20% more than ridesourcing in 2018 according to the statistics of the dataset. Thus, the taxi data contains more rich and complete features of human mobility than the ridesourcing data does.

E. Comparisons With Baselines

1) Overall Performance: In this part, we conduct the experiments of our model and all other baselines on not only New York datasets but also Chicago datasets to demonstrate its generalization. Comparison results are shown in Table IV. The smallest error values in each column are highlighted in bold. We can observe that:

- The performance of **HA** and **FC-LSTM** are the worst among all baselines, though **FC-LSTM** is slightly better than **HA**. These two methods consider only the temporal correlations and ignore the dependencies between regions. Their high error values express the importance and necessity of modeling the spatial dependencies.
- The methods based on the adaptive graph generally perform better than the methods based on the predefined graph. More specifically, methods, such as **DCRNN**, **T-GCN**, **ASTGCN** and **STSGCN**, need an adjacency matrix to be input and further model the spatial dependencies. By contrast, **MTGNN**, **CCRNN** and **GMSDR** construct a adaptive adjacency matrix that can discover the hidden correlations among regions from data. It helps to model more complicated spatial dependencies and achieve the higher prediction accuracy eventually.
- In general, co-prediction methods are better than the methods that consider only one mode. Our method **TSIN** achieves the best performance with the lowest errors on New York datasets. Both **CoGNN** and **TSIN** integrate the prediction tasks of two modes into one framework and model the interaction between modes. Such additional information from other mode could help to improve the demand prediction. While **CoGNN** focuses on the interaction of modes from the spatial perspective only, our **TSIN** models the inter-mode dependencies from both spatial and temporal perspectives. In this sense, it is more capable of learning the spatial-temporal features, and that leads to better results consequently.
- The experiments on CHI datasets verify the generalization of our model. The results are presented on the right side of Table IV. The MAE and RMSE values of **TSIN** on Chicago taxi and ridesourcing datasets are the lowest in their corresponding columns. It shows that our model achieve the lowest absolute errors. However, the MAPE values are relatively high, especially on the taxi dataset. The reason could be that, the MAPE is sensitive to the actual demand number and the taxi demand of Chicago is more sparse compared to the ridesourcing demand. And the small demand value leads to significant MAPE value as mentioned. As a matter of fact, this high MAPE value happen to not only **TSIN** but also all other baselines.

2) Computational Efficiency: Last but not least, we evaluate the computational efficiency based on NYC datasets from three perspectives: the number of parameters, the training time and the inference time. Our model and all neural network based baselines are evaluated under the same conditions for a fair

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON NEW YORK AND CHICAGO DATASETS

Method	NYC-Taxi			NYC-Ridesourcing			CHI-Taxi			CHI-Ridesourcing		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
HA	34.353	133.361	56.259	34.483	122.612	49.387	5.151	84.477	24.905	29.769	70.228	92.097
FC-LSTM	30.975	103.701	49.169	28.899	66.016	35.295	4.454	104.992	18.524	34.211	121.086	97.158
DCRNN	9.559	29.346	16.217	9.111	23.897	14.613	1.836	55.986	6.265	9.024	36.994	24.429
T-GCN	10.735	36.933	17.820	9.765	27.217	15.011	2.738	80.412	8.058	10.712	52.028	27.115
ASTGCN	9.614	29.706	16.048	9.082	22.993	14.041	1.735	53.676	5.428	8.339	30.619	20.516
STSGCN	10.284	31.107	17.179	9.602	24.608	14.789	1.737	52.367	5.824	8.701	33.141	21.970
AGCRN	9.508	26.412	16.672	9.063	22.076	14.385	1.690	53.259	5.710	8.877	31.495	22.906
MTGNN	9.609	27.287	16.134	8.979	22.359	13.537	1.662	**48.462**	5.179	8.659	39.973	20.487
CCRNN	9.334	26.697	15.905	8.847	22.607	13.807	1.670	53.534	5.341	8.665	35.602	26.364
GMSDR	9.357	27.222	15.969	8.695	21.715	13.250	1.818	50.338	5.772	8.408	34.570	20.535
CoGNN	9.320	25.693	15.979	8.909	20.985	13.762	1.679	56.995	5.387	8.167	**29.331**	20.409
TSIN	**8.971**	**24.382**	**15.495**	**8.424**	**20.191**	**12.855**	**1.587**	51.541	**5.036**	**7.865**	32.549	**19.067**

TABLE V
RESULTS ON COMPUTATIONAL EFFICIENCY

Model	#Parameters	Training time (s/epoch)	Inference time (ms)
FC-LSTM	259K	2.29	0.27
DCRNN	218K	7.85	12.93
T-GCN	12K	3.47	2.81
ASTGCN	62K	5.38	4.05
STSGCN	684K	11.56	15.11
AGCRN	728K	11.52	16.61
MTGNN	28K	5.61	4.71
CCRNN	74K	5.55	7.25
GMSDR	97K	8.08	13.49
CoGNN	298K	10.94	6.08
TSIN	163K	5.39	3.56

comparison. The results are reported in Table V. It can be observed that:

- Although **FC-LSTM** achieves the shortest training time and inference time, it has the lowest prediction performance among all these neural based models (i.e., the highest MAE, MAPE and RMSE values on all datasets in Table IV except **HA**).
- **TSIN** has a moderate model complexity and relatively short training and inference time. It is slightly longer than **T-GCN**, which has the smallest model size with the fewest parameters. However, **T-GCN** shows the second worst performance among neural network based models in Table IV.
- Both **CoGNN** and **TSIN** are designed for multi-task co-prediction, but **TSIN** is better since it is almost two times faster than **CoGNN** in terms of training and inference (2.03 and 1.71 times, respectively) with only 54% of **CoGNN**'s model size.

VI. DISCUSSION

In this section, we discuss the generalization performance of the proposed **TSIN** model in coupling more transportation modes. Specifically, we intend to integrate the bike demand data from NYC and conduct the co-prediction task for taxi, ridesourcing, bike simultaneously. The *NYC-Bike* dataset includes 13,488,880 trip records of Citi Bike stations in Manhattan during 2018. It is important to note that the bike data is *station-centric*, thus we aggregate trips from all bike stations within each region to count region-level bike demands, and then perform the same data preprocessing as *NYC-Taxi* dataset. Moreover, the bike stations cover only 56 regions of Manhattan, therefore, to ensure that multi-mode demands are spatially aligned, we use only taxi and ridesourcing demands from these 56 regions when co-predicting.

Due to the similarity among spatial-temporal prediction tasks, the *bike* twin can directly reuse the same neural network structure as the other two twins, that is, no further extra network design is required. In the intertwined component, we just need to add several relatively small and lightweight spatial-temporal modules to link the twins to each other for sharing information. Thus, it is safe to claim our **TSIN** model is highly flexible and easy extendable when coupling more types of demand data.

Furthermore, we aim to address the following two issues: (1) for n transportation modes in the urban transportation system, how many coupling combinations in total; and (2) for the three specific demand data, which coupling combination yields the best co-prediction result and why.

Theorem 1: The problem of determining the number of coupling combinations (B_n) for n transportation modes is equivalent to the one of determining the number of partitions for a set containing n elements.

Proof: The theorem is easy to understand and the proof is quite straightforward. Thus, the number of coupling combinations can be computed using the following recursive formula:

$$B_{n+1} = \sum_{k=0}^{n} C_n^k B_{n-k}, \tag{22}$$

where $B_0 = B_1 = 1$. ■

According to Eq. 22, there would be 5 coupling combinations in total when $n = 3$, listed as follows: **Combination I:** {{Taxi}, {Ridesourcing}, {Bike}}, which indicates each mode works independently and perform the prediction individually; **Combination II:** {{Taxi, Ridesourcing}, {Bike}}, which indicates taxi and ridesourcing modes work together in the co-prediction task while bike mode works independently and performs the prediction individually; **Combination III:** {{Taxi, Bike}, {Ridesourcing}}, which indicates taxi and bike

TABLE VI
PERFORMANCE COMPARISON OF DIFFERENT COMBINATIONS ON 56 REGIONS OF MANHATTAN

Combination	NYC-Taxi			NYC-Ridesourcing			NYC-Bike		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
I	9.977	24.998	16.224	9.045	19.869	13.553	2.877	46.068	4.648
II	**9.838**	**21.475**	16.207	**8.824**	**18.134**	13.337	2.877	46.068	4.648
III	9.925	23.154	16.220	9.045	19.869	13.553	**2.832**	43.460	**4.577**
IV	9.977	24.998	16.224	8.993	19.971	13.482	2.835	43.845	4.615
V	9.843	22.364	**16.188**	8.837	19.223	**13.324**	2.841	**43.321**	4.667

modes work together and perform the co-prediction while ridesourcing mode works independently and performs the prediction individually; **Combination IV:** {{Taxi}, {Bike, Ridesourcing}}, which indicates bike and ridesourcing modes work together and perform the co-prediction while taxi mode works independently and performs the prediction individually; and **Combination V:** {{Taxi, Ridesourcing, Bike}}, which indicates all three modes work together and perform the co-prediction simultaneously.

Table VI reports the performance comparison of different coupling combinations. The colored numbers in each row represent that the corresponding mode works independently in that coupling combination, and the best results in each column are highlighted in bold. Please note that the study area in Table VI contains only 56 regions in Manhattan, such a sub-area does not fully reflect the spatial dependencies within original travel demands for the entire Manhattan. This may be the reason for the worse results in Table VI compared to IV. Further, we can find that:

- All best results do not appear in the first row of the table, indicating that coupling different types of demands into a unified co-prediction framework is indeed effect. In other words, it confirms our proposition of "coupling makes better" in this work.
- On the contrary, most best results appear in the second row of the table, clearly demonstrating the best combination happens to the case of coupling taxi and ridesourcing, and leaving bike alone. This is probably due to that passengers taking bike for quite different purposes (e.g., connecting trip, short travel) comparing to taxi and ridesourcing. Thus, it is reasonable that we couple taxi and ridesourcing for co-prediction, although our original decision is based on the domain knowledge.
- Just a few best results appear in the last row of the table, implying that "many could be better than all". Nevertheless, how to discover the best coupling combination for co-prediction is non-trivial and it should be a separated research problem that is worth exploring in the future.

VII. CONCLUSION AND FUTURE WORK

In this paper, we propose a multi-task learning framework named **TSIN** for multi-mode co-prediction. Specifically, the framework contains a taxi twin component and ridesourcing twin component, which performs the prediction for each mode, respectively. The twin components have exactly the same structure that contain a number of stacked blocks composed of temporal and spatial convolutional layers, to capture the intra-mode dependencies. Furthermore, we also design a temporal and spatial intertwined component to bridge the twins for learning the inter-mode spatial temporal dependencies. Extensive experiments conducted on four real-world datasets demonstrate the effectiveness and superiority of **TSIN**.

In the future, we plan to broaden and deepen this work in the following directions. First, we plan to incorporate mode-shifting behaviours such as 'jockeying' explicitly into the deep model in a theory-guided/physical informed manner. Second, we plan to design twin components with heterogeneous network structures. Generally, different modes of transportation services have heterogeneous traffic nodes (such as bus stations and taxi zones), and the spatial-temporal patterns of different modes could also be diverse, thus it may be better to design the mode-specific neural network architecture for the mode-specific demand prediction [31]. Moreover, the most suitable coupling method between the different transport modes also needs to be elaborately designed. Finally, we intend to investigate the impact of zone/grid size on multi-mode demand prediction if the spatially fine-grained traffic data is available.

REFERENCES

[1] L. Bai, L. Yao, S. S. Kanhere, X. Wang, and Q. Z. Sheng, "STG2Seq: Spatial–temporal graph to sequence model for multi-step passenger demand forecasting," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1981–1987.

[2] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 17804–17815.

[3] K. Bandara, C. Bergmeir, and H. Hewamalage, "LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1586–1599, Apr. 2021.

[4] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 62, pp. 21–34, Jan. 2016.

[5] P. S. Castro, D. Zhang, C. Chen, S. Li, and G. Pan, "From taxi GPS traces to social and community dynamics: A survey," *ACM Comput. Surv.*, vol. 46, no. 2, pp. 1–34, Nov. 2013.

[6] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6164–6173, Apr. 2009.

[7] D. Chai, L. Wang, and Q. Yang, "Bike flow prediction with multi-graph convolutional networks," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2018, pp. 397–400.

[8] C. Chen, Q. Liu, X. Wang, C. Liao, and D. Zhang, "Semi-Traj2Graph identifying fine-grained driving style with GPS trajectory data via multi-task learning," *IEEE Trans. Big Data*, vol. 8, no. 6, pp. 1550–1565, Dec. 2022.

[9] C. Chen, D. Zhang, Y. Wang, and H. Huang, *Enabling Smart Urban Services With GPS Trajectory Data*. Cham, Switzerland: Springer, 2021.

[10] D. Gammelli, I. Peled, F. Rodrigues, D. Pacino, H. A. Kurtaran, and F. C. Pereira, "Estimating latent demand of shared mobility through censored Gaussian processes," *Transp. Res. C, Emerg. Technol.*, vol. 120, Nov. 2020, Art. no. 102775.

[11] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI*, vol. 33, 2019, pp. 922–929.

[12] S. Guo et al., "A simple but quantifiable approach to dynamic price prediction in ride-on-demand services leveraging multi-source urban data," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 3, pp. 1–24, Sep. 2018.

[13] J. Jiao and F. Wang, "Shared mobility and transit-dependent population: A new equity opportunity or issue?" *Int. J. Sustain. Transp.*, vol. 15, no. 4, pp. 294–305, Feb. 2021.

[14] J. Ke, S. Feng, Z. Zhu, H. Yang, and J. Ye, "Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach," *Transp. Res. C, Emerg. Technol.*, vol. 127, Jun. 2021, Art. no. 103063.

[15] J. Ke, H. Zheng, H. Yang, and X. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 591–608, Dec. 2017.

[16] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.

[17] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1003–1012.

[18] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[19] Y. Liang, G. Huang, and Z. Zhao, "Joint demand prediction for multimodal systems: A multi-task multi-relational spatiotemporal graph neural network approach," *Transp. Res. C, Emerg. Technol.*, vol. 140, Jul. 2022, Art. no. 103731.

[20] C. Liu, C.-X. Chen, and C. Chen, "META: A city-wide taxi repositioning framework based on multi-agent reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13890–13895, Aug. 2022.

[21] D. Liu, J. Wang, S. Shang, and P. Han, "MSDR: Multi-step dependency relation networks for spatial temporal forecasting," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 1042–1050.

[22] M. Liu, B. Du, and L. Sun, "Co-prediction of multimodal transportation demands with self-learned spatial dependence," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 824–833.

[23] T. Liu, W. Wu, Y. Zhu, and W. Tong, "Predicting taxi demands via an attention-based convolutional recurrent neural network," *Knowl.-Based Syst.*, vol. 206, Oct. 2020, Art. no. 106294.

[24] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi–passenger demand using streaming data," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1393–1402, Sep. 2013.

[25] L. Rayle, D. Dai, N. Chan, R. Cervero, and S. Shaheen, "Just a better taxi? A survey-based comparison of taxis, transit, and ridesourcing services in San Francisco," *Transp. Policy*, vol. 45, pp. 168–178, Jan. 2016.

[26] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial–temporal synchronous graph convolutional networks: A new framework for spatial–temporal network data forecasting," in *Proc. 34th AAAI Conf. Artif. Intell.*, vol. 34, no. 1. New York, NY, USA: AAAI Press, Apr. 2020, pp. 914–921.

[27] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 124–132, Mar. 2006.

[28] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 3104–3112.

[29] 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, Apr. 2022.

[30] Y. Tong et al., "The simpler the better: A unified approach to predicting original taxi demands based on large-scale online platforms," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1653–1662.

[31] L. Wang, D. Chai, X. Liu, L. Chen, and K. Chen, "Exploring the generalizability of spatio-temporal traffic prediction: Meta-modeling and an analytic framework," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3870–3884, Apr. 2023.

[32] Q. Wang et al., "Learning shared mobility-aware knowledge for multiple urban travel demands," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 7025–7035, May 2022.

[33] S. Wang, J. Cao, and P. S. Yu, "Deep learning for spatio-temporal data mining: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3681–3700, Aug. 2022.

[34] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 753–763.

[35] Z. Wu, D. Zheng, S. Pan, Q. Gan, G. Long, and G. Karypis, "TraverseNet: Unifying space and time in message passing for traffic forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 14, 2022, doi: 10.1109/TNNLS.2022.3186103.

[36] P. Xie et al., "Spatio-temporal dynamic graph relation learning for urban metro flow prediction," *IEEE Trans. Knowl. Data Eng.*, early access, Apr. 25, 2023, doi: 10.1109/TKDE.2023.3269771.

[37] H. Xu, T. Zou, M. Liu, Y. Qiao, J. Wang, and X. Li, "Adaptive spatiotemporal dependence learning for multi-mode transportation demand prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18632–18642, Oct. 2022.

[38] J. Xu, R. Rahmatizadeh, L. Bölöni, and D. Turgut, "Real-time prediction of taxi demand using recurrent neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2572–2581, Aug. 2018.

[39] H.-F. Yang, T. S. Dillon, and Y. P. Chen, "Optimized structure of the traffic flow forecasting model with a deep learning approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2371–2381, Oct. 2017.

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

[41] J. Ye, L. Sun, B. Du, Y. Fu, X. Tong, and H. Xiong, "Co-prediction of multiple transportation demands based on deep spatio-temporal neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 305–313.

[42] J. Ye, L. Sun, B. Du, Y. Fu, and H. Xiong, "Coupled layer-wise graph convolution for transportation demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, 2021, pp. 4617–4625.

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

[44] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 468–478, Mar. 2020.

[45] K. Zhang, Z. Liu, and L. Zheng, "Short-term prediction of passenger demand in multi-zone level: Temporal convolutional neural network with multi-task learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1480–1490, Apr. 2020.

[46] J. Zhao, C. Chen, H. Huang, and C. Xiang, "Unifying uber and taxi data via deep models for taxi passenger demand prediction," *Pers. Ubiquitous Comput.*, vol. 27, no. 3, pp. 523–535, Jun. 2023.

[47] J. Zhao et al., "2F-TP: Learning flexible spatiotemporal dependency for flexible traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 3, 2022, doi: 10.1109/TITS.2022.3146899.

[48] K. Zhao, D. Khryashchev, and H. Vo, "Predicting taxi and uber demand in cities: Approaching the limit of predictability," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2723–2736, Jun. 2021.

[49] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[50] X. Zhou, Y. Shen, Y. Zhu, and L. Huang, "Predicting multi-step citywide passenger demands using attention-based neural networks," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 736–744.

[51] Z.-H. Zhou, "Rehearsal: Learning from prediction to decision," *Frontiers Comput. Sci.*, vol. 16, no. 4, p. 164352, 2022.

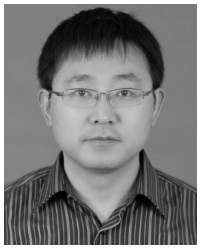
Jie Zhao (Graduate Student Member, IEEE) received the B.S. degree from the School of Computer and Information, Anhui Polytechnic University, Anhui, China, in 2017. He is currently pursuing the Ph.D. degree with the College of Computer Science, Chongqing University, China. His research interests include spatial–temporal data mining, traffic prediction, and graph representation learning.

Chao Chen (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in control science and control engineering from Northwestern Polytechnical University, Xi'an, China, in 2007 and 2010, respectively, and the joint Ph.D. degree from Sorbonne University and Institut Mines-Télécom/Télécom SudParis, France, in 2014. He is currently a Full Professor with the College of Computer Science, Chongqing University, Chongqing, China. He has published over 100 papers, including 40 ACM/IEEE TRANSACTIONS. His work on taxi trajectory data mining was featured by IEEE SPECTRUM in 2011, 2016, and 2020, respectively. He was also a recipient of the Best Paper Runner-Up Award at MobiQuitous 2011. His research interests include pervasive computing, mobile computing, urban logistics, data mining from large-scale GPS trajectory data, and big data analytics for smart cities. He is a Senior Member of CCF.

Wanyi Zhang received the M.Sc. degree in computer science from Jilin University, China, and the Ph.D. degree from the Department of Information Engineering and Computer Science, University of Trento, Italy, in 2022. She is currently a Post-Doctoral Research Associate with Chongqing University, China. Her research interests include ubiquitous computing, mobile crowd sensing systems, personal context recognition, user behavior study, and the design of skeptical learning framework dealing with annotations from unreliable users.

Ruiyuan Li received the B.E. and M.S. degrees from Wuhan University, China, in 2013 and 2016, respectively, and the Ph.D. degree from Xidian University, China, in 2020. He was the Head of the Spatio-Temporal Data Group, JD Intelligent Cities Research, leading the research and development of JUST (JD Urban spatio-temporal data engine). Before joining JD, he had interned in Microsoft Research Asia from 2014 to 2017. He is currently an Associate Professor with Chongqing University, China. His research interests include spatio-temporal data management and urban computing.

Fuqiang Gu (Member, IEEE) received the Ph.D. degree from The University of Melbourne. He is currently a Professor with the College of Computer Science, Chongqing University, China. Before joining the Chongqing University, he has successively worked as a Researcher with RWTH-Aachen University, University of Toronto, and National University of Singapore. His main research interests include positioning and navigation, robotics, autonomous systems, and machine learning. He is a member of ACM and CCF.



Songtao Guo (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer software and theory from Chongqing University, Chongqing, China, in 1999, 2003, and 2008, respectively. From 2011 to 2012, he was a Professor with Chongqing University. He was a Senior Research Associate with the City University of Hong Kong from 2010 to 2011 and a Visiting Scholar with Stony Brook University, New York, from 2011 to 2012. He is currently a Full Professor with Chongqing University. His research interests include wireless sensor networks, wireless ad hoc networks, data center networks, and mobile edge computing. He has published more than 100 scientific papers in leading refereed journals and conferences. He has received many research grants as a principal investigator from the National Science Foundations of China and Chongqing and the Post-Doctoral Science Foundation of China. He is a Senior Member of ACM/CCF.

Jun Luo received the B.S. and M.S. degrees in mechanical engineering from Henan Polytechnic University, Jiaozuo, China, in 1994 and 1997, respectively, and the Dr.Eng. degree from the Research Institute of Robotics, Shanghai Jiao Tong University, Shanghai, China, in 2000. His research interests include robot sensing, sensory feedback, mechatronics, man machine interfaces, and special robotics.

Yu Zheng (Fellow, IEEE) is currently the Vice President and the Chief Data Scientist with the JD Finance Group, passionate about using big data and AI technology to tackle urban challenges. He also leads the JD Urban Computing Business Unit as the President and serves as the Director of the JD Intelligent Cities Research. Before joining JD, he was a Senior Research Manager of Microsoft Research. He is also a Chair Professor with Shanghai Jiao Tong University and an Adjunct Professor with The Hong Kong University of Science and Technology. His research interests include big data analytics, spatio-temporal data mining, machine learning, and artificial intelligence.