Semi-Supervised City-Wide Parking Availability Prediction via Hierarchical Recurrent Graph Neural Network

Weijia Zhang, Hao Liu, *Member, IEEE,* Yanchi Liu, *Member, IEEE,* Jingbo Zhou, *Member, IEEE,* Tong Xu, *Member, IEEE,* and Hui Xiong, *Fellow, IEEE*

Abstract—The ability to predict city-wide parking availability is crucial for the successful development of Parking Guidance and Information (PGI) systems. The effective prediction of city-wide parking availability can boost parking efficiency, improve urban planning, and ultimately alleviate city congestion. However, it is a non-trivial task for city-wide parking availability prediction because of three major challenges: 1) the non-Euclidean spatial autocorrelation among parking lots, 2) the dynamic temporal autocorrelation inside of and between parking lots, and 3) the scarcity of information about real-time parking availability obtained from real-time sensors (*e.g.*, camera, ultrasonic sensor, and bluetooth sensor). To this end, we propose a *Semi-supervised Hierarchical Recurrent Graph Neural Network-X* (SHARE-X) to predict parking availability of each parking lot within a city. Specifically, we first propose a hierarchical graph convolution module to model the non-Euclidean spatial autocorrelation among parking lots. Along this line, a contextual graph convolution block and a multi-resolution soft clustering graph convolution block are respectively proposed to capture local and global spatial dependencies between parking lots. Moreover, we devise a hierarchical attentive recurrent network module to incorporate both short and long-term dynamic temporal dependencies of parking lots. Additionally, a parking availability approximation module is introduced to estimate missing real-time parking availabilities from both spatial and temporal domains. Finally, experiments on two real-world datasets demonstrate that SHARE-X outperforms eight state-of-the-art baselines in parking availability prediction.

Index Terms—Parking availability prediction, Graph neural network, Semi-supervised learning, Urban computing.

1 Introduction

In recent years, we have witnessed significant development of various Intelligent Transportation Systems (ITS) [1], e.g., parking guidance and information (PGI) system. According to a survey by the International Parking Institute (IPI)[1], over 30% of cars on the road are searching for parking, and these cruising cars contribute up to 40% traffic jams in urban areas [2]. Thus, city-wide parking availability prediction is of great importance to help drivers efficiently find parking space, help governments for urban planning, and alleviate the city's traffic congestion.

Due to its importance, city-wide parking availability prediction has attracted much attention from both academia and industry. On one hand, Google Maps predicts parking difficulty on a city-wide scale based on users' survey and trajectory data [3], and Baidu Maps estimates real-time city-wide parking availability based on environmental contextual features (*e.g.*, Point of Interest (POI), map queries) [4], [5]. The above mentions make city-wide parking availability prediction based on biased and indirect input signals (*e.g.*, user's feedback is noisy and lagged), which may induce inaccurate prediction results. On the other hand, in recent years, we have witnessed real-time sensor devices such as camera, ultrasonic sensor, and bluetooth sensor become ubiquitous, which can significantly improve the prediction accuracy of parking availability [6], [7], [8]. However, for economic and privacy concerns, it is difficult to install real-time sensors covering all parking lots of a city.

In this paper, we propose to *simultaneously* predict the availability for all the parking lots within a city, based on both environmental contextual data (*e.g.*, POI distribution, population) and partially observed real-time parking availability data. By integrating both datasets, we can make a better parking availability prediction at city-scale. However, this is a non-trivial task which faces the following three major challenges. (1) *Spatial autocorrelation*. The availability of a parking lot is not only affected by the occupancy of nearby parking lots but may also synchronize with distant parking lots [9], [10], [11]. The first challenge is how to model the irregular and non-Euclidean autocorrelation between parking lots. (2) *Temporal autocorrelation*. Future availability of a parking lot is correlated with its availability of previous time periods [12], [13], including short and long-term temporal dependencies [14]. Besides, the spatial autocorrelation between parking lots may also vary over time [15], [16]. How to model short and long-term dynamic temporal autocorrelation of each parking lot is another

- *Weijia Zhang and Tong Xu are with Anhui Province Key Lab of Big Data Analysis and Application, School of Computer Science, University of Science and Technology of China. E-mail: wjzhang3@mail.ustc.edu.cn, tongxu@ustc.edu.cn.*
- *Hao Liu and Jingbo Zhou are with Business Intelligence Lab, Baidu Research, National Engineering Laboratory of Deep Learning Technology and Application, China. E-mail: liuhaoscut@gmail.com, zhoujingbo@baidu.com.*
- *Yanchi Liu and Hui Xiong are with Rutgers, the State University of New Jersey, USA. E-mail: yanchi.liu@rutgers.edu, hxiong@rutgers.edu.*
- *Hao Liu and Hui Xiong are corresponding authors.*

1. https://www.parking.org/wp-content/uploads/2015/12/Emerging-Trends-2012.pdf

challenge. (3) *Parking availability data scarcity*. Only a small portion of parking lots are equipped with real-time sensors. According to one of the largest map service applications, there are over $70,000$ parking lots in Beijing, however, only 6.12% of them have real-time parking availability data. The third challenge is how to utilize the scarce real-time parking availability information.

To tackle the above challenges, we have done some preliminary work in [17], which proposes the Semi-supervised Hierarchical Recurrent Graph Neural Network (SHARE) framework to incorporate both environmental contextual factors and sparse real-time parking availability data for city-wide parking availability prediction. Specifically, we first propose a hierarchical graph convolution module to capture non-Euclidean spatial correlations among parking lots. It consists of a contextual graph convolution block and a soft clustering graph convolution block for local and global spatial dependencies modeling, respectively. Second, we propose a parking availability approximation module to estimate missing real-time parking availabilities of parking lots without sensor monitoring. Specifically, we introduce a propagating convolution block and reuse the temporal module to approximate missing parking availabilities from both spatial and temporal domains, then fuse them through an entropy-based mechanism. Then, we evaluate SHARE on two real-world datasets collected from BEIJING and SHENZHEN, two metropolises in China. The results demonstrate our model achieves the best prediction performance against seven baselines.

In this paper, we propose the *Semi-supervised Hierarchical Recurrent Graph Neural Network-X* (SHARE-X), which further extends SHARE for more effective city-wide parking availability prediction. We further make the following four major contributions:

- We propose a multi-resolution soft clustering graph convolution block for global spatial autocorrelation modeling. The multi-resolution soft clustering graph convolution block is a generalized version of soft clustering graph convolution, where arbitrary layers soft clustering graph convolution operation can be stacked for parking availability prediction.
- We devise a hierarchical attentive recurrent network module for temporal autocorrelation modeling, which integrates both short-term temporal dependency and long-term periodicity for parking availability prediction.
- We provide a systematic complexity analysis of SHARE-X and its basic variant SHARE.
- We conduct extensive experiments on two real-world datasets, and the results demonstrate the effectiveness of SHARE-X and its components.

2 Related Work

Parking availability prediction. Previous studies on parking availability prediction mainly fell into two categories, contextual data based prediction and real-time sensor based prediction. For contextual data based prediction, Google-parking [3] and Du-parking [4] predicted parking availability based on indirect signals (*e.g.*, user feedbacks and contextual factors), which might induce an inaccurate prediction result. For real-time sensor based prediction, the study in [12] proposed an auto-regressive model and the study in [7] proposed a boosting method for parking availability inference. Above approaches were limited by economic and privacy concerns and were hard to be scaled to all parking lots in a city. Moreover, all the above approaches didn't fully exploit non-Euclidean spatial autocorrelations between parking lots, which limited their prediction performance.

Graph neural network. Graph neural network (GNN) extended the well-known convolution neural network to non-Euclidean graph structures, where the representation of each node was derived by first aggregating and then transforming representations of its neighbors [18], [19], [20]. Note that our soft clustering graph convolution is connected to [21], but our objective is to capture global spatial correlation for node-level prediction. Due to its effectiveness, GNN has been successfully applied to several spatiotemporal forecasting tasks, such as traffic flow forecasting [22], [23], [24], [25], [26] and taxi demand forecasting [27], [28], [29]. However, we argue these approaches overlooked either contextual factors or global spatial dependencies, and were not tailored for parking availability prediction.

3 Preliminaries

Consider a set of parking lots $P = P_l \cup P_u = \{p_1, p_2, \ldots, p_N\}$, where N is the total number of parking lots, P_l and P_u denote a set of parking lots with and without real-time sensors (*e.g.*, camera, ultrasonic sensor, bluetooth sensor), respectively. Let $\mathbf{X}^{c,t} = [\mathbf{x}_1^{c,t}, \mathbf{x}_2^{c,t}, \ldots, \mathbf{x}_N^{c,t}] \in \mathcal{R}^{N \times M}$ denote observed M dimensional contextual feature vectors (*e.g.*, POI distribution, population) for all parking lots in P at time t. Next we begin the problem definition of parking availability prediction with the definition of parking availability.

Definition 1. ***Parking availability (PA)****. Given a parking lot $p_i \in P$, at time step t, the parking availability of p_i, denoted y_i^t, is defined as the number of vacant parking spots in p_i.*

Specifically, we use $\mathbf{y}_{P_l}^t = [y_1^t, y_2^t, \ldots, y_{|P_l|}^t]$ to denote observed PAs of parking lots in P_l at time step t. In this paper, we are interested in predicting PAs for all parking lots $p_i \in P$ by leveraging the contextual data of P and partially observed real-time parking availability data of P_l.

Problem 1. ***Parking availability prediction****. Given a historical time window T, contextual features for all parking lots $\boldsymbol{\mathcal{X}}^c = (\mathbf{X}^{c,t-T+1}, \mathbf{X}^{c,t-T+2}, \ldots, \mathbf{X}^{c,t})$, and partially observed real-time PAs $\boldsymbol{\mathcal{Y}}_{P_l} = (\mathbf{y}_{P_l}^{t-T+1}, \mathbf{y}_{P_l}^{t-T+2}, \ldots, \mathbf{y}_{P_l}^{t})$, our problem is to predict PAs for all $p_i \in P$ over the next τ time steps,*

$$f(\boldsymbol{\mathcal{X}}^c; \boldsymbol{\mathcal{Y}}_{P_l}) \rightarrow (\hat{\mathbf{y}}^{t+1}, \hat{\mathbf{y}}^{t+2}, \ldots, \hat{\mathbf{y}}^{t+\tau}), \tag{1}$$

where $\hat{\mathbf{y}}^{t+1} = \hat{\mathbf{y}}_{P_l}^{t+1} \cup \hat{\mathbf{y}}_{P_u}^{t+1}$, $f(\cdot)$ is the mapping function we aim to learn.

4 Framework Overview

The architecture of SHARE-X is shown in Figure 1, where the inputs are contextual features as well as partially observed real-time PAs, and the outputs are the predicted PAs of all parking lots in next τ time steps. There are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3034140, IEEE Transactions on Knowledge and Data Engineering

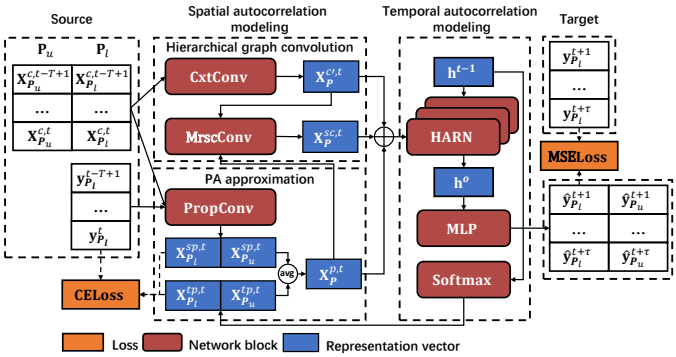


Fig. 1. The framework overview of SHARE-X.

three major components in SHARE-X. First, the *Hierarchical graph convolution* module models spatial autocorrelations among parking lots, including the *Contextual Graph Convolution* (CxtConv) operation and the *Multi-Resolution Soft Clustering Graph Convolution* (MrscConv) operation. The CxtConv block captures local spatial dependencies between parking lots through rich contextual features (*e.g.*, POI distribution, regional population), and the MrscConv block captures global spatial correlations among distant parking lots by softly assigning each parking lot to a set of latent cluster nodes. Second, for the temporal autocorrelation modeling module, we devise the *hierarchical attentive recurrent network* (HARN) to model both short and long-term temporal dependencies among each parking lot. Third, the *PA approximation* module estimates distributions of missing PAs for parking lots in P_u, from both spatial and temporal domains. In the spatial domain, the *Propagating Graph Convolution* (PropConv) block propagates observed real-time PAs to approximate missing PAs based on the contextual similarity of each parking lot. In the temporal domain, we reuse the *Gated Recurrent Unit* (GRU) block to approximate current PA distributions based on its output in previous time period. Two estimated PA distributions are then fused through an entropy-based mechanism and fed to MrscConv block and HARN module.

5 Hierarchical Graph Convolution

We first introduce the hierarchical graph convolution module, including the contextual graph convolution block and the multi-resolution soft clustering graph convolution block. In this section, we assume all graph convolution operations are applied on time t and omit the time superscript to ease the presentation.

5.1 Contextual graph convolution

In the spatial domain, the PA of nearby parking lots are usually correlated and mutually influenced by each other. For example, when there is a big concert, the PAs of parking lots near the concert hall are usually low, and the parking demand usually gradually diffuses from nearby to distant. Inspired by the recent success of graph convolution network [18], [30] on processing non-Euclidean graph structures, we first introduce the CxtConv block to capture local spatial dependencies solely based on contextual features.

We model the local spatial correlations among parking lots as a graph $G = (V, E, A)$, where $V = P$ is the set of parking lots, E is a set of edges indicating connectivity among parking lots, and A denotes the proximity matrix of G [31]. Specifically, we define the connectivity constraint $e_{ij} \in E$ as

$$e_{ij} = \begin{cases} 1, & dist(v_i, v_j) \leq \epsilon \\ 0, & otherwise \end{cases}, \tag{2}$$

where $dist(\cdot)$ is the road network distance between parking lots p_i and p_j, ϵ is a distance threshold.

Since the influence of different nearby parking lots may vary non-linearly, we employ an attention mechanism to compute the coefficient between parking lots, defined as

$$c_{ij} = Attn_{cxt}(\mathbf{W}_a \mathbf{x}_i^c, \mathbf{W}_a \mathbf{x}_j^c), \tag{3}$$

where $\mathbf{x}_i^c$ and $\mathbf{x}_j^c$ are current contextual features of parking lot p_i and p_j, $\mathbf{W}_a$ is a learnable weighted matrix shared over all edges, and $Attn_{cxt}(\cdot)$ is a shared attention function (*e.g.*, dot-product, concatenation) [32]. The proximity score between p_i and p_j is further defined as

$$\alpha_{ij} = \frac{exp(c_{ij})}{\sum_{k \in \mathcal{N}_i} exp(c_{ik})}. \tag{4}$$

In general, the above attention mechanism is capable of computing pair-wise proximity scores for all $p_i \in P$. However, this formulation will lead to quadratic complexity. To weigh more attention on neighboring parking lots and help faster convergence, we inject the adjacency constraint where the attention operation only operates on adjacent nodes $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ is a set of neighboring parking lots of p_i in G. Note that the influence of nearby parking lots at different time steps may also vary. Therefore we learn a different proximity score for each different time step.

Once $\alpha_{ij} \in A$ is obtained, the contextual graph convolution operation updates representation of current parking lot by aggregating and transforming its neighbors, defined as

$$\mathbf{x}_i^{c'} = \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}_c \mathbf{x}_j^c), \tag{5}$$

where σ is a non-linear activation function, and $\mathbf{W}_c$ is a learnable weighted matrix shared over all parking lots. Note that we can stack l identical contextual graph convolution layers to capture l-hop local dependencies, where the output of the $(l-1)$-th convolution layer is the input of l-th layer.

5.2 Multi-resolution soft clustering graph convolution

Besides local spatial correlation, distant parking lots may also be correlated. For example, distant parking lots in similar functional areas may show similar PA, *e.g.*, business areas may have lower PA at office hour, and residential areas may have higher PA at the same time. However, CxtConv only captures local spatial correlation. [33] shows that when l goes large, the representation of all parking lots tends to be similar, therefore losing discriminative power. To this end, we propose the Multi-Resolution Soft Clustering Graph Convolution (MrscConv) to capture global spatial correlations between parking lots, as illustrated in Figure 2. The intuition behind MrscConv is two-fold. First, distant parking lots may have similar contextual features and PAs,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3034140, IEEE Transactions on Knowledge and Data Engineering

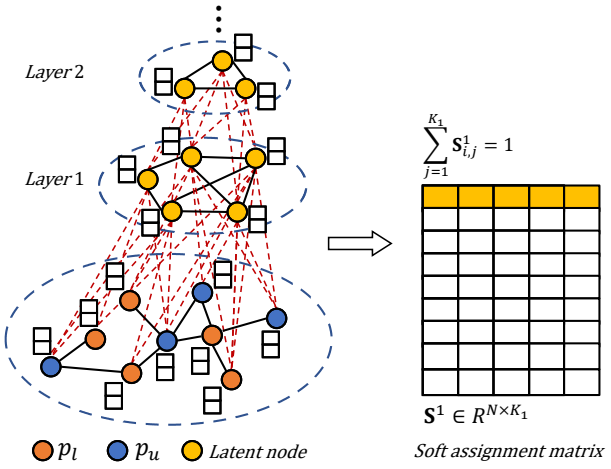


Fig. 2. Multi-resolution soft clustering graph convolution.

therefore should have similar representations. The shared latent node representation can be viewed as a regularization for the prediction task. Second, one parking lot may be mapped to multiple latent nodes. If we view each latent node as a different functionality class in different granularity, a parking lot may serve for several functionalities. For example, a parking lot in a recreational center may be occupied by external visitors from a nearby office building.

5.2.1 Primitive soft clustering graph convolution

We first present Soft Clustering Graph Convolution (SCConv), the primitive operation in MrscConv for specified soft clustering convolution. Specifically, SCConv defines a set of latent nodes and learns the representation of latent nodes based on learned representations of each parking lot. Rather than cluster each parking lot into a specific cluster, we learn a soft assignment matrix so that each parking lot has a chance to belong to multiple latent nodes with different probabilities. Define K as the number of latent nodes, let $\mathbf{S} \in \mathcal{R}^{N \times K}$ denote the soft assignment matrix, where $\mathbf{S}_{i,j} \in \mathbf{S}$ denotes the probability of the i-th parking lot p_i maping to j-th latent node. Specifically, let $\mathbf{S}_{i,\cdot}$ denote the i-th row and $\mathbf{S}_{\cdot,j}$ denote the j-th column of $\mathbf{S}$. Define $\mathbf{x}_i^{cp}$ as the combination of CxtConv representation $\mathbf{x}_i^{c'}$ and approximated PA distribution $\mathbf{x}_i^p$ (details of $\mathbf{x}_i^{cp}$ and $\mathbf{x}_i^p$ will be discussed in Section 7.3), each row of $\mathbf{S}$ is computed as

$$\mathbf{S}_{i,\cdot} = Softmax(\mathbf{W}_s \mathbf{x}_i^{cp}), \tag{6}$$

which guarantees that a given parking lot belonging to each latent node follows a probability distribution, and $\mathbf{W}_s$ are learnable parameters.

Once $\mathbf{S}$ is obtained, we further denote $\mathbf{S}^\top$ as the transpose of $\mathbf{S}$, then the representation of each latent node $\mathbf{x}_i^s \in \mathbf{X}^s$ can be derived by

$$\mathbf{x}_i^s = \sum_{j=1}^{N} \mathbf{S}_{i,j}^\top \mathbf{x}_j^{cp}. \tag{7}$$

Given the representation of each latent node, similar to CxtConv, we apply graph convolution operation to capture the dependency between each latent node,

$$\mathbf{x}_i^{s'} = \sigma\Big(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^s \mathbf{W}_l \mathbf{x}_j^s\Big), \tag{8}$$

where $\mathbf{W}_l$ are learnable parameters, σ is a non-linear activation function, and α_{ij}^s is the proximity score between two latent nodes. Rather than introducing an extra attention parameter as in CxtConv, we derive the proximity score between latent nodes based on adjacency constraint between parking lots,

$$\alpha_{ij}^s = \sum_{m=1}^{N} \sum_{n=1}^{N} \mathbf{S}_{i,m}^\top a_{mn} \mathbf{S}_{n,j}. \tag{9}$$

where a_{mn} equals one if parking lots p_m and p_n are connected in connectivity constraints of CxtConv (Eq. (2)) or PropConv (Eq. (25)). With learned latent node representation, we generate the soft clustering representation for each parking lot by aggregating all latent node representations from the top layer to the bottom,

$$\mathbf{x}_i^{sc} = \sum_{j=1}^{K} \mathbf{S}_{i,j} \mathbf{x}_j^{s'}. \tag{10}$$

5.2.2 Multi-resolution generalization

Then, we extend SCConv to multiple resolutions. The MrscConv further improves the predictive power by explicitly modeling the latent hierarchy among distant parking lots.

Assume there are F SCConv layers in total. Consider the $(f-1)$-th and f-th SCConv layers, let $\mathbf{x}_i^{s',(f-1)}$ denote the latent representation derived from the $(f-1)$-th layer SCConv, and $\mathbf{S}^f \in \mathcal{R}^{K_{f-1} \times K_f}$ denote the corresponding soft assignment matrix, where K_f is the number of latent nodes in the f-th layer. In particular, we set $K_0 = N$, where N is the number of parking lots. We can derive the corresponding soft assignment matrix $\mathbf{S}^f$, latent representation $\mathbf{x}_i^{s',f}$, proximity score $\alpha_{ij}^{s,f}$ for the f-th SCConv layer based on Equation (6), Equation (7), (8), and Equation (9), respectively. Note that $a_{mn}^f = \alpha_{mn}^{s,f-1}$ when $f > 1$.

With the learned latent representations in different resolutions, one intermediate problem is how to generate the unified soft clustering representation for each parking lot. To this end, we further define the soft transition matrix through iterative multiplication of soft assignment matrix in each lower SCConv layer. For the f-th SCConv layer, the soft transition matrix is defined as

$$\mathbf{T}^f = \prod_{i=1}^{f} \mathbf{S}^i, \tag{11}$$

where $\mathbf{T}^f \in \mathcal{R}^{N \times K_f}$, $\mathbf{S}^i$ is the i-th layer soft assignment matrix such that $i \leq f$. $\mathbf{T}_{i,j}^f \in \mathbf{T}^f$ can be viewed as the probability that i-th parking lot p_i maps to the j-th latent node in the f-th SCConv layer. With the soft transition matrix, we can obtain the f-th layer soft clustering representation of each parking lot

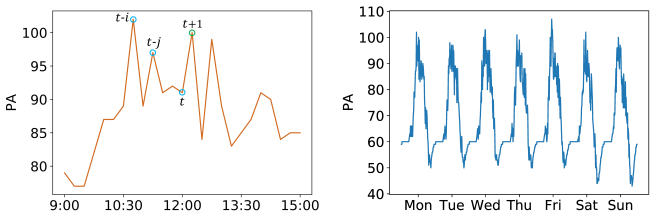
$$\mathbf{x}_i^{sc,f} = \sum_{j=1}^{K_f} \mathbf{T}_{i,j}^f \mathbf{x}_j^{s',f}. \tag{12}$$

Finally, we have the unified soft clustering representation,

$$\mathbf{x}_i^{sc} = [\mathbf{x}_i^{sc,1} \oplus \mathbf{x}_i^{sc,2} \oplus \cdots \oplus \mathbf{x}_i^{sc,F}], \tag{13}$$

where $\oplus$ is the concatenation operation. $\mathbf{x}_i^{sc}$ combines the soft clustering representation in each SCConv layers, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3034140, IEEE Transactions on Knowledge and Data Engineering



(a) PA in different time steps (b) Daily periodicity

Fig. 3. (a) PA in different time steps has a diversified correlation. (b) Daily periodicity of PA.

the latent tree structure adaptively regularizes distant correlated parking lots.

6 Hierarchical Attentive Recurrent Network

In the temporal domain, the future availability of a parking lot is correlated with its availability of previous time periods. Specifically, we identify two types of temporal dependency, the short-term dependency and long-term periodicity, as illustrated in Figure 3. Along this line, we devise a Hierarchical Attentive Recurrent Network (HARN) module to model short-term and long-term temporal dependencies simultaneously. The architecture of HARN is shown in Figure 4.

6.1 Base recurrent network

We leverage the Gated Recurrent Unit (GRU) [34], a simple yet effective variant of recurrent neural network (RNN), as the base block of HARN. We define $\mathbf{x}_i^t$ as the integrated representation of each parking lot at time t (details of $\mathbf{x}_i^t$ will be discussed in Section 7.3). Given representations of parking lot p_i in previous T time steps, $(\mathbf{x}_i^{t-T+1}, \mathbf{x}_i^{t-T+2}, \cdots, \mathbf{x}_i^t)$, we denote the hidden states of p_i at time step $t-1$ and t as $\mathbf{h}_i^{t-1}$ and $\mathbf{h}_i^t$, respectively. The temporal dependency between $\mathbf{h}_i^{t-1}$ and $\mathbf{h}_i^t$ can be modeled by

$$\mathbf{h}_i^t = (1 - \mathbf{z}_i^t) \circ \mathbf{h}_i^{t-1} + \mathbf{z}_i^t \circ \widetilde{\mathbf{h}}_i^t, \tag{14}$$

where $\mathbf{z}_i^t$ and $\widetilde{\mathbf{h}}_i^t$ are defined as

$$\begin{cases} \mathbf{r}_i^t = \sigma(\mathbf{W}_r[\mathbf{h}_i^{t-1} \oplus \mathbf{x}_i^t] + \mathbf{b}_r) \\ \mathbf{z}_i^t = \sigma(\mathbf{W}_z[\mathbf{h}_i^{t-1} \oplus \mathbf{x}_i^t] + \mathbf{b}_z) \\ \widetilde{\mathbf{h}}_i^t = \tanh(\mathbf{W}_{\widetilde{h}}[\mathbf{r}_i^t \circ \mathbf{h}_i^{t-1} \oplus \mathbf{x}_i^t] + \mathbf{b}_{\widetilde{h}}) \end{cases}, \tag{15}$$

where $\boldsymbol{W}_r$, $\boldsymbol{W}_z$, $\boldsymbol{W}_{\widetilde{h}}$, $\boldsymbol{b}_r$, $\boldsymbol{b}_z$, $\boldsymbol{b}_{\widetilde{h}}$ are learnable parameters, and $\circ$ denotes the Hadamard product.

6.2 Hierarchical attentive recurrent network

Although the basic GRU module captures the recent temporal dependency, it still has two major drawbacks. First, take Figure 3(a) for example, when predicting PA of the parking lot at time step $t+1$, the time steps $t-i$ and $t-j$ show higher correlations with $t+1$. However, the GRU ignores the diversified importance of previous time steps. Second, as shown in Figure 3(b), the PA shows strong daily periodicity, which can be utilized to improve the prediction performance. However, the GRU suffers from the gradient

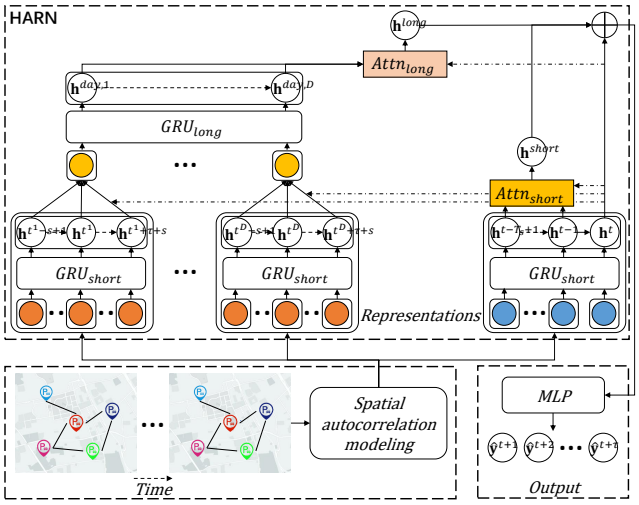


Fig. 4. The architecture of HARN. Given the output of hierarchical graph convolution and approximated PA as the input, the low-level GRU_{short} and $Attn_{short}$ captures diversified short-term dependency, while the high-level GRU_{long} and $Attn_{long}$ captures the long-term periodicity.

vanishing problem [35], and therefore fails to characterize the long-term temporal periodicity.

To tackle the first problem, we introduce a short-term attentive operation to quantify the influence of previous T_s time steps,

$$\alpha_{tj} = \frac{exp(Attn_{short}(\mathbf{h}_i^t, \mathbf{h}_i^j))}{\sum_{k=t-T_s+1}^{t} exp(Attn_{short}(\mathbf{h}_i^t, \mathbf{h}_i^k))}, \tag{16}$$

where T_s is the short-term time step length, $Attn_{short}(\cdot)$ is a shared attention function. Then, we can derive the short-term representation as

$$\mathbf{h}_i^{short} = \sum_{j=t-T_s+1}^{t} \alpha_{tj} \mathbf{h}_i^j. \tag{17}$$

$\mathbf{h}_i^{short}$ adaptively aggregates representations in previous T_s time steps and particularly pays more attention on highly correlated time steps.

For the long-term temporal periodicity, we consider PAs in a longer time period. Specifically, to reduce the length of the input sequence, for predicting next τ time steps, HARN incorporates the same time steps in previous D days as the input. Besides, to alleviate the temporal shifting problem [16] (*e.g.*, the lowest PA appeared at 9:00-9:15 today but appeared at 9:15-9:30 yesterday), HARN further incorporates s consecutive time slots before and after the next time step. For example, to predict next time step 9:00-9:15, we not only consider the PA at 9:00-9:15 in previous D days, but also incorporate time slots 8:30-9:00 and 9:15-9:45 in each previous day if $s = 2$. In this way, for the d-th day, we have $2s+\tau$ hidden states $(\mathbf{h}_i^{t^d-s+1}, \mathbf{h}_i^{t^d-s+2}, \cdots, \mathbf{h}_i^{t^d+\tau+s})$ (t^d denotes t-th time step in the d-th day). We reuse the short-term attentive operation and derive the d-th day representation,

$$\mathbf{x}_i^{day,d} = \sum_{j=t^d-s+1}^{t^d+\tau+s} \alpha_{tj} \mathbf{h}_i^j. \tag{18}$$

Then, we employ another GRU block (GRU_{long}) to model the temporal dependency between previous D days,

$$\mathbf{h}_i^{day,d} = GRU_{long}(\mathbf{x}_i^{day,d}, \mathbf{h}_i^{day,d-1}), \tag{19}$$

and derive the long-term representation via a long-term attentive operation,

$$\mathbf{h}_i^{long} = \sum_{d=1}^{D} \alpha_{td} \mathbf{h}_i^{day,d}, \tag{20}$$

$$\alpha_{td} = \frac{exp(Attn_{long}(\mathbf{h}_i^t, \mathbf{h}_i^d))}{\sum_{k=1}^{D} exp(Attn_{long}(\mathbf{h}_i^t, \mathbf{h}_i^k))}, \tag{21}$$

where $Attn_{long}$ is the attention function for long-term temporal dependencies modeling.

Finally, we obtain the overall representation by

$$\mathbf{h}_i^o = [\mathbf{h}_i^t \oplus \mathbf{h}_i^{short} \oplus \mathbf{h}_i^{long}]. \tag{22}$$

$\mathbf{h}_i^o$ can be directly used to predict PAs of next τ time steps,

$$(\hat{y}_i^{t+1}, \hat{y}_i^{t+2}, \ldots, \hat{y}_i^{t+\tau}) = \sigma(\mathbf{W}_o \mathbf{h}_i^o), \tag{23}$$

where $\mathbf{W}_o \in \mathcal{R}^{\tau \times |\mathbf{h}_i^o|}$.

7 Parking Availability Approximation

The real-time PA is a strong signal for future PA prediction. However, only a small portion (*e.g.*, 6.12% in Beijing) of real-time PAs can be obtained through real-time sensors, which prevents us from directly using real-time PA as an input feature. To leverage the information hidden in partially observed real-time PA, we approximate missing PAs from both spatial and temporal domains. The proposed method consists of three blocks, *i.e.*, the spatial PropConv block, the temporal GRU block, and the fusion block. Note that rather than approximate a scalar PA $\hat{y}_i$, we learn the distribution of $\hat{y}_i$, denoted as $\mathbf{x}_i^p$, for better information preservation. Given a PA y_i, we discretize its distribution to a q dimensional one hot vector $\mathbf{y}_i^p \in \mathcal{R}^p$. The objective of the PA approximation is to minimize the difference between $\mathbf{y}_i^p$ and $\mathbf{x}_i^p$.

7.1 Spatial based PA approximation

Similar to CxtConv, for each $p_i \in P_u$, the PropConv operation is defined as

$$\mathbf{x}_i^{sp} = \sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{y}_j^p, \tag{24}$$

where $\mathbf{x}_i^{sp}$ is the obtained PA distribution, α_{ij} is the proximity score between p_i and p_j. Different from CxtConv, the estimated PA is only aggregated from nearby parking lots with real-time PA, and we preserve the aggregated vector representation without extra activation function. The proximity score is computed via the same attention mechanism in Eq. (4), but with a relaxed connectivity constraint

$$e_{ij} = \begin{cases} 1, & dist(v_i, v_j) \leq max(\epsilon, dist_{knn}(v_i)), i \neq j \\ 0, & otherwise \end{cases}, \tag{25}$$

where $dist_{knn}(v_i)$ denotes the road network distance between parking lot p_i and its k-th nearest parking lot $p_j \in P_l$. The relaxed adjacency constraint improves node connectivity for more sufficient propagation of observed PAs, and therefore alleviates the data scarcity problem.

7.2 Temporal based PA approximation

We reuse the output of the GRU block to approximate real-time PA from the temporal domain. The difference between current PA approximation and future PA prediction is here we employ a different $Softmax$ function. Remember that in the previous step, we have obtained the hidden state $\mathbf{h}_i^{t-1}$ from GRU, thus we directly approximate distribution of PA at t by

$$\mathbf{x}_i^{tp,t} = Softmax(\mathbf{W}_{tp} \mathbf{h}_i^{t-1}). \tag{26}$$

This step doesn't introduce extra computation for GRU, and the *Softmax* layer normalizes $\mathbf{x}_i^{tp,t}$ to a distribution.

7.3 Approximated PA fusion

Rather than directly averaging $\mathbf{x}_i^{sp}$ and $\mathbf{x}_i^{tp}$, we propose an entropy-based mechanism to fuse the two PA distributions. Specifically, we weigh more on the approximation with less uncertainty [36], *i.e.*, the one with smaller entropy. Given an estimated PA distribution $\mathbf{x}_i$, its entropy is

$$H(\mathbf{x}_i) = -\sum_{j=1}^{p} \mathbf{x}_i(j) \log \mathbf{x}_i(j), \tag{27}$$

where $\mathbf{x}_i(j)$ represents the j-th dimension of $\mathbf{x}_i$. We fuse two PA distributions $\mathbf{x}_i^{sp}$ and $\mathbf{x}_i^{tp}$ as follow:

$$\mathbf{x}_i^p = \frac{exp(-H(\mathbf{x}_i^{sp}))\mathbf{x}_i^{sp} + exp(-H(\mathbf{x}_i^{tp}))\mathbf{x}_i^{tp}}{\mathbf{Z}_i}, \tag{28}$$

where $\mathbf{Z}_i = exp(-H(\mathbf{x}_i^{sp})) + exp(-H(\mathbf{x}_i^{tp}))$.

The approximated PA distribution $\mathbf{x}_i^p$ is applied for two tasks. First, it is concatenated with the learned representation of the CxtConv, $\mathbf{x}_i^{cp} = [\mathbf{x}_i^{c'} \oplus \mathbf{x}_i^p]$, and then fed to the MrscConv block for latent node representation learning. Second, it is combined with the output of the CxtConv and MrscConv, $\mathbf{x}_i^t = [\mathbf{x}_i^{c',t} \oplus \mathbf{x}_i^{sc,t} \oplus \mathbf{x}_i^{p,t}]$. We use $\mathbf{x}_i^t$ as the integrated representation for each parking lot $p_i \in P$ at time step t, and feed it into HARN for the overall representation learning.

8 Model Training

Since only parking lots P_l have observable labels, following the semi-supervised learning paradigm, SHARE-X aims to minimize the *mean square error* (MSE) between the predicted PA and the observed PA

$$O_1 = \frac{1}{\tau |P_l|} \sum_{i=1}^{|P_l|} \sum_{j=1}^{\tau} (\hat{y}_i^{t+j} - y_i^{t+j})^2. \tag{29}$$

Additionally, in PA approximation, we introduce extra cross entropy (CE) loss to minimize the error between the observed PA and approximated PA distributions (*i.e.*, the spatial and temporal based PA distribution approximation $\mathbf{x}_i^{sp,t}$ and $\mathbf{x}_i^{tp,t}$) in current time step t,

$$O_2 = -\frac{1}{|P_l|} \sum_{i=1}^{|P_l|} \mathbf{y}_i^{p,t} \log \mathbf{x}_i^{sp,t}, \tag{30}$$

$$O_3 = -\frac{1}{|P_l|} \sum_{i=1}^{|P_l|} \mathbf{y}_i^{p,t} \log \mathbf{x}_i^{tp,t}. \tag{31}$$

By considering both MSE loss and CE loss, SHARE-X aims to jointly minimize the following objective

$$O = O_1 + \beta(O_2 + O_3), \tag{32}$$

where β is the hyper-parameter that controls the importance of two CE losses.

9 Complexity Analysis

In this section, we first analyze the computational complexity of SHARE, then discuss the overall complexity of SHARE-X.

Complexity of SHARE. In each prediction, the computational cost of SHARE comes from both the spatial and temporal autocorrelation modeling modules. Specifically, the spatial autocorrelation modeling module consists of three blocks: CxtConv, PropConv and SCConv. For CxtConv, the complexity at each time step is

$$\mathcal{T}_{CxtConv} = \mathcal{O}(l(|V|F^2 + |E^C|F)), \tag{33}$$

where F and l are the number of features and stacked CxtConv layers [18]. $|V|$ and $|E^C|$ denote the number of nodes and edges in the contextual graph. Similarly, the complexity of PropConv is

$$\mathcal{T}_{PropConv} = \mathcal{O}(|V|F^2 + |E^P|F). \tag{34}$$

Then, we compute the complexity of SCConv by

$$\mathcal{T}_{SCConv} = \mathcal{O}(|E|K + |V|K^2 + |V|KF), \tag{35}$$

where K is the number of latent nodes, $|E| = |E^C \cup E^P|$ is the number of connected edges among parking lots. For the temporal autocorrelation modeling module, the complexity of the GRU block [37] at each time step is

$$\mathcal{T}_{GRU} = \mathcal{O}(|V|F^2). \tag{36}$$

The total complexity of SHARE is the combination of three spatial blocks and the GRU block,

$$\begin{aligned}\mathcal{T}_{SHARE} = \mathcal{O}(T_s((l|E^C| + |E^P|)F + l|V|F^2 \\ + |E|K + |V|K^2 + |V|KF)),\end{aligned} \tag{37}$$

where T_s is the short-term input time step length.

Complexity of SHARE-X. Compared with SHARE, the increased computational cost of SHARE-X comes from two optimized blocks, *i.e.*, the MrscConv block and the HARN module. On one hand, as the generalized version of SCConv, the complexity of MrscConv is

$$\begin{aligned}\mathcal{T}_{MrscConv} = \mathcal{O}(\sum_{f=1}^{F}(|E_f|K_f + K_{f-1}K_f^2 + |V|K_fF \\ + |V|K_{f-1}K_fI(f > 1))),\end{aligned} \tag{38}$$

where F is the total number of SCConv layers, K_f and $|E_f|$ are the number of latent nodes and connected edges in layer f, $I(f > 1)$ equals one if condition $f > 1$ is satisfied, and zero otherwise. In practice, as setting fewer latent nodes in higher levels is effective to capture the high-level correlations ($K_f = 0.1K_{f-1}$ in our model), the complexity of MrscConv is on par with SCConv. On the other hand, since the representation learning in different days can be executed in parallel, the complexity of HARN can be written as,

$$\mathcal{T}_{HARN} = \mathcal{O}(max(T_s, T_d)|V|F^2 + D|V|F^2), \tag{39}$$

where T_d is the input time steps length in each previous day (*i.e.*, $T_d = \tau + 2s$) and D is the number of previous days. Overall, the complexity of SHARE-X is,

$$\begin{aligned}\mathcal{T}_{SHARE-X} = \mathcal{O}(max(T_s, T_d)((l|E^C| + |E^P|)F + l|V|F^2 \\ + |E|K_1 + |V|K_1^2 + |V|K_1F) + D|V|F^2).\end{aligned} \tag{40}$$

When $T_s \geq T_d$, the increased complexity primarily depends on the GRU_{long} block (*i.e.*, $\mathcal{O}(D|V|F^2)$). In this work, we set $D = 3$ to reduce the computational overhead induced by the GRU_{long} block. However, when $T_s < T_d$, the increased complexity from the representation learning of previous day is critical. How to incorporate larger T_d more efficiently is an interesting problem, and is left as future work.

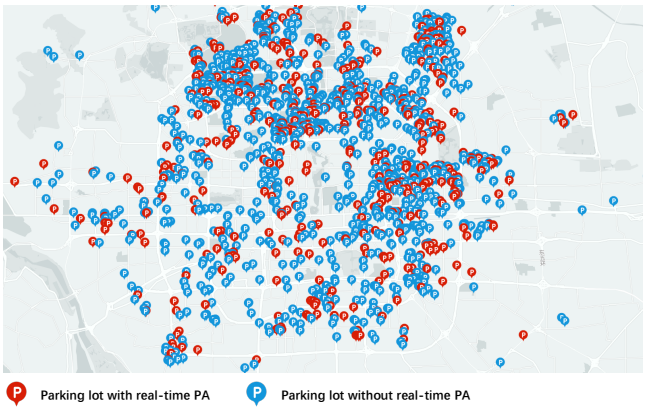


Fig. 5. Spatial distribution of parking lots in BEIJING.

10 Experiments

10.1 Experimental setup

10.1.1 Data description

We use two real-world datasets collected from BEIJING and SHENZHEN, two metropolises in China. Both datasets range from April 20, 2019, to May 20, 2019. All PA records are crawled every 15 minutes from a publicly accessible app, in which all parking occupancy information is collected by real-time sensors. POI and check-in data are collected through Baidu Maps Place API and location SDK [38], [39], [40]. Similar to [4], we associate POI distribution [41], [42] to each parking lot and aggregate the number of check-in records in the past 15 minutes nearby each parking lot to derive the population data. In this work, we regard POI distribution as static features, while the population is dynamic across different time steps. We chronologically order the above data, take the first 60% as the training set, the following 20% for validation, and the rest as the test set. In each dataset, 70% of parking lots are masked as unlabeled. The spatial distribution of parking lots in BEIJING in our datasets are shown in Figure 5. The statistics of the datasets are summarized in Table 1.

TABLE 1
Statistics of datasets.

Description	BEIJING	SHENZHEN
# of parking lots	1,965	1,360
# of PA records	5,847,840	4,047,360
Average # of parking spots	210.24	185.36
# of check-ins	9,436,362,579	3,680,063,509
# of POIs	669,058	250,275
# of POI categories	197	188

10.1.2 Implementation details

All experiments are performed on a Linux server with 26-core Intel(R) Xeon(R) Gold 5117 CPU @ 2.00 GHz and NVIDIA Tesla P40 GPU. Following previous work [22], [43], the PA is normalized before input and scaled back to absolute PA in output. We choose $T_s = 12$, $\tau = 3$ and select $D = 3$, $s = 2$ for prediction, and set $\epsilon = 1$km and $k = 10$ to connect parking lots. The dimensions of $\mathbf{x}^{c'}$, $\mathbf{x}^{sc}$ and $\mathbf{h}^t$ are fixed to 32, 64 and 64, respectively, q is fixed to 50. The number of CxtConv layers is 2. We use dot-product attention in CxtConv and PropConv, use general attention [44] (*i.e.*, $Attn(\mathbf{h}^i, \mathbf{h}^j) = \mathbf{h}^i \mathbf{W} \mathbf{h}^j$) in HARN. In MrscConv, the ratio of latent nodes is set to 0.1 (*i.e.*, $K_f = 0.1 K_{f-1}$) and F is set to 2. The activation functions in CxtConv and MrscConv are LeakyReLU ($\alpha = 0.2$), and Sigmoid in other places. We employ Adam optimizer to train our model, fix the learning rate to 0.001, and set β to 0.5. For neural network based baselines, we also employ Adam optimizer for training but with tuned learning rate, and early stop training if the loss doesn't decrease lower on validation set over 30 epochs.

10.1.3 Evaluation metrics

We adopt *Mean Absolute Error* (MAE) and *Rooted Mean Square Error* (RMSE), two widely used metrics [15] for evaluation.

10.1.4 Baselines

We compare our full approach with the following eight baselines and a basic variant of SHARE-X. For a fair comparison, the short-term inputs of all these algorithms are identical (*i.e.*, features in previous T_s historical steps). We carefully tuned major hyper-parameters of each baseline via grid search based on their recommended settings.

- **LR** uses logistic regression for parking availability prediction. We concatenate previous T_s steps historical features as the input and predict each parking lot separately. The learning rate is set to $5\mathrm{e}{-4}$.
- **GBRT** is a variant of boosting tree for regression tasks. It is widely used in practice and performs well in many data mining challenges. We use the version in XGboost [45]. We concatenate input features as done in LR, set learning rate to 0.3, maximal tree depth and minimal child weight to 3.
- **GRU** [34] predicts the PA of each parking lot without considering spatial dependency. We train two GRUs for P_l and P_u separately. All hyper-parameters are the same as the GRU's setting in SHARE-X.
- **Google-Parking** [3] is the parking difficulty prediction model deployed on Google Maps. It uses a feed-forward deep neural network for prediction. We concatenate input features as done in LR. The number of hidden layers is set to 3, with $5\mathrm{e}{-4}$ learning rate.
- **Du-Parking** [4] is the parking availability estimation model used on Baidu Maps. It fuses several LSTMs to capture various temporal dependencies. The period sequences length is 3, with 0.001 learning rate.
- **STGCN** [43] is a graph neural network model for traffic forecasting. It models both spatial and temporal dependency with convolution structures. The input graph is constructed as described in the original paper but keeps the same graph connectivity with our CxtConv. Both the graph and temporal convolution kernels size are set to 3, the learning rate is fixed to 0.001.
- **DCRNN** [22] is another graph convolution network based model, which models spatial and temporal dependency by integrating graph convolution and GRU. The input graph and learning rate are the same as STGCN. The number of recurrent layers of GRU and the graph convolution diffusion step are set to 2, scheduled sampling probability in decoder is set to 0.5, and learning rate is 0.001.
- **GMAN** [26] is a state-of-the-art graph network model for traffic forecasting. It incorporates spatio-temporal attention block into both encoder and decoder to model spatial and temporal correlations. We use two ST-Attention blocks, set both the number of attention heads and their dimensions to 8, and learning rate is 0.001.
- **SHARE** [17] is a basic variant of SHARE-X, including CxtConv block, SCConv block, and PA approximation module. The hyper-parameters are the same as described in the original paper.

10.2 Overall performance

Table 2 reports the overall results of our method and all the compared baselines on two datasets with respect to MAE and RMSE. We run all methods 5 times with different random seeds and report the mean and standard deviation (SD) of the results. As can be seen, our model together with its variant outperform all other baselines using both metrics, which demonstrates the advance of SHARE-X. Specifically, SHARE-X achieves $(28.4\%, 28.3\%, 27.4\%)$ and $(29.0\%, 28.2\%, 28.5\%)$ improvements beyond the state-of-the-art approach (GMAN) on MAE and RMSE on BEIJING for $(15min, 30min, 45min)$ prediction, respectively. Similarity, the improvement of *MAE* and *RMSE* on SHENZHEN are $(25.5\%, 26.5\%, 27.1\%)$ and $(20.6\%, 20.3\%, 20.1\%)$. We also observe significant improvement by comparing SHARE-X with its variant SHARE. By taking advantage of both MrscConv and HARN, SHARE-X achieves $(10.3\%, 9.9\%, 11.3\%)$ and $(7.1\%, 6.9\%, 7.8\%)$ improvements beyond SHARE on MAE and RMSE on BEIJING and the improvement on SHENZHEN is consistent. Besides, we conduct Welch's t-test and all the p-values between our model and each baseline are smaller than 0.01, indicating the statistical significance of improvements. All of the above results demonstrate the effectiveness of our model.

Looking further into the results, we observe all graph-based models (*i.e.*, STGCN, DCRNN, GMAN, SHARE

TABLE 2
Parking availability prediction error given by *MAE* and *RMSE* on BEIJING and SHENZHEN. All the improvements are statistically significant according to Welch's t-test at level 0.01 by comparing SHARE-X to other baselines.

Algorithm	BEIJING (15/ 30/ 45 min/ SD)		SHENZHEN (15/ 30/ 45 min/ SD)	
	MAE	RMSE	MAE	RMSE
LR	29.89 / 30.29 / 30.69 / 0.05	69.83 / 71.01 / 72.21 / 0.16	24.48 / 24.76 / 25.16 / 0.04	50.95 / 52.00 / 52.98 / 0.24
GBRT	17.34 / 17.94 / 18.46 / 0.08	45.01 / 48.83 / 51.74 / 0.29	14.01 / 14.58 / 14.84 / 0.05	35.60 / 38.23 / 38.34 / 0.35
GRU	18.49 / 18.77 / 19.39 / 0.19	53.69 / 54.43 / 56.64 / 1.91	16.52 / 16.74 / 17.17 / 0.34	46.89 / 47.11 / 47.31 / 0.69
Google-Parking	21.11 / 21.38 / 22.39 / 0.55	57.58 / 59.12 / 60.24 / 0.46	16.84 / 17.77 / 18.36 / 0.38	47.73 / 48.52 / 49.25 / 0.13
Du-Parking	17.60 / 17.67 / 17.97 / 0.05	51.02 / 51.52 / 52.76 / 0.92	13.99 / 14.23 / 14.48 / 0.06	42.37 / 42.94 / 43.54 / 0.40
STGCN	15.93 / 16.01 / 16.57 / 0.55	49.14 / 49.39 / 50.24 / 1.26	13.48 / 13.83 / 14.07 / 0.41	39.07 / 40.02 / 40.37 / 1.47
DCRNN	15.64 / 15.82 / 16.03 / 0.15	47.24 / 48.01 / 49.04 / 0.45	13.14 / 13.26 / 13.88 / 0.14	42.75 / 43.43 / 44.39 / 0.14
GMAN	13.43 / 13.72 / 14.01 / 0.15	41.93 / 42.36 / 43.69 / 0.92	11.58 / 11.82 / 12.26 / 0.13	36.49 / 36.94 / 37.89 / 0.64
SHARE	10.72 / 10.92 / 11.46 / 0.17	32.01 / 32.65 / 33.89 / 0.63	9.26 / 9.42 / 9.63 / 0.12	30.56 / 30.94 / 31.76 / 0.29
SHARE-X (ours)	**9.62 / 9.84 / 10.17** / 0.15	**29.75 / 30.40 / 31.23** / 0.24	**8.63 / 8.69 / 8.94** / 0.09	**28.97 / 29.44 / 30.27** / 0.17

and SHARE-X) outperform other deep learning based approaches (*i.e.*, Google-Parking and Du-parking), which consistently reveals the advantage of incorporating spatial dependency for parking availability prediction. Among these baselines, GMAN achieves the best performances, for it can model dynamic spatial correlations among parking lots, and capture the diversified correlations of different time steps. Remarkably, GBRT outperforms Google-parking, GRU, LR, and achieves a similar result with Du-parking, which validates our exception that GBRT is a simple but effective approach for regression tasks. One extra interesting finding is that both MAE and RMSE of all methods on SHENZHEN is relatively smaller than on BEIJING. This is possible because the parking lots are denser and more evenly distributed in SHENZHEN. Therefore they are easier to predict.

10.3 Ablation study

Next we conduct ablation studies on SHARE-X to further verify the effectiveness of each component. Due to the page limit, we only report the results on BEIJING using MAE.

10.3.1 Effect of spatial dependency modeling

We first investigate the effectiveness of spatial blocks. Specifically, we evaluate three variants of SHARE-X, (1) *noCxt* excludes the CxtConv block, (2) *noMrsc* excludes the MrscConv block, and (3) *noProp* excludes the PropConv block. The ablation results are reported in Figure 6(a). As can be seen, compared with the overall SHARE-X for $(15min, 30min, 45min)$ prediction, removing CxtConv block, MrscConv block and PropConv yield $(22.5\%, 24.8\%, 27.0\%)$, $(17.3\%, 16.5\%, 16.9\%)$, $(22.3\%, 21.0\%, 18.0\%)$ performance degradation, respectively. The above results verify the effectiveness of three spatial blocks on capturing spatial dependencies. Moreover, we notice the CxtConv block and PropConv block have a greater impact on performance, as they are used to incorporate environmental contextual data and sparse real-time parking availability data for parking availability prediction.

10.3.2 Effect of temporal dependency modeling

To examine the importance of each temporal component, we evaluate two variants: (1) *noHARN* removes the hierarchical attentive recurrent network module, and (2) *noHA* removes the hierarchical attentive structure in HARN. As reported in Figure 6(b), there is $(4.1\%, 5.4\%, 5.4\%)$ performance degradation compared with SHARE-X when we remove HARN, and there is $(2.7\%, 3.0\%, 3.0\%)$ performance degradation if we only remove the hierarchical attentive structure. The above results demonstrate the effectiveness of integrating long-term temporal dependencies and diversified importance of previous time steps in PA prediction.

10.3.3 Effect of PA approximation

To verify the effectiveness of PA approximation, we consider three variants of SHARE-X: (1) *noApx* removes the whole PA approximation module, (2) *noProp* only removes the spatial PA approximation operation, and (3) *noTeA* only removes the temporal PA approximation operation. As reported in Figure 6(c), there is dramatic $(28.6\%, 26.0\%, 24.1\%)$ performance degradation compared to SHARE-X when we remove the whole PA approximation module, demonstrating the importance of PA for prediction. Besides, we also notice that the performance degradation of noProp is much more serious than noTeA, indicating that the PA approximation from spatial domain is more critical.

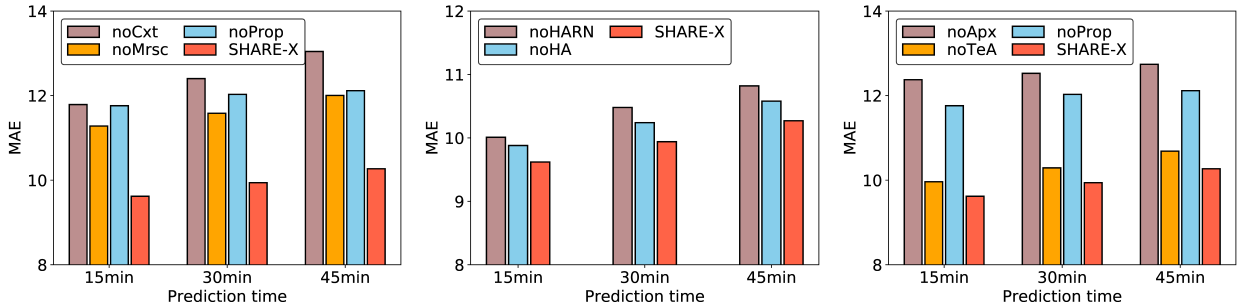
10.4 Parameter sensitivity

Here we report the hyper-parameters sensitivity, including the impact of the ratio of labeled parking lots (*i.e.*, $|P_l|/N$), the number of stacked layers in CxtConv, the effect of distance threshold ϵ, the effect of top-k nearest parking lots, the effect of the importance of CE losses β, the number of MrscConv layers F, the number of daily input length D, and the effect of additional s time slots. As shown in Figure 7, all experiments are performed on SHARE-X, except the ratio of labeled parking lot which is tested on all algorithms in Table 2. Each time we vary a hyper-parameter, set others to their default values. We evaluate the performance of different hyper-parameters using MAE on BEIJING.

We first vary the ratio of the labeled parking lots from 0.06 to 0.9. We perform this evaluation for our model and all baselines, as reported in Figure 7(a). The results are unsurprising: equipping more real-time sensors in parking lots leads to more accurate PA prediction. However, equipping more sensors leads to extra economic costs and may be constrained by each parking lot's policies. In addition, we also observe our model has consistent improvement against other baselines under different ratios. The improvement is especially remarkable when the labeled parking lots are less than 0.5. The above results demonstrate the effectiveness of our model for parking availability prediction even with extreme data scarcity.

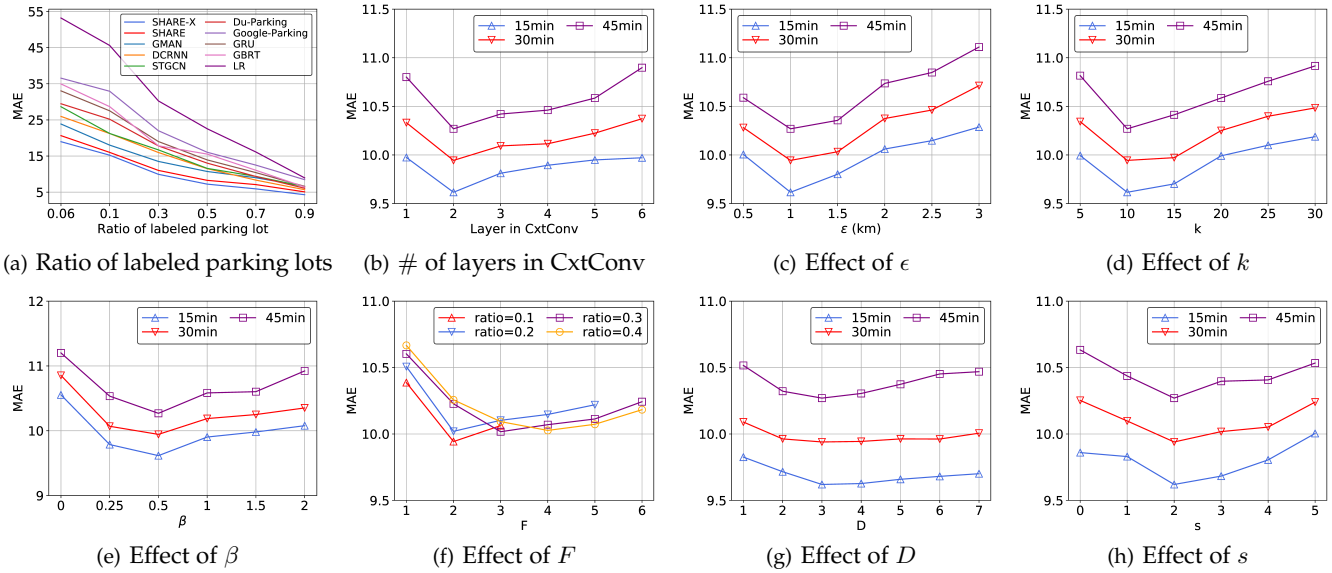
Then we vary the number of stacked CxtConv layers (defined in Eq. (5)) from 1 to 6. The results are reported in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3034140, IEEE Transactions on Knowledge and Data Engineering



(a) Effect of spatial modeling (b) Effect of temporal modeling (c) Effect of PA approximation

Fig. 6. Ablation study on BEIJING.



(a) Ratio of labeled parking lots (b) # of layers in CxtConv (c) Effect of ϵ (d) Effect of k

(e) Effect of β (f) Effect of F (g) Effect of D (h) Effect of s

Fig. 7. Parameter sensitivity on BEIJING.

Figure 7(b). As can be seen, by setting two stacked layers in CxtConv achieves the best performance. Further decrease or increase stacked layers lead to performance degradation. This is because too few layers cannot aggregate sufficient information, whereas too many layers make the model lose discriminative power.

Besides, we vary the distance threshold ϵ (defined in Eq. (2)) from 0.5 to 3. The results are reported in Figure 7(c). SHARE-X achieves the best performance when $\epsilon = 1$km. This makes sense since too few neighbors limit the information propagation, whereas too many neighbors introduce extra noises in the information propagation process. Furthermore, we vary the parameter k (defined in Eq. (25)) when connecting parking lots from 5 to 30. The results are reported in Figure 7(d). Overall, SHARE-X achieves the best performance when $k = 10$. We observe consistent performance degradation when decrease or increase k. The reason is similar to the effect of ϵ.

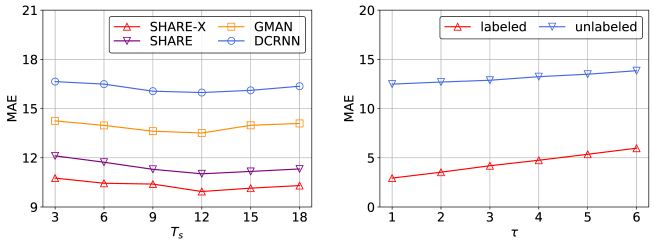
Figure 7(e) reports the effect of CE losses β (defined in Eq. (32)) on BEIJING. We can make the following observations. (1) Adding CE loss can remarkably improve the prediction performance. (2) SHARE-X achieves the best performance when β is 0.5, and the performance degrades when we further decrease or increase β. Based on the above observations, we choose $\beta = 0.5$ in the overall experiment.

Furthermore, We vary the number of MrscConv layers F (defined in Section 5.2.2) from 1 to 6 with different ratios of latent nodes vary from 0.1 to 0.4. The results are reported in Figure 7(f) (missing values indicate no latent nodes in the corresponding layer). As can be seen, SHARE-X achieves better performance on average (*i.e.*, mean MAE of 15, 30 and 45min) when F is greater than one, which demonstrates the effectiveness of MrscConv to model global spatial correlations in diverse resolutions. We also note that when increase the latent node ratio from 0.1 to 0.4, the best performances for F are 2, 2, 3 and 4, respectively. This is because increasing the ratio of latent nodes requires to stack more SCConv layers to obtain the same expressive power for regularization.

After that, we test the impact of long-term daily periodicity input length D (defined in Eq. (20)) on SHARE-X. The results are reported in Figure 7(g). We observe SHARE-X achieves the best performance when $D = 3$, and the errors grow when we decrease or increase D. One possible reason is that a short input cannot provide sufficient daily periodicity information, whereas too long input introduces more noises for long-term temporal periodicity modeling and also leads to hard convergence for model training.

Finally, we vary the additional time slots parameter s (defined in Eq. (18)). The results are reported in Figure 7(h). As can be seen, SHARE-X achieves the best

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3034140, IEEE Transactions on Knowledge and Data Engineering



(a) Effect of T_s (b) Effect of τ

Fig. 8. Temporal robustness study on BEIJING.

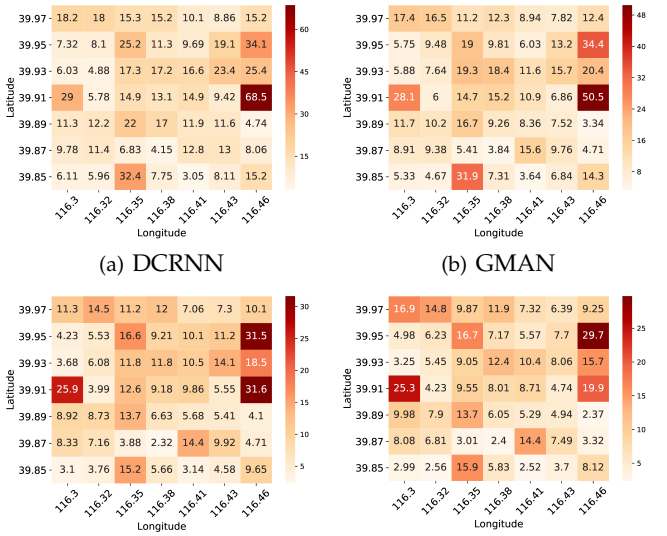
performance when $s = 2$, and has a notable performance improvement compared with $s = 0$. This reveals that considering temporal shifting is useful for PA prediction. We observe consistent performance degradation by further increase s from 2 to 5. This is perhaps because too long additional time steps introduce more noises for periodic temporal dependency modeling.

10.5 Effectiveness on different time steps

In this section, we study the effect of different input and output time step length. First, we test the impact of short-term input time step length T_s (defined in Eq. (16)) on SHARE-X and SHARE as well as two competitive baselines (*i.e.*, GMAN and DCRNN). The results are reported in Figure 8(a). We observe all methods consistently achieve the best performance when $T_s = 12$ (consider the previous 3 hours as input), and the errors grow both when we decrease or increase T_s. One possible reason is that an excessively short input cannot provide sufficient temporal correlated information, whereas too long input introduces more noises for temporal dependency modeling and also leads to hard training. Furthermore, to study the impact of the prediction step, we vary τ (defined in Eq. (23)) from 1 (predict future 15 minutes) to 6 (predict future 90 minutes) on SHARE-X. The results are reported in Figure 8(b). We separate the result of labeled and unlabeled parking lots. Overall, labeled parking lots are much easier to predict. Besides, by increasing τ, the errors of all parking lots increases consistently. However, we can observe the errors of labeled parking lots are increasing faster, because the temporal dependency between observed PA and future PA becomes lower when τ goes large.

10.6 Effectiveness on different regions

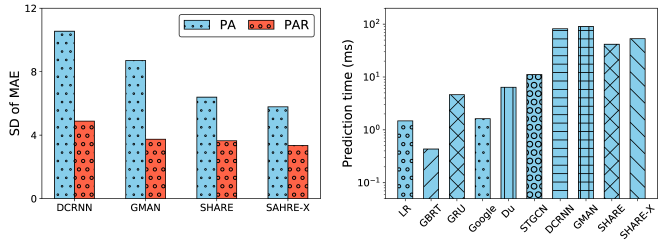
To evaluate the performance of algorithms on different regions, we partition BEIJING into a set of disjoint grids based on longitude and latitude, test the performance of SHARE-X and SHARE as well as two competitive baselines (*i.e.*, GMAN and DCRNN) on each region. Figure 9 shows the averaged MAE of algorithms on each region in BEIJING. Overall, the MAE of SHARE-X is smaller than other algorithms in most regions, and Figure 10(a) shows the standard deviation (SD) of regions MAE of SHARE-X is also smaller than other algorithms, which demonstrates the spatial robustness of our model. The MAE in each region is even except for several outliers. We find that these algorithms' performances are highly correlated with the number of parking spots of each parking lot. Because if we calculate the averaged MAE of Parking Availability



(a) DCRNN (b) GMAN

(c) SHARE (d) SHARE-X

Fig. 9. Spatial robustness study on MAE on BEIJING.



(a) SD of regions MAE (b) Efficiency of algorithms

Fig. 10. Spatial robustness study and efficiency analysis on BEIJING.

Ratio (PAR, PAR = PA/# of parking spots) on each region and then scale them to the same mean value as the averaged MAE of PA, shown in Figure 10(a), the SD of regions MAE decreases for all algorithms. This is possible because for the same fluctuation on PAR, the parking lot with a larger number of parking spots will have larger error on PA. This result indicates further optimization can be applied to these large parking lots to improve the overall performance.

10.7 Efficiency analysis

Finally, we evaluate the efficiency of each model. We report the averaged prediction time of all time steps in the test set on BEIJING. The results of each baseline and our model are reported in Figure 10(b). As can be seen, deep learning models take longer time than statistical learning models (*e.g.*, GBRT). Besides, all graph-based models (including SHARE-X and SHARE) take a longer time than other deep models. Note that our models achieve significant latency reduction compared with DCRNN and GMAN. SHARE-X takes 53.1ms, and SHARE takes 41.6ms (to predict all parking lots once), which are over 35% faster than DCRNN (81.8ms) and GMAN (90.2ms). The performance gain is mainly because our models simplify repetitive graph convolution layers in DCRNN and global attention operations in GMAN, without sacrificing prediction accuracy. Moreover, we observe SHARE-X (53.1ms) is only slightly slower than SHARE (41.6ms), demonstrating the computational efficiency of the additional MrscConv and HARN

module. The above results also validate our theoretical analysis in Section 9.

11 Conclusion

In this paper, we presented SHARE-X, a city-wide parking availability prediction framework based on both environmental contextual data and partially observed real-time parking availability data. We first proposed a hierarchical graph convolution module to capture both local and global spatial dependencies. Then, we adopted a hierarchical attentive recurrent network module to capture dynamic short and long-term temporal dependencies of each parking lot. Besides, a parking availability approximation module was proposed for parking lots without real-time parking availability information. Extensive experimental results on two real-world datasets showed that the performance of SHARE-X for parking availability prediction significantly outperformed eight state-of-the-art baselines.

Acknowledgments

This research is supported in part by grants from the National Natural Science Foundation of China (Grant No.91746301, 71531001).

References

[1] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[2] D. C. Shoup, "Cruising for parking," *Transport Policy*, vol. 13, no. 6, pp. 479–486, 2006.

[3] N. Arora, J. Cook, R. Kumar, I. Kuznetsov, Y. Li, H.-J. Liang, A. Miller, A. Tomkins, I. Tsogsuren, and Y. Wang, "Hard to park?: Estimating parking difficulty at scale," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 2296–2304.

[4] Y. Rong, Z. Xu, R. Yan, and X. Ma, "Du-parking: Spatio-temporal big data tells you realtime parking availability," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 646–654.

[5] H. Zhu, E. Chen, H. Xiong, H. Cao, and J. Tian, "Mobile app classification with enriched contextual information," *IEEE Transactions on mobile computing*, vol. 13, no. 7, pp. 1550–1563, 2013.

[6] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, "Parknet: drive-by sensing of road-side parking statistics," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, 2010, pp. 123–136.

[7] R. Fusek, K. Mozdřeň, M. Šurkala, and E. Sojka, "Adaboost for parking lot occupation detection," in *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, 2013, pp. 681–690.

[8] J. Zhou and A. K. Tung, "Smiler: A semi-lazy time series prediction system for sensors," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015, pp. 1871–1886.

[9] P. Wang, Y. Fu, G. Liu, W. Hu, and C. Aggarwal, "Human mobility synchronization and trip purpose detection with mixture of hawkes processes," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 495–503.

[10] Y. Liu, C. Liu, X. Lu, M. Teng, H. Zhu, and H. Xiong, "Point-of-interest demand modeling with human mobility patterns," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 947–955.

[11] Z. Jiang, "A survey on spatial prediction methods," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1645–1664, 2018.

[12] T. Rajabioun and P. A. Ioannou, "On-street and off-street parking availability prediction using multivariate spatiotemporal models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2913–2924, 2015.

[13] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 468–478, 2019.

[14] J. Gu, Q. Zhou, J. Yang, Y. Liu, F. Zhuang, Y. Zhao, and H. Xiong, "Exploiting interpretable patterns for flow prediction in dockless bike sharing systems," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[15] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "Geoman: Multi-level attention networks for geo-sensory time series prediction." in *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI*, 2018, pp. 3428–3434.

[16] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019, pp. 5668–5675.

[17] W. Zhang, H. Liu, Y. Liu, J. Zhou, and H. Xiong, "Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020, pp. 1186–1193.

[18] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations, ICLR*, 2018.

[19] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[20] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[21] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Advances in Neural Information Processing Systems*, 2018, pp. 4800–4810.

[22] 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.

[23] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019, pp. 922–929.

[24] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[25] 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 Transactions on Intelligent Transportation Systems*, 2019.

[26] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020, pp. 1234–1241.

[27] 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 *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019, pp. 3656–3663.

[28] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, "Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 1227–1235.

[29] T. Liu, W. Wu, Y. Zhu, and W. Tong, "Predicting taxi demands via an attention-based convolutional recurrent neural network," *Knowledge-Based Systems*, p. 106294, 2020.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR*, 2017.

[31] Z. Yuan, H. Liu, Y. Liu, D. Zhang, F. Yi, N. Zhu, and H. Xiong, "Spatio-temporal dual graph attention network for query-poi matching," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 629–638.

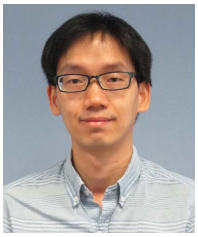
[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 6000–6010.

[33] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018, pp. 3538–3545.

[34] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.
[35] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (indrnn): Building a longer and deeper rnn," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5457–5466.
[36] H.-P. Hsieh, S.-D. Lin, and Y. Zheng, "Inferring air quality for station location recommendation based on urban big data," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 437–446.
[37] R. Rana, "Gated recurrent unit (gru) for emotion classification from noisy speech," *arXiv preprint arXiv:1612.07778*, 2016.
[38] H. Liu, T. Li, R. Hu, Y. Fu, J. Gu, and H. Xiong, "Joint representation learning for multi-modal transportation recommendation," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019, pp. 1036–1043.
[39] S. Li, J. Zhou, T. Xu, H. Liu, X. Lu, and H. Xiong, "Competitive analysis for points of interest," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020, pp. 1265–1274.
[40] H. Liu, Y. Li, Y. Fu, H. Mei, J. Zhou, X. Ma, and H. Xiong, "Polestar: An intelligent, efficient and national-wide public transportation routing engine," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020, pp. 2321–2329.
[41] H. Liu, Y. Tong, P. Zhang, X. Lu, J. Duan, and H. Xiong, "Hydra: A personalized and context-aware multi-modal transportation recommendation system," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 2314–2324.
[42] H. Liu, Y. Tong, J. Han, P. Zhang, X. Lu, and H. Xiong, "Incorporating multi-source urban data for personalized and context-aware multi-modal transportation recommendation," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020.
[43] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI*, 2018.
[44] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015, pp. 1412–1421.
[45] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

Weijia Zhang is currently a master student in School of Computer Science, University of Science and Technology of China (USTC), Hefei, China. He is working in Anhui Province Key Laboratory of Big Data Analysis and Application, USTC. His research interests include the spatio-temporal data mining and urban computing.

Hao Liu received the PhD degree from the Hong Kong University of Science and Technology (HKUST), in 2017 and the BSc degree from the South China University of Technology (SCUT), in 2012. He is currently working as a research scientist at the Business Intelligence Lab, Baidu Research. His research interests include the areas of spatio-temporal data mining and large-scale data management. He is a member of the IEEE.



Yanchi Liu received the PhD in Information Technology from Rutgers, the State University of New Jersey and the PhD in Management Science from the University of Science and Technology Beijing. His research interests include data mining, artificial intelligence, business intelligence, and recommender systems. He has published prolifically in top conferences/journals in the data mining and artificial intelligence communities, such as KDD, IJCAI, WWW, TCYB.

Jingbo Zhou is a staff research scientist at Business Intelligent Lab of Baidu Research, working on machine learning problems for both scientific research and business applications, with a focus on spatio-temporal data mining, user behavior study and knowledge graphs. He obtained his Ph.D. degree from National University of Singapore in 2014, and B.E. degree from Shandong University in 2009. He has published several papers in top venues, such as SIGMOD, KDD, VLDB, ICDE, TKDE and AAAI.

Tong Xu (M'17) received the Ph.D. degree in University of Science and Technology of China (USTC), Hefei, China, in 2016. He is currently working as an Associate Professor of the Anhui Province Key Laboratory of Big Data Analysis and Application, USTC. He has authored more than 60 journal and conference papers in the fields of social network and social media analysis, including IEEE TKDE, IEEE TMC, IEEE TMM, KDD, AAAI, ICDM, etc.

Hui Xiong is currently a Full Professor at Rutgers University, where he received the 2018 Ram Charan Management Practice Award as the Grand Prix winner from the Harvard Business Review, RBS Deans Research Professorship (2016), the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence (2009), the IEEE ICDM Best Research Paper Award (2011), and the IEEE ICDM Outstanding Service Award (2017). He received the Ph.D. degree in Computer Science from the University of Minnesota-Twin Cities, USA, in 2005. He is a co-Editor-in-Chief of Encyclopedia of GIS, an Associate Editor of IEEE TBD, ACM TKDD, and ACM TMIS. He has served regularly on the organization committees of numerous conferences, such as a Program Co-Chair for ACM KDD 2018 (research track), ACM KDD 2012 (industry track), IEEE ICDM 2013, and a General Co-Chair for IEEE ICDM 2015. He is an IEEE Fellow and an ACM Distinguished Scientist.