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

Weijia Zhang[1*], **Hao Liu**[2*†], **Yanchi Liu**[3], **Jingbo Zhou**[2], **Hui Xiong**[2†]

[1]University of Science and Technology of China, Hefei, China, [2]Business Intelligence Lab, Baidu Research,
National Engineering Laboratory of Deep Learning Technology and Application, Beijing, China, [3]Rutgers University, USA
wjzhang3@mail.ustc.edu.cn, {liuhao30, zhoujingbo}@baidu.com, yanchi.liu@rutgers.edu, xionghui@gmail.com

Abstract

The ability to predict city-wide parking availability is crucial for the successful development of Parking Guidance and Information (PGI) systems. Indeed, the effective prediction of city-wide parking availability can improve parking efficiency, help urban planning, and ultimately alleviate city congestion. However, it is a non-trivial task for predicting city-wide parking availability 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 GPS). To this end, we propose *Semi-supervised Hierarchical Recurrent Graph Neural Network* (SHARE) for predicting city-wide parking availability. Specifically, we first propose a hierarchical graph convolution structure to model non-Euclidean spatial autocorrelation among parking lots. Along this line, a contextual graph convolution block and a soft clustering graph convolution block are respectively proposed to capture local and global spatial dependencies between parking lots. Additionally, we adopt a recurrent neural network to incorporate dynamic temporal dependencies of parking lots. Moreover, we propose a parking availability approximation module to estimate missing real-time parking availabilities from both spatial and temporal domain. Finally, experiments on two real-world datasets demonstrate the prediction performance of SHARE outperforms seven state-of-the-art baselines.

Introduction

In recent years, we have witnessed significant development of Intelligent Transportation Systems (ITS) (Zhang et al. 2011). Parking guidance and information (PGI) systems, especially parking availability prediction, is an indispensable component of ITS. According to a survey by the International Parking Institute (IPI)[1], over 30% cars on the road are searching for parking, and these cruising cars contribute up to 40% traffic jams in urban areas (Shoup 2006). Thus, city-wide parking availability prediction is of great importance

to help drivers efficiently find parking, 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 (Arora et al. 2019), and Baidu Maps estimates real-time city-wide parking availability based on environmental contextual features (*e.g.*, Point of Interest (POI), map queries, *etc.*) (Rong et al. 2018). The above mentions make city-wide parking availability prediction based on biased and indirect input signals (*e.g.*, user's feedback are 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 GPS become ubiquitous, which can significantly improve the prediction accuracy of parking availability (Mathur et al. 2010; Fusek et al. 2013; Zhou and Tung 2015). However, for economic and privacy concerns, it is difficult to be scaled up to cover all parking lots of a city.

In this paper, we propose to *simultaneously* predict the availability of each parking lot of 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 a city-scale. However, it is a non-trivial task faced with the following three major challenges. (1) *Spatial autocorrelation*. The availability of a parking lot is not only effected by the occupancy of nearby parking lots but may also synchronize with distant parking lots (Wang et al. 2017; Liu et al. 2017). 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 (Rajabioun and Ioannou 2015). Besides, the spatial autocorrelation between parking lots may also vary over time (Liang et al. 2018; Yao et al. 2019). How to model dynamic temporal autocorrelation of each parking lot is another challenge. (3) *Parking availability scarcity*. Only a small portion of parking lots

---

*Equal contribution.

†Corresponding author.

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

are equipped with real-time sensors. According to one of the largest map service application, 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 and incomplete real-time parking availability information.

To tackle above challenges, in this paper, we present *Semi-supervised Hierarchical Recurrent Graph Neural Network* (SHARE) for city-wide parking availability prediction. Our major contributions are summarized as follows:

- We propose a semi-supervised spatio-temporal learning framework to incorporate both environmental contextual factors and sparse real-time parking availability data for city-wide parking availability prediction.

- We 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.

- 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 domain, then fuse them through an entropy-based mechanism.

- 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.

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, GPS, *etc.*), respectively. Let $\mathbf{X}^t = \{\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_N^t\} \in \mathcal{R}^{N \times M}$ denote observed M dimensional contextual feature vectors (*e.g.*, POI distribution, population, *etc.*) for all parking lots in P at time t. We begin the formal 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 spot 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 problem. Given historical time window T, contextual features for all parking lots $\boldsymbol{\mathcal{X}} = (\mathbf{X}^{t-T+1}, \mathbf{X}^{t-T+2}, \ldots, \mathbf{X}^t)$, 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,*

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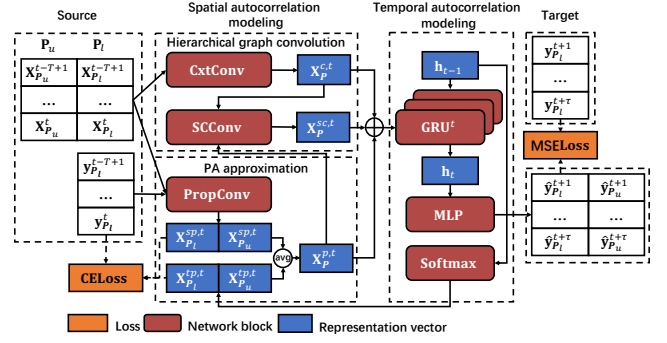

Figure 1: The framework overview of SHARE.

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.

Framework overview

The architecture of SHARE is shown in Figure 1, where the inputs are contextual features as well as partially observed real-time PAs, and the output are the predicted PAs of all parking lots in next τ time steps. There are three major components in SHARE. First, the *Hierarchical graph convolution* module models spatial autocorrelations among parking lots, where the *Contextual Graph Convolution* (CxtConv) block captures local spatial dependencies between parking lots through rich contextual features (*e.g.*, POI distribution, regional population, *etc.*), while the *Soft Clustering Graph Convolution* (SCConv) block captures global correlations among distant parking lots by softly assigning each parking lot to a set of latent cluster nodes. Second, the temporal autocorrelation modeling module employs the *Gated Recurrent Unit (GRU)* to model dynamic temporal dependencies of each parking lot. Third, the *PA approximation* module estimates distributions of missing PAs for parking lots in P_u, from both spatial and temporal domain. In the spatial domain, the *Propagating Graph Convolution* (Prop-Conv) 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 GRU module 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 feed to SCConv block and GRU module for final prediction.

Hierarchical spatial dependency modeling

We first introduce the hierarchical graph convolution module, including the contextual graph convolution block and the soft clustering graph convolution block.

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 (Kipf and Welling 2017; Velickovic et al. 2018) 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 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 (Ma et al. 2019). Specifically, we define the connectivity constraint $e_{ij} \in E$ as

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

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

$$c_{ij} = Attn(\mathbf{W}_a \mathbf{x}_i^c, \mathbf{W}_a \mathbf{x}_j^c), \qquad (3)$$

where $\mathbf{x}_i^c$ and $\mathbf{x}_j^c$ are current contextual representations of parking lot p_i and p_j, $\mathbf{W}_a$ is a learnable weighted matrix shared over all edges, and $Attn(\cdot)$ is a shared attention mechanism (e.g., dot-product, concatenation, etc.) (Vaswani et al. 2017). The proximity score between p_i and p_j is further defined as

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

In general, the above attention mechanism is capable of computing pair-wise proximity score 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 operate 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 lot at different time step may also vary, we learn a different proximity score for each different time steps.

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

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

where σ is a non-linear activation function, and $\mathbf{W}_c \in \mathcal{R}^{d \times d}$ 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, and $\mathbf{x}_j^c$ is the raw contextual feature in the first CxtConv layer.

Soft clustering graph convolution

Besides local 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
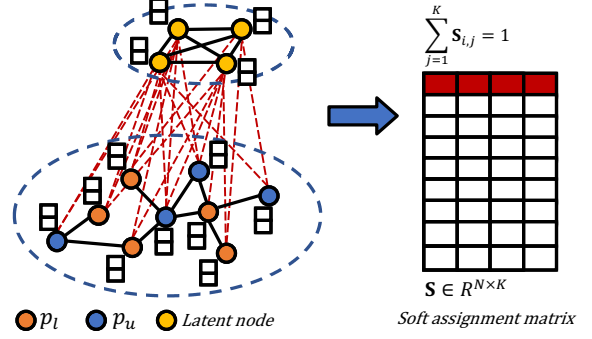

Figure 2: Hierarchical soft clustering.

higher PA at the same time. However, CxtConv only captures local spatial correlation. (Li, Han, and Wu 2018) shows when l goes large, the representation of all parking lots tends to be similar, therefore losses discriminative power. To this end, we propose the SCConv block to capture global correlations between parking lots. Specifically, SCConv defines a set of latent nodes and learns the representation of each latent node 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 have a chance to belong to multiple clusters with different probabilities (but with total probability equal to one), as shown in Figure 2.

The intuition behind SCConv is two-fold. First, distant parking lots may have similar contextual features and PAs, 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, 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.

The key component in SCConv is the soft assignment matrix. Given that there are K latent nodes, let $\mathbf{S} \in \mathcal{R}^{N \times K}$ denotes the soft assignment matrix, where $\mathbf{S}_{i,j} \in \mathbf{S}$ denotes the probability of i-th parking lot p_i maps to j-th latent node. Specifically, we use $\mathbf{S}_{i,\cdot}$ denote the i-th row and $\mathbf{S}_{\cdot,j}$ denote the j-th column of $\mathbf{S}$. Given the learned representation of each parking lot $\mathbf{x}_i$, each row of $\mathbf{S}$ is computed as

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

which guarantees that the probabilities that a given parking lot belongs to each latent node sum equals one.

Once $\mathbf{S}$ is obtained, the representation of each latent node $\mathbf{x}_i^s \in \mathbf{X}^s$ can be derived by

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

Given the representation of each latent node, similar to CxtConv, we apply soft clustering convolution operation to

capture the dependency between each latent node,

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

where σ is non-linear activation function, and α_{ij}^s is the proximity score between two latent nodes. Rather than introduce extra attention parameter as in CxtConv, we derive proximity score between latent nodes based on adjacency constraint between parking lots,

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

where a_{mn} equals one if parking lots p_m and p_n are connected. With learned latent node representation, we generate the soft clustering representation for each parking lot as a reverse process of latent node representation generation,

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

Temporal dependency modeling

We leverage the Gated Recurrent Unit (GRU) (Chung et al. 2014), a simple yet effective variant of recurrent neural network (RNN), to model the temporal dependency. Consider previous T step inputs of parking lot p_i, $(\mathbf{x}_i^{t-T+1}, \mathbf{x}_i^{t-T+2}, \cdots, \mathbf{x}_i^t)$, we denote the status 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

$$\mathbf{h}_i^t = (1 - \mathbf{z}_i^t) \circ \mathbf{h}_i^{t-1} + \mathbf{z}_i^t \circ \widetilde{\mathbf{h}}_i^t, \qquad (11)$$

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

$$\begin{cases} \mathbf{r}_i^t = \sigma(\mathbf{W}_r[\mathbf{h}_i^{t-1} \oplus \mathbf{x}_i^t] + \mathbf{b}_r) \\ \mathbf{z}_i^t = \sigma(\mathbf{W}_z[\mathbf{h}_i^{t-1} \oplus \mathbf{x}_i^t] + \mathbf{b}_z) \\ \widetilde{\mathbf{h}}_i^t = \tanh(\mathbf{W}_{\widetilde{h}}[\mathbf{r}_i^t \circ \mathbf{h}_i^{t-1} \oplus \mathbf{x}_i^t] + \mathbf{b}_{\widetilde{h}}) \end{cases}, \qquad (12)$$

where $\mathbf{W}_r, \mathbf{W}_z, \mathbf{W}_{\widetilde{h}}, \mathbf{b}_r, \mathbf{b}_z, \mathbf{b}_{\widetilde{h}}$ are learnable parameters, $\oplus$ is the concatenation operation, and $\circ$ denotes Hadamard product. Then the hidden state $\mathbf{h}_i^t$ is directly used to predict PAs of next τ time steps,

$$(\hat{y}_i^{t+1}, \hat{y}_i^{t+2}, \dots, \hat{y}_i^{t+\tau}) = \sigma(\mathbf{W}_o \mathbf{h}_i^t), \qquad (13)$$

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

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 directly apply real-time PA as a part of input feature. To leverage the information hidden in partially observed real-time PA, we approximate missing PAs from both spatial and temporal domain. 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 $\hat{y}$, we learn the distribution of PA, $\mathbf{x}^p = P(\hat{y})$, for better information preservation. Given a PA y, we discretize its distribution to a p dimensional one hot vector $\mathbf{y} \in \mathcal{R}^p$. The objective of the PA approximation is to minimize the difference between $\mathbf{y}$ and $\mathbf{x}^p$.

Spatial based PA approximation

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

$$\mathbf{x}_i^{sp} = \sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{y}_j, \qquad (14)$$

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 through same attention mechanism in Equation (4), but with a relaxed connectivity constraint

$$e_{ij} = \begin{cases} 1, & dist(v_i, v_j) \leq max(\epsilon, dist_{knn}(v_i)), i \neq j \\ 0, & otherwise \end{cases}, \qquad (15)$$

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 PA, and therefore alleviates the data scarcity problem.

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 previous step, we have obtained hidden state $\mathbf{h}_i^{t-1}$ from GRU, we directly approximate distribution of PA at t by

$$\mathbf{x}_i^{tp,t} = Softmax(\mathbf{W}_{tp} \mathbf{h}_i^{t-1}). \qquad (16)$$

This step doesn't introduce extra computation for GRU, and the $Softmax$ layer normalizes $\mathbf{x}_i^{tp,t}$ sum equals one.

Approximated PA fusion

Rather than directly averaging $\mathbf{x}_i^{sp}$ and $\mathbf{x}_i^{tp}$, we propose an entropy-based mechanism to fuse two PA distributions. Specifically, we weigh more on the approximation less uncertainty (Hsieh, Lin, and Zheng 2015), i.e., the one with smaller entropy. Given an estimated PA distribution $\mathbf{x}_i$, its entropy is

$$H(\mathbf{x}_i) = -\sum_{j=1}^{p} \mathbf{x}_i(j) \log \mathbf{x}_i(j), \qquad (17)$$

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:

$$\mathbf{x}_i^p = \frac{exp(-H(\mathbf{x}_i^{sp}))\mathbf{x}_i^{sp} + exp(-H(\mathbf{x}_i^{tp}))\mathbf{x}_i^{tp}}{\mathbf{Z}_i}, \qquad (18)$$

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 and fed to the SCConv block for latent node representation learning. Second, it is combined with the output of the CxtConv and SCConv, $\mathbf{x}_i^t = \mathbf{x}_i^{c,t} \oplus \mathbf{x}_i^{sc,t} \oplus \mathbf{x}_i^{p,t}$. We use $\mathbf{x}_i^t$ as the overall representation for each parking lot $p_i \in P$ at time step t, and feed it into the GRU module to generate final PA prediction results.

Model training

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

$$O_1 = \frac{1}{\tau |P_l|} \sum_{i=1}^{|P_l|} \sum_{j=1}^{\tau} (\hat{y}_i^{t+j} - y_i^{t+j})^2. \qquad (19)$$

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,

$$O_2 = -\frac{1}{|P_l|} \sum_{i=1}^{|P_l|} \mathbf{y}_i^t \log \mathbf{x}_i^{sp,t}, \qquad (20)$$

$$O_3 = -\frac{1}{|P_l|} \sum_{i=1}^{|P_l|} \mathbf{y}_i^t \log \mathbf{x}_i^{tp,t}. \qquad (21)$$

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

$$O = O_1 + \beta(O_2 + O_3), \qquad (22)$$

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

Experiments

Experimental setup

Data description. We use two real-world datasets collected from BEIJING and SHENZHEN, two metropolises in China. Both datasets are ranged 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 are collected by real-time sensors. We associate POI distribution (Liu et al. 2019b; Zhu et al. 2016) to each parking lot and aggregate check-in records nearby each parking lot in every 15 minutes as the population data. POI and check-in data are collected through Baidu Maps Place API and location SDK (Liu et al. 2019a). 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% parking lots are masked as unlabeled. The spatial distribution of parking lots in BEIJING are shown in Figure 3. The statistics of the datasets are summarized in Table 1.

Implementation details. Our model and all seven baselines are implemented with PaddlePaddle. Following previous work (Li et al. 2018; Yu, Yin, and Zhu 2018), the PA is normalized before input and scaled back to absolute PA in output. We choose $T = 12$ and select $\tau = 3$ for prediction. We set $\epsilon = 1$Km and $k = 10$ to connect parking lots. The dimension of $\mathbf{x}^c$ and $\mathbf{x}^{sc}$ are fixed to 32, p is fixed to 50. The layer of CxtConv, SCConv, and PropConv are $2, 1, 1$, respectively. We use dot-product attention in this paper. In SC-Conv, the number of latent nodes is set to $K = 0.1N$, where
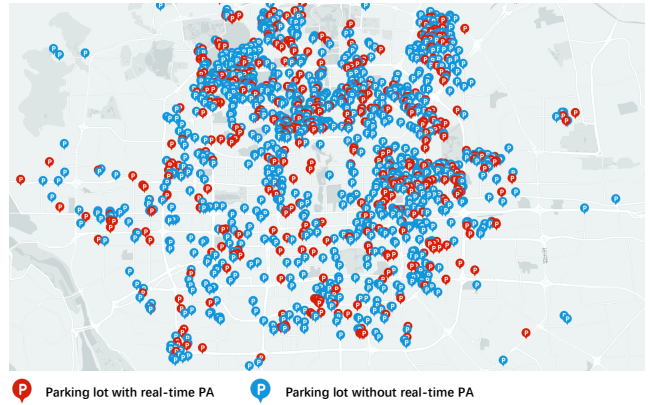

Parking lot with real-time PA   Parking lot without real-time PA

Figure 3: Spatial distribution of parking lots in BEIJING.

Table 1: Statistics of datasets.

Description	BEIJING	SHENZHEN
# of parking lots	1,965	1,360
# of PA records	5,847,840	4,047,360
Averaged # 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

N is the total number of parking lots. The activation function in CxtConv and SCConv are LeakyReLU ($\alpha = 0.2$), and Sigmoid in other layers. We employ the Adam optimizer for training, fix the learning rate to 0.001 and set β to 0.5. For a fair comparison, all parameters of each baseline are carefully tuned based on the recommended settings.

Evaluation metrics. We adopt *Mean Average Error* (MAE) and *Rooted Mean Square Error* (RMSE), two widely used metrics (Liang et al. 2018) for evaluation.

Baselines. We compare our full approach with the following seven baselines and two variants of SHARE:

- **LR** uses logistic regression for parking availability prediction. We concatenate previous T steps historical features as the input and predict each parking lot separately.

- **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 (Chen and Guestrin 2016), and the input is the same as LR.

- **GRU** (Chung et al. 2014) predicts the PA of each parking lot without considering spatial dependency. We train two GRUs for P_l and P_u separately.

- **Google-Parking** (Arora et al. 2019) is the parking difficulty prediction model deployed on Google Maps. It uses a feed-forward deep neural network for prediction.

- **Du-Parking** (Rong et al. 2018) is the parking availability estimation model used on Baidu Maps. It fuses several LSTMs to capture various temporal dependencies.

- **STGCN** (Yu, Yin, and Zhu 2018) is a state-of-the-art graph neural network model for traffic forecasting. It

Table 2: Parking availability prediction error given by *MAE* and *RMSE* on BEIJING and SHENZHEN.

Algorithm	BEIJING (15/ 30/ 45 min)		SHENZHEN (15/ 30/ 45 min)	
	MAE	RMSE	MAE	RMSE
LR	29.90 / 30.27 / 30.58	69.74 / 70.95 / 72.00	24.59 / 24.80 / 25.09	51.31 / 52.36 / 52.80
GBRT	17.29 / 17.81 / 18.40	44.60 / 48.50 / 51.59	13.90 / 14.67 / 14.71	35.05 / 37.98 / 38.09
GRU	18.51 / 18.78 / 19.73	55.43 / 55.92 / 58.64	16.73 / 16.88 / 17.14	46.92 / 47.26 / 47.56
Google-Parking	21.49 / 21.68 / 22.85	57.26 / 59.25 / 60.48	17.10 / 18.33 / 18.69	47.30 / 48.45 / 49.34
Du-Parking	17.67 / 17.70 / 18.03	50.17 / 50.63 / 51.75	13.91 / 14.17 / 14.39	42.66 / 43.24 / 43.56
STGCN	16.57 / 16.44 / 17.10	50.79 / 51.04 / 52.61	13.46 / 13.59 / 13.88	39.26 / 39.96 / 40.29
DCRNN	15.66 / 15.97 / 16.30	46.28 / 47.80 / 48.87	13.11 / 13.19 / 13.89	42.74 / 43.37 / 44.27
CxtGNN (ours)	15.29 / 15.69 / 16.15	45.55 / 46.69 / 47.78	12.39 / 12.73 / 13.09	36.31 / 36.92 / 37.46
CAGNN (ours)	12.45 / 12.77 / 13.20	39.99 / 40.81 / 41.31	10.50 / 10.62 / 10.98	31.86 / 32.12 / 32.83
SHARE (ours)	**10.68 / 10.97 / 11.43**	**32.00 / 32.78 / 33.78**	**9.23 / 9.41 / 9.66**	**30.44 / 30.90 / 31.70**

(a) Ratio of labeled parking lot   (b) Ratio of latent node   (c) Effect of T (d) Effect of τ
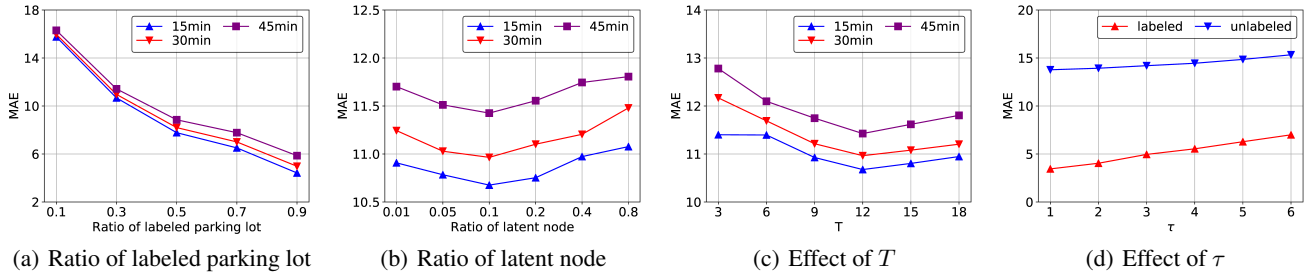
Figure 4: Parameter sensitivity on BEIJING.

models both spatial and temporal dependency with convolution structure. The input graph is constructed as described in the original paper but keeps same graph connectivity with our CxtConv.

- **DCRNN** (Li et al. 2018) is another graph convolution network based model, which models spatial and temporal dependency by integrating graph convolution and GRU. The input graph is the same as STGCN.

- **CxtGNN** is a basic version of SHARE, without including PA approximation and soft clustering graph convolution.

- **CAGNN** is another variant of SHARE but without including the soft clustering graph convolution block.

Overall performance

Table 2 reports the overall results of our methods and all the compared baselines on two datasets with respect to MAE and RMSE. As can be seen, our model together with its variants outperform all other baselines using both metrics, which demonstrates the advance of SHARE. Specifically, SHARE achieves $(31.8\%, 31.3\%, 29.9\%)$ and $(30.9\%, 31.5\%, 30.9\%)$ improvements beyond the state-of-the-art approach (DCRNN) on MAE and RMSE on BEIJING for $(15min, 30min, 45min)$ prediction, respectively. Similarity, the improvement of *MAE* and *RMSE* on SHENZHEN are $(29.6\%, 28.7\%, 30.5\%)$ and $(28.8\%, 28.8\%, 28.4\%)$. Moreover, we observe significant improvement by comparing SHARE with its variants (*i.e.*, CxtGNN and CAGNN). For example, by adding the PA approximation module, CAGNN achieves $(18.6\%, 18.6\%, 18.3\%)$ lower MAE and

$(12.2\%, 12.6\%, 13.5\%)$ lower RMSE than CxtGNN on BEIJING, respectively. By further adding the SCConv block, SHARE achieves $(14.2\%, 14.1\%, 13.4\%)$ lower MAE and $(20\%, 19.7\%, 18.2\%)$ lower RMSE than CAGNN on BEIJING. The improvement in SHENZHEN are consistent. All above results demonstrate effectiveness of the PA approximation and the hierarchical graph convolution architecture.

Looking further in to the results, we observe all graph convolution based models (*i.e.*, STGCN, DCRNN and SHARE) 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. 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 spatial distribution of parking lots is more dense and evenly distributed in SHENZHEN; therefore they are easier to predict.

Parameter sensitivity

Due to space limitations, here we report the impact of the ratio of labeled parking lot (*i.e.*, $|P_l|/N$), the proportion of latent nodes in the soft clustering graph convolution with respect to the total number of parking lot (*i.e.*, K/N), the input time step T and the prediction time step τ using MAE on BEIJING. Each time we vary a parameter, set others to their default values. The results on BEIJING using RMSE
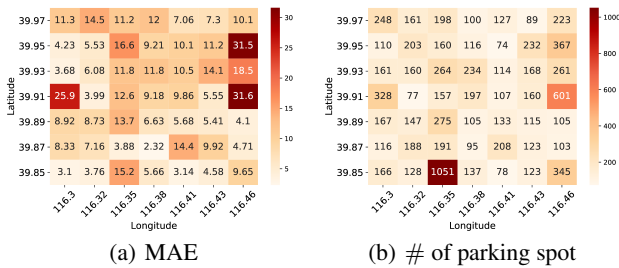
(a) MAE   (b) # of parking spot

Figure 5: Robustness study on BEIJING.

and on SHENZHEN using both metrics are similar.

First, we vary the ratio of the labeled parking lot from 0.1 to 0.9. The results are reported in Figure 4(a). The results are unsurprising: equipping more real-time sensors in parking lots enables us to more accurately predict PA. However, equipping more sensors lead to extra economic cost and may be constrained by policies of each parking lot. Finding the most cost-effective ratio and exploring optimal sensor distribution are important problems in the future study.

Then, we vary the ratio of the latent nodes from 0.01 to 0.8. For example, there are $1,965$ parking lots on BEIJING, 0.01 corresponds to 20 latent nodes. The results are reported in Figure 4(b). As can be seen, there is a performance improvement by increasing the ratio of latent node form 0.01 to 0.1, but a performance degradation by further increasing the ratio of the latent node from 0.1 to 0.8. The reason is that heavily reduce the number of latent nodes reduces the discriminative power of learned latent representation, whereas too many latent nodes reduces the regularization power of learned latent representation.

To test the impact of input length, we vary T from 3 to 18. The results are reported in Figure 4(c). SHARE achieves least errors when $T = 12$. One possible reason is that an excessively short-term input can not provide sufficient temporal correlated information, whereas too long input introduces more noises for temporal dependency modeling.

Finally, to test the impact of prediction step, we vary τ from 1 to 6. The results are reported in Figure 4(d). We separate the result of labeled and unlabeled parking lots separately. Overall, labeled parking lots are much easier to predict. Besides, by increase τ, the error of all parking lots increases consistently. However, we can observe the error of labeled parking lots are increasing faster, this makes sense because the temporal dependency between observed PA and future PA becomes lower when τ goes large.

Effectiveness on different regions

To evaluate the performance of SHARE on different regions, we partition BEIJING into a set of disjoint grid based on longitude and latitude, and test the performance of SHARE on each region. Figure 5(a) and Figure 5(b) plot the averaged MAE of SHARE and averaged number of parking spot in each region on BEIJING, respectively. Overall, the MAE in each region is even except for several outliers. We find the performance of SHARE is highly correlated with the averaged number of parking spots in each re-

gion. For example, the MAE on region $(116.46, 39.91)$ and $(116.46, 39.95)$ are 31.6 and 31.5, which are greater than the overall MAE 10.68. Meanwhile, the averaged parking spot of these two regions are 601 and 367, significantly greater than overall averaged parking spot 210.24. This is possible because for the same ratio of parking availability fluctuate, parking lot with a larger number of parking spot will have larger MAE. This result indicates in the future further optimization can be applied to these large parking lots to improve the overall performance.

Related Work

Parking availability prediction. Previous studies on parking availability prediction mainly fall in two categories, contextual data based prediction and real-time sensor based prediction. For contextual data based prediction, Google-parking (Arora et al. 2019) and Du-parking (Rong et al. 2018) predict parking availability based on indirect signals (*e.g.*, user feedbacks and contextual factors), which may induce an inaccurate prediction result. For real-time sensor based prediction, study in (Rajabioun and Ioannou 2015) proposes an auto-regressive model and study in (Fusek et al. 2013) proposes a boosting method for parking availability inference. Above approaches are limited by economic and privacy concerns and are hard to be scaled to all parking lots in a city. Moreover, all the above approaches don't fully exploit non-Euclidean spatial autocorrelations between parking lots, which limits their prediction performance.

Graph neural network. Graph neural network (GNN) extends the well-known convolution neural network to non-Euclidean graph structures, where the representation of each node is derived by first aggregating and then transforming representations of its neighbors (Velickovic et al. 2018). It is worth to point out that the idea of our soft clustering graph convolution is partially inspired by (Ying et al. 2018), 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 (Li et al. 2018; Guo et al. 2019) and taxi demand forecasting (Geng et al. 2019; Wang et al. 2019). However, we argue these approaches either overlook contextual factors or global spatial dependency and are not tailored for parking availability prediction.

Conclusion

In this paper, we present SHARE, a city-wide parking availability prediction framework based on both environmental contextual data and partially observed real-time parking availability data. We first propose a hierarchical graph convolution module to capture both local and global spatial correlations. Then, we adopt a simple yet effective GRU module to capture dynamic temporal autocorrelations of each parking lot. Besides, a parking availability approximation module is proposed for parking lots without real-time parking availability information. Extensive experimental results on two real-world datasets show that the performance of SHARE for parking availability prediction significantly outperforms seven state-of-the-art baselines.

Acknowledgement

References

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

Chen, T., and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 785–794.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

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

Geng, X.; Li, Y.; Wang, L.; Zhang, L.; Yang, Q.; Ye, J.; and Liu, Y. 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 3656–3663.

Guo, S.; Lin, Y.; Feng, N.; Song, C.; and Wan, H. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 922–929.

Hsieh, H.-P.; Lin, S.-D.; and Zheng, Y. 2015. 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*, 437–446.

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

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *6th International Conference on Learning Representations, ICLR 2018*.

Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 3538–3545.

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

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

Liu, H.; Li, T.; Hu, R.; Fu, Y.; Gu, J.; and Xiong, H. 2019a. Joint representation learning for multi-modal transportation recommendation. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 1036–1043.

Liu, H.; Tong, Y.; Zhang, P.; Lu, X.; Duan, J.; and Xiong, H. 2019b. 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*, 2314–2324.

Ma, S.; Hu, R.; Wang, L.; Lin, X.; and Huai, J.-P. 2019. An efficient approach to finding dense temporal subgraphs. *IEEE Transactions on Knowledge and Data Engineering*.

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

Rajabioun, T., and Ioannou, P. A. 2015. On-street and off-street parking availability prediction using multivariate spatiotemporal models. *IEEE Transactions on Intelligent Transportation Systems* 16(5):2913–2924.

Rong, Y.; Xu, Z.; Yan, R.; and Ma, X. 2018. Du-parking: Spatiotemporal big data tells you realtime parking availability. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 646–654.

Shoup, D. C. 2006. Cruising for parking. *Transport Policy* 13(6):479–486.

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

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018*.

Wang, P.; Fu, Y.; Liu, G.; Hu, W.; and Aggarwal, C. 2017. 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*, 495–503.

Wang, Y.; Yin, H.; Chen, H.; Wo, T.; Xu, J.; and Zheng, K. 2019. 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*, 1227–1235.

Yao, H.; Tang, X.; Wei, H.; Zheng, G.; and Li, Z. 2019. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*.

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

Yu, B.; Yin, H.; and Zhu, Z. 2018. 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*.

Zhang, J.; Wang, F.-Y.; Wang, K.; Lin, W.-H.; Xu, X.; and Chen, C. 2011. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems* 12(4):1624–1639.

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

Zhu, H.; Xiong, H.; Tang, F.; Liu, Q.; Ge, Y.; Chen, E.; and Fu, Y. 2016. Days on market: Measuring liquidity in real estate markets. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 393–402. ACM.