Predicting Carpark Availability in Singapore with Cross-Domain Data: A New Dataset and A Data-Driven Approach

Huaiwu Zhang[1*] , **Yutong Xia**[2*] , **Siru Zhong**[1] , **Kun Wang**[2] , **Zekun Tong**[2] ,
Qingsong Wen[3] , **Roger Zimmermann**[2] and **Yuxuan Liang**[1,4†]

[1]The Hong Kong University of Science and Technology (Guangzhou), China
[2]National University of Singapore, Singapore
[3]Squirrel AI, USA
[4]State Key Laboratory of Resources and Environmental Information System, China
{huaiwuzhang1998, yutong.x, yuxliang}@outlook.com; rogerz@comp.nus.edu.sg

Abstract

The increasing number of vehicles highlights the need for efficient parking space management. Predicting real-time Parking Availability (PA) can help mitigate traffic congestion and the corresponding social problems, which is a pressing issue in densely populated cities like Singapore. In this study, we aim to collectively predict future PA across Singapore with complex factors from various domains. The contributions in this paper are listed as follows: *(1) A New Dataset:* We introduce the `SINPA` dataset, containing a year's worth of PA data from 1,687 parking lots in Singapore, enriched with various spatial and temporal factors. *(2) A Data-Driven Approach:* We present DeepPA, a novel deep-learning framework, to collectively and efficiently predict future PA across thousands of parking lots. *(3) Extensive Experiments and Deployment:* DeepPA demonstrates a 9.2% reduction in prediction error for up to 3-hour forecasts compared to existing advanced models. Furthermore, we implement DeepPA in a practical web-based platform to provide real-time PA predictions to aid drivers and inform urban planning for the governors in Singapore. We release the dataset and source code at https://github.com/yoshall/SINPA.

1 Introduction

Urban traffic congestion significantly contributes to air pollution and poses health risks [Kelly and Fussell, 2015; Liang *et al.*, 2017]. The uneven distribution of parking spaces and inadequate parking decision-making by drivers are significant factors contributing to traffic congestion [Tilahun *et al.*, 2017]. The challenge of finding parking spaces is expected to grow as the population increases, thus the resulting social and environmental problems will be especially more acute in densely populated countries such as Singapore [Shamsuzzoha *et al.*, 2021]. To address these problems, there is an urgent need to optimize this issue by employing precise and efficient prediction of **Parking Availability (PA)** using smart city technologies [Xie *et al.*, 2022].
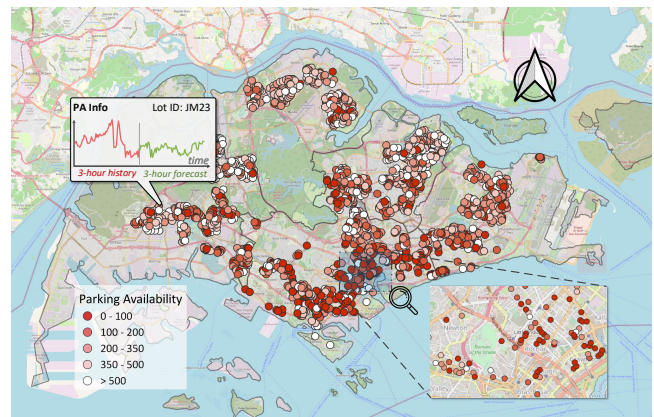

Figure 1: Distribution of 1,687 parking lots throughout Singapore. Each lot reports real-time parking availability every 15 minutes.

In this study, we focus on collectively predicting PA across thousands of parking lots in Singapore, as shown in Figure 1. This undertaking is confronted with several noteworthy challenges. Firstly, the availability of parking spaces within a parking lot is intricately influenced by multiple *complex factors*, including both temporal and spatial dimensions. Temporal external factors (e.g., weather conditions) wield substantial influence in shaping travel preferences and, consequently, parking patterns. Meanwhile, spatial factors, such as land use and road density, exert a noteworthy impact on the parking behavior of residents, consequently affecting PA. Therefore, effectively capturing the effects of temporal and spatial factors on PA is a necessary and formidable challenge.

Secondly, the *complex spatial correlations* among parking lots exhibit a non-linear relationship concerning geospatial distance measured in Euclidean space. This non-linearity is further compounded by the inherent disparities in their attributes, such as land functions and road network structures. In essence, the PA patterns of two proximate parking lots may diverge significantly due to distinctions in their intrinsic spa-

---

*H. Zhang and Y. Xia contributed equally to this work.

†Corresponding author. Email: yuxliang@outlook.com

tial characteristics. Additionally, comprehending the interconnected dynamics among various parking lots, influenced by the relatively stable total number of vehicles in the region, is imperative for a better understanding of PA patterns.

Thirdly, an approach that captures long-range (i.e., global) spatial dependencies [Zhang *et al.*, 2022] comes with another challenge in terms of *computational efficiency*. This challenge is particularly pronounced at fine spatial granularity, i.e., thousands of carparks in our task. Common methods like Graph Convolutional Networks[Kipf and Welling,] capture global spatial dependencies by stacking many local convolutional layers, which significantly increases computational costs. More advanced methods [Wu *et al.*, 2019; Wu *et al.*, 2020] using an adaptive adjacency matrix also result in a high computational complexity i.e., $\mathcal{O}(N^2)$, and the computational load of methods like Multi-head Self-Attention (MSA)[Vaswani *et al.*, 2017] also increases quadratically with the number of nodes N. Therefore, efficiently modeling the global spatial dependencies remains an unresolved issue.

Given the challenges outlined above, we aim to develop a model that not only incorporates a variety of cross-domain factors but also effectively captures the complex spatial correlations among parking lots to predict future PA accurately. To achieve this, we first crawl and process the **Sin**gapore **PA** dataset, i.e., `SINPA`, including PA data along with various external spatio-temporal factors, such as meteorological and land use data, specific to Singapore.

We then propose a data-driven model **DeepPA**, for collectively forecasting the future PA readings across thousands of carparks in Singapore while maintaining a manageable computational load. Specifically, we introduce the Graph Cosine Operator (GCO), a new mechanism for dynamically capturing spatial dependencies across extensive parking lots while mitigating computational complexity through discrete cosine transform. In addition, considering the crucial role of modeling temporal dependencies [Liang *et al.*, 2018; Guo *et al.*, 2019; Zheng *et al.*, 2020], we innovatively treat temporal information as a distinct "parking lot" entity. This allows us to integrate spatial data into position encoding effectively. Then we utilize causal MSA [Oord *et al.*, 2016; Liang *et al.*, 2023] to adhere to temporal sequences, thereby facilitating the learning of time-sensitive patterns in a manner that is both efficient and interpretable.

In summary, our contributions lie in the following aspects:

- **A New Dataset**. We crawl, process, and introduce `SINPA`, a large-scale parking availability dataset incorporating cross-domain data in Singapore. To the best of our knowledge, this dataset is the first publicly available dataset in the field of PA values forecasting, providing diverse applications within spatio-temporal domain research. The dataset is publicly accessible at https://github.com/yoshall/SINPA. It includes data obtained from Data.gov.sg, Urban Redevelopment Authority, and Land Transport Authority, which are made available under the Singapore Open Data Licence[1]

- **A Data-Driven Approach**. Leveraging insights gained from the analysis of multi-domain features, we develop a

---

[1]https://beta.data.gov.sg/open-data-license.

simple yet effective deep learning framework, DeepPA, to predict PA readings. DeepPA uses Graph Cosine Operator and Casual MSA to capture the complex spatial and temporal correlations among parking lots while ensuring an acceptable computational load.

- **Extensive Experiments**. We evaluate DeepPA on our dataset. The empirical results demonstrate the model's high accuracy, efficiency, and adaptability. These attributes make DeepPA an ideal candidate for real-world deployment for real-time PA prediction applications.

2 Preliminary

Parking Availability (PA) refers to the remaining space for parking in a parking lot. At the time t, the PA of the parking lots can be denoted as $\mathbf{X}^t = \left\{ x_1^t, x_2^t, ..., x_N^t \right\} \in \mathbb{R}^N$, where N refers to the total number of parking lots. Each entry x_n^t indicates the count of remaining parking spaces of n-th parking lots at time t. Given the historical PA of all carparks from the past T time steps and corresponding external features, we aim to learn a function $f(\cdot)$ that predicts their PA readings over the next τ steps:

$$f(\mathbf{X}^{1:T}; \mathbf{F}_t^{1:T}, \mathbf{F}_s) \rightarrow \mathbf{Y}^{1:\tau}, \qquad (1)$$

where $\mathbf{X}^{1:T} \in \mathbb{R}^{T \times N}$ indicates historical PA of all carparks; $\mathbf{F}_t^{1:T} = \left\{ F_{mete}^{1:T}, F_{time}^{1:T} \right\} \in \mathbb{R}^{T \times C_t}$ denotes external temporal features and C_t represent its dimension; $\mathbf{F}_s = \left\{ F_{loc}, F_{pln}, F_{use}, F_{rd} \right\} \in \mathbb{R}^{N \times C_s}$ denotes the spatially related feature and C_s represent its dimension; $\mathbf{Y}^{1:\tau} \in \mathbb{R}^{\tau \times N}$ is the future PAs.

2.1 Related Work

Intelligent Parking

Intelligent parking has attracted considerable attention from both academia and industry. Regarding the prediction of PA, Sii-MOBILITY[Badii *et al.*, 2018] integrates weather and traffic flow predicting PA based on Bayesian regularization neural networks. Subsequently, [Zhang *et al.*, 2020b] proposes a novel method SHARE based on Gated Recurrent Units[Chung *et al.*, 2014] and contextual graph convolution to capture both local and global spatial dependencies within parking lots, enabling estimation of missing PA in both temporal and spatial dimensions.

In addition, intelligent parking systems have other downstream tasks. [Rong *et al.*, 2018a] employs Long Short-Term Memory (LSTM) to model the PA-related temporal closeness, period, and current general influence to achieve parking availability imputation. Utilizing PA prediction combined with recommendation algorithms suggested the optimal parking lots near the user's specified destination [Xu *et al.*, 2023]. Moreover, models such as Random Forest and LSTM have identified the influence of weather factors on drivers' parking behavior [Zhang *et al.*, 2020a]. Besides, unlike enclosed parking lots that are independent of the street, it has also achieved the prediction of the availability of curbside parking spaces [Roman *et al.*, 2018].
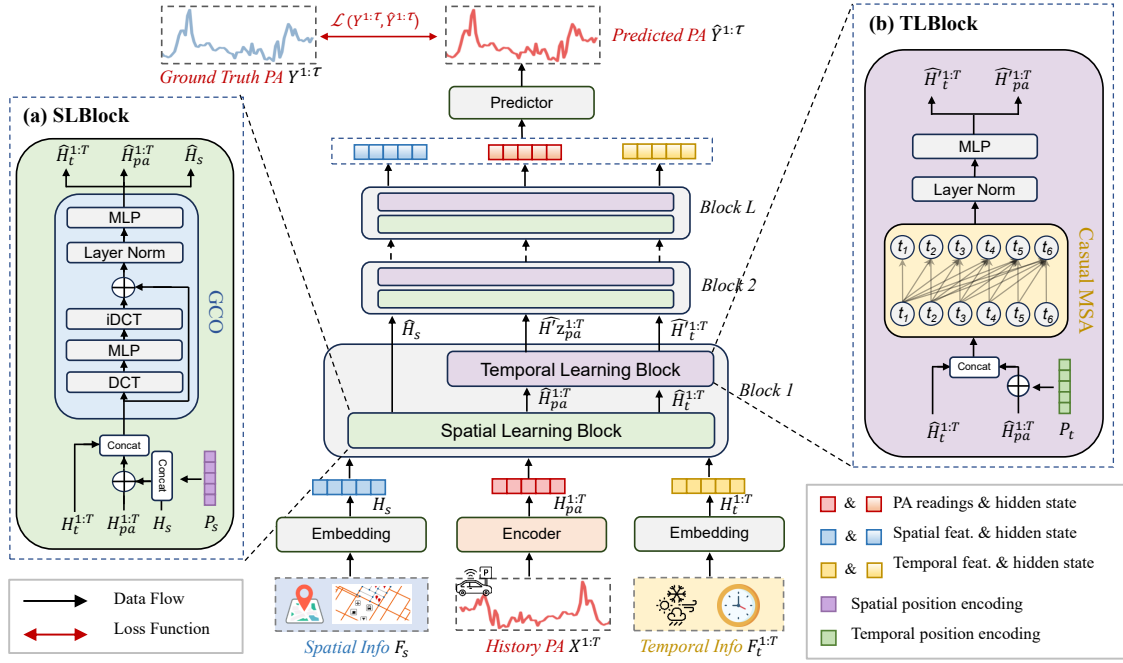
Figure 2: The framework of our proposed DeepPA. Initially, historical PA readings and spatio-temporal information are encoded and mapped into hidden states, respectively. Subsequently, PA will interact with spatio-temporal information in two sequential stages: (a) Spatial Learning Block (SLBlock) captures spatial dependence among parking lots, in which we introduce a novel structure Graph Cosine Operator (GCO) to enhance interaction efficiency; (b) TLBlock simultaneously investigates the temporal dynamics of PA and temporal information adhering to temporal patterns through Causal Multi-Head Attention (Causal MSA). Finally, DeepPA will integrate the feature space to forecast future PA. Concat: Concatenate. Info:Information

Data-Driven Spatio-Temporal Forecasting

Spatio-Temporal (ST) forecasting has garnered significant attention in the past few years [Jin *et al.*, 2023]. Conventional models, e.g., ARIMA and VAR, have enjoyed widespread adoption, while their performance tends to be suboptimal with highly volatile datasets[Liang *et al.*, 2018; Guo *et al.*, 2019; Zheng *et al.*, 2020; Liang *et al.*, 2021]. Building upon Graph Neural Networks (GNN) [Kipf and Welling,], Spatio-Temporal Graph Neural Networks (STGNN) [Wang *et al.*, 2021] have emerged for ST data by integrating temporal components, e.g. Temporal Convolutional Networks [Bai *et al.*, 2018] or Recurrent Neural Networks [Graves, 2013]. Pioneering examples include DCRNN [Li *et al.*, 2018], STGCN [Yu *et al.*, 2018], and ST-MGCN [Geng *et al.*, 2019]. Subsequent studies GWNet [Wu *et al.*, 2019] and AGCRN [Bai *et al.*, 2020], leverage adaptive adjacency matrices to enhance predictive performance. Meanwhile, ASTGCN [Guo *et al.*, 2019], GMAN [Zheng *et al.*, 2020], and STTN[Xu *et al.*, 2020] employ attention mechanisms to discern dynamic ST dependencies. STGODE [Fang *et al.*, 2021], STGNCDE [Choi *et al.*, 2022], and MixRNN [Liang *et al.*, 2022] capture continuous ST dynamics by employing neural ordinary differential equations. Causality-based methods such as CaST [Xia *et al.*, 2024] resolve the distribution shift problem via causal tools. Nonetheless, these methods are not tailored to the parking problem while concurrently addressing the above-mentioned challenges.

3 Methodology

The spatial and temporal external factors (i.e., weather and geo-location) play an important role in PA forecasting. To address the challenges of complex spatial correlations and computational efficiency, we have developed DeepPA, a deep learning framework for efficiently predicting future PA throughout Singapore.

3.1 Overall Framework

To address the challenges of complex spatial correlations and computational efficiency, we develop DeepPA, a deep learning framework for efficiently predicting future PA throughout Singapore, shown in Figure 2. We first transform the historical PA $\mathbf{X}^{1:T}$, along with spatial features $\mathbf{F}_s$ and temporal features $\mathbf{F}_t^{1:T}$ into the latent feature space using encoders, implemented through Multi-Layer Perceptrons (MLPs) and embedding layers. The hidden features are then fed to L DeepPA blocks to interact with each other to capture ST dependencies within data. Each DeepPA block consists of two sub-blocks as follows.

- *Spatial Learning Block (SLBlock)*: This module is utilized to capture intricate spatial relationships between parking lots, adopting a unique perspective by treating time information as a new node. To efficiently capture the non-Euclidean spatial relationship between thousands of parking lots, we introduce a novel Graph Cosine Operator.

- *Temporal Learning Block (TLBlock)*: This transformer-structure module is dedicated to capturing the periodic patterns that emerge over time at each parking lot. Specifically, it simultaneously model the hidden state of PA $\mathbf{H}^{1:T}$ and temporal features $\mathbf{F}_t^{1:T}$. To preserve the sequential order, we incorporate a mask in the MSA mechanism, ensuring that each state focuses only on its preceding states.

3.2 Spatial Learning Block (SLBlock)

Considering PA across different parking lots is interdependent, we aim to develop a module that effectively captures the specific spatial dependencies among thousands of parking lots while ensuring lower computational costs.

Overview

As depicted in the left part of Figure 2, $\mathbf{H}_s \in \mathbb{R}^{B \times T \times N \times C_{hs}}$ is concatenated with the spatial position encoding $\mathbf{P_s} \in \mathbb{R}^{B \times T \times N \times (C_p - C_{hs})}$ and then added to $\mathbf{H}_{pa}^{1:T}$:

$$\mathbf{H}^{1:T} = \mathbf{H}_{pa}^{1:T} + \text{concat}(\mathbf{H}_s, \mathbf{P_s}), \quad (2)$$

where $\mathbf{H}^{1:T} \in \mathbb{R}^{B \times T \times N \times C_p}$ is the PA with integrated the spatial features. In order to achieve interaction with temporal information, we then map the temporal information to the same dimension C_p as $\mathbf{H}^{1:T}$, treating it as an auxiliary node, i.e., a virtual node which acts as the $(N + 1)$-th parking lot, and concatenating it with $\mathbf{H}^{1:T}$:

$$\mathbf{H}^{1:T} = \text{concat}(\mathbf{H}^{1:T}, \mathbf{H}_t^{1:T}), \quad (3)$$

where $\mathbf{H}^{1:T} \in \mathbb{R}^{BT \times (N+1) \times C_p}$ represents the PA that integrates spatio-temporal information. After obtaining the initial features from both spatial and temporal domains, we jointly capture local and global spatial correlations of $\mathbf{H}^{1:T}$ via our proposed *Graph Cosine Operator (GCO)* by $\hat{\mathbf{H}}^{1:T} = \text{GCO}(\mathbf{H}^{1:T})$, which will be described in the subsequent part.

Graph Cosine Operator (GCO)

Since utilizing the attention mechanism in our dataset is computationally extensive, we design GCO to mitigate this issue by simulating how spatial dependencies induce changes $\Delta \hat{\mathbf{H}}^{1:T}$ in the hidden state of the parking lot. For a single parking lot, the variation of PA at each moment is known to be affected by all other parking lots, which can be expressed by the *heat conduction formula* [Wang *et al.*, 2007]:

$$\Delta \hat{H}_i^t = \sum k_{ji}(\hat{H}_j^t - \hat{H}_i^t) \ (i, j \in N , t \in T) \quad (4)$$

where k_{ji} represents the thermodynamic coefficient, which, in our study, signifies a learnable correlation coefficient between $\hat{H}_j^t$ and $\hat{H}_i^t$. Thus, at the same time instant t, the changes in the hidden states of all parking lots, i.e., $\Delta \hat{\mathbf{H}}^t$, can be represented as follows:

$$\Delta \hat{\mathbf{H}}^t = \begin{bmatrix} \Delta \hat{H}_1^t \\ \Delta \hat{H}_2^t \\ ... \\ \Delta \hat{H}_N^t \end{bmatrix} = \begin{bmatrix} \sum k_{j1}(\hat{H}_j^t - \hat{H}_1^t) \\ \sum k_{j2}(\hat{H}_j^t - \hat{H}_2^t) \\ ... \\ \sum k_{jN}(\hat{H}_j^t - \hat{H}_N^t) \end{bmatrix}, \quad (5)$$

where $\Delta \hat{H}_i^t$ is the i-th row of $\Delta \hat{\mathbf{H}}^t$.

Since the number of vehicles remains essentially constant throughout the entire range, we can assume that the correlation between two parking lots is consistent, i.e., $k_{ji} = k_{ij}$. After applying the distributive property to Eq. 5, we can obtain the following expression:

$$\Delta \hat{\mathbf{H}}^t = \begin{bmatrix} k_{11} & \cdots & k_{1N} \\ \vdots & \ddots & \vdots \\ k_{N1} & \cdots & k_{NN} \end{bmatrix} \hat{\mathbf{H}}^t - \begin{bmatrix} K_1 & & \\ & \cdots & \\ & & K_N \end{bmatrix} \hat{\mathbf{H}}^t \quad (6)$$

where K_i represents $\sum k_{ji}, j \in N$. It can be observed that the two learnable matrices in the equation are respectively a symmetric matrix and a diagonal matrix. Hence, they can be regarded as a special form of degree matrix $\hat{\mathbf{D}}$ and adjacency matrix $\hat{\mathbf{A}}$. Therefore, Eq. 6 can be expressed in the following form:

$$\Delta \hat{\mathbf{H}}^t = (\hat{\mathbf{D}} - \hat{\mathbf{A}})\hat{\mathbf{H}}^t = \hat{\mathbf{L}}\hat{\mathbf{H}}^t = \sigma \mathbf{L}\hat{\mathbf{H}}^t \quad (7)$$

where $\mathbf{L}$ represents the Laplacian matrix [Merris, 1994], We aim to fit $\hat{\mathbf{L}}$ by adding a set of trainable weights σ to $\mathbf{L}$. Let λ and $\mathbf{v}$ be the eigenvalues and eigenvectors of $\mathbf{L}$, respectively. Then Eq. 7 can be expressed as follows:

$$\hat{\mathbf{L}}\hat{\mathbf{H}}^t = \sigma \mathbf{L}\mathbf{v} = \sigma \lambda \mathbf{v} \quad (8)$$

Left-multiplying by the Laplacian matrix is equivalent to solving for the second derivative [Koren, 2003]. Therefore, for Fourier bases, the transformation can be denoted as follows:

$$\sigma \mathbf{L} e^{-j\omega t} = \sigma \frac{\partial^2 e^{-j\omega t}}{\partial t^2} = \sigma(-\omega^2)e^{-j\omega t} \quad (9)$$

where $-\omega^2$ represents the the eigenvalues of the Laplacian matrix. Defining a set of eigenvalues of $\mathbf{L}$ as:

$$\Lambda = (\lambda_1, \lambda_2, ..., \lambda_N) \in (-\omega^2) \quad (10)$$

The corresponding unit eigenvectors are denoted as:

$$\mathbf{U} = (\vec{u_1}, \vec{u_2}, ..., \vec{u_N}) \in e^{-j\omega t} \quad (11)$$

Thus Eq. 8 can be rewritten as:

$$\hat{\mathbf{L}}\hat{\mathbf{H}}^t = \sigma \mathbf{U}\Lambda \mathbf{U}^T \hat{\mathbf{H}}^t = \sigma(\mathcal{F}^{-1}(\mathcal{F}(\Lambda) \cdot \mathcal{F}(\hat{\mathbf{H}}^t))) \quad (12)$$

where $\mathcal{F}(\cdot)$ represents Discrete Fourier Transform (DFT). Inspired by Fourier Neural Operator (FNO)[Li *et al.*, 2020], we parameterize σ and $\mathcal{F}(\Lambda)$ via MLP layers. Previous work employed neural operators to approximate the MSA[Guibas *et al.*, 2021]. According to Euler's formula, the Fourier transform maps data from the time domain to the frequency domain, converting real numbers into complex numbers:

$$e^{-j\omega t} = \cos \omega t + j \sin \omega t \quad (13)$$

After this transformation, complex numbers are decomposed into their real and imaginary components. When multiplied by weights, it is crucial to adhere to the distributive property of multiplication, thus MLPs exhibit a fourfold increase in computational time when operating on complex numbers compared to real numbers. In the case of retaining only the real part, we use the Discrete Cosine Transform (DCT) to approximate DFT to reduce computational complexity while preserving performance efficacy, as demonstrated in Table 2. For a clearer comprehension, the overall operational steps of GCO for $\mathbf{H}^{1:T}$ are as illustrated in Algorithm 1.

Algorithm 1 Graph Cosine Operator

Input: $\mathbf{H}^{1:T} \in \mathbb{R}^{BT \times (N+1) \times C_p}$;
Output: Updated hidden states $\hat{\mathbf{H}}^{1:T}$;
1: Step 1: Projection into the frequency domain.
   $\hat{\mathbf{H}}^{1:T} = \text{DCT}(\mathbf{H}^{1:T})$
2: Step 2: Fitting the basis functions.
   $\hat{\mathbf{H}}^{1:T} = \text{MLP}(\hat{\mathbf{H}}^{1:T})$
3: Step 3: Returning to the spatial domain.
   $\hat{\mathbf{H}}^{1:T} = \text{iDCT}(\hat{\mathbf{H}}^{1:T})$
4: Step 4: Normalizing the output.
   $\hat{\mathbf{H}}^{1:T} = \text{LayerNorm}(\hat{\mathbf{H}}^{1:T})$
5: Step 5: Fitting the Laplacian matrix.
   $\hat{\mathbf{H}}^{1:T} = \text{MLP}(\hat{\mathbf{H}}^{1:T})$

Discussion

SLBlock is designed by jointly considering the following factors: 1) *Spatial dependencies*. Given that the strength of mutual influence between stations is not solely dependent on distance, we introduced a dynamic correlation coefficient k_{ji} to calculate the relationship between parking lots. 2) *Efficiency*. Via the GCO, we have managed to reduce the computational complexity. 3) *Temporal factor interation*: We treat the overall temporal information as a new node interacting with the PA to take temporal external factors into account.

3.3 Temporal Learning Block (TLBlock)

Accurately capturing the periodic relationship is crucial for the prediction. Meanwhile, variations in weather conditions and the impact of holidays can alter the travel patterns of residents, thereby impacting this periodicity. Therefore, we designed a novel module TLBlock to capture the periodicity while taking into account the impact of special time variables.

Capturing Temporal Correlation

First, to enable position-aware MSA, we infuse learnable position encoding into the input PA of TLBlock. Then, we decouple $\hat{\mathbf{H}}_t^{1:T}$ and $\hat{\mathbf{H}}^{1:T}$, where $\hat{\mathbf{H}}^{1:T}$ is considered as the $(N+1)$th parking lot in the SLBlock: $\hat{\mathbf{H}}^{1:T}, \hat{\mathbf{H}}_t^{1:T} = \text{split}(\hat{\mathbf{H}}^{1:T})$. Next, we merge the B and N dimensions of $\hat{\mathbf{H}}^{1:T}$ and concatenate them with $\hat{\mathbf{H}}_t^{1:T}$:

$$\hat{\mathbf{H}}^{1:T} = \text{concat}(\hat{\mathbf{H}}^{1:T}, \hat{\mathbf{H}}_t^{1:T}), \hat{\mathbf{H}}^{1:T} + \mathbf{P_t} \in \mathbb{R}^{(BN+B) \times T \times C_p}$$

where $\mathbf{P_t}$ represents temporal position encoding. Subsequently, temporal factors are modeled alongside PA using the same encoder to enable the model will take into account the influence of temporal information to a greater extent.

Temporal Causality

Given that the PA at a given step is not contingent on its future, inspired by [Oord *et al.*, 2016; Liang *et al.*, 2023], we introduce causality into MSA to ensure the model adheres to the temporal sequence of the input data, implemented by masking specific entries in the attention map:

$$\text{Causal MSA}(\hat{\mathbf{H}}^{1:T}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{\alpha}} + \mathbf{M})\mathbf{V} \quad (14)$$

where $\mathbf{Q} = \mathbf{W_q}\hat{\mathbf{H}}^{1:T}$, $\mathbf{K} = \mathbf{W_k}\hat{\mathbf{H}}^{1:T}$, and $\mathbf{V} = \mathbf{W_v}\hat{\mathbf{H}}^{1:T}$ are the queries, keys and values.

For a single state, $\hat{H}^i$, applying a mask $\mathbf{M} \in \mathbb{R}^{\alpha \times T \times T \times C_p/\alpha}$ involves setting its attention weights towards temporally later states $\hat{H}^j$ to $-\inf$, where $j \in (t+1, T)$.

3.4 Prediction & Optimization

DeepPA makes predictions based on the hidden state of the last DeepPA block $\hat{\mathbf{H}}^{1:T}$ using a predictor, which is implemented via an MLP layer in our implementation:

$$\hat{\mathbf{Y}}^{1:\tau} = \text{MLP}(\hat{\mathbf{H}}^{1:T}). \quad (15)$$

To train our model, we minimize the loss function $\mathcal{L}_{MAE}$. Here, $\mathcal{L}_{MAE}$ is the Mean Absolute Error for evaluating the errors between our prediction $\hat{\mathbf{Y}}^{1:\tau}$ and the corresponding ground truth $\mathbf{Y}^{1:\tau}$.

4 Experiments

We then conduct a series of experiments on SINPA to validate the efficacy of DeepPA. These experiments are designed to address the following Research Questions (RQ):

- **RQ1**: How does DeepPA compare in performance to existing PA forecasting approaches?

- **RQ2**: How does each module within DeepPA contribute to improving overall model performance?

- **RQ3**: What efficiency gains are achieved through the implementation of the GCO module?

- **RQ4**: Can DeepPA be effectively applied in practical scenarios for real-time online prediction?

4.1 Dataset Description

We crawled over three-year real-time PA data (2018/03/26 to 2021/08/11) every 5 minutes from 1,921 parking lots throughout Singapore from https://data.gov.sg. For our experiments, we re-sampled the raw dataset into the 15-minute interval and chose lots with a missing rate of PA of less than 30%. In addition, due to the temporal distribution shift, we only use one-year data (2020/07/01 to 2021/06/30), and the ratio of training: validation: testing sets is set as 10:1:1. We then remove parking lots with obvious distribution shift (i.e., high KL divergence). After sample filtering, it remains 1,687 parking lots with stationary data distributions. We also crawl external attributes for these lots, including meteorological data (i.e., temperature, humidity, and wind speed), panning areas, utilization type, and road networks data from Data.gov.sg, the Urban Redevelopment Authority (URA)[2] and the Land Transport Authority (LTA) website[3].

4.2 Experimental Settings

We implement our model by PyTorch 1.10 using a Quadro RTX A6000 GPU. The Adam optimizer is utilized to train our model, and the batch size is 8. The learning rate starts from 1×10^{-3}, halved every three epochs. For the hidden dimension C in SLBlock and SLBlock, we conduct a grid

---

Model	#Param	Time / epo (s) ↓	Average (1∼12 steps)		0-1h (1∼4 steps)		1-2h (5∼8 steps)		2-3h (9∼12 steps)	
			MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓
HA	-	-	45.28	69.68	44.41	68.11	43.68	67.89	45.62	71.44
VAR	-	-	54.12	105.64	53.10	93.98	56.86	94.46	55.71	130.00
DCRNN	392	896	$12.68_{\pm0.02}$	$30.33_{\pm0.05}$	$9.53_{\pm0.04}$	$27.16_{\pm0.06}$	$12.80_{\pm0.04}$	$30.04_{\pm0.09}$	$15.71_{\pm0.03}$	$33.83_{\pm0.06}$
STGCN	583	<u>221</u>	$12.70_{\pm0.11}$	$31.17_{\pm0.20}$	$9.64_{\pm0.06}$	$27.71_{\pm0.09}$	$12.94_{\pm0.18}$	$31.03_{\pm0.23}$	$15.65_{\pm0.23}$	$34.84_{\pm0.31}$
GWNET	808	303	$12.76_{\pm0.21}$	$31.14_{\pm0.18}$	$9.54_{\pm0.14}$	$27.73_{\pm0.15}$	$12.91_{\pm0.27}$	$30.88_{\pm0.26}$	$15.84_{\pm0.28}$	$34.86_{\pm0.16}$
MTGNN	207	361	$12.41_{\pm0.24}$	$30.62_{\pm0.18}$	$9.31_{\pm0.16}$	$27.37_{\pm0.10}$	$12.50_{\pm0.25}$	$30.37_{\pm0.25}$	$15.42_{\pm0.34}$	$34.16_{\pm0.33}$
ASTGCN	11,587	540	<u>$12.06_{\pm0.23}$</u>	<u>$30.22_{\pm0.31}$</u>	$9.88_{\pm0.12}$	$28.37_{\pm0.19}$	<u>$12.27_{\pm0.24}$</u>	$30.21_{\pm0.32}$	<u>$14.02_{\pm0.34}$</u>	<u>$32.10_{\pm0.41}$</u>
Du-Parking	392	683	$12.55_{\pm0.04}$	$30.41_{\pm0.05}$	<u>$9.22_{\pm0.02}$</u>	<u>$27.09_{\pm0.07}$</u>	$12.62_{\pm0.04}$	$30.12_{\pm0.08}$	$15.80_{\pm0.07}$	$34.05_{\pm0.08}$
SHARE	476	402	$12.57_{\pm0.05}$	$31.02_{\pm0.21}$	$9.33_{\pm0.04}$	$27.46_{\pm0.03}$	$12.70_{\pm0.05}$	$30.74_{\pm0.17}$	$15.69_{\pm0.07}$	$34.90_{\pm0.43}$
DeepPA (ours)	197	**219**	**10.95***$_{\pm0.10}$	**29.54***$_{\pm0.06}$	**9.11***$_{\pm0.09}$	$28.00_{\pm0.07}$	**11.18***$_{\pm0.09}$	**29.57***$_{\pm0.08}$	**12.56***$_{\pm0.15}$	**31.06***$_{\pm0.11}$

Table 1: 5-run results. The **bold**/<u>underlined</u> font means the best/the second best result. #Param: the number of parameters, expressed in thousands (Kilo). * denotes the improvement over the second best model is statistically significant at level 0.05 .

search over $\{8, 16, 32, 64, 128\}$, and $C = 64$ obtains the best result. The block number of SLBlock and TLBlock is 2.

We compare our DeepPA with the following baselines that belong to the following four categories: (1) *Classical methods*: **HA** [Zhang *et al.*, 2017] and **VAR** [Toda, 1991]. (2) *STGNN variants*: **DCRNN** [Li *et al.*, 2018], **STGCN** [Yu *et al.*, 2018], **GWNET** [Wu *et al.*, 2019], and **MTGNN** [Wu *et al.*, 2020]. (3) *Attention-based models*: **ASTGCN** [Guo *et al.*, 2019]. (4) *Deep learning for PA prediction*: **SHARE** [Zhang *et al.*, 2020b] and **Du-parking** [Rong *et al.*, 2018b]. Note that some strong baselines **STTN** [Xu *et al.*, 2020] **GMAN** [Zheng *et al.*, 2020], are omitted due to Out-Of-GPU-Memory (OOM) caused by a large number of carparks within `SINPA`.

4.3 Model Comparison (RQ1)

To address **RQ1**, following previous studies, we perform a model comparison in terms of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE)[Jean *et al.*, 2016]. We run each method five times and report the average metric of each model. As shown in Table 1, DeepPA significantly outperforms all competing baselines on both metrics according to the Student's T-test at level 0.05.

In terms of performance, we can observe that: 1) Deeplearning-based approaches show a substantial improvement over classical methods like HA and VAR due to their enhanced learning capacity. 2) Our proposed model exhibits a pronounced advantage over the SHARE and Du-parking models in the domain of parking space prediction, thereby providing additional validation for the influential role of the collected spatio-temporal information on PA prediction. 3) In comparison to the second-ranked approach (i.e., ASTGCN), DeepPA demonstrates a remarkable 9.2% decrease in MAE for long-term future prediction within the time interval of 3h. 4) Shifting the focus to individual hour predictions, the performance improvements for a single hour are observed to be 0.1%, 8.8%, and 10.4%, respectively.

Moreover, we have evaluated the efficiency of each method by reporting both the number of parameters and the training time per epoch, as detailed in Table 1. Our proposed model stands out due to its minimal parameter count and the fastest training duration among all baselines. This underscores our model's suitability for practical deployment: the reduced pa-
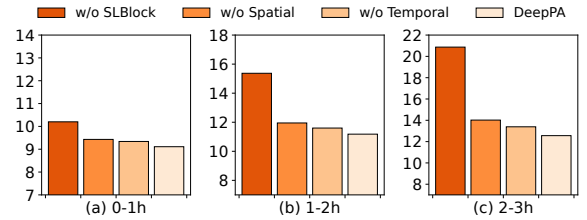

Figure 3: Effects of SLBlock on MAE.

rameter requirement minimizes memory demands, making it ideal for resource-constrained environments; the rapid training capability highlights its ability to quickly adapt to user needs, significantly enhancing user experience.

These attributes collectively make our model an optimal choice for real-world applications where *accuracy*, *efficiency*, and *adaptability* are of paramount importance.

4.4 Ablation Study (RQ2)

Effects of SLBlock

To investigate the effects of SLBlock, we consider the following variables for comparison: a) **w/o SLBlock**: we replace SLBlock with regular convolution solely operating on the PA. b) **w/o spatial**: we do not treat spatial information as a distinct entity for interaction with the PA. c) **w/o temporal**: we do not incorporate time information as a separate node for interaction with PA. The results (see Figure 3) reveal that removing either SLBlock or the ST information within the module leads to a significant decrease in MAE, underscoring the critical role of modeling spatial dependencies.

Effects of TLBlock

We then compare our model with its variants integrated with various temporal modules to assess the effectiveness of our TLBlock as well as the components in it: a) **w/o TLBlock**: we replace TLBlock with regular convolution solely operating on the PA. b) **MSA**: we replace Causal MSA with vanilla MSA. c) **w/o PE**: we remove the temporal position encoding from TLBlock. The results (see Figure 4) show that all the variants incorporating temporal modules demonstrate superior performance compared to the variants without TLBlock, confirming the importance of temporal information.
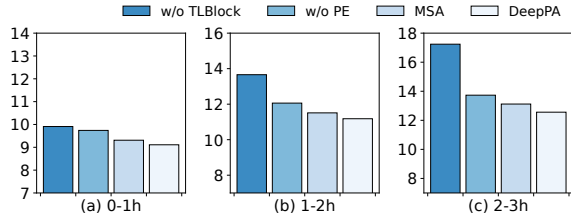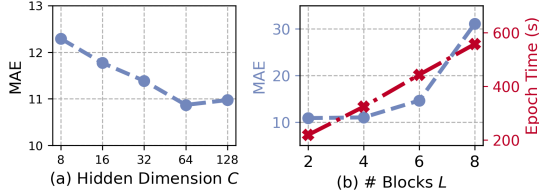
Figure 4: Effects of TLBlock on MAE.

Figure 5: Effects of (a) hidden dimension C and (b) the number of blocks L.

Hyperparameter Study

We explore the effects of the number of DeepPA blocks L and the hidden dimension C. First, we fix the block size at 2 and increase the hidden dimension starting from 8. Results in Figure 5a show that once the hidden dimension reached 64, the performance did not improve further. We thus set C as 64 and increase L starting from 2 (see Figure 5b) and found that when L increases from 2 to 4, there is no significant improvement in the performance. However, the training time per epoch increases by nearly double. Furthermore, when L exceeds 4, the model's performance deteriorates sharply, possibly due to overfitting and other factors.

4.5 Efficiency Study (RQ3)

To address **RQ3**, we evaluate the efficiency gains achieved by the GCO compared with other methods, then examine how the performance varies with the retention of different low-frequency signals.

Efficiency of GCO

To investigate the efficiency improvements introduced by the GCO, we replace this module with the following modules: a) **MSA**: while effective in capturing relationships across all parking lots, MSA is known for its high theoretical time complexity. b) **AFNO**: conceptually similar to GCO, AFNO [Guibas *et al.*, 2021] uses neural operators to approximate MSA. Previous research has shown its effectiveness in addressing the high time complexity issues in Vision Transformers with increased image resolution. Table 2 shows the training time per epoch for each variant. Compared to MSA and AFNO, GCO not only maintains performance but also significantly improves efficiency. Notably, after removing the SLBlock, the model's training time per epoch is around 115s, indicating that GCO reduces the training time by about 3.20 times and 2.37 times compared to MSA and AFNO, respectively, demonstrating GCO's efficient capability.

Variant	1-3h	0-1h	1-2h	2-3h	#Param	Time(s)
MSA	**10.87**	**9.07**	**11.10**	**12.44**	227	448
AFNO	10.90	9.11	11.12	12.48	199	361
GCO	10.95	9.11	11.18	12.56	197	**219**

Table 2: Comparative efficiency of GCO against other methods, measured in MAE. #Param: the number of parameters, expressed in thousands (Kilo).

4.6 Practicality (RQ4)

Regarding **RQ4**, leveraging the demonstrated efficiency and accuracy of our model, we deploy it for real-time parking availability forecasting. An illustrative example of our system in action is shown in Figure 6. The web is built on the Mapbox platform [Rzeszewski, 2023]. After selecting the target parking space, the system will predict future PA and present it in the form of a line graph. The shown graph displays a notable correlation between the predicted (represented by the blue line) and actual (represented by the red line) PA readings, demonstrating our model's high accuracy in predicting PA trends and variations effectively. For a direct experience of this interactive forecasting tool, please visit our demo web-based platform at https://sinpa.netlify.app.
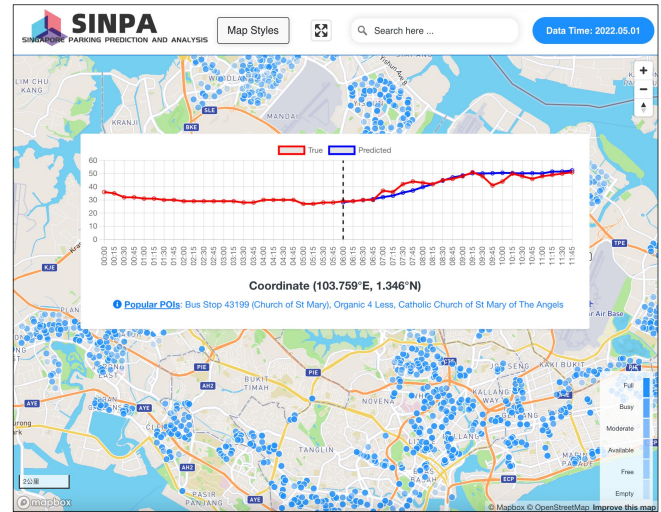

Figure 6: Our web-based PA monitoring and prediction system.

5 Conclusion and Future Work

In this paper, we crawled, processed, and released the SINPA dataset, a comprehensive collection of one-year PA readings from over 1,600 parking lots across Singapore, along with various external factors. To the best of our knowledge, SINPA is the first dataset of its kind to be made publicly available. Considering the complex relationships and the important role of external features, we presented DeepPA, a deep-learning model tailored for PA prediction. Our evaluations demonstrate the accuracy, efficiency, and adaptability of our model. In the future, we plan to explore reinforcement learning to improve parking recommendation services.

Acknowledgements

References

[Badii *et al.*, 2018] Claudio Badii, Paolo Nesi, and Irene Paoli. Predicting available parking slots on critical and regular services by exploiting a range of open data. *IEEE Access*, 6:44059–44071, 2018.

[Bai *et al.*, 2018] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*, 2018.

[Bai *et al.*, 2020] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

[Choi *et al.*, 2022] Jeongwhan Choi, Hwangyong Choi, Jeehyun Hwang, and Noseong Park. Graph neural controlled differential equations for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6367–6374, 2022.

[Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[Fang *et al.*, 2021] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 364–373, 2021.

[Geng *et al.*, 2019] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *AAAI*, volume 33, pages 3656–3663, 2019.

[Graves, 2013] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[Guibas *et al.*, 2021] John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Efficient token mixing for transformers via adaptive fourier neural operators. In *International Conference on Learning Representations*, 2021.

[Guo *et al.*, 2019] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 922–929, 2019.

[Jean *et al.*, 2016] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

[Jin *et al.*, 2023] Guangyin Jin, Yuxuan Liang, Yuchen Fang, Zezhi Shao, Jincai Huang, Junbo Zhang, and Yu Zheng. Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[Kelly and Fussell, 2015] Frank J Kelly and Julia C Fussell. Air pollution and public health: emerging hazards and improved understanding of risk. *Environmental geochemistry and health*, 37:631–649, 2015.

[Kipf and Welling,] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

[Koren, 2003] Yehuda Koren. On spectral graph drawing. In *International Computing and Combinatorics Conference*, pages 496–508. Springer, 2003.

[Li *et al.*, 2018] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR '18)*, 2018.

[Li *et al.*, 2020] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[Liang *et al.*, 2017] Yuxuan Liang, Zhongyuan Jiang, and Yu Zheng. Inferring traffic cascading patterns. In *Proceedings of the 25th acm sigspatial international conference on advances in geographic information systems*, pages 1–10, 2017.

[Liang *et al.*, 2018] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. Geoman: Multi-level attention networks for geo-sensory time series prediction. In *IJCAI*, 2018.

[Liang *et al.*, 2021] Yuxuan Liang, Kun Ouyang, Yiwei Wang, Ye Liu, Junbo Zhang, Yu Zheng, and David S Rosenblum. Revisiting convolutional neural networks for citywide crowd flow analytics. In *ECML-PKDD*, 2021.

[Liang *et al.*, 2022] Yuxuan Liang, Kun Ouyang, Yiwei Wang, Zheyi Pan, Yifang Yin, Hongyang Chen, Junbo Zhang, Yu Zheng, David S Rosenblum, and Roger Zimmermann. Mixed-order relation-aware recurrent neural networks for spatio-temporal forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Liang *et al.*, 2023] Yuxuan Liang, Yutong Xia, Songyu Ke, Yiwei Wang, Qingsong Wen, Junbo Zhang, Yu Zheng, and Roger Zimmermann. Airformer: Predicting nationwide air quality in china with transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14329–14337, 2023.

[Merris, 1994] Russell Merris. Laplacian matrices of graphs: a survey. *Linear algebra and its applications*, 197:143–176, 1994.

[Oord *et al.*, 2016] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[Roman *et al.*, 2018] Cristian Roman, Ruizhi Liao, Peter Ball, Shumao Ou, and Martin de Heaver. Detecting on-street parking spaces in smart cities: Performance evaluation of fixed and mobile sensing systems. *IEEE transactions on intelligent transportation systems*, 19(7):2234–2245, 2018.

[Rong *et al.*, 2018a] Yuecheng Rong, Zhimian Xu, Ruibo Yan, and Xu Ma. Du-parking: Spatio-temporal big data tells you realtime parking availability. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 646–654, 2018.

[Rong *et al.*, 2018b] Yuecheng Rong, Zhimian Xu, Ruibo Yan, and Xu Ma. Du-parking: Spatio-temporal big data tells you realtime parking availability. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 646–654, 2018.

[Rzeszewski, 2023] Michał Rzeszewski. Mapbox. In *Evaluating Participatory Mapping Software*, pages 21–40. Springer, 2023.

[Shamsuzzoha *et al.*, 2021] Ahm Shamsuzzoha, Juha Nieminen, Sujan Piya, and Kendall Rutledge. Smart city for sustainable environment: A comparison of participatory strategies from helsinki, singapore and london. *Cities*, 114:103194, 2021.

[Tilahun *et al.*, 2017] Surafel Luleseged Tilahun, Giovanna Di Marzo Serugendo, et al. Cooperative multiagent system for parking availability prediction based on time varying dynamic markov chains. *Journal of Advanced Transportation*, 2017, 2017.

[Toda, 1991] Hiroyuki Toda. *Vector autoregression and causality*. Yale University, 1991.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[Wang *et al.*, 2007] Liqiu Wang, Xuesheng Zhou, and Xiaohao Wei. *Heat conduction: mathematical models and analytical solutions*. Springer Science & Business Media, 2007.

[Wang *et al.*, 2021] Sheng Wang, Zhifeng Bao, J Shane Culpepper, and Gao Cong. A survey on trajectory data management, analytics, and learning. *ACM Computing Surveys (CSUR)*, 54(2):1–36, 2021.

[Wu *et al.*, 2019] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *IJCAI*, pages 1907–1913, 7 2019.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 753–763, 2020.

[Xia *et al.*, 2024] Yutong Xia, Yuxuan Liang, Haomin Wen, Xu Liu, Kun Wang, Zhengyang Zhou, and Roger Zimmermann. Deciphering spatio-temporal graph forecasting: A causal lens and treatment. *Advances in Neural Information Processing Systems*, 36, 2024.

[Xie *et al.*, 2022] Hongbin Xie, Xuan Song, and Haoran Zhang. Maas and iot: Concepts, methodologies, and applications. *Big Data and Mobility as a Service*, pages 229–243, 2022.

[Xu *et al.*, 2020] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.

[Xu *et al.*, 2023] Minrui Xu, Jiajie Xu, Rui Zhou, Jianxin Li, Kai Zheng, Pengpeng Zhao, and Chengfei Liu. Empowering a* algorithm with neuralized variational heuristics for fastest route recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[Yu *et al.*, 2018] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *IJCAI*, 2018.

[Zhang *et al.*, 2017] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI*, pages 1655–1661, 2017.

[Zhang *et al.*, 2020a] Feng Zhang, Ningxuan Feng, Yani Liu, Cheng Yang, Jidong Zhai, Shuhao Zhang, Bingsheng He, Jiazao Lin, and Xiaoyong Du. Pewlstm: Periodic lstm with weather-aware gating mechanism for parking behavior prediction. In *IJCAI*, pages 4424–4430, 2020.

[Zhang *et al.*, 2020b] Weijia Zhang, Hao Liu, Yanchi Liu, Jingbo Zhou, and Hui Xiong. Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1186–1193, 2020.

[Zhang *et al.*, 2022] Dalin Zhang, Kaixuan Chen, Yan Zhao, Bin Yang, Lina Yao, and Christian S Jensen. Design automation for fast, lightweight, and effective deep learning models: A survey. *arXiv preprint arXiv:2208.10498*, 2022.

[Zheng *et al.*, 2020] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1234–1241, 2020.