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

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

\textsuperscript{1}The Hong Kong University of Science and Technology (Guangzhou), China
\textsuperscript{2}National University of Singapore, Singapore
\textsuperscript{3}Squirrel AI, USA
\textsuperscript{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].

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-
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, 2016] 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., $O(N^2)$, 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 Singapore 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.

- **A Data-Driven Approach.** Leveraging insights gained from the analysis of multi-domain features, we develop a 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 $X^t = \{x^t_1, x^t_2, \ldots, x^t_N\} \in \mathbb{R}^N$, where $N$ refers to the total number of parking lots. Each entry $x^t_n$ indicates the count of remaining parking spaces of $n$-th parking lot 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 $\tau$ steps:

$$f(X^{1:T}; F^{1:T}_x, F_s) \rightarrow Y^{1:\tau},$$

where $X^{1:T} \in \mathbb{R}^{T \times N}$ indicates historical PA of all carparks; $F^{1:T}_x = \{F^{1:T}_{\text{meter}}, F^{1:T}_{\text{time}}\} \in \mathbb{R}^{T \times C_t}$ denotes external temporal features and $C_t$ represents its dimension; $F_s = \{F_{\text{loc}}, F_{\text{plan}}, F_{\text{use}}, F_{\text{rd}}\} \in \mathbb{R}^{N \times C_s}$ denotes the spatially related feature and $C_s$ represents its dimension; $Y^{1:\tau} \in \mathbb{R}^{\tau \times N}$ 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].
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, 2016], 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 ACRN [Bai et al., 2020], leverage adaptive adjacency matrices to enhance predictive performance. Meanwhile, ASTGCN [Guo et al., 2019], GMAN [Zheng et al., 2020], and STTNCXu 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 aforementioned 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 $X_{1:T}$, along with spatial features $F_s$ and temporal features $F_{t:T}^p$ 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.

![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.](image-url)
• **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 $H^{1:T}$ and temporal features $F^{i:T}_t$. 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, $H_s$ ∈ $\mathbb{R}^{B \times T \times N \times C_h}$ is concatenated with the spatial position encoding $P_s$ ∈ $\mathbb{R}^{B \times T \times N \times (C_p - C_h)}$ and then added to $H_{pa}^{1:T}$:

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

where $H^{1:T} \in \mathbb{R}^{B \times T \times N \times C_p}$ 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 $H^{1:T}$, treating it as an auxiliary node, i.e., a virtual node which acts as the $(N+1)$-th parking lot, and concatenating it with $H^{1:T}$:

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

where $H^{1:T} \in \mathbb{R}^{B \times T \times (N+1) \times C_p}$ 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 $H^{1:T}$ via our proposed **Graph Cosine Operator (GCO)** by $\hat{H}^{1:T} = \text{GCO}(H^{1:T})$, 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 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 via our proposed Graph Cosine Operator (GCO) to jointly capture local and global spatial correlations of $H^{1:T}$.

The changes in the hidden states of all parking lots, i.e., $\Delta \hat{H}^i$, can be represented as follows:

$$\Delta \hat{H}^i = \begin{bmatrix} \Delta \hat{H}^1_i \\ \Delta \hat{H}^2_i \\ \vdots \\ \Delta \hat{H}^N_i \end{bmatrix} = \begin{bmatrix} \sum k_{ji}(\hat{H}^j_i - \hat{H}^j_t) \\ \sum k_{ji}(\hat{H}^j_i - \hat{H}^j_t) \\ \vdots \\ \sum k_{ji}(\hat{H}^j_i - \hat{H}^j_t) \end{bmatrix}, \quad (5)$$

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

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 obtain the following expression:

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

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 $\hat{D}$ and adjacency matrix $\hat{A}$. Therefore, Eq. 6 can be expressed in the following form:

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

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

$$\hat{L}\hat{H}^i = \sigma L\hat{V} = \sigma \lambda \hat{V} \quad (8)$$

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:

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

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

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

The corresponding unit eigenvectors are denoted as:

$$U = (\vec{u}_1, \vec{u}_2, ..., \vec{u}_N) \in e^{-j\omega t} \quad (11)$$

Thus Eq. 8 can be rewritten as:

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

where $F(\cdot)$ represents Discrete Fourier Transform (DFT). Inspired by Fourier Neural Operator (FNO)[Li et al., 2020], we parameterize $\sigma$ and $F(\Lambda)$ 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 $H^{1:T}$ are as illustrated in Algorithm 1.
Temporal factor interaction
Via the GCO, we have managed to reduce the computational
calculate the relationship between parking lots. 2) Efficiency

Discussion
SLBlock is designed by jointly considering the following fac-
tors: 1) Spatial dependencies. Given that the strength of mu-
tual influence between stations is not solely dependent on dis-
tance, we introduced a dynamic correlation coefficient \(k_{ij}\) to
calculate the periodicity between parking lots. 2) Efficiency.
Via the GCO, we have managed to reduce the computational
complexity. 3) Temporal factor interaction: We treat the over-
total 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
developed 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 po-
position encoding into the input PA of TLBlock. Then, we de-
couple \(\hat{H}^1;T\) and \(\hat{H}^1;T\), where \(\hat{H}^1;T\) is considered as the \((N+1)\)th
parking lot in the SLBlock: \(\hat{H}^1;T, \hat{H}^1;T = \text{split}(\hat{H}^1;T)\).
Next, we merge the \(B\) and \(N\) dimensions of \(\hat{H}^1;T\) and con-
catenate them with \(\hat{H}^1;T\):
\[
\hat{H}^1;T = \text{concat}(\hat{H}^1;T, \hat{H}^1;T, \hat{H}^1;T + P_t) \in \mathbb{R}^{(BN+B) \times T \times C_p}
\]
where \(P_t\) represents temporal position encoding. Subse-
sequently, temporal factors are modeled alongside PA using the
same encoder to enable the model to 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 fu-
ture, 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:
\[
\text{Causal MSA}(\hat{H}^1;T) = \text{softmax}(\frac{A^T \hat{H}^1;T + M}{\sqrt{d}})V
\]
where \(Q = W_q \hat{H}^1;T\), \(K = W_k \hat{H}^1;T\), and \(V = W_v \hat{H}^1;T\)
are the queries, keys and values.

Discussion
SLBlock is designed by jointly considering the following fac-
tors: 1) Spatial dependencies. Given that the strength of mu-
tual influence between stations is not solely dependent on dis-
tance, we introduced a dynamic correlation coefficient \(k_{ij}\) to
calculate the periodicity between parking lots. 2) Efficiency.
Via the GCO, we have managed to reduce the computational
complexity. 3) Temporal factor interaction: We treat the over-
total 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
developed 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 po-
position encoding into the input PA of TLBlock. Then, we de-
couple \(\hat{H}^1;T\) and \(\hat{H}^1;T\), where \(\hat{H}^1;T\) is considered as the \((N+1)\)th
parking lot in the SLBlock: \(\hat{H}^1;T, \hat{H}^1;T = \text{split}(\hat{H}^1;T)\).
Next, we merge the \(B\) and \(N\) dimensions of \(\hat{H}^1;T\) and con-
catenate them with \(\hat{H}^1;T\):
\[
\hat{H}^1;T = \text{concat}(\hat{H}^1;T, \hat{H}^1;T, \hat{H}^1;T + P_t) \in \mathbb{R}^{(BN+B) \times T \times C_p}
\]
where \(P_t\) represents temporal position encoding. Subse-
sequently, temporal factors are modeled alongside PA using the
same encoder to enable the model to 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 fu-
ture, 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:
\[
\text{Causal MSA}(\hat{H}^1;T) = \text{softmax}(\frac{A^T \hat{H}^1;T + M}{\sqrt{d}})V
\]
where \(Q = W_q \hat{H}^1;T\), \(K = W_k \hat{H}^1;T\), and \(V = W_v \hat{H}^1;T\)
are the queries, keys and values.

For a single state, \(\hat{H}^1\), applying a mask \(M \in \mathbb{R}^{G \times T \times C_p}\) involves setting its attention weights towards
temporally later states \(\hat{H}_{j}^1\) 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 \(\hat{H}^1;T\) using a predictor, which is imple-
mented via an MLP layer in our implementation:
\[
\hat{Y}_{1;T}^1 = \text{MLP}(\hat{H}^1;T).
\]

To train our model, we minimize the loss function \(L_{MAE}\).
Here, \(L_{MAE}\) is the Mean Absolute Error for evaluating the
errors between our prediction \(\hat{Y}_{1;T}^1\) and the corresponding
ground truth \(Y_{1;T}^1\).

4 Experiments
We then conduct a series of experiments on SINPA to val-
date the efficacy of DeepPA. These experiments are designed
to address the following Research Questions (RQ):
• RQ1: How does DeepPA compare in performance to exist-
ing PA forecasting approaches?
• RQ2: How does each module within DeepPA contribute to
improving overall model performance?
• RQ3: What efficiency gains are achieved through the im-
plementation of the GCO module?
• RQ4: Can DeepPA be effectively applied in practical sce-
narios 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 ex-
periments, 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 meteoro-
logical 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) website³.

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 \(1 \times 10^{-3}\), halved every three epochs. For the hidden
dimension \(C\) in SLBlock and TLBlock, we conduct a grid

²https://www.ura.gov.sg/
³https://datamall.lta.gov.sg/content/datamall/en.html
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] and 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) Deep-learning-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 parameter 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)

To investigate the effects of SLBlock, we consider the following variables for comparison: a) \textit{w/o SLBlock}: we replace SLBlock with regular convolution solely operating on the PA. b) \textit{w/o spatial}: we do not treat spatial information as a distinct entity for interaction with the PA. c) \textit{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 SLBlock

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) \textit{w/o TLBlock}: we replace TLBlock with regular convolution solely operating on the PA. b) \textit{MSA}: we replace Causal MSA with vanilla MSA. c) \textit{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.
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 5 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.

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.
Acknowledgements
This work is mainly supported by a grant from State Key Laboratory of Resources and Environmental Information System and the Guangzhou-HKUST(GZ) Joint Funding Program (No. 2024AO30620). This work is also funded by the Advanced Research and Technology Innovation Centre (ARTIC), the National University of Singapore under Grant (project number: A-8000969-00-00).

References


