Collaborative Parking Vacancy Prediction for Cities With Partial Sensors Missing

Beiyu Song, Huachi Zhou, Xiao Huang, and Wei Ma, *Member, IEEE*

***Abstract*— City-wide short-term parking vacancy (PV) prediction is essential to transportation management. In many cities, only partial parking lots are equipped with sensors, causing severe data missing issues. Predicting PVs for parking lots without sensors is attractive but challenging. First, PV prediction itself is a nontrivial task since spatial dependencies among different regions in a city are complex and dynamic. By connecting parking lots based on geographical closeness using pre-designed rules, state-of-the-art PV prediction models achieve reasonable performance. However, those connectivities are not able to adapt to capture the changing dependencies in the graph. Second, node-to-node-based spatial and temporal dependencies will no longer be reliable since neighboring parking lot sensors may be intensely missing, which deteriorates the ability of the model to accurately impute PVs, leveraging the complex dependencies. To this end, we propose a novel framework named Collaborative graph Learning for pArking vacancy Prediction (CLaP). Specifically, to overcome the limitation of traditional pre-designed node connections, we propose a collaborative training method incorporating node attributes into graph augmentation, thus enhancing the ability to capture dynamic spatial dependencies through message aggregation. Besides, a recurrent PV recovering module is developed to impute missing embeddings by deeply coupling spatial-temporal dependencies. Experimental results on two real-world datasets demonstrate that CLaP outperforms state-of-the-art PV prediction models on city-wide PV prediction and imputation precision for parking lots without sensors.**

***Index Terms*— Parking vacancy prediction, graph neural network, traffic prediction with data missing.**

I. Introduction

WITH the development of Intelligent Transportation Systems (ITS) and the deployment of advanced sensing technologies, smart transportation challenges can be solved in a more data-driven manner [1]. Specifically, city-wide distributed sensors are deployed to provide real-time parking space information [2]. It enables parking vacancy prediction, helping drivers find nearby parking lots without cruising [3].

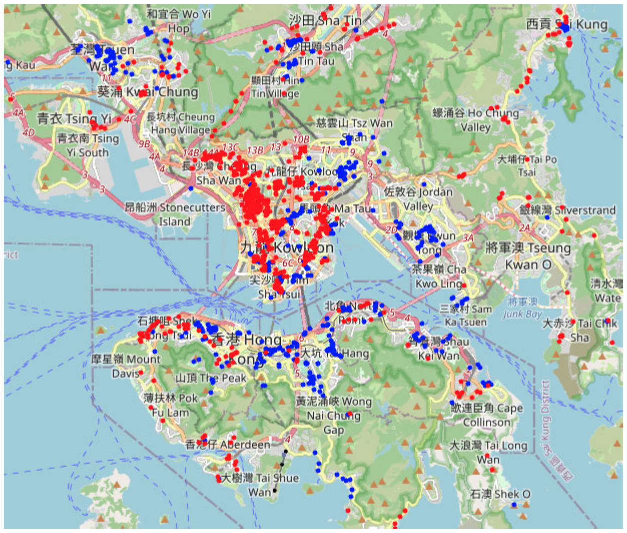


Fig. 1. Spatial distribution of parking lots in Hong Kong. Blue spots refer to parking lots with real-time PV data, and red spots refer to parking lots with PV data missing.

Existing PV prediction studies [4], [5], [6] rely on the availability of complete PV sequences of all parking lots to track spatial-temporal evolution tendencies and previse future behaviors. However, PVs may not be continuously accessible due to sensors partially missing. In many cities, sensors are not installed due to high cost or may be broken because of poor maintenance [7]. According to data[1] collected from sensors deployed by the Hong Kong government, we observe that 60% of parking lots are incapable of recording PVs across consecutive time steps, as shown in Fig. 1. The severely missing sensors lead to the high absence of PVs in both spatial and temporal dimensions, which makes it challenging to achieve a satisfactory prediction. Traditional preprocessing imputation methods, e.g., k-Nearest Neighbor (KNN), could be utilized to approximate the missing PVs before the implementation of prediction models. But these methods fail to simulate real-world spatial-temporal patterns comprehensively so that the imputed PVs may be highly biased from ground truths.

Manuscript received 16 May 2023; revised 28 September 2023 and 16 January 2024; accepted 8 March 2024. This work was supported in part by the Smart Traffic Fund of the Transport Department of Hong Kong Special Administrative Region, China, under Grant PSRI/07/2108/PR; and in part by the Grant from the Research Grants Council of Hong Kong, SAR, under Project PolyU 25208322. The Associate Editor for this article was M. Shafie-Khah. *(Corresponding author: Beiyu Song.)*

Beiyu Song is with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong, SAR, China (e-mail: beiyu.song@connect.polyu.hk).

Huachi Zhou and Xiao Huang are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, SAR, China (e-mail: huachi666.zhou@connect.polyu.hk; xiaohuang@comp.polyu.edu.hk).

Wei Ma is with the Department of Civil and Environmental Engineering and the Research Institute for Sustainable Urban Development, The Hong Kong Polytechnic University, Hong Kong, SAR, China (e-mail: wei.w.ma@polyu.edu.hk).

Digital Object Identifier 10.1109/TITS.2024.3377734

[1] https://data.gov.hk/en-data/dataset/hk-td-msd_1-metered-parking-spaces-data

We notice that recent efforts [8], [9] aggregate nearest neighbors at each time step and then transmit to recurrent neural network (RNN) components to approximate the missing PVs of the next time steps. Meanwhile, they capture spatial dependencies through a static graph, which connects parking lots based on human-designed criteria. Nevertheless, there are two limitations. First, the precision of approximation relies on the similarity of PV patterns among the nearest nodes with sensors to the target node. The PV patterns may evolve as time progresses, and the similarity of PV patterns between the nearest nodes with sensors and target nodes may change at different historical time spans. Iteratively generated last-time step output does not explicitly model spatial-temporal dependencies at different lengths of historical time slices. Thus it may not provide sufficient context to impute the missing PVs accurately. Second, to model spatial dependencies, graph construction based on road connection or geographical closeness is reasonable. But it may not be able to adapt to capture the changing dependencies of complex real-time parking vacancy patterns in the graph, which may be influenced by multiple factors: Point-of-Interest (POI) distributions [10], traffic conditions [11], etc. For example, it's easy to find parking lots around shopping malls, while hard for nearby office buildings on weekdays.

In summary, real-time parking vacancy prediction with sensors partially missing faces two significant challenges. (i) The first challenge is the dynamic spatial dependencies faced by PV prediction tasks. The PV patterns can change rapidly and vary based on time. For example, at some time step, the neighboring nodes may exhibit dissimilar PV patterns while distant nodes may exhibit similar ones. A feasible solution is adding connections to those parking lots that maximally express dynamic spatial dependencies [12]. Nevertheless, it is non-trivial to identify appropriate connections and pre-design the graph structure due to non-obvious parking vacancy patterns changing with time. (ii) The second challenge is how to leverage the severely corrupted spatial-temporal dependencies caused by missing PVs to do imputation. The PVs of neighboring nodes may be highly possible absent, while the PV patterns of distant nodes without missing sensors may have different distributions at different time spans. To address this intricate case, an available approach is to make approximations considering spatial and temporal dependencies one after the other. But it fails to model the deep interactions between them simultaneously and does not provide sufficient context to impute missing data preciously.

To tackle those challenges, we propose a novel framework: Collaborative graph Learning for pArking vacancy Prediction (CLaP), to predict parking vacancies with sensors partially missing. The contributions can be summarized as follows:

- We design a deep learning framework tailored to the specific data missing issue for predicting city-wide parking vacancy.
- We develop a collaborative graph training mechanism to capture dynamic spatial dependencies between parking lots. The graph-based spatial connections are learned by incorporating node attributes into graph augmentation.
- We propose a parking vacancy recovering module to impute missing PVs. To deeply couple spatial-temporal dependencies, pseudo PV embeddings are approximated by jointly leveraging neighbor representations at different historical time spans.
- We conduct extensive experiments on two real-world datasets. Compared with several baselines, our model outperforms significantly on PV prediction tasks, including both city-wide predictions and predictions for parking lots without sensors.

TABLE I

SUMMARY OF IMPORTANT NOTATIONS

Notation	Description
$\lvert P_r \rvert$	number of parking lots having PVs
$\mathcal{Y}_r^T \in \mathbb{R}^{\lvert P_r \rvert \times T}$	input PV matrix
$X_{aug} \in \mathbb{R}^{N \times d_c}$	augmented POI features
$\overline{Y}^t \in \mathbb{R}^{N \times d_p}$	recovered PV embeddings
$Y_{aug}^t \in \mathbb{R}^{N \times (d_c + d_p)}$	augmented spatial representations
$\hat{Y} \in \mathbb{R}^{N \times T'}$	predicted PV matrix

II. PRELIMINARIES

In this section, we introduce two key definitions and formulate the problem statement of real-time PV prediction with sensors partially missing. The important notations throughout the paper are listed in Table I.

- Definition 1: Parking vacancy (PV). Let $P = P_r \cup P_s = \{p_1, p_2, \ldots, p_N\}$ be a set of parking lots with the total number of N, where P_r refers to parking lots with sensors while P_s without. The PV of each parking lot p_i is represented as y_i^t, indicating the available parking space at time t. All involved y_i^t denoted as $Y^t = Y_r^t \cup Y_s^t$ are parking vacancies of all parking lots, where Y_r^t represents observed PVs from P_r while Y_s^t represents missing ones from P_s. Parking vacancy (PV) embeddings denote high-dimensional node representations.
- Definition 2: Parking lot network. The parking lots are modeled as an undirected graph $G = (P, E, A, X)$, where P is the set of all parking lots and E is a set of edges indicating the connection between them. $A \in \mathbb{R}^{N \times N}$ denotes the binary-valued adjacency matrix, where $A_{i,j} = 1$ when the geographical distance between p_i and p_j is smaller than a geographical threshold δ. $X \in \mathbb{R}^{N \times d}$ refers to POI features of parking lots, which can be learned through model training.

A. Problem Statement

For inference, given partially observed PV sequences $\mathcal{Y}_r^T = \left(Y_r^{t-T+1}, Y_r^{t-T+2}, \ldots, Y_r^t\right) \in \mathbb{R}^{|P_r| \times T}$ from P_r at historical T time steps, the objective is to predict PV sequences in the next T' time steps $\mathcal{Y}^{T'} = \left(Y^{t+1}, Y^{t+2}, \ldots, Y^{t+T'}\right) \in \mathbb{R}^{N \times T'}$ of all parking lots P.

III. METHODOLOGY

The framework of CLaP is shown in Fig. 2. It is composed of three major modules: attention-based POI feature extraction

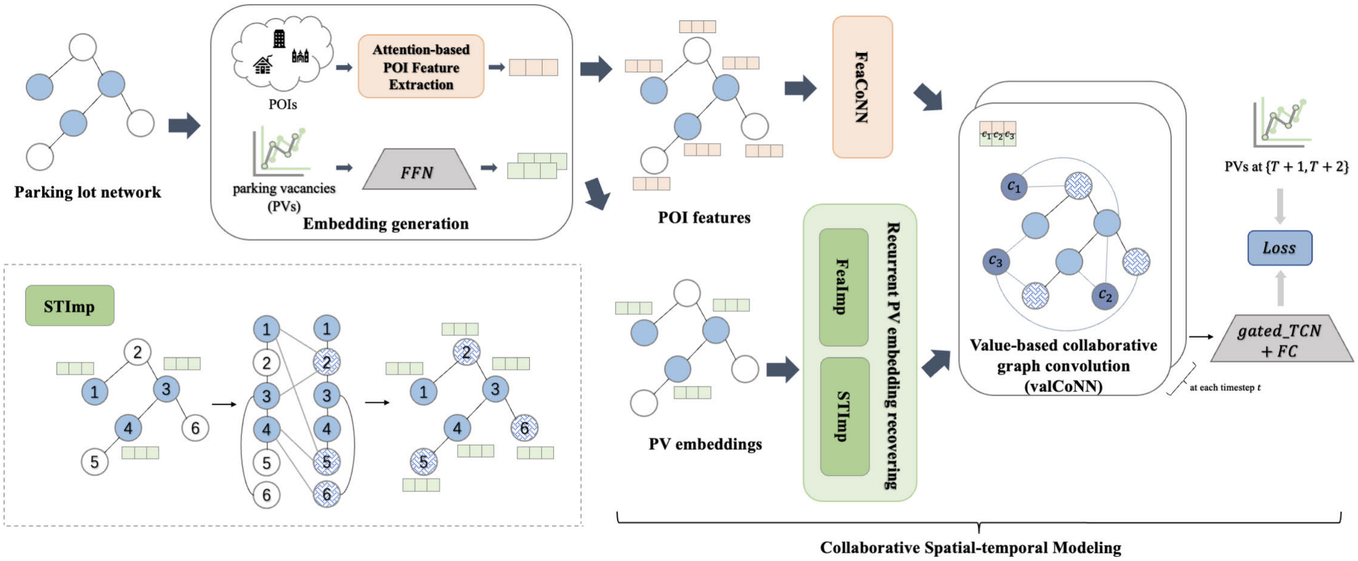


Fig. 2. The architecture of the proposed framework - Collaborative graph Learning for pArking vacancy Prediction (CLaP).

(*POIAttn*), recurrent parking vacancy embedding recovering, and collaborative spatial-temporal modeling. (i) The first module extracts POI features X for all parking lots as additional node attributes besides PVs. This module guarantees parking lots without sensors also have node attributes for the recovery of PV embeddings in the second module and the construction of an augmented graph in the third module. (ii) To reconstruct severely corrupted spatial-temporal dependencies for each node, we first recover missing PV embeddings for sensorless nodes. The second module achieves this by employing two submodules: Spatial-Temporal Imputation (*STImp*) and Feature-Based Augmented Imputation (*feaImp*). These submodules leverage two types of information: neighbors' historical PV embeddings and surrounding POI features separately to achieve more precise imputed PV embeddings. (iii) To capture the intricate spatial dependencies, we design the collaborative graph aggregation mechanism and develop two submodules: Feature-based collaborative graph aggregation (*feaCoNN*) and value-based collaborative graph aggregation (*valCONN*). They learn augmented graphs and propagate PV embeddings to model dynamic spatial dependencies at different time steps on the extracted POI features and recovered PV embeddings. Then the dilated temporal convolution block with a gating mechanism (*gated_TCN*) models the temporal dependencies in the aggregated PV embeddings at the last T time steps.

A. Attention-Based POI Feature Extraction

In the parking lot graph, each node is endowed with node attributes representing parking vacancies and POIs. POI information refers to a specific point location of potential interest with geographical coordinates. It serves essential node attributes to characterize parking lots.

Extracting POI features may be beneficial to explore spatial dependencies between parking lots [4], e.g., parking lots are more occupied around shopping malls than office buildings at weekends. To selectively generate meaningful POI categorical information as additional node features, we utilize self-attention mechanism [13] to capture the spatial dependencies between parking lots and surrounding POIs, defined as:

$$X = Attention\,(Q, K, V) = softmax\left(\frac{QK^{\top}}{\sqrt{d}}\right)V, \qquad (1)$$

where $X \in \mathbb{R}^{N \times d}$ refers to the generated POI features; d represents the extracted POI feature dimension; Q, K refers to geographical information (latitude and longitude) of parking lots and surrounding POIs, respectively; V refers to categorical information of POIs. $\top$ denotes the transpose operation here. If a POI is closer to a parking lot, the attention coefficient learned by similar geographical information will weigh more when aggregating POI-based features compared with other POIs.

This way, parking lots described by similar POI distributions around will exhibit similar POI features for nodes. Instead of concerning each POI's influence on parking lots, we focus on a set of POIs categories within a geographical range in the city map for computing efficiency.

B. Recurrent Parking Vacancy Embedding Recovering

The recurrent PV embedding recovery module generates pseudo PVs for all parking lots over the past T time steps based on node attributes. It consists of two parallel submodules for this purpose: spatial-temporal imputation (*STImp*), which leverages neighboring nodes with historical PV values, and feature-based augmented imputation (*feaImp*), which utilizes extracted POI features from the *POIAttn* module.

1) Spatial-Temporal Imputation: For missing embedding recovering, current efforts model spatial and temporal dependencies separately and design divided components to do imputation. The potential spatial-temporal interactions between them may not be fully captured. Due to the sensors being severely missing, it is necessary for the target node to rely on distant nodes to impute the missing PVs. The distant nodes may not follow closely similar PV patterns to the target one; that is to say, the importance of these nodes may vary over time. As a result, the imputed data will inevitably be biased from ground truths. Additionally, the errors may iteratively propagate across consecutive time steps. In order to accurately capture such intricate dependencies, it is necessary to analyze the relationship between the nearest nodes with sensors and the target node at different historical time spans. Inspired by this, we target jointly modeling historical neighbor node PVs at different time spans to impute missing data.

Due to the possible sparse connection between nodes, we introduce a more tightly connected adjacency matrix for PV embedding recovering: $\hat{A} = A \mid A'$ where $A'_{i,j} = 1$ when the euclidean distance between p_i and p_j is among the top K minimum in P_r. $\mid$ refers to OR operation and hyper-parameter K controls the sparsity of the graph. $\hat{A}$ guarantees that each parking lot has at least K neighbors having PV embeddings. It eliminates the possibility of inconsequential information aggregation arising from adjacent nodes lacking PV embeddings. In other words, even if parking lots in a city suffer from extreme sparsity, originally isolated nodes can also have at least K edges connected to other nodes with sensors in the tighter adjacency matrix.

The transition process of spatial-temporal dependencies modeling can be viewed as a special Markov process [14] where current states of all parking lots P can be inferred by previous ones with:

$$\dot{Y}^t = \beta(\hat{A} \odot W^{t-1})\dot{Y}^{t-1}, \tag{2}$$

with

$$\dot{Y}^{t-1} = Y^{t-1} \odot m + \dot{Y}^{t-1} \odot (1-m)\,, \tag{3}$$

where $W^{t-1} \in \mathbb{R}^{N \times N}$ is a learnable parameter matrix; β is a trainable weight; $\odot$ refers to element-wise product. Masking assignment $m \in \{0,1\}^N$ is set to 1 if the corresponding node has features, while 0 if it is missing. $\dot{Y}^{t-1}$ refers to estimated PV distributions while Y^{t-1} refers to real observed ones at time $t-1$. The Eq.(3) allows for updating of missing PVs while preserving available PVs through the aggregation of every adjacent timestamp.

Through this way, the states of neighbor nodes over different time spans before t have the possibility of becoming the optimal dependent ones of the target node. Based on the main idea of the Markov process that the current state only depends on the previous one and with the recurrent update on the information of the current time step, Eq.(2) is updated by:

$$\dot{Y}^t = \sum_{z=1}^{\tau} \beta^z (\prod_{j=1}^{z} \hat{A} \odot W^{t-j})(Y^{t-z} \odot (1-m)), \tag{4}$$

where τ denotes the selected historical length and is set as a hyper-parameter. The aggregation guarantees that the target node jointly considers the PVs of neighbor nodes over different time spans. Fig. 3 represents the whole picture of our *STImp*.

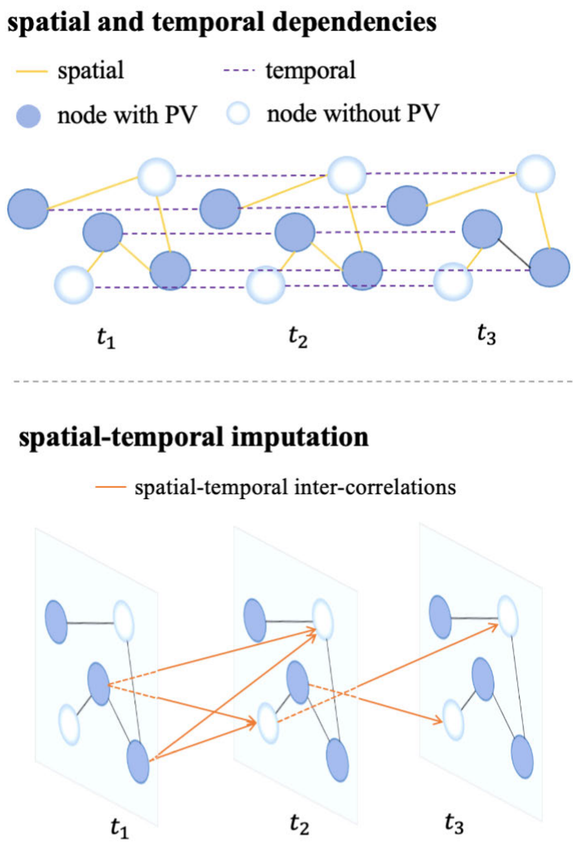

spatial and temporal dependencies
spatial
temporal
node with PV
node without PV
t_1
t_2
t_3
spatial-temporal imputation
spatial-temporal inter-correlations
t_1
t_2
t_3

Fig. 3. Spatial-temporal imputation block (*STImp*). The picture above shows the spatial and temporal dependencies commonly concerned in the PV prediction task. The picture below represents our idea of deeply coupling spatial-temporal dependencies from adjacent nodes at previous time steps.

2) Feature-Based Augmented Imputation: STImp generates pseudo PVs for all parking lots using historical neighboring nodes with PV values. The accurate approximation is still challenging in scenarios with severely missing sensors. We notice nodes with similar POI features may share similar parking vacancy patterns. The extracted POI features may serve as the beneficial signal to assist better PV approximation. This motivates us to harness the generated POI features from *POIAttn* within the parking network to generate improved approximations of PV distributions concurrently. And we introduce an attention-based graph convolution [15] to obtain pseudo PV embeddings $\tilde{Y}^t$ guided by POI features at each time step. The message passing function defined on the parking lot p_i at the first layer is as follows:

$$\tilde{y}_i^{(1)} = \sigma\left(\sum\nolimits_{j \in N_i} a_{ij} W y_j^{(0)}\right), \tag{5}$$

where $y_j^{(0)} \in \mathbb{R}^{d_p}$ denotes PV embeddings for p_j; σ is the sigmoid activation function; $W \in \mathbb{R}^{d_p \times d_p}$ is a learnable matrix; $N_i = \{j | \hat{A}_{i,j} \neq 0\}$ denotes the nearest neighbor node set with sensors of p_i. The attentive weight a_{ij} emphasizes the neighboring parking lot with relevant extracted POI features defined as:

$$a_{ij} = \frac{exp(DP(x_i, x_j))}{\sum_{u \in N_i} exp(DP(x_i, x_u))}, \tag{6}$$

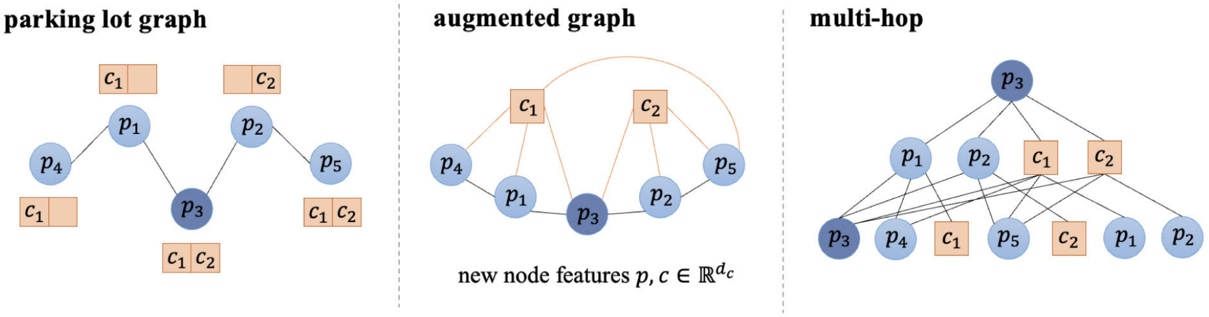


Fig. 4. Example of *feaCoNN*. Parking lot graph models the original graph structure. Augmented graph models the new graph structure considering feature-based nodes c_1 and c_2.

where $DP(x_i, x_j)$ represents the scaled dot-product function between POI features of p_i and p_j. By stacking multiple propagation layers, we obtain the updated pseudo PV embeddings for all parking lots P considering surrounding POI's influence, and denote them as $\tilde{Y}^t$.

We concatenate outputs from two submodules $\overline{Y}^t = \dot{Y}^t + \tilde{Y}^t \in \mathbb{R}^{N\times d_p}$ as the generated pseudo PV embeddings.

C. Collaborative Spatial-Temporal Modeling

After being processed by the recurrent parking vacancy embedding recovery module, each parking lot has real or pseudo PV embeddings for the last T time steps. We aim to capture intricate dynamic dependencies by introducing a collaborative graph aggregation mechanism. This mechanism introduces additional nodes for each dimension in node attributes, i.e., POI features and PV embeddings at each time step. These newly established nodes will prioritize PVs related to the corresponding dimension and propagate them to all parking lots based on their relevance. After the module aggregates neighbor information at each time step, a subsequent gated temporal convolution block performs time step aggregation, evolving to predict the next time step.

1) Collaborative Graph Aggregation in Spatial Domain: Current efforts only model spatial dependencies in a pre-designed static graph. However, in real-world parking scenarios, spatial-temporal correlations between parking lots are more intricate and may change over time spans. For example, it is common that two distant parking lots share similar PV patterns at some time steps while quite different at others. It drives us to deeply integrate node attributes into the GNN aggregation to capture dynamic dependencies, that is, to make nodes potentially sharing similar PV distributions more tightly connected.

Based on that, we propose a novel graph training mechanism. First, it augments parking networks with learned node features as newly created nodes. Second, it links parking lots with corresponding newly-added feature nodes. To provide a clearer explanation of the operational details, the example is illustrated in Fig. 4. Nodes p_4 and p_5 have the same element 1 at the first feature dimension (c_1). The augmented graph connects them to node c_1 by an edge with weight 1. In this way, all parking lots not only aggregate adjacent node messages through the pre-designed graph, but also attend to distant nodes based on node attribute similarity.

We design two parallel submodules, feature-based collaborative graph aggregation (*feaCoNN*) and value-based collaborative graph aggregation (*valCoNN*), with this mechanism. The *feaCoNN* exploits the extracted POI feature as input to construct the augmented graph, while *valCoNN* exploits the real or pseudo PV embeddings. To save layout space, we take *feaCoNN* as an example. The adjacency matrix of the augmented graph is constructed as follows:

$$\hat{A} = \begin{bmatrix} A & X \\ X^\top & 0 \end{bmatrix} \in \mathbb{R}^{(N+d)\times(N+d)}, \tag{7}$$

where $X \in \mathbb{R}^{N\times d}$ refers to POI features.

The node pairwise edge strength indicates the closeness of spatial dependency. We capture the intricate spatial dependency by prioritizing the neighbors with the edge strength in $\hat{A}$. We proceed to conduct message aggregation on the newly built augmented graph. Since the original POI features of all parking lots are converted to edges, all node features are randomly set to prevent the inclusion of duplicate information. We use $\hat{H}^{(0)} \in \mathbb{R}^{N\times d_c}$ to denote the freshly initialized features of parking lot nodes P, while $\tilde{H}^{(0)} \in \mathbb{R}^{d\times d_c}$ denotes features of feature nodes Q. Then the parking lot p_i could aggregate messages from P and Q together. Its feature is updated by:

$$\hat{H}_i^{(1)} = \sigma\left(\alpha \textstyle\sum_{j\in N_i^G} \hat{b}_{ij}\hat{H}_j^{(0)} + (1-\alpha)\sum_{q\in Q}\tilde{b}_{iq}\tilde{H}_q^{(0)}\right), \tag{8}$$

with normalizing edge weights by a softmax function:

$$\hat{b}_{ij} = \frac{exp(\hat{A}_{ij})}{\sum_{u\in N_i^G} exp(\hat{A}_{iu})}, \tilde{b}_{iq} = \frac{exp(\hat{A}_{iq})}{\sum_{u\in Q} exp(\hat{A}_{iu})}, \tag{9}$$

where adjacency weight α is a trade-off factor; N_i^G formulates the neighbor node set of p_i in parking networks G. After multiple steps of message aggregation, we obtain the smoothed POI-based features $X_{aug} = \hat{H}^{(k)} \in \mathbb{R}^{N\times d_c}$.

ValCoNN takes $X_{aug} \oplus \overline{Y}^t \in \mathbb{R}^{N\times(d_c+d_p)}$ as input and generates $Y_{aug}^t \in \mathbb{R}^{N\times(d_c+d_p)}$ at each time step recurrently for further temporal modeling. The graph augmentation and message aggregation of *valCoNN* are the same as *feaCoNN*, and we omit them for brevity.

By applying these two submodules, node representations can be aggregated based on learnable spatial dependencies. And the connection is dynamically determined by real-time PV embeddings and POI features. They guarantee parking lots having similar features to share tighter spatial connection even if the two nodes are separated by a large number of edges in the original graph, which benefits PV prediction for large-scale cities with a mass of parking lots.

2) Dilated Convolution in Temporal Domain: The spatial representations generated by collaborative graph aggregation for last T time steps is $Y_{aug} \in \mathbb{R}^{N\times T\times(d_c+d_p)}$. It effectively captures the parking vacancy patterns of each parking lot over time. To make accurate predictions, we extract both short-term and long-term temporal dependency within the patterns by leveraging 1D dilated causal convolution [16]. This convolution operation has a large receptive field which allows it to extract extensive information from Y_{aug}. The dilated convolution operation is defined as:

$$\hat{Y} = \left(\theta_1 \star Y_{aug} + b_1\right) \odot \sigma\left(\theta_2 \star Y_{aug} + b_2\right), \tag{10}$$

where $\star$ refers to dilated causal convolution; θ_1, θ_2, b_1 and b_2 refer to weight and bias parameters separately.

We stack two convolutions with a gating mechanism [17] to extend the receptive fields of historical PVs and obtain aggregated representations $\hat{Y} \in \mathbb{R}^{N\times T\times 1}$. It contains both spatial and temporal information, which could be utilized to guide the next T' time step PV prediction.

D. Joint Training

We adopt the mean square error (MSE) loss to compare the difference between imputed PVs and real historical PVs for parking lots among P_r denoted as:

$$\mathcal{L}_{pseudo} = \frac{1}{T}\frac{1}{|P_r|}\sum_{t=1}^{T}\sum_{i=1}^{|P_r|}\left(y_i^t - \overline{y}_i^t\right)^2. \tag{11}$$

The $\mathcal{L}_{pseudo}$ loss is introduced to minimize the error between the imputed PV distributions and observed one. Then we compute the PV prediction loss of the next T' time steps also for parking lots among P_r:

$$\mathcal{L}_{train} = \frac{1}{T'}\frac{1}{|P_r|}\sum_{t=1}^{T'}\sum_{i=1}^{|P_r|}\left(y_i^t - \hat{y}_i^t\right)^2. \tag{12}$$

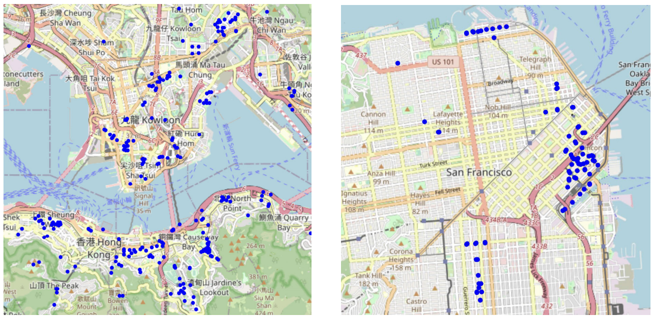
Finally, we integrate two losses into a joint training objective:

$$\mathcal{L}_{all} = \mathcal{L}_{train} + \gamma\mathcal{L}_{pseudo}, \tag{13}$$

where γ controls the magnitude of recovering the loss.

IV. Experiments

We conduct extensive experiments to investigate the effectiveness of our proposed model. Four research questions are waiting to be answered. **Q1**: How does CLaP perform on PV prediction tasks compared with state-of-the-art baselines, especially for those parking lots without sensors? **Q2**: How does each core component of CLaP contribute to the whole model performance? **Q3**: How does CLaP perform with different hyper-parameters? **Q4**: How do missing PVs affect model performance? **Q5**: How do POIs influence PV prediction from different perspectives?



(a) Hong Kong (b) San Francisco

Fig. 5. Spatial distributions of sensors in two cities.

A. Two Real-World Datasets

- **HK (Hong Kong)**. The dataset is collected from DATA.GOV.HK,[2] recording the occupancy status of parking lots with sensing meters in Hong Kong. We use Map-Matching to match all parking lots with road segments collected from OpenStreetMap.[3] Then the objective becomes to investigate the total parking vacancies on each road segment. The time is November 2021.
- **SF (San Francisco)**. The dataset is collected from HARVARD Dataverse,[4] recording the measured parking availabilities of parking lots in San Francisco. The period was from June to July, 2013.

Figs. 5a and 5b show the spatial distributions of sensors in Hong Kong and San Francisco. In Hong Kong, most sensors are deployed on main streets in commonly visited business and residential areas with large traffic flows; while in San Francisco, sensors are located in a small business district somewhere around a pier. The number of tested nodes is larger and the sensors' connectivities are much more complicated in Hong Kong compared with those in San Francisco. We group and transform parking records into PVs every 5 minutes. We split the city map into 40 × 40 and 30 × 30 grids for HK and SF datasets respectively, and count categorical POI distribution [18] in each grid [19]. We will explain the detailed description in the following POI analysis. The geographical information of POIs that fall in corresponding grids in *POIAttn* is recorded as the grid center's latitude and longitude. The example is shown in Fig. 6.

To validate the performance of our model, especially on parking lots with sensors missing (P_s), we select nodes with complete PV information. Notably, when training, we incorporate a $N \times T$ masking matrix with P_s nodes set to 0 and P_r nodes set to 1 to guarantee parking lots with missing sensors will not be included during the training process of the proposed framework. The implementation of training is shown in Fig. 7a. Besides, PVs are transformed using min-max normalization before model implementation. Remarkably, the

[2] https://data.gov.hk/en-data/dataset/hk-td-msd_1-metered-parking-spaces-data

[3] https://www.openstreetmap.org/

[4] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/YLWCSU

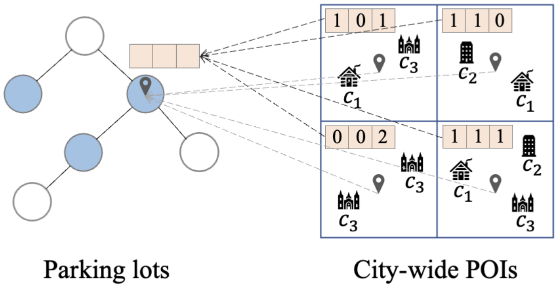


Fig. 6. Example of *POIAttn*. POI features are generated based on the attention score queried by locations. Example of POI attribute of one node. Each element in the feature vector represents how many times a location or place, e.g., c_1 appears in the geographical range. The POI feature is generated based on the attention score queried by the location.

predicted PVs are scaled back to make the comparison. The detailed dataset description is shown in Table II.

B. Implementation Details

The model is implemented by PyTorch and run on NVIDIA GeForce RTX 3090, 24GB memory. We use Adam optimizer with a learning rate of 0.001. Early stopping is adopted based on validation performance with a threshold of 30 epochs.

The observed PV sequences are split into training, validation, and test sets with a ratio of 6:2:2. Within each set, we choose $T = 12$ and $T' = 12$ time steps. With a time step corresponding to 5 minutes' PV information, we use the past one hour's PV to predict the next one hour's PV.

The hyper-parameters are tuned to be optimal based on the performance of the validation set. Specifically, we set the distance threshold $\delta = 1000\ m$ and $\delta = 200\ m$ to connect parking lots for HK and SF datasets separately. K for constructing $\hat{A}$ is set to be 50 and 20 for HK and SF datasets. The searched hyper-parameters are listed below. d, d_p are set to 16 and 32. Specifically, high-dimensional POI features are generated from *POIAttn*, whose dimension d is aligned with the dimension of POI categories. POI categories are mapped to dimension d from a linear function. High-dimensional PV embeddings d_p are generated through the feed-forward network (FFN). d_c is set to 32 and 16 in HK and SF datasets. α, β, γ equal 0.5, 0.9, and 1. τ is 6, that is to say, it considers the historical 6 time steps when recovering missing PV embeddings. The optimal propagation step of graph convolution in *feaImp*, *feaImp*, and *valCoNN* is 1, 2, and 2. We run all models five times repeatedly and report the average results.

For other baselines, we all incorporate POIs into parking lots as node features for a fair comparison. We utilize the source code released by the authors and adopt the reported optimal hyper-parameter configuration.

C. Experimental Settings

We analyze the model performance over two tasks: PV prediction for all parking lots and parking lots without sensors. The implementation of evaluation is shown in Fig. 7b.

- **Task 1: PV prediction for all parking lots.** This task aims to predict PVs for all parking lots among P in common scenarios where the sensor's missing issue is not severe. The missing ratio is fixed at 40%, and we report the average predicted results of the first 15 minutes, 30 minutes, 45 minutes, and 60 minutes, respectively.
- **Task 2: PV prediction for parking lots without sensors.** This task mimics some real-world scenarios, where the parking lot network suffers from a severe sensor missing problem. The missing ratio is set to 60% and 80% in this task for all datasets. Notably, in this task, we only calculate all model performance on parking lots among P_s, to examine how effectively our model can predict missing PVs under a high missing ratio. We report the average results of the predicted next one hour.

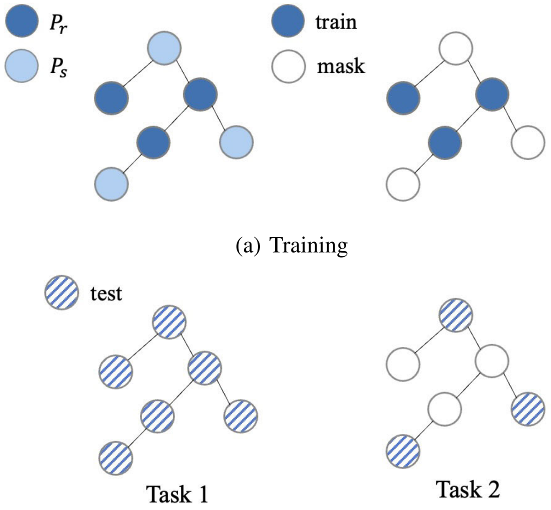
The selection of parking lots without sensors can affect model performance since the significance of PVs varies across nodes in the graph. For a fair comparison, P_s is randomly selected and shared by all the models before every run of the model. Given that some baselines are incapable of handling missing data, we average PVs of k nearest neighbors to impute missing one within the geographical range δ defined before.

D. Evaluation Metrics

We adopt two evaluation metrics, mean average error (MAE) and rooted mean square error (RMSE), for benchmark comparison. Smaller numbers represent the estimated values closer to the ground truths, indicating better performance.

TABLE II
DATASET DESCRIPTION

Discription	HK	SF
No. of nodes	517	86
No. of time steps	8,640	12,096
No. of POIs	367,244	113,993
No. of POI categories	803	146



(a) Training

(b) Example of two evaluation tasks. Task 1 aims to test the prediction accuracy of all parking lots among P. Task 2 aims to test the imputation precision of parking lots without sensors among P_s.

Fig. 7. The implementation of training and testing dataset.

TABLE III

PERFORMANCE COMPARISON ON PREDICTION ACCURACY OF P WITH DIFFERENT PREDICTION TIME STEPS

Dataset	**Models**	15min		30min		45min		60min	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
HK	HA	1.41	2.89	1.45	2.90	1.48	2.91	1.50	2.92
	STGCN	1.16	2.48	1.21	2.54	1.23	2.60	1.28	2.66
	DCRNN	1.15	2.99	1.27	3.08	1.36	3.18	1.43	3.27
	ASTGCN	1.24	2.45	1.22	2.51	1.27	2.57	1.37	2.68
	Graph WaveNet	1.14	2.99	1.23	3.09	1.31	3.19	1.37	3.28
	AGCRN	1.04	2.37	1.10	2.43	1.15	2.50	1.19	2.57
	STGODE	1.25	2.96	1.35	3.05	1.41	3.15	1.48	3.25
	TGNM	1.04	2.44	1.15	2.53	1.22	2.61	1.30	2.67
	CLaP (ours)	**0.94**	**2.32**	**0.98**	**2.40**	**1.02**	**2.46**	**1.06**	**2.53**
	CLaP-SP	1.16	2.46	1.20	2.52	1.23	2.58	1.26	2.64
	CLaP-GCN	1.03	2.42	1.06	2.49	1.09	2.55	1.16	2.62
SF	HA	2.23	3.65	2.32	3.69	2.38	3.73	2.44	3.77
	STGCN	1.95	3.34	2.08	3.48	2.21	3.60	2.32	3.71
	DCRNN	1.85	3.59	1.98	3.66	2.08	3.72	2.17	3.78
	ASTGCN	1.87	3.38	1.99	3.54	2.14	3.69	2.26	3.82
	Graph WaveNet	1.85	3.59	1.99	3.68	2.09	3.75	2.18	3.83
	AGCRN	1.84	3.52	2.00	3.67	2.13	3.80	2.24	3.91
	STGODE	1.91	3.57	2.02	3.64	2.10	3.69	2.16	3.73
	TGNM	1.85	3.50	2.11	3.66	2.27	3.80	2.40	3.92
	CLaP (ours)	**1.32**	**2.79**	**1.50**	**2.99**	**1.68**	**3.19**	**1.81**	**3.33**
	CLaP-SP	1.74	3.01	1.89	3.20	1.97	3.34	2.12	3.54
	CLaP-GCN	1.51	3.01	1.68	3.20	1.82	3.36	1.95	3.50

E. Baselines

To compare with CLaP on task 1, we introduce the following baselines:

- **HA** utilizes the average PVs of historical time steps as the predicted one.
- **STGCN** captures spatial-temporal correlations through building stacked ST-Conv blocks, each containing temporal gated convolutions and spatial graph convolution neural networks [20].
- **DCRNN** uses bidirectional random walks on graphs to model spatial complexity and implements an encoder-decoder model to capture temporal dependencies [21].
- **ASTGCN** introduces a spatial-temporal attention mechanism to capture correlations through modeling recent, daily-periodic and weekly-periodic dependencies [22].
- **Graph WaveNet** introduces an adaptive graph learning module to capture temporal dependencies considering long historical sequences [17].
- **AGCRN** proposes a data-learning spatial graph construction method and utilizes GRU to capture temporal dependencies [23].
- **STGODE** adds semantic adjacency matrix into graph construction to better understand spatial correlations [24].
- **TGNM** captures the dynamic spatial associations on road structures through a BERT-extended module and mines traffic flow patterns based on a temporal graph neural network [25].

Among those, **STGCN**, **DCRNN**, **ASTGCN**, **Graph WaveNet**, **AGCRN** and **STGODE** are widely-used baselines for traffic prediction. Traffic prediction and PV prediction share commonalities to some extent. We choose those baselines to test the effectiveness of our model designed for data completely missing.

We also include two variants of our model to evaluate the effectiveness of the proposed core modules:

- **CLaP-SP** excludes the spatial-temporal imputation block (SPImp). Only *feaImp* is left in the recurrent recovering block.
- **CLaP-GCN** replaces collaborative graph aggregation with traditional GCN in *feaCoNN* and *valCoNN*.

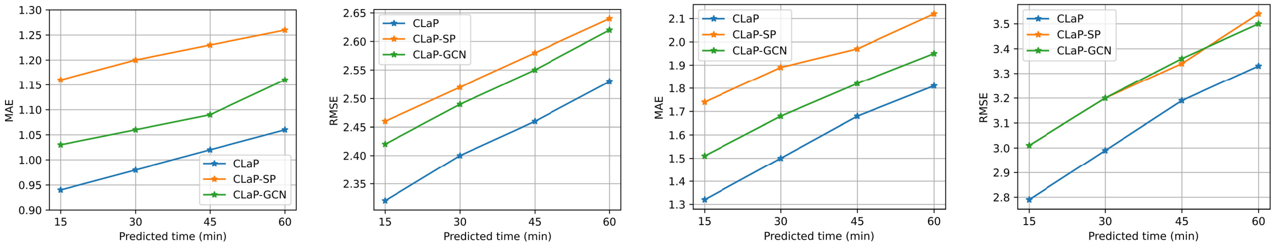
To demonstrate the superiority of CLaP on task 2, we introduce some baselines aiming to handle missing PVs:

- **KNN** is used to impute data based on neighbors within the range δ. Due to the limited available deep learning-based models handling missing PVs, we select KNN as one traditional imputation method to make a comparison.
- **BTTF** proposes a Bayesian temporal factorization for modeling large-scale spatiotemporal data in the presence of missing values and making a prediction [26].
- **GCN-LSTM** leverages traditional GCN and LSTM to capture spatial and temporal dependencies. The LSTM temporal module is reused to recover missing data considering temporal influences [9].
- **SHARE** proposes a hierarchical graph convolution structure to model spatial correlations and a parking availability approximation component to handle missing PVs [8].

F. Effectiveness Evaluation (Q1 & Q2)

We evaluate the overall performance of all baselines over two datasets to answer the first and second questions.

Performance on Task 1. The result of task 1 is shown in Table III. We notice that as the length of the predicted time step increases, all models show a slight decline in prediction accuracy. It may be due to the reason that it is still hard to handle long sequential information. The table



(a) MAE on the HK dataset (b) RMSE on the HK dataset (c) MAE on the SF dataset (d) RMSE on the SF dataset

Fig. 8. Effectiveness of proposed modules.

TABLE IV

PERFORMANCE COMPARISON ON IMPUTATION PRECISION OF P_s WITH DIFFERENT MISSING RATIOS

Dataset	Models	60%		80%	
		MAE	RMSE	MAE	RMSE
HK	KNN	2.34	3.89	2.75	4.65
	ASTGCN	1.66	3.70	1.67	3.77
	GCN-LSTM	1.69	3.53	1.71	3.66
	SHARE	1.70	3.55	1.74	3.67
	TGNM	1.74	4.22	1.83	4.24
	BTTS	1.80	3.85	2.13	4.45
	CLaP (ours)	**1.25**	**3.43**	**1.41**	**3.59**
SF	KNN	3.58	4.73	**3.54**	4.84
	ASTGCN	4.43	5.32	6.27	7.34
	GCN-LSTM	5.06	5.96	9.20	10.17
	SHARE	4.55	5.44	8.59	9.37
	TGNM	3.76	4.81	3.86	4.93
	BTTS	3.19	4.49	4.96	6.86
	CLaP (ours)	**2.89**	**3.74**	3.55	**4.57**

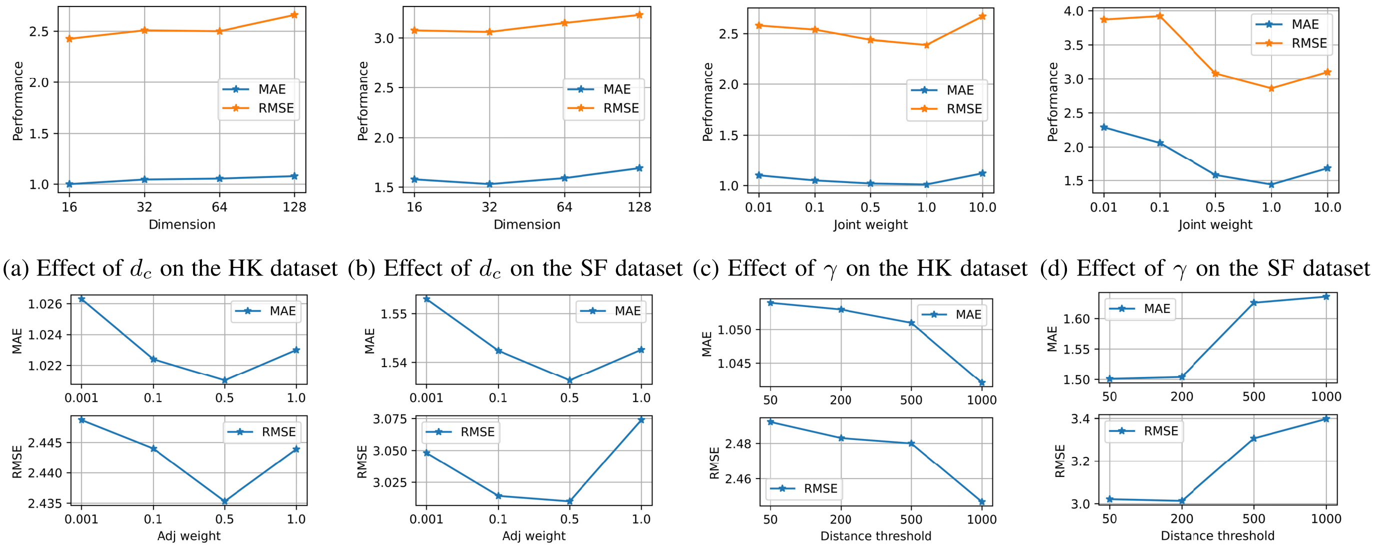
shows that CLaP significantly outperforms all other baselines on two datasets, demonstrating our model's superiority. The exceeding is because these models are not designed to handle inputs with missing PVs. Traditional data imputation before model implementation introduces slight differences from real-observed PVs, aggravating the model's prediction accuracy. Besides, both MAE and RMSE scores of the HK dataset are relatively smaller than those of the SF dataset. The possible reason for the small testing scores may be the larger number of trainable nodes and the denser spatial distributions of nodes in Hong Kong. Test results on the HK dataset achieve an average 19.8% improvement on MAE and an average 13.0% improvement on RMSE, which demonstrate a less significant enhancement compared with the 22.6% and 16.1% improvement tested on the SF dataset. Because of the fewer obtainable neighbor nodes based on the spatial graph in the SF dataset, traditional data imputation methods will bring more errors, and deep-learning-based PV prediction models will also fail to aggregate useful node attribute information, thus leading to worse prediction results. CLaP solves both the data missing and the sparse spatial distribution problems, which are much more severe in the SF dataset, and hence it demonstrates a more significant improvement in SF.

Performance on Task 2. The result of task 2 is shown in Table IV. From the table, we have the following observations. **First, a larger missing ratio leads to worse imputation precision for those parking lots without sensors.** It may be due to severely corrupted spatial and temporal dependencies. Under this scenario, the PVs of neighbors will be highly inaccessible while distant nodes may not have similar PV patterns. Thus the irrelevant PV patterns introduce bias for the imputation process, causing a significant prediction error. **Second, the performance of deep learning-based models handling missing PVs loses to CLaP, especially on the SF dataset.** The performance improvement is attributed to jointly modeling consecutive historical time steps in PV imputation and dynamic spatial dependencies. So CLaP performs consistently better than other baselines on two datasets. **Third, CLaP performs the most stable as increasing missing ratio.** Although the baselines capture the spatial dependencies to some extent, they do not model deep interactions between spatial and temporal dependencies while jointly considering the POI influence. In the HK experiment setting, since there exists a relatively large quantity of nodes and the spatial connectivities between nodes are comparatively dense, an extreme 80% missing ratio can still construct a spatial graph, which helps make PV prediction. That is the reason why other models perform not so badly when testing on HK with an 80% missing ratio. However, the number of nodes in the SF graph is much fewer, and an extreme 80% missing ratio will severely corrupt the spatial correlations, which makes the prediction a task for time-series modeling with few trainable samples. As a result, KNN even performs the best in this extreme case, and other deep learning-based models demonstrate extraordinarily bad results. From the results, CLaP is more robust to the large sensor missing ratio, demonstrating the effectiveness of the proposed imputation module.

Ablation Study. The ablation study result is shown in Fig. 8. The performance gap between the two variants and CLaP shows the effectiveness of our proposed modules. CLaP-SP excludes *STImp*, which does not consider coupling the spatial-temporal dependencies when dealing with missing data. The superiority of CLaP compared with CLaP-SP demonstrates that coupling the dependencies between real-time PVs is beneficial for imputing missing data. CLaP-GCN replaces collaborative graph learning with traditional graph convolution networks and loses to CLaP by a wide margin, which verifies the effectiveness of incorporating node attributes into graph augmentation to model the dynamic spatial dependencies.

G. Parameter Sensitivity Analysis (Q3)

In this subsection, we now explore the impact of different hyper-parameters, including dimension d_c over *feaCoNN* and



(a) Effect of d_c on the HK dataset (b) Effect of d_c on the SF dataset (c) Effect of γ on the HK dataset (d) Effect of γ on the SF dataset

(e) Effect of α on the HK dataset (f) Effect of α on the SF dataset (g) Effect of δ on the HK dataset (h) Effect of δ on the SF dataset

Fig. 9. Parameter sensitivity analysis.

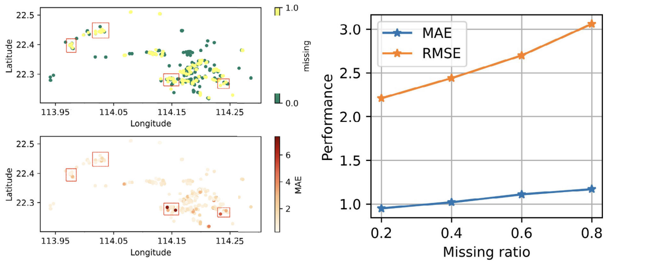
valCoNN, joint loss weight γ, adjacency α, and distance threshold δ to investigate the third question. The results are collected from all parking lots P on the two datasets.

CoNN dimension d_c represents the initialized node feature dimension of the augmented graph. Figs. 9a and 9b show that the increasing embedding dimension generally harms model performance. And setting $d_c = 16$ and 32 for HK and SF datasets are enough to achieve the best model performance.

Joint weight γ serves as a balancing factor between PV prediction loss and imputation precision loss. And the results are put on Figs. 9c and 9d. We notice that CLaP performs better than drops as γ keeps rising. We infer that when γ is small, raising it pushes the model to attend to a more precise PV approximation and then gets good performance. In contrast, when it is large, raising the weight makes the model focus on adapting to the random fluctuations in PVs, causing the overfitting problem. Therefore an appropriate γ selection is vital to model performance.

Adjacency weight α refers to a trade-off factor between interaction from the neighbor node set and interaction from the feature node set for aggregation. The value range is from 0 to 1. A value of either 0 or 1 corresponds to the exclusion of one of the two interactions. We display the results on two datasets in Figs. 9e and 9f and find that the curves are U-shaped. The best performance is achieved when the adjacency weight equals around 0.5, demonstrating the necessity of the two interactions.

Distance threshold δ defines how parking lots are connected in the network based on adjacency matrix A, where $A_{i,j} = 1$ when the geographical distance between p_i and p_j is smaller than the threshold δ. δ is difficult to define since parking lots are distributed dissimilarly in different cities. We decide to select the distance threshold based on the performance of the validation set. From Figs. 9g and 9h, δ is chosen to be 1000 and 200 meters to achieve the best performance.

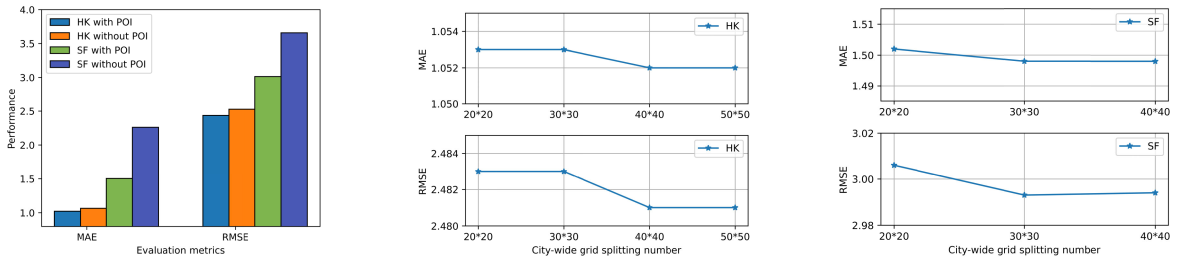


(a) PV distribution in Hong Kong (b) Missing ratio's effect on PV prediction

Fig. 10. Studies on missing PVs. Fig. 10b shows the missing PV distribution in our experiment setting above, and the corresponding predicted accuracy (MAE) below.

H. Impact of Missing Data (Q4)

We conduct two groups of experiments to investigate how missing ratios would affect model performance on the HK dataset. In the first group, we visualize all parking lots on a spatial map based on their geographic coordinates and plot the result in Fig. 10a. In the top subfigure, green circles represent the parking lots with sensors, while yellow circles represent those without sensors. In the bottom subfigure, the shades of the color depict the MAE results predicted by the well-trained CLaP model, wherein a lighter shade signifies superior performance. We note that it is challenging to accurately predict future PVs for parking lots without sensors, especially for those whose neighbors also suffer from the severe sensors missing problem. In the second group, we vary the missing ratio and test the prediction performance of all parking lots for



(a) Effectiveness of POI (b) Splitting number in the HK dataset (c) Splitting number in the SF dataset

Fig. 11. POI-related analysis.

CLaP. It is reasonable that the increasing number of equipped sensors will improve the ability to predict future PVs, which is shown in Fig. 10b.

I. Impact of Point-of-Interests (Q5)

To add additional contextual information on parking lots, we select POIs and generate rich node features in *POIAttn*. In this subsection, we conduct several experiments to test the effectiveness of introducing POIs on the HK and SF datasets.

POIs positively impact the model performance. As shown in Fig. 11a, both the MAE and RMSE scores demonstrate an improvement in the introduction of POI information for the two datasets. We notice a relatively notable improvement in the SF dataset after dynamic POI information is incorporated, which demonstrates the importance of additional node attributes when PV features are missing in the PV prediction scene.

To incorporate rich POI features for each parking lot in an efficient way, we split the city map into several areas and generate weighted POI distributions based on the distance between parking lots and areas. It is difficult to decide the splitting criteria in reality because of the unknown relationships between POIs and parking patterns. We choose the best splitting criteria based on the accuracy performance tested on the validation dataset. Based on the evaluation scores shown in Figs. 11b and 11c, we select 40 × 40 and 30 × 30 for splitting with the sizes around 0.7 and 0.2 square kilometers for HK and SF, respectively.

V. Related Work

A. Graph Convolution Network

Convolution in graphs shows a more flexible extraction of local patterns than traditional convolution implemented on grid-based data. Recently, spatial and spectral-based graph convolution [15], [27] have shown significant results in various downstream tasks [28], [29], [30](e.g., node representation, link prediction, etc.). Thanks to GCN's effectiveness, many existing works targeting spatial-temporal prediction [31], [32], [33], [34] leverage the combination of GCN and RNN-based methods, showing promising results. However, traditional graph convolution only considers complex and dynamic spatial correlations as closeness-based static graph structures locally, which cannot handle complicated global information transfer in our research setting.

B. Traffic Prediction

Traffic prediction has recently become a hot topic, referring to the prediction of traffic-related data, including speed, volumes, congestion, etc. Traffic prediction relies on modeling the spatial-temporal correlations given historical information. STGCN [20], DCRNN [21], ASTGCN [22], Graph WaveNet [17], AGCRN [23] and STGODE [24] design different graphs and propagation methods to capture spatial dependencies and utilize RNN and its variant to model temporal dependencies. Recently, efforts have been involved in traffic prediction with data missing. TGNM [25] imputes missing nodes by capturing the dynamic spatial associations on road structures through a BERT-extended module and mines traffic flow patterns. Traffic prediction shares similar model backbones and experimental settings with parking vacancy prediction to some extent, and can be viewed as a more generalized topic in traffic.

C. Parking Vacancy Prediction

The early practices consider parking vacancy prediction a classification problem [35] and conduct a comparative analysis using traditional machine learning techniques, including K-Nearest Neighbors, Random Forest, and Voting Classifier. Recently, deep learning-based models [36], [37], [38] have been extensively applied to PV prediction. Du-parking [39] incorporates mobility and meteorological features to parking lots to optimize the prediction of spatial-temporal correlations between parking lots, which targets fine-grained classification. A hybrid model that stacks two RNN components is proposed to take advantage of historical knowledge [40]. Current efforts and Mepark [4], [5] combine both GCN and Long short-term memory (LSTM) to capture spatial and temporal correlations based on observed historical data and other external factors. CPPM [41] captures spatial correlations based on similarities between parking lots with the aid of street image information and generates related features to model spatial-temporal correlations. ALL [42] utilizes reinforcement learning and federated learning to deal with the small sample issues faced by PV prediction. However, those methods fail to investigate intricate spatial correlations in city-wide parking scenarios and are not able to handle parking lots without sensors since complete historical PV sequences are required for model implementation.

VI. Conclusion

In this paper, we introduce CLaP, an effective framework for parking vacancy prediction with sensors partially missing. Our contributions are two-fold: (i) We explore to model the dynamic spatial dependencies by incorporating node attributes to augment parking networks. (ii) We jointly leverage neighbor representations at different historical time spans to impute missing PV embeddings. The extensive experiments demonstrate that our model outperforms state-of-the-art baselines regarding stability and effectiveness. Our model also enjoys high flexibility. For example, *feaCoNN* can be conducted recurrently by other related dynamic external factors (e.g., POI check-ins, trajectories) and is expected to show better prediction results. Our future work is to explore the influence of those dynamic external factors on model performance. Besides, since missing is an important issue in our work, we are interested in investigating whether and how including "key nodes" will influence the prediction performance. Moreover, our model can be utilized in other downstream spatial-temporal prediction tasks facing data scarcity issues besides PV prediction.

Acknowledgment

The authors thank the Transport Department of the Government of Hong Kong, SAR, for providing the on-street parking data. The contents of this article reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein.

References

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

[2] G. S. Tewolde, "Sensor and network technology for intelligent transportation systems," in *Proc. IEEE Int. Conf. Electro/Information Technol.*, May 2012, pp. 1–7.

[3] T. Shen, Y. Hong, M. M. Thompson, J. Liu, X. Huo, and L. Wu, "How does parking availability interplay with the land use and affect traffic congestion in urban areas? The case study of Xi'an, China," *Sustain. Cities Soc.*, vol. 57, Jun. 2020, Art. no. 102126.

[4] S. Yang, W. Ma, X. Pi, and S. Qian, "A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources," *Transp. Res. C, Emerg. Technol.*, vol. 107, pp. 248–265, Oct. 2019.

[5] D. Zhao et al., "MePark: Using meters as sensors for citywide on-street parking availability prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7244–7257, Jul. 2022.

[6] Y. Feng, Y. Xu, Q. Hu, S. Krishnamoorthy, and Z. Tang, "Predicting vacant parking space availability zone-wisely: A hybrid deep learning approach," *Complex Intell. Syst.*, vol. 8, no. 5, pp. 4145–4161, Oct. 2022.

[7] X. Yu and P. D. Prevedouros, "Performance and challenges in utilizing non-intrusive sensors for traffic data collection," *Adv. Remote Sens.*, vol. 2, no. 2, pp. 45–50, 2013.

[8] W. Zhang, H. Liu, Y. Liu, J. Zhou, and H. Xiong, "Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 1186–1193.

[9] W. Zhong, Q. Suo, X. Jia, A. Zhang, and L. Su, "Heterogeneous spatio-temporal graph convolution network for traffic forecasting with missing values," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2021, pp. 707–717.

[10] Y. Liu, C. Liu, X. Lu, M. Teng, H. Zhu, and H. Xiong, "Point-of-interest demand modeling with human mobility patterns," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 947–955.

[11] C. Zhu, A. Mehrabi, Y. Xiao, and Y. Wen, "CrowdParking: Crowdsourcing based parking navigation in autonomous driving era," in *Proc. Int. Conf. Electromagn. Adv. Appl. (ICEAA)*, Sep. 2019, pp. 1401–1405.

[12] J. Ye, J. Zhao, K. Ye, and C. Xu, "How to build a graph-based deep learning architecture in traffic domain: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 3904–3924, May 2022.

[13] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–12.

[14] Z. Cui, L. Lin, Z. Pu, and Y. Wang, "Graph Markov network for traffic forecasting with missing data," *Transp. Res. C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102671.

[15] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *Stat*, vol. 1050, p. 20, Oct. 2017.

[16] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 933–941.

[17] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 1907–1913.

[18] M. Ye, P. Yin, W. Lee, and D. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proc. Int. Conf. SIGIR*, 2011, pp. 325–334.

[19] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[20] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3634–3640.

[21] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.

[22] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 922–929.

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

[24] Z. Fang, Q. Long, G. Song, and K. Xie, "Spatial–temporal graph ODE networks for traffic flow forecasting," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 364–373.

[25] P. Wang, Y. Zhang, T. Hu, and T. Zhang, "Urban traffic flow prediction: A dynamic temporal graph network considering missing values," *Int. J. Geographical Inf. Sci.*, vol. 37, no. 4, pp. 885–912, Apr. 2023.

[26] X. Chen and L. Sun, "Bayesian temporal factorization for multidimensional time series prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 4659–4673, Sep. 2022.

[27] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[28] S. Pan, R. Hu, S.-F. Fung, G. Long, J. Jiang, and C. Zhang, "Learning graph embedding with adversarial training methods," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2475–2487, Jun. 2020.

[29] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 4805–4815.

[30] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, "MGAE: Marginalized graph autoencoder for graph clustering," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 889–898.

[31] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 1, pp. 1234–1241.

[32] M. Li and Z. Zhu, "Spatial–temporal fusion graph neural networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, 2021, pp. 4189–4196.

[33] K. Tian, J. Guo, K. Ye, and C.-Z. Xu, "ST-MGAT: Spatial–temporal multi-head graph attention networks for traffic forecasting," in *Proc. IEEE 32nd Int. Conf. Tools Artif. Intell. (ICTAI)*, 2020, pp. 714–721.

[34] C. Tang, J. Sun, Y. Sun, M. Peng, and N. Gan, "A general traffic flow prediction approach based on spatial–temporal graph attention," *IEEE Access*, vol. 8, pp. 153731–153741, 2020.

[35] F. M. Awan, Y. Saleem, R. Minerva, and N. Crespi, "A comparative analysis of machine/deep learning models for parking space availability prediction," *Sensors*, vol. 20, no. 1, p. 322, Jan. 2020.

[36] A. Camero, J. Toutouh, D. H. Stolfi, and E. Alba, “Evolutionary deep learning for car park occupancy prediction in smart cities,” in *Proc. 12th Int. Conf.*, Kalamata, Greece. Cham, Switzerland: Springer, Jun. 2019, pp. 386–401.

[37] E. I. Vlahogianni, K. Kepaptsoglou, V. Tsetsos, and M. G. Karlaftis, “A real-time parking prediction system for smart cities,” *J. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 192–204, Mar. 2016.

[38] F. Yu, J. Guo, X. Zhu, and G. Shi, “Real time prediction of unoccupied parking space using time series model,” in *Proc. Int. Conf. Transp. Inf. Saf. (ICTIS)*, 2015, pp. 370–374.

[39] Y. Rong, Z. Xu, R. Yan, and X. Ma, “Du-parking: Spatio-temporal big data tells you realtime parking availability,” in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 646–654.

[40] C. Zeng, C. Ma, K. Wang, and Z. Cui, “Parking occupancy prediction method based on multi factors and stacked GRU-LSTM,” *IEEE Access*, vol. 10, pp. 47361–47370, 2022.

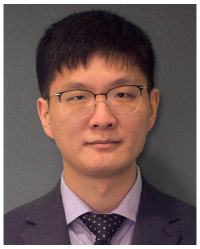
[41] S. Gong et al., “Spatio-temporal parking occupancy forecasting integrating parking sensing records and street-level images,” *Int. J. Appl. Earth Observ. Geoinf.*, vol. 118, Apr. 2023, Art. no. 103290.

[42] H. Qu, S. Liu, J. Li, Y. Zhou, and R. Liu, “Adaptation and learning to learn (ALL): An integrated approach for small-sample parking occupancy prediction,” *Mathematics*, vol. 10, no. 12, p. 2039, Jun. 2022.

Beiyu Song received the bachelor’s degree in information security from Beijing University of Posts and Telecommunications in 2020 and the master’s degree from the Department of Applied Mathematics, The Hong Kong Polytechnic University, in 2023, where she is currently pursuing the M.Phil. degree with the Department of Electrical and Electronic Engineering.

Huachi Zhou received the B.S. degree in software engineering from East China Normal University in 2020, and the master’s degree from the Department of Applied Mathematics, The Hong Kong Polytechnic University (PolyU), in 2022, where he is currently pursuing the Ph.D. degree with the Department of Computing. Before studying with PolyU, he was an Intern with Morgan Stanley, Bilibili, and TCL Corporate Research. He has published several papers in top conferences/journals, e.g., SIGIR, CIKM, and IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.

Xiao Huang received the B.S. degree in engineering from Shanghai Jiao Tong University in 2012 and the Ph.D. in computer engineering from Texas A&M University in 2020. He is an Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University (PolyU). Before joining PolyU, he was a Research Intern with Microsoft Research and Baidu USA. He has published over 40 papers in top conferences/journals. He is a Program Committee Member of NeurIPS (2021–2023), AAAI (2021–2023), ICLR (2022–2023), KDD (2019–2023), TheWebConf (2022–2023), ICML (2021–2023), IJCAI (2020–2023), CIKM (2019–2022), WSDM (2021–2023), SDM (2022), and ICKG (2020–2021).



Wei Ma (Member, IEEE) received the bachelor’s degree in civil engineering and mathematics from Tsinghua University, China, and the master’s degree in machine learning and civil and environmental engineering and the Ph.D. degree in civil and environmental engineering from Carnegie Mellon University, USA. He is currently an Assistant Professor with the Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University (PolyU). His research focuses on intersection of machine learning, data mining, and transportation network modeling, with applications for smart and sustainable mobility systems. He has received the 2020 Mao Yisheng Outstanding Dissertation Award, the Best Paper Award (theoretical track) at INFORMS Data Mining and Decision Analytics Workshop, and the 2022 Kikuchi Karlaftis Best Paper Award of the TRB Committee on Artificial Intelligence and Advanced Computing Applications.