9-2021

Holistic prediction for public transport crowd flows: A spatio dynamic graph network approach

Bingjie HE

Shukai LI

Chen ZHANG

Baihua ZHENG
Singapore Management University, bhzheng@smu.edu.sg

Fugee TSUNG

Citation

1

Holistic Prediction for Public Transport Crowd Flows: A Spatio Dynamic Graph Network Approach

Bingjie He[1], Shukai Li[2], Chen Zhang[1](✉), Baihua Zheng[3], and Fugee Tsung[4]

[1] Industrial Engineering, Tsinghua University, Beijing, China
hebj20@mails.tsinghua.edu.cn, zhangchen01@tsinghua.edu.cn
[2] State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China shkli@bjtu.edu.cn
[3] School of Information Systems, Singapore Management Universtiy, Singapore
bhzheng@smu.edu.sg
[4] Industrial Engineering and Decision Analytics, The Hong Kong University of Science and Technology, Hong Kong season@ust.hk

Abstract. This paper targets at predicting public transport in-out crowd flows of different regions together with transit flows between them in a city. The main challenge is the complex dynamic spatial correlation of crowd flows of different regions and origin-destination (OD) paths. Different from road traffic flows whose spatial correlations mainly depend on geographical distance, public transport crowd flows significantly relate to the region's functionality and connectivity in the public transport network. Furthermore, influenced by commuters' time-varying travel patterns, the spatial correlations change over time. Though there exist many works focusing on either predicting in-out flows or OD transit flows of different regions separately, they ignore the intimate connection between the two tasks, and hence lose efficacy. To solve these limitations in the literature, we propose a Graph spAtio dynamIc Network (GAIN) to describe the dynamic non-geographical spatial correlation structures of crowd flows, and achieve holistic prediction for in-out flows of each region together with OD transit flow matrix between different regions. In particular, for spatial correlations, we construct a dynamic graph convolutional network for the in-out flow prediction. Its graph structures are dynamically learned from the prediction of OD transit flow matrix, whose spatial correlations are further captured via a multi-head graph attention network. For temporal correlations, we leverage three blocks of gated recurrent units, which capture minute-level, daily-level and weekly-level temporal correlations of crowd flows separately. Experiments on real-world datasets are used to demonstrate the efficacy and efficiency of GAIN.

Keywords: crowd flows prediction · origin-destination matrix · dynamic spatial correlation · public transport system · graph attention network.

1 Introduction

The public transport system is the backbone for human mobility in urban areas. Accurate prediction of public transport flow is greatly important to both system operators and passengers. It allows operators to conduct better train operation planning, detect potential abnormal traffic flows and render fast remedial strategies. It also provides real-time traffic information to passengers for better travel planning. Currently, the Automated Fare Collection System (AFC), widely used in urban public transport network, provides a convenient way for passenger travel data collection. The AFC data (aka smart card data) record passengers' each trip information, including the boarding and alighting time and corresponding stations, and offer big support for crowd flow analysis. Generally, two types of crowd flows are of interest. The first is the in-out flow of each region, which measures the number of passengers entering/leaving the region via public transport systems for each time step. The second is the finer-grain transit flow from each origin region to another destination region, i.e., Origin-Destination (OD) matrix prediction. How to utilize AFC data to predict these two levels of crowd flows has been raising researchers' interest in data mining for intelligent transportation system construction. There are several challenges involved.

The first and most critical is the complex spatial correlation of crowd flow data. Unlike road traffic flows, where the spatial correlations for different regions are based on the "first law of geography", i.e., near things are more related than distant things. However, for public transport systems, the spatial correlations are not fully based on geographical distance, but also the connectivity structure of the public transportation network and the region functionality. For example, though two non-adjacent regions are far away from each other on the map, they could be directly connected by a metro line or located in the same functional regions, and consequently share highly correlated crowd flow patterns. As such, traditional models trying to capture geographically neighboring spatial features are not suitable.

Furthermore, the spatial correlations are time-dynamic. For example, in the morning peak hour, the transport system carries hundreds of thousands of commuters from residential areas to the central business district (CBD), which leads to a high correlation between outflows of residential districts and inflows of CBD. This correlation dies down after the morning peak. In the evening, the outflows of CBD become more related to inflows of commercial regions, since people go for entertainment and leisure after work. At night, the outflows of commercial regions begin to relate to inflows of residential regions. However, most current studies assume that spatial correlations are static, and hence lose the prediction accuracy.

Last but foremost, predicting the two granularities of crowd flows, i.e., in-out flows and OD transit flows, have intimate connections with each other. The sum of passenger flows from one region to others is the outflow of that region. Likewise, destinations of OD transit flows determine the inflow of the corresponding regions. Consequently, accurate prediction of outflows in one region can help predict the transition flows from it to other regions more accurately, vice versa. Consequently, the OD transit flow matrix and in-out flows of regions mutually influence each other and a holistic prediction model with consideration of their connections is expected to have better prediction performance.

To address above issues, in this paper, we consider the connections between in-out flows and OD transit flows of different regions, and develop a genuine holistic framework for crowd flows prediction of the public transport system. In particular, we treat each region as a node in the graph, and propose a dynamic graph-based neural network framework to capture the dynamic spatial correlations of crowd flows of different nodes. First, we formulate a multi-head graph attention network (GAT) block for OD matrix prediction. The GAT model could dynamically leverage features from spatial correlated regions using the attention mechanism, and track the OD transit patterns of different regions accurately. Consider different kinds of spatial correlations may co-exist simultaneously, we apply the multi-head technique. In addition, the learned attention graphs of GAT can be really good representations of spatial correlation structures of different regions. Hence we further use them as the dynamic input graphs of the graph convolution network (GCN) block for in-out flow prediction of different regions. Consequently, the graph structure of GCN is not required to be pre-defined or dependent on any prior information, but is dynamically learned from the prediction process of the OD transit flow matrix. In this way, the two tasks are intimately interrelated with each other for joint training. Last, data from urban railway transit systems of Hong Kong and Shenzhen validate our proposed methodology. Extensive experiments and comparisons with state-of-the-art methods demonstrate the out-performance of our proposed method.

2 Literature Review

2.1 In-out Flow Prediction

Generally, crowd flows prediction is referred as in-out flow prediction of a region. It has been extensively studied in many literature works, among which deep learning-based methods are the current mainstream tools since they could effectively model temporal and spatial correlations of crowd flows simultaneously. Various network structures have been proposed and applied to solve different problems.

For temporal correlation description with neural network models, recurrent neural network (RNN) and its variants, e.g. long short-term memory (LSTM) and gated recurrent unit (GRU) have been widely applied. For example, in [1], LSTM units are built to model peak-hour and post-accident traffic state. In [2], the authors extended the fully-connected LSTM to have convolutional structures such that it can handle spatio-temporal data. In [3], a periodically shifted attention mechanism is introduced to handle the long-term periodic temporal shifting. Different methods have demonstrated competitive performances in different data sets.

For spatial correlation description, lines of study adopt convolutional neural network (CNN)-based structures to model nonlinear spatial characteristics [4, 5].

They learned traffic flows as heat map images and utilized convolutions to capture spatial correlations. The typical applications include taxi trajectory prediction, bike rent/return prediction, etc, where geographically nearby regions are important to help predict the target region. However, CNN is not suitable for public transport crowd flows data where correlation structures of flows between two regions are generally not only related to their geographical distance, but also a lot of other factors, such as spatial structures of the transportation network. Substantial research generalized the convolution operator to non-Euclidean data [6]. Among them, Graph Convolutional Neural Network (GCN) is a significant stride [7, 8]. GCN-based methods assume each region as a node in the graph, and spatial correlations between different regions are denoted as edge weights between nodes. GCN has been an appealing choice for public transport flow forecasting, where the graph is defined based on station connectivity, geo-graph attributes, contextual features (point-of-interest) [7], flow profile similarity, etc. One limitation of these works is that the prediction performance is greatly influenced by the pre-defined graph. Yet how to choose these various kinds of graphs case-by-case is a practically difficult problem depending on the specific application purpose. It's also difficult to evaluate which kind of graph is better, and some specific types of graphs are not suitable in general cases. There are also some attempts using attention strategy [8]. However, they still rely on pre-defined graph structures.

Last but foremost, all the above methods only predict the in-out flows of different regions, yet ignore OD transit flows between different regions.

2.2 OD Transit Flow Matrix Prediction

OD matrix forecasting aims at predicting transit flows between different regions. In [9], the authors proposed a contextualized spatial-temporal network, which incorporated diverse contextual information to predict taxi OD demand. In [10], the authors formulated the OD matrix together with other geographical features as tensors and developed a multi-scale convolutional LSTM for predicting future OD traffic demand. In [11], Multi-Perspective Graph Convolutional Networks (MPGCN) with LSTM is proposed to extract temporal features for OD matrix prediction. In [12], a matrix factorization-embedded graph CNN is proposed for road OD matrix prediction.

Yet these methods only consider OD matrix prediction, without taking in-out flow prediction into account. In [13], the authors first considered multi-task learning of OD matrix and in-out flows together. It developed a grid-embedding based multi-task learning framework to predict OD passenger demands, together with in-out flows of each region. Yet the two tasks are merely added together as one objective function without any information sharing. In [14], the authors further proposed a better information fusion framework. It first designed two separate CNN modules to extract features of OD matrix and in-out flows. Then the two tasks' features were concatenated together in a fusion module. However, this simple concatenation does not consider task differences carefully and does not design which features are shared and which ones should be task-specific. In [15], an adversarial network is proposed for OD matrix and in-out flow prediction. It adopts a shared-private framework which contains both private and shared spatial-temporal encoders and decoders. A discriminative loss on task classification and an adversarial loss on shared feature extraction are incorporated to reduce information redundancy. However, these methods target at road traffic flow prediction and assume neighboring regions are more correlated. Consequently, they are not suitable for public transport flow prediction. And none of them consider dynamic spatial correlations and hence have limited prediction power. Last, these models define one region's inflow (outflow) as directed sum of OD transit flows over all the destination (origin) regions. This indicates only OD trips completed in the same time step are considered for counting in-out flows, and the learned model is forced to capture the concurrent spatial correlations between in-out flow data and OD transit data. However, for trips in public transport systems, they generally take longer time than one time step and simple summation of OD transit flows cannot substitute in-out flows, leading the above models to fail to give accurate prediction, as shown in our case studies later.

3 Problem Formulation

We first introduce some basic notations and define the public transport crowd flows prediction problem formally.

Definition 1 *(Region)*: We partition the city into N non-overlapping regions. Each region denoted as $g(n), n = 1, \ldots, N$, can have irregular figure and different size, depending on geography of the public transport system. The whole grid map is represented by $\mathcal{M}_N$.

This definition is a bit different from some previous studies [4], which assume each region should be a rectangular, with in total of $I \times J$ grids based on longitude and latitude dimensions. This is because our analysis focuses on public transport systems, whose spatial connectivity is not critically dependent on the geographical distance between different regions. The city map could be partitioned according to functionality of different regions, points of interest, volume of crowd flows, etc. Furthermore, we need to remove certain regions without public transport stations inside.

Definition 2 *(Node)*: Given the map $\mathcal{M}_N$ with partitioned regions, we define the city graph with $V = \{v_1, v_2, \ldots, v_N\}$ as the node set. Each node corresponds to one region in $g(n), n = 1, \ldots, N$.

Definition 3 *(Inflow/Outflow)*: Let (τ, l) be spatial-temporal coordinates, with τ denotes a time step and l denotes a location. Define $\mathcal{P}$ as a set of trip data. Each trip is denoted by its origination information $o = (\tau_o, l_o)$ and destination information $d = (\tau_d, l_d)$. Here τ_o and τ_d represent the trip starting time and ending time respectively; l_o and l_d represent the origin and destination location respectively. Given the corresponding city graph, the in-out flow of node $v_n \in V$, whose corresponding region in $\mathcal{M}_N$ is $g(n)$, is defined as $\boldsymbol{y}_t^n \in \mathbb{R}^2$:

$$(\boldsymbol{y}_t^n)_1 = \left|\{(o,d) \in \mathcal{P} : l_d \in v_n \wedge \tau_d \in t\}\right|, \tag{1}$$

$$(\boldsymbol{y}_t^n)_2 = \left|\{(o,d) \in \mathcal{P} : l_o \in v_n \wedge \tau_o \in t\}\right|, \tag{2}$$

where $(\boldsymbol{y}_t^n)_1$ and $(\boldsymbol{y}_t^n)_2$ represent the inflow and outflow of region $g(n)$ respectively. The symbol $|\cdot|$ denotes the cardinality of the set. With abuse of notation, we further define $\boldsymbol{y}_t^{\text{in}} = [(\boldsymbol{y}_t^1)_1, \ldots, (\boldsymbol{y}_t^N)_1]^T \in \mathbb{R}^N$, $\boldsymbol{y}_t^{\text{out}} = [(\boldsymbol{y}_t^1)_2, \ldots, (\boldsymbol{y}_t^N)_2]^T \in \mathbb{R}^N$, and $\mathbf{Y}_t = [\boldsymbol{y}_t^{\text{in}}, \boldsymbol{y}_t^{\text{out}}] \in \mathbb{R}^{N\times 2}$.
Definition 4 *(OD transit flow matrix)*: Similarly, given data $\mathcal{P}$, and grid map $\mathcal{M}_N$ in time step t with corresponding city graph with nodes V, the OD transit flow matrix in time step t is defined as $\mathbf{S}_t \in \mathbb{R}^{N\times N}$:

$$(\mathbf{S}_t)_{mn} = \left|\{(o,d) \in \mathcal{P} : l_o \in v_n \wedge l_d \in v_m \wedge \tau_d \in t\}\right|, \tag{3}$$

where the corresponding region of v_n and v_m in $\mathcal{M}_N$ are $g(n)$ and $g(m)$, respectively. $(\mathbf{S}_t)_{mn}$ represents OD transit flow from node v_n to node v_m. Abusing notation a bit, the m^{th} row of OD transit flow matrix $\mathbf{S}_t$ is denoted as $\boldsymbol{s}_t^m$, which describes the OD transit flows from all other nodes to node m in time step t.
Problem: Our goal is to provide a holistic prediction framework for the in-out flows of each region and the OD transit flow matrix. Specifically, given the city region nodes $V = \{v_1, v_2, \ldots, v_N\}$, current time step t, and historical data $\mathbf{Y}_{t-s}, \ldots, \mathbf{Y}_t$, $\mathbf{S}_{t-s}, \ldots, \mathbf{S}_t$ for $s = 0, \ldots, t-1$, we propose a model to collectively predict $\mathbf{Y}_{t+1}$ and $\mathbf{S}_{t+1}$.

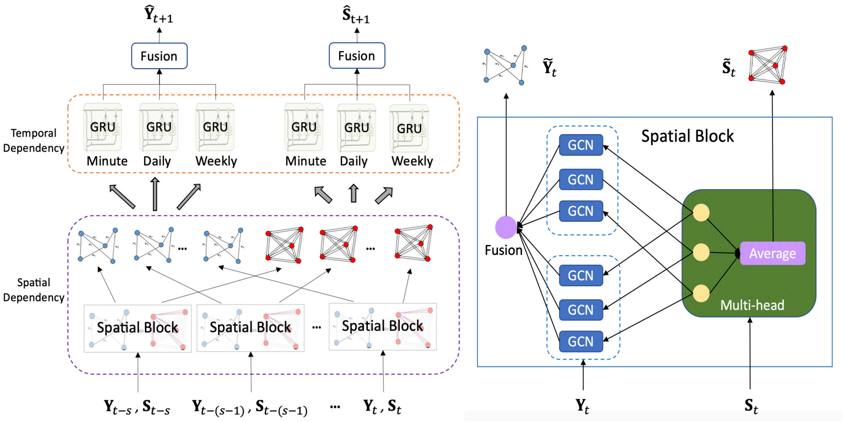
4 Methodology

The framework of the proposed Graph spAtio dynamIc Network (GAIN) is shown in Fig. 1a. For spatial correlations, we use the multi-head GAT block for OD flow prediction. Its learned dynamic attention network can effectively capture non-adjacent spatial correlations of different regions, and hence can be fed into the GCN block for in-out flow prediction. The two blocks cooperate with each other and achieve joint prediction. For temporal correlations, we connect the above spatial blocks to three blocks of GRU, which capture the minute-level, daily-level and weekly-level correlations respectively. Last, the outputs of the three blocks are fused together in the output layer for final prediction.

4.1 Spatial Correlation

We utilize graph networks to capture non-adjacent spatial correlations. Fig. 1b represents the structure of this block.

Spatial Correlation for OD Transit Flow Prediction First, we propose a GAT module to capture the spatial correlations of $\boldsymbol{s}_t^i$. To be specific, we first define a graph $G_t = \{V, E, \mathbf{A}_t\}$, where each node v_i is one region (as in Definition 2), E is the edge set, $\mathbf{A}_t$ is the adjacency matrix, and each weight represents the pairwise spatial relationship between two regions. We perform the information aggregation with the graph-based attention encoder to preserve high-order region-wise crowd relations from a global perspective. Its general idea is to learn which regions are able to attend in terms of their crowd flow patterns in a dynamic



(a) (b)

Fig. 1. (a) The structure of Graph spAtio dynamIc Network (GAIN); (b) Inner structure in Spatial Block.

way, i.e., how to aggregate both self-features and neighbor features for prediction for different time steps dynamically. Yet unlike GCN which requires to pre-define the adjacency matrix $\mathbf{A}_t$, GAT only requires the prior connectivity information E, i.e., whether there is an edge from v_i to v_j. As to the weight of the edge, it can be automatically learned via attention mechanisms.

In particular, to enhance the expressive power of feature representations during the graph-based aggregation process, we first perform linear transformation on the input feature $\boldsymbol{s}_t^i \in \mathbb{R}^N$ of node v_i with a shared parameterized weight matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$, i.e., $\mathbf{W}\boldsymbol{s}_t^i$. Then we compute a pairwise attention coefficient between v_i and v_j by concatenating the projected embeddings $\mathbf{W}\boldsymbol{s}_t^i$ and $\mathbf{W}\boldsymbol{s}_t^j$, and taking a dot product of them with a weight vector $\boldsymbol{c}$, i.e., $\alpha_{ij,t} = \boldsymbol{c}^T \left[\mathbf{W}\boldsymbol{s}_t^i \| \mathbf{W}\boldsymbol{s}_t^j\right]$, where $\|$ is the concatenation operation. The activation function of LeakyReLU and the softmax are further applied to generate the attention coefficient:

$$a_{ij,t} = \text{softmax}_j\left(LeakyReLU(\alpha_{ij,t})\right) = \frac{\exp\left(LeakyReLU(\alpha_{ij,t})\right)}{\sum_{v_k \in neigh(i)} \exp\left(LeakyReLU(\alpha_{ik,t})\right)}, \tag{4}$$

where $neigh(i)$ denotes the neighbour set of v_i defined by E. In this paper, we can suppose the graph is fully-connected without taking into account any prior information about the connectivity property of different regions. Alternatively, we can also use the connectivity structure of the public transport system, such as the urban railway transit map, as a reasonable prior of the connectivity property.

In addition, to capture different types of spatial correlations and improve the fitting ability of the self-attention, multi-head attention is employed in the mechanism, which uses the average of K parallel attention results as the updated features:

$$\tilde{\boldsymbol{s}}_t^i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{v_j \in neigh(i)} a_{ij,t}^k \mathbf{W}^k \boldsymbol{s}_t^j\right), \tag{5}$$

where $a_{ij,t}^k$ are normalized attention coefficients computed by the k^{th} attention mechanism. We denote the k^{th}-head attention graph as $G_{k,t} = \{V, E, \mathbf{A}_t^k\}$, and the output from GAT that has captured the spatial correlations of $\mathbf{S}_t$ as $\tilde{\mathbf{S}}_t$, with $\tilde{\boldsymbol{s}}_t^i$ as its i^{th} row.

Dynamic Spatial Correlation for Inflow/Outflow Prediction To capture dynamic and non-Euclidean spatial correlation structures, we design a dynamic GCN module. GCN conducts convolution over a graph with an adjacency matrix $\mathbf{A}$ where each element a_{ij} represents the spatial correlation between v_i and v_j. The general idea of GCN is to learn node representations by exchanging information among its correlated neighbours, and consequently extract the patterns hidden in the graphs.

It is noted that the learned edge weights $\alpha_{ij,t}^k$ from GAT can be regarded as good representations of dynamical spatial correlation structures of different regions. Thus we use $\mathbf{A}_t^k, k = 1, \ldots, K$ as graph inputs of GCN for in-out flow prediction. Specifically, we also employ the multi-head for the GCN block with the k^{th}-head adjacency matrix as $(\mathbf{A}_t^k)_{ij} = \alpha_{ij,t}^k$. As the OD transit flows evolve over time, $\alpha_{ij,t}^k$ also changes over time. Consequently, the dynamic correlation structures of both $\mathbf{Y}_t$ and $\mathbf{S}_t$ have been successfully described in the collaborative model.

Furthermore, consider the correlation structure between inflows of regions is different from that between outflows of regions, two GCNs are conducted for inflow and outflow, respectively. Take inflow for example, the input node features are $\boldsymbol{y}_t^{\text{in}}$, then the spectral convolutions on graph are defined as:

$$\widetilde{\boldsymbol{y}}_t^{\text{in},k} = g_\theta *_{G_{k,t}} \boldsymbol{y}_t^{in} = \mathbf{U}_t^k \mathrm{g}_\theta(\boldsymbol{\Lambda}_t^k) \mathbf{U}_t^{k^T} \boldsymbol{y}_t^{\text{in}}, \tag{6}$$

where $\mathbf{D}_t^k \in \mathbb{R}^{N \times N}$ is the diagonal degree matrix with the i^{th} diagonal element as $(\mathbf{D}_t^k)_{ii} = \sum_j (\mathbf{A}_t^k)_{ij}$; $\mathbf{L}_t^k = (\mathbf{D}_t^k)^{-1}\left(\mathbf{D}_t^k - \mathbf{A}_t^k\right)$ is the Laplacian matrix; $\boldsymbol{\Lambda}_t^k \in \mathbb{R}^{N \times N}$ and $\mathbf{U}_t^k$ are results of the eigenvalue decomposition of $\mathbf{L}_t^k = \mathbf{U}_t^k \boldsymbol{\Lambda}_t^k \mathbf{U}_t^{k^T}$. $\mathrm{g}_\theta(\boldsymbol{\Lambda}_t^k)$ is a function of the eigenvalues of $\mathbf{L}_t^k$, and can be localized in space and reduce learning complexity by a polynomial filter [6]. The GCN construction for outflow data $\boldsymbol{y}_t^{\text{out}}$ can be conducted in the same way, and get the output $\widetilde{\boldsymbol{y}}_t^{\text{out},k}$. Then we fuse the results from these GCNs:

$$\widetilde{\mathbf{Y}}_t = \mathrm{ReLU}\left(\sum_{k=1}^{K} \widetilde{\boldsymbol{y}}_t^{\text{in},k} \mathbf{W}^{\text{in},k} + \sum_{k=1}^{K} \widetilde{\boldsymbol{y}}_t^{\text{out},k} \mathbf{W}^{\text{out},k}\right), \tag{7}$$

where $\widetilde{\mathbf{Y}}_t \in \mathbb{R}^{N \times 2}$ are outputs from the GCNs, and $\mathbf{W}^{\text{in},k}, \mathbf{W}^{\text{out},k} \in \mathbb{R}^{1 \times 2}$ are parameters to be learned.

4.2 Temporal Correlation

Now we talk about temporal correlation modeling for $\mathbf{Y}_t$ and $\mathbf{S}_t$. Consider training long-term temporal information is a nontrivial task. To address this issue, we explicitly model relative historical time steps by capturing the minute-level, daily-level and weekly-level correlations separately [4]. For each time level, we construct the GRU cells using $\tilde{\mathbf{Y}}_t$ and $\tilde{\mathbf{S}}_t$ as input.

Take daily-level feature extraction of $\widetilde{\mathbf{Y}}_t$ for example. After capturing spatial correlation features in the GCN block, we first use a flatten layer to transform $\widetilde{\mathbf{Y}}_t \in \mathbb{R}^{N\times 2}$ to a feature vector $\tilde{\boldsymbol{y}}_t \in \mathbb{R}^{2N}$. The sequence to be inputted in GRU is $\{\tilde{\boldsymbol{y}}_{t+1-l_d\cdot d}, \tilde{\boldsymbol{y}}_{t+1-(l_d-1)\cdot d}, ...\tilde{\boldsymbol{y}}_{t+1-d}\}$, where d is the number of time steps in one day and l_d is the considered maximum lag for daily-level feature extraction. Then the GRU captures the temporal correlations of $\tilde{\boldsymbol{y}}_t$ as:

$$\boldsymbol{z}_t^d = \sigma\left(\mathbf{W}_z^d \tilde{\boldsymbol{y}}_t + \mathbf{U}_z^d \boldsymbol{h}_{t-1}^d + \boldsymbol{b}_z^d\right), \tag{8}$$

$$\boldsymbol{r}_t^d = \sigma\left(\mathbf{W}_r^d \tilde{\boldsymbol{y}}_t + \mathbf{U}_r^d \boldsymbol{h}_{t-1}^d + \boldsymbol{b}_r^d\right), \tag{9}$$

$$\tilde{\boldsymbol{h}}_t^d = \tanh\left(\mathbf{W}_h^d \tilde{\boldsymbol{y}}_t + \mathbf{U}_h^d \left(\boldsymbol{r}_t^d \circ \boldsymbol{h}_{t-1}^d\right) + \boldsymbol{b}_h^d\right), \tag{10}$$

$$\boldsymbol{h}_t^d = (1 - z_t) \circ \boldsymbol{h}_t^d + z_t \circ \widetilde{\boldsymbol{h}}_{t-1}^d. \tag{11}$$

The output of the last layer GRU, denoted as $\boldsymbol{h}_{t+1}^d \equiv \hat{\boldsymbol{y}}_{t+1}^d$, represents the daily-level temporal feature. And then we reshape it into $\hat{\mathbf{Y}}_{t+1}^d \in \mathbb{R}^{N\times 2}$. Similarly, we can input the minute-level sequence $\{\tilde{\boldsymbol{y}}_{t+1-l_m\cdot}, \tilde{\boldsymbol{y}}_{t+1-(l_m-1)}, ..., \tilde{\boldsymbol{y}}_t\}$ where l_m is the considered maximum lag for minute-level feature extraction, and get $\hat{\mathbf{Y}}_{t+1}^m$. We input the weekly-level sequence $\{\tilde{\boldsymbol{y}}_{t+1-l_w\cdot w}, \tilde{\boldsymbol{y}}_{t+1-(l_w-1)\cdot w}, ..., \tilde{\boldsymbol{y}}_{t+1-w}\}$ where w equals the number of time steps in one week and l_w is the maximum lag for weekly-level feature extraction, and get $\hat{\mathbf{Y}}_{t+1}^w$. Likewise, we can construct another three GRU blocks to capture temporal relationship of $\mathbf{S}_{t+1}$ and get the minute-level, daily-level and weekly-level components $\hat{\mathbf{S}}_{t+1}^m$, $\hat{\mathbf{S}}_{t+1}^d$, $\hat{\mathbf{S}}_{t+1}^w$ respectively.

4.3 Fusion

Last, we combine results from the three GRUs together for final prediction by parametric-matrix-based fusion with tanh hyperbolic function:

$$\widehat{\mathbf{Y}}_{t+1} = \tanh\left(\mathbf{W}_m^1 \circ \hat{\mathbf{Y}}_{t+1}^m + \mathbf{W}_d^1 \circ \hat{\mathbf{Y}}_{t+1}^d + \mathbf{W}_w^1 \circ \hat{\mathbf{Y}}_{t+1}^w\right), \tag{12}$$

$$\widehat{\mathbf{S}}_{t+1} = \tanh\left(\mathbf{W}_m^2 \circ \hat{\mathbf{S}}_{t+1}^m + \mathbf{W}_d^2 \circ \hat{\mathbf{S}}_{t+1}^d + \mathbf{W}_w^2 \circ \hat{\mathbf{S}}_{t+1}^w\right), \tag{13}$$

where $\circ$ is element-wise multiplication, and $\mathbf{W}_m^1$, $\mathbf{W}_d^1$, $\mathbf{W}_w^1$, $\mathbf{W}_m^2$, $\mathbf{W}_d^2$, $\mathbf{W}_w^2$ are parameters to be learned to represent impacts of different components.

The final loss function adopts mean squared error between the true flows and the predicted ones:

$$\mathcal{L}(\theta) = \lambda_{\text{region}} \left\|\mathbf{Y}_{t+1} - \widehat{\mathbf{Y}}_{t+1}\right\|^2 + \lambda_{OD} \left\|\mathbf{S}_{t+1} - \widehat{\mathbf{S}}_{t+1}\right\|^2, \tag{14}$$

where λ_{region} and λ_{OD} are adjustable hyper-parameters, and θ indicates all parameters in GAIN.

Remark 1. Note that from Definition 3 and 4, the outflow of a region can be computed by summing all the OD transit flows whose origin is that region. As such, we may add a regularization term to penalize the difference between the predicted $\widehat{\boldsymbol{y}}_{t+1}^{out}$ and $\widehat{\mathbf{S}}_{t+1}\mathbf{1}$ where $\mathbf{1} \in \mathbb{R}^{N\times 1}$ is a vector with all components equal to 1, i.e., $\lambda_{\text{lim}} \left\| \widehat{\boldsymbol{y}}_{t+1}^{out} - \widehat{\mathbf{S}}_{t+1}\mathbf{1} \right\|^2$ in (14) with adjustable hyper-parameter λ_{lim}.

5 Experiments and Results

5.1 Experimental Settings

Data In our experiment, we consider two large-scale real-world datasets for performance evaluation, which contain smart card data from the corresponding AFC systems as follows.

- **Hong Kong Dataset** (HK): The dataset contains passengers' railway trip records in HK from Jan 1^{st} 2017 to Feb 28^{th} 2017. We use the first 52 days for training, and the remained 7 days for testing. We split the whole city as 40×60 regions, $N = 92$ of which have at least one station. The length of each time step is set as 10 minutes.
- **Shenzhen Dataset** (SZ): The dataset contains passengers' railway trip records in SZ from Dec 1^{st} 2015 to Dec 30^{st} 2015. The previous 23 days are used for training, and the rest 7 days are for testing. We split the whole city as 10×10 regions, $N = 36$ of which have at least one station. The length of each time step is set as 10 minutes.

Evaluation Metric Two metrics are used for performance evaluation: Rooted Mean Squared Error (RMSE) and Mean Absolute Error (MAE).

Baselines We compared GAIN with the following state-of-the-art methods. The parameters for all the methods are well tuned with the best performance reported. It is noted that for GEML and MDL, they also aim at joint prediction of in-out flows and OD transit flows.

- **AR:** We build AR models for the inflow and outflow of each region, and transit flow of each OD path separately. Each model's lag order is tuned by Akaike information criterion (AIC).
- **ARIMA:** We build ARIMA models for the inflow and outflow of each region, and transit flow of each OD path separately. Each model's lag order is tuned by Akaike information criterion (AIC).
- **GRU:** All crowd flows, including in-out flows of each region and transit flows of all the OD paths, are stacked together as a matrix with rows as time step (the size of which equals look-back window K=6) and columns as different crowd flow variables. The matrix is inputted into GRU for prediction.

- **CNN:** In-out flows of all the regions in each time step are viewed as two images inputted into CNN. The temporal information is modeled as features and we set look-back window K as 6.
- **ConvLSTM [6]:** In-out flows of all the regions are mapped into city grids. The LSTM structure is comprised with 2 ConvLSTM layers and 1 convolutional layer, and the look-back time window is set to 6.
- **DeepST [4]:** In-out flows of all the regions are mapped into city grids. 6 convolution layers are used, and the sequence lag length is set as 6, 3, 1 for modules of temporal closeness, period and trend dependencies, respectively.
- **ST-ResNet [5]:** Three residual units are stacked, and each is with two combinations of "ReLU+Convolution".
- **ASTGCN [8]:** Two ST blocks are stacked, and the look-back time window is set to 6. Inflow and outflow are fed in as features, and predicted respectively.
- **GEML [13]:** One layer GCN and Periodic-skip LSTM are conducted, with the length of the skipped time steps set as the number of time steps in one day.
- **MDL [14]:** All crowd flows are mapped into city grids. Three residual units and two convolution layers are stacked for OD transit flow network and in-out flow network, respectively.

Experiment Settings For GAIN, tanh activation function is used. Min-Max normalization is used to standardize data into range $[-1, 1]$. In the evaluation, we apply inverse Min-Max transformation obtained on the training set to recover flow values. For temporal correlation, we set $l_m = 6$, $l_d = 3$ and $l_w = 1$. For the spatial block, we set $K = 3$ (the number of attention heads in GAT) and $L = 1$ (the number of GCN layers). The order of polynomials of the Laplacian is set as 1. The batch size is set to 10. 80% of the training samples are selected for training each model and the remaining are in validation set for parameter tuning. We use Adam as our optimizer and the epoch is set as 100. We also use early-stopping to avoid overfitting in all experiments, with patience set to be 20, and we reduce learning rate when a metric has stopped improving, with patience set to be 5 and factor be 0.1.

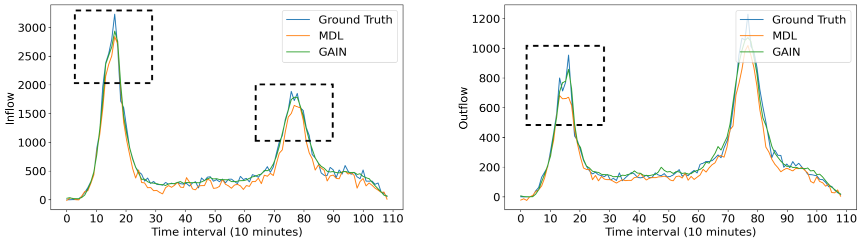
5.2 In-out Flow Prediction

We first compare GAIN with baseline methods for in-out flow prediction. As shown in Table 1, GAIN achieves the best results among all approaches for both inflow and outflow prediction of the two datasets.

As for AR, ARIMA, and GRU, they perform poorly as they do not consider the spatial correlations into the model. CNN performs bad as it simply models the spatial information as features. Furthermore, the performance of ConvLSTM is quite unsatisfactory. One possible reason is that the temporal pattern in our data is not very complex. Yet LSTM is over complicated and tends to overfit the data a lot, leading to a even worse performance than AR, ARIMA and GRU. Two spatio-temporal deep-learning based models, i.e., DeepST and ST-ResNet,

Table 1. In-out flow prediction of different methods.

Method	HK				SZ			
	Inflow		Outflow		Inflow		Outflow	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
AR	108.25	64.37	78.48	47.84	109.82	60.48	81.44	44.35
ARIMA	106.27	63.01	77.27	47.27	107.72	60.00	80.73	44.32
GRU	90.44	51.42	68.60	39.71	80.22	49.75	73.30	43.11
CNN	110.09	61.14	77.86	45.59	95.12	56.55	83.02	49.80
ConvLSTM	129.43	74.33	121.17	79.56	144.76	86.50	147.32	85.44
DeepST	94.31	54.87	74.54	45.17	94.11	56.22	90.67	55.28
ST-Resnet	82.27	48.98	64.73	40.24	83.31	52.60	76.20	48.33
ASTGCN	102.84	60.40	98.27	56.78	112.10	66.00	97.05	56.98
GEML	110.09	65.53	111.44	71.08	99.95	61.60	108.33	68.66
MDL	92.97	55.01	88.57	50.46	79.15	51.48	65.89	43.04
GAIN	**81.81**	**45.86**	**58.48**	**36.00**	**68.68**	**42.00**	**61.35**	**37.01**



(a) (b)

Fig. 2. Inflow and outflow results for 24 Dec. 2015 of SZ dataset: (a) Region 21 (Daxin, Taoyuan, Yuehaimen, Shenzhen University); (b) Region 14 (Honglang North, Xingdong, Liuxiandong).

still perform worse than GAIN. This is because their spatial correlation is based on geographical distance, which works for road traffic prediction, such as for taxi or bicycles. However, they are not good at public transport crowd flow prediction. ASTGCN performs even worse than ST-ResNet. One reason is that its spatial correlation highly depends on the pre-defined network graph, which is not helpful for prediction.

As to the two joint prediction models, GEML and MDL, they perform worse than GAIN. For GEML, it is because its network structure mainly targets at OD flows. Yet the in-out flows are less conscientiously calculated by simply weighted sum of features of OD flows, and hence have less prediction accuracy. For MDL, it is because it assumes geographically close regions are more correlated, and thus is not suitable for public transport flow prediction.

Table 2. OD flow prediction results of different methods.

Method	HK					SZ				
	ARIMA	GRU	GEML	MDL	**GAIN**	ARIMA	GRU	GEML	MDL	**GAIN**
RMSE	4.71	5.03	6.04	4.63	**4.83**	7.81	6.83	8.76	6.86	**6.78**
MAE	1.86	1.87	1.95	2.26	**1.80**	3.71	3.36	4.18	3.80	**3.33**

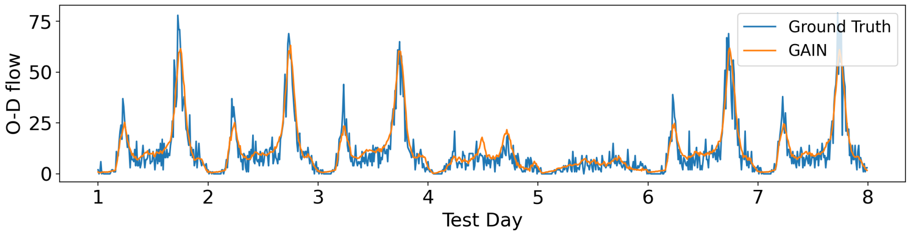


Fig. 3. The OD transit flow prediction from Region 73 (Quarry Bay) to Region 48 (Kowloon Bay) in HK dataset.

To better demonstrate the prediction performance, we randomly select one day and plot the prediction results of GAIN and MDL (the best baseline in SZ dataset) against the ground truth inflow and outflow. Fig.2 shows that GAIN is closer to the ground truth than MDL for most time steps. Especially, for peak hours with extreme high crowd flows, GAIN performs much better than MDL, as shown in the framed time windows in Fig.2. In some time steps with sudden crowd flow changes, however, both methods could not predict well. One possible reason is that we ignore some external features like weather, due to lack of data.

5.3 OD Transit Flow Prediction

Now we evaluate the performance of OD transit flow matrix prediction of GAIN. Here we select some representative baselines: GEML and MDL which two are targeted at OD flow prediction, ARIMA and GRU which are basic models and can be easily applied into OD flow prediction. As to other works in the literature, their original papers aim at in-out flow prediction and cannot be easily extended for OD matrix prediction, so we do not compare with them. The results are shown in Table 2. Clearly, GAIN has overall the best performance. Though MDL outperforms GAIN a bit for HK dataset in terms of RMSE, their differences are insignificant, and GAIN even performs better in terms of MAE. Furthermore, GAIN also overwhelmingly outperforms MDL for in-out flow prediction. Combining results of Table 1, we can conclude the joint prediction framework of GAIN is more efficient than GEML and MDL. As to ARIMA, surprisingly it performs well for HK dataset, but generally ARIMA performs much worse than others for SZ dataset and for in-out flows. Fig.3 shows the ground truth and the prediction results of GAIN for one selected OD path in one week. We can see that the predicted curve can capture the various passenger flow patterns in different days accurately.

5.4 Sensitivity Analysis

To better evaluate the connection between in-out flow prediction and OD transit flow matrix prediction, we conduct parametric analysis for GAIN by tuning the hyper-parameters λ_{region} and λ_{OD} in the loss function. The ratio of λ_{region} and λ_{OD} adjusts the importance weight of in-out flows and OD transit flows. If $\lambda_{\text{region}} = 0$ or $\lambda_{\text{OD}} = 0$, then the model only predicts in-out flows, or the OD transit flow matrix, respectively. This means our dual-task prediction model changes to single-task model. Table 3 shows the prediction results under different combinations of λ_{region} and λ_{OD} on SZ dataset. Clearly, the joint prediction model performs better than the single-task model. It verifies that the OD matrix and in-out flows of regions mutually influence each other and a holistic prediction model with consideration of their intimate connections tends to increase prediction performance. Furthermore, when $\lambda_{\text{OD}}/\lambda_{\text{region}}$ increases, the prediction performance of both tasks improve. This means if we adjust more importance weights on OD transit flow prediction, its prediction accuracy becomes better and also results in better in-out flow prediction. This further demonstrates these two tasks mutually influence each other. In contrast, when $\lambda_{\text{region}}/\lambda_{\text{OD}}$ increases, the prediction performance of in-out flows improves insignificantly, while OD matrix prediction becomes much worse. As the ratio increases more, both tasks perform worse. This indicates the bottleneck of the multi-task prediction is OD matrix prediction, whose worse performance also deteriorates prediction of in-out flows.

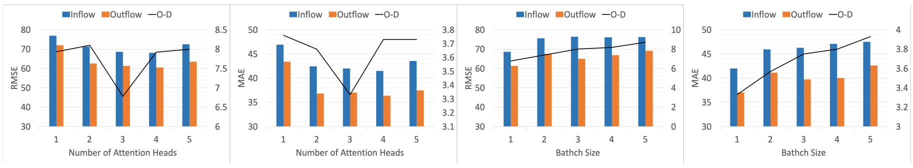
We also evaluate the effect of attention head number on the performance. The number of heads represents how many different spatial correlations are captured in GAT. Fig.4a and Fig.4b show when head number is 3 or 4, both RMSE and MAE achieve the lowest equivalently for in-out flow prediction. When head number is 3, both RMSE and MAE achieve the lowest for OD transit flow prediction. This indicates there may be three kinds of spatial correlations between different regions. We guess they represent the correlations between inflows of regions, correlations between outflows of regions, and interactive correlations between inflows and outflows of regions. This further demonstrates GAIN can extract dynamic and complex spatial features adaptively. As more than 3 heads are included, the model complexity increases and tends to be over-fitting. Last, Fig.4c and Fig.4d show the effect of batch size. Smallest batch size achieves the best generalization performance. This is because large batch size leads the model to make large gradient updates and consequently reach local minimum, while small batch size is noisy, offering more randomness and lower generation error.

6 Conclusions

This work proposes a holistic prediction framework for in-out flows and OD transit flow matrix for public transport network based on graph neural networks. It uses

Table 3. The impact of hyperparameter ratio for GAIN on SZ dataset.

Hyperparameter		Inflow		Outflow		Transition	
λ_{region}	λ_{OD}	RMSE	MAE	RMSE	MAE	RMSE	MAE
1	10	68.68	42.00	61.35	37.01	6.78	3.33
1	5	73.29	43.37	62.18	37.24	6.98	3.44
1	1	74.40	44.09	63. 65	37.91	7.41	3.62
0	1	\	\	\	\	8.13	3.80
1	0	85.94	52.98	78.60	48.36	\	\
5	1	73.18	44.66	63.58	38.31	7.96	3.66
10	1	74.05	45.09	65.34	39.32	8.16	3.71



(a) (b) (c) (d)

Fig. 4. Effect of parameter settings on SZ dataset: (a) RMSE and (b) MAE on different numbers of attention head; (c) RMSE and (d) MAE on different batch sizes.

dynamic GCNs for in-out flow prediction. The graph structures are dynamically learned from the prediction process of OD transit flow matrix, where a multi-head GAT model is used to capture spatial correlations. The above spatial blocks are further inputted into three GRU blocks for minute-level, daily-level and weekly-level temporal correlation description separately. Experiments on two real-world datasets show that our model outperforms several state-of-the-arts.

Acknowledgments

This work was supported in part by the NSFC Grant 71901131, 71932006 and 71931006, in part by the ASFC Grant 2020Z063058001, in part by the Tsinghua GuoQiang Research Center Grant 2020GQG1014 and Tsinghua University Intelligent Logistics & Supply Chain Research Center Grant THUCSL20182911756-001, in part by the Ministry of Education, Singapore, AcRF Tier 2 Funding Grant MOE2019-T2-2-116, and in part by the Hong Kong RGC GRF Grant 16216119 and 16201718.

References

1. Yu, R., Li, Y., Shahabi, C., Demiryurek, U., Liu, Y.: Deep learning: A generic approach for extreme condition traffic forecasting. In: Proceedings of the 2017 SIAM international Conference on Data Mining. pp. 777–785. SIAM (2017)

2. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems. pp. 802–810 (2015)
3. Yao, H., Tang, X., Wei, H., Zheng, G., Li, Z.: Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 5668–5675 (2019)
4. Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X.: Dnn-based prediction model for spatio-temporal data. In: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. pp. 1–4 (2016)
5. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. arXiv preprint arXiv:1610.00081 (2016)
6. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems **29**, 3844–3852 (2016)
7. Pan, Z., Liang, Y., Wang, W., Yu, Y., Zheng, Y., Zhang, J.: Urban traffic prediction from spatio-temporal data using deep meta learning. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1720–1730 (2019)
8. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 922–929 (2019)
9. Liu, L., Qiu, Z., Li, G., Wang, Q., Ouyang, W., Lin, L.: Contextualized spatial–temporal network for taxi origin-destination demand prediction. IEEE Transactions on Intelligent Transportation Systems **20**(10), 3875–3887 (2019)
10. Chu, K.F., Lam, A.Y., Li, V.O.: Deep multi-scale convolutional lstm network for travel demand and origin-destination predictions. IEEE Transactions on Intelligent Transportation Systems **21**(8), 3219–3232 (2019)
11. Shi, H., Yao, Q., Guo, Q., Li, Y., Zhang, L., Ye, J., Li, Y., Liu, Y.: Predicting origin-destination flow via multi-perspective graph convolutional network. In: ICDE. pp. 1818–1821. IEEE (2020)
12. Hu, J., Yang, B., Guo, C., Jensen, C.S., Xiong, H.: Stochastic origin-destination matrix forecasting using dual-stage graph convolutional, recurrent neural networks. In: ICDE. pp. 1417–1428. IEEE (2020)
13. Wang, Y., Yin, H., Chen, H., Wo, T., Xu, J., Zheng, K.: Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1227–1235 (2019)
14. Zhang, J., Zheng, Y., Sun, J., Qi, D.: Flow prediction in spatio-temporal networks based on multitask deep learning. IEEE Transactions on Knowledge and Data Engineering **32**(3), 468–478 (2019)
15. Wang, S., Miao, H., Chen, H., Huang, Z.: Multi-task adversarial spatial-temporal networks for crowd flow prediction. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 1555–1564 (2020)