MG-ASTN: Multigraph Framework With Attentive Spatial–Temporal Networks for Crowd Mobility Prediction

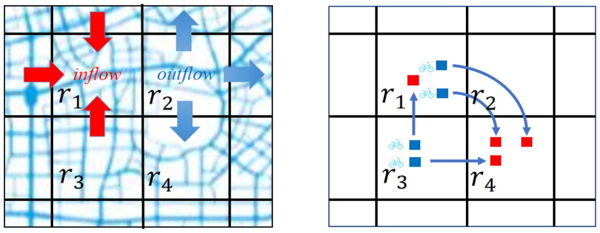
Cheng Luo, Rusheng Cai, Haizhou Guo, Siting Luo, Rui Mao, Landu Jiang, and Dian Zhang, *Member, IEEE*

***Abstract*—Predicting urban crowd patterns/flows is a challenging task due to complex spatial–temporal (ST) dependencies. In this article, we aim to examine and report the capability and effectiveness of the current most widely used graph convolutional networks (GCNs) on mobile data analysis in ST networks for crowd flow forecasting. Specifically, we propose a novel dual-stream framework leveraging multigraph with attentive ST networks (MG-ASTN) to simultaneously predict crowd in–out flow and origin–destination (OD) flow based on the trajectory data collected by on-board devices (e.g., GPS). MG-ASTN utilizes multi-GCNs encoding non-Euclidean correlations to explore pairwise relationships among regions. In addition, we further apply a cross-channel attention mechanism with 3-D temporal convolutional network to address the heterogeneity of ST features and capture more meaningful data representations for multitask learning. In the evaluation, we conduct experiments based on two real-world data sets and verify most well-known state-of-the-art methods for crowd flow prediction. The results demonstrate that MG-ASTN could outperform other solutions—in–out flow prediction with lowest RMSE and MAE, and OD flow prediction beyond others in most cases, thus it has great potential in modeling the complex correlations among regions in ST networks and enabling accurate prediction in urban computing.**

***Index Terms*—Flow prediction, graph convolutional networks (GCNs), mobile data analysis, urban computing.**

I. Introduction

THE ACCURATE urban flow forecasting in spatial–temporal (ST) networks plays an important role in city-wide human mobility analysis and intelligent transportation management [1]. As shown in Fig. 1, there are two typical urban crowd flow representations in ST networks: 1) in–out flow: the traffic volume in/out a grid during a given time slot, and 2) origin–destination (OD) flow: the set of trips from origin to destination grids during a given time slot. In–out flow prediction, for example, is able to facilitate utilizing emergency mechanisms to mitigate heavy traffic jam at specific regions and avoid potential accident hazards (e.g., stampede). While the OD flow prediction could facilitate estimating the volume between affected regions and improving the emerging mobility-on-demand (MOD) services.



Inflow/Outflow

Origin-Destination Flow

Fig. 1. Examples of in–out flow and OD flow.

Due to the complex correlations and heterogeneity of data, the accurate prediction of urban flows in ST networks is very challenging. Early work for forecasting the human mobility or the taxi–passenger demands are mainly based on statistical models or machine learning models [2]. However, most of them only consider limited spatial dependencies (e.g., transition probabilities) among regions. It will be less practical when facing large-scale complex spatial data [3]. Powered by recent advances in big data and deep learning techniques, there have been considerable research efforts leveraging CNN, RNN or long-short term memory (LSTM), as well as GNN-based models on capturing ST dependencies for flow prediction tasks [4]. However, these approaches failed to consider interactions between spatial and temporal dependencies and RNN-based architectures may not capture the dynamics in the temporal correlations accurately [5].

Recently, multitask learning frameworks have been proposed for simultaneously predicting in–out flows and OD/transitions among nodes in ST networks [6], [7]. It shows that in–out flows and OD flows are highly correlated and

Manuscript received 28 December 2022; revised 12 April 2023; accepted 15 May 2023. Date of publication 31 May 2023; date of current version 24 October 2023. This work was supported in part by NSFC under Grant 61872247, Grant 62072311, and Grant U2001212; in part by the Shenzhen Project under Grant JCYJ20220531091407016 and Grant JCYJ20210324094402008; in part by the Shenzhen Peacock Talent under Grant 827-000175; in part by the Guangdong Natural Science Fund under Grant 2019A1515011064; in part by the Key Technology Research and Development Program of China under Grant 2019YFE0196800; in part by the Overseas High-Level Talents of Shenzhen under Grant KQJSCX20180329191021388; and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2020B1515120028. *(Corresponding authors: Landu Jiang; Dian Zhang.)*

Cheng Luo, Rusheng Cai, Haizhou Guo, Rui Mao, Landu Jiang, and Dian Zhang are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: 2070276165@email.szu.edu.cn; 1910272060@email.szu.edu.cn; 1800271045@email.szu.edu.cn; mao@szu.edu.cn; landu.jiang@szu.edu.cn; zhangd@szu.edu.cn).

Siting Luo is with the College of Computer Science and Technology, Xidian University, Xi'an 710100, China (e-mail: 2321170490@qq.com).

Digital Object Identifier 10.1109/JIOT.2023.3281648

share some common latent features. These two tasks are complementary to each other and their performance can be mutually enhanced. On the other hand, graph convolutional networks (GCNs)-based methods have been widely applied in ST network data analysis and achieved promising results in urban flow prediction [5], [8]. However, existing GCN-based solutions do not fully address how do geographic relations of nodes affect the mutual influence of in–out flows and OD flows.

Though Wang et al. [7] proposed a semantic spatio-temporal (ST) graph to learn common features across prediction tasks, they did not fully address the spatial correlations of crowd flows at a regional level, the pairwise relationships among regions were not fully taken into account in their approach. Therefore, further research is needed to better explore these spatial correlations and their impact on crowd flow prediction.

In this article, we propose a novel dual-stream framework leveraging multigraph with attentive ST networks (MG-ASTN) that simultaneously forecasts region in–out volume and OD flows. More specifically, we utilize multi-GCNs (MGCNs) on both two learning tasks to explore pairwise relationships by encoding different types of non-Euclidean correlations among regions. More importantly, we employ a cross-channel attention [9] mechanism to fuse the feature learning of MGCNs on in–out flow and OD flow. The proposed attention module does not only exploit channelwise attention that better address the heterogeneity of spatial correlations generated by GCNs on each task, but also use multihead attention that captures more meaningful representations of ST and external context data for multitask learning. We then apply a 3-D temporal convolutional network (TCN) processing attentively selected ST features for the final crowd flow prediction.

The contributions of this article are summarized as follows.

1) We propose MG-ASTN, a new dual-stream framework for crowd (in–out and OD) flow prediction in ST networks, which uses Multigraph CNN with attention mechanisms addressing the heterogeneity of different graph models on both two tasks.
2) We explore pairwise relationships by encoding different types of non-Euclidean correlations as well as leveraging the cross-channel attention module (CAM). In order to simultaneously predict in–out flows and OD flows, a 3-D TCN is used for synchronously processing attentively selected ST features.
3) We conduct experiments based on two real-world data sets and evaluate most well-known state-of-the-art methods for crowd flow prediction in ST networks. The results demonstrate that GCN-based approach could outperform other solutions—in–out flow prediction with lowest RMSE and MAE, and OD flow prediction beyond others in most cases. We also provide a study to examine the effectiveness of hyperparameter in multitask GCN models.

The reminder of this article is organized as follows. Section II gives a formal problem definition and Section III presents the proposed framework design. Section IV shows the evaluation results based on real-world data sets. Finally, Section V reviews the related work and Section VI concludes this article.

II. Data and Problem Definition

In this section, we first briefly introduce the definition of grids in graph, in–out flow, as well as OD flow between regions, and then state our crowd flow prediction problems.

A. Notations

Definition 1 (Grid): A city is partitioned into $M \times N$ nonoverlapping grids based on the longitude and latitude, denoted by $G = \{g_{1,1}, \dots, g_{i,j}, \dots, g_{M,N}\}$, where each grid $g_{i,j}$ represents the ith row and jth column cell in the region map. In this article, we partition the target area into 16 × 16 grids.

Definition 2 (In–Out Flow): The movement of crowd into and out of a specific grid $g_{i,j}$ within a time slot t. Let $\mathcal{P}$ be a collection of flow trajectories in $g_{i,j}$, than we can represent inflow $x^{\text{in}}_{t,i,j}$ and outflow $x^{\text{out}}_{t,i,j}$ in time slot t as

$$x^{\text{in}}_{t,i,j} = \sum_{\tau_r \in \mathcal{P}} \left| \left\{ u > 1 \mid p_{u-1} \notin g_{i,j} \wedge p_u \in g_{i,j} \right\} \right| \tag{1}$$

$$x^{\text{out}}_{t,i,j} = \sum_{\tau_r \in \mathcal{P}} \left| \left\{ u > 1 \mid p_u \in g_{i,j} \wedge p_{u+1} \notin g_{i,j} \right\} \right| \tag{2}$$

where $\tau_r : p_1 \to p_2 \to \cdots \to p_{\tau_r}$ is one of the trajectory in $\mathcal{P}$, p_u is a geospatial coordinate and $p_u \in g_{i,j}$ means p_u is within region $g_{i,j}$.

Definition 3 (OD Flow): An OD flow captures the number of travels for specific OD pairs in a particular time period, commonly represented by an OD matrix [7]. We define the OD matrix at tth time step as: $Y_t \in \mathcal{R}^{L\times L}$, where $L = M \times N$ equals the total number of grids, and each element $y_t^{l,k}$ in the matrix $\mathcal{R}$ denotes the volume of flows starting from the lth grid and ending at kth grid.

B. Problem Definition

Given a time slot τ, the in–out flow of an area can be represented as $\{X_t^c \mid t = \tau - l + 1, \dots, \tau - 1, \tau\}$, while the history of OD flows (OD flow matrix) is $\{M_t \mid t = \tau - l + 1, \dots, \tau - 1, \tau\}$, where l is the length of time periods. For any given time τ, our purpose is to simultaneously predict the in–out flow $X^c_{\tau+1}$ and OD flow $M_{\tau+1}$.

III. Methodology

In this section, we introduce the details of MG-ASTN, our multigraph framework for crowd flow prediction. MG-ASTN consists of three main steps: 1) *trajectory data processing*; 2) *multigraph-based spatial encoding*; and 3) *ST modeling and flow prediction*, as illustrated in Fig. 2.

A. Trajectory Data Processing

Following the definitions in Section II-A, we divide the map into $m \times n$ grids. There are two channels for the volume of inflow and outflow, thus the size of in–out flow image is $m \times n \times 2$. Similarly, we can build the OD flow matrix with the

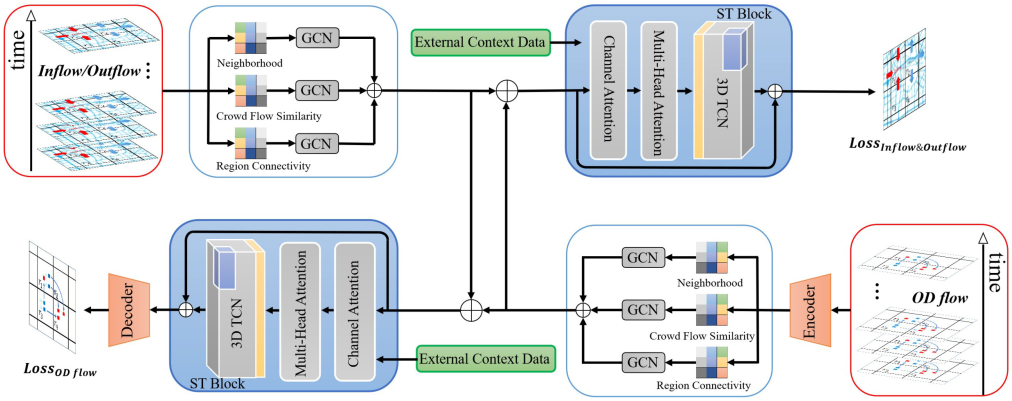


Fig. 2. System architecture of the proposed MGCN-based attentive ST network.

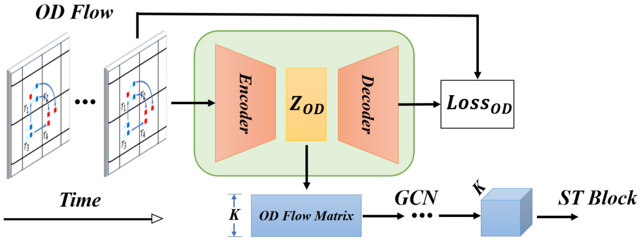


Fig. 3. Architecture of AE on OD flow prediction.

size $N \times N$, $(N = m \times n)$ based on the OD trips of the raw trajectories. In addition, we input the OD flow matrix into auto encoder (AE) for data compression, this is because the transformation that actually occurs in the next time interval (between a location and the rest of places) may be a small part of the probability of N^2 (the entire map), which may lead to unreliable and ineffective traffic prediction due to the sparsity of the OD matrix data.

As shown in Fig. 3, the actual architecture of AE has a symmetrical encoder and decoder with multiple fully connected layers.

We can encode OD flow data into latent layer with a lower dimension size $N \times K$. As illustrated in Fig. 3, we are able to extract meaningful features to identify patterns and trends, and reduce the dimensionality of the data to making it easier to analyze and process. We set K values as 64, 32, 16, 8, 2 to compared the influence of hyper parameter K in Section IV.

B. Multigraph-Based Spatial Encoding

We present a graph-based model of the city map, denoted by $G = (V, A)$, where V represents the set of all vertices corresponding to grids in the map and A is the adjacency matrix ($A \in \boldsymbol{R}^{|V| \times |V|}$), whose entries indicate the connections between vertices. In this section, we introduce multigraph convolution networks that model various correlations among grids.

1) Crowd Flow Similarity: The crowd flow similarity graph $G_F = (V, A_F)$ plays a crucial role in capturing the similarity of historical flow patterns of grids [10].

Particularly, the functionality of a region is characterized based on its crowd flow patterns, namely, inflow and outflow, and the similarity between regions is measured by computing the correlations (e.g., Pearson correlation coefficient, discussed in Section IV) using the historical data. For two grids g_i and g_j, the (i, j)th cell in the adjacency matrix A_F is defined as

$$A_{F,ij} = \left[\text{Similarity}^{\text{in}}_{g_i,g_j}, \text{Similarity}^{\text{out}}_{g_i,g_j}\right]. \tag{3}$$

Crowd flow similarity is defined as $\text{Similarity}^{\text{in/out}}_{g_i,g_j} = \text{Pearson}(F^{\text{in/out}}_{0\sim t}(g_i), F^{\text{in/out}}_{0\sim t}(g_j))$, where $F^{\text{in}}_{0\sim t}(gi)$ represents the historical crowd inflow sequence of region g_i from time 0–t while $F^{\text{out}}_{0\sim t}(gi)$ denotes the crowd outflow. We computed the correlation between each region and all other regions, resulting in a symmetric matrix with size of 16×16 in our case.

2) Neighborhood: Neighborhood graph $G_N = (V, A_N)$, which encodes the geographical proximity. We construct the graph by considering the geographical neighbors of a region in real world based on the spatial proximity. The neighborhood of a region in the graph is defined as its surrounding adjacent regions in a 3×3 grid [10]. Then we have

$$A_{N,ij} = \begin{cases} 1, & g_i \text{ and } g_j \text{ are neighbors} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

3) Region Connectivity: Region connectivity graph $G_C = (V, A_C)$, which represents the node connectivity based on the transitions between distant regions, can be modeled using the strength of the OD flows between different regions. The neighborhood graph measures the intrinsic closeness of nearby regions, while modeling the mobility relationships between the origins and destinations in the network can correlate geographically distant yet conveniently reachable regions. To this end, we define two regions, g_i and g_j, as semantically connected neighbors of each other if there is at least one trip from one region to the other

$$A_{C,ij} = \text{conn}\left(g_i, g_j\right) = \sum_{0}^{\tau} \text{OD}^{i,j}_{t}. \tag{5}$$

The connectivity between these regions is quantified by the accumulated amount of trips $\text{OD}^{i,j}_{t}$, which is used to define the edge AC, ij in the graph $G_C = (V, A_C)$. Here, $\text{conn}(g_i, g_j)$ is the indicator function of the connectivity between regions g_i and g_j.

C. Spectral Convolutions on Graphs

According to the definitions in [5], [10], and [11], with the multiple graph constructed, we can generalize the graph convolution for a given signal with C channels $X \in \mathbb{R}^{n\times C}$ as: $Z = \tilde{D}^{-(1/2)}\tilde{A}\tilde{D}^{-(1/2)}X\Theta$

In this article, we consider a two-layer GCN for ST network data analysis with the adjacency matrix A. We first define $\hat{A} = \tilde{D}^{-(1/2)}\tilde{A}\tilde{D}^{-(1/2)}$, and then take the simple form into the forward model for flow prediction

$$Z = f(X, A) = \sigma\left(\hat{A}\sigma\left(\hat{A}XW^0\right)W^1\right). \tag{6}$$

The neural network weights W^0 and W^1 are trained using gradient descent, σ denotes the activation function (ReLU in this article). In addition, with multiple graphs constructed, the multigraph convolution for spatial dependency modeling can be defined as

$$X_{l+1} = \sigma\left(\bigcup_{A\in\mathbb{A}} f(A)X_l W_l\right) \tag{7}$$

where X_l and X_{l+1} are feature vectors in layers l and $l+1$, and $\cup$ denotes aggregation function for graphs, $\mathbb{A}$ denotes the set of graphs: $G_F = (V, A_F)$, $G_N = (V, A_N)$ and $G_C = (V, A_C)$ as we use MGCNs taking into account three types of region-wise relationships presented above. Thus, $f(A)$ represents the aggregation matrix of different samples based on graphs. What is more, as we are using an architecture that learns the latent representations of in–out flow (Z^{IO}) and OD flow (Z^{OD}) simultaneously. By fusing them together, we have the final vector representations $Z = Z^{IO} \oplus Z^{\mathrm{OD}}$, Z will be fed into the 3-D TCN for ST dependency modeling.

D. Spatial–Temporal Modeling and Prediction

We use ST block to capture the data representations of crowd flow, utilizing the features extracted by the MGCNs. In order to overcome the heterogeneity of input data, we employ a module that infers attention maps along separate dimensions-channel and ST to fuse the two-stream MGCN features.

Particularly, we integrate the cross-channel attention mechanism inspired by [9] in ST block, which explored the spatial correlations among regions, followed by combining the ST features and 1-D external context. Additionally, we adopt the CBAM module in each subnetwork to emphasize extracting features along both the channel and spatial axes. As illustrated in Fig. 2, the temporal information can be extracted through the iteration and connection of multilayer 3-D TCNs. As we have a long axis of time, it is necessary to reduce the time step through convolution. While at the same time, we use two technologies, including strided convolution and padding to implement the proposed 3-D TCN module. Among them, strided convolution is used to deal with the increase in the computational load with the rise of dimensions, while avoiding the massive calculation of the traditional TCNs [3]. Strided convolutions are cost effective which can reduce the dimension of temporal representation while cannot align the time slot τ. Padding is one of the solution to keep causality by padding zeros to the head of temporal representation.

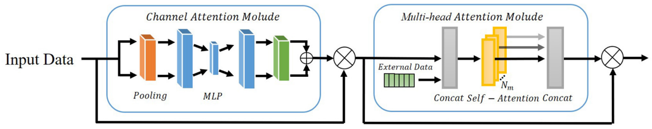


Fig. 4. Attention module.

E. Cross Channel Attention Module

By applying CBAM [9] in each CNN subnetwork. We are able to extract informative features from both channel and spatial axes (as shown in the left part of Fig. 4). For a given feature map Υ_{s1}, CBAM calculates its channel and spatial weight matrix sequentially, then refines the feature map based on these two weight matrix. The module is summarized in following equation:

$$\Upsilon'_{s2} = \Upsilon_{s1} \otimes M_c^{C'}(\Upsilon_{s1}) \tag{8}$$

$$\Upsilon_{s2} = \Upsilon'_{s2} \otimes M_s^{H'\times W'}(\Upsilon'_{s2}) \tag{9}$$

where $M_c^{C'}$ is the channel attention matrix processed by CAM and $M_s^{H'\times W'}$ is the spatial attention matrix from spatial attention module (SAM).

To generalize an attention matrix M_c of feature map Υ_{s1} in channel dimension, CBAM squeezes the spatial dimension of the spectrogram feature map by applying both average-pooling $\mathcal{P}_{\mathrm{avg}}$ and max-pooling $\mathcal{P}_{\max}$. The pooling features are connected by a shared convolutional network Ξ

$$M_c = \sigma\left[\Xi\left(\mathcal{P}_{\mathrm{avg}}(\Upsilon_{s1})\right) + \Xi\left(\mathcal{P}_{\max}\Upsilon_{s1})\right)\right] \tag{10}$$

where σ represents the Sigmoid function.

The SAM is to emphasize the inter spatial features of the adjusted spectrogram feature map Υ'_{s2}. Different from CAM, it refines the feature map along channel dimension by leveraging average pooling operations and max pooling operations. The efficient spatial features of feature map Υ'_{s2} is extracted efficient information through a two channel convolution layer. Spatial attention matrix M_s is

$$M_s = \sigma\left(\xi\left[\mathcal{P}_{\mathrm{avg}}(\Upsilon_{s1})\right); \mathcal{P}_{\max}\Upsilon_{s1}\right]\right) \tag{11}$$

where ξ is the convolution function with a $(7, 7)$ kernel size.

In addition, the multihead attention is also deployed to emphasize the inter ST features of the adjusted spectrogram feature map. Through parallel operation of multiple attention modules, concatenating the independent attention outputs into an expected dimension. In our design, we extract features containing spatio-temporal correlation from different GCNs (such as neighborhood, regional connectivity, and crowd flow similarity), and dynamically calculate the weight of different time steps in the model (as shown in the right part of Fig. 4).

There are multiple causal self-attention layers [3], [12] taking the input X_t and multiplies it by using three same size matrices to form Q, K and V, which represent query, key and value, respectively. The process of self-attention in each module can be defined as follows:

$$\mathtt{Attention}(Q, K, V) = \mathtt{softmax}\left(\frac{QK^T}{q_k}\right)V \tag{12}$$

where QK^T is a square matrix, and q_k is the temporal dimension of Q. Since the calculation of Q in the equation needs to be expressed by the predicted results, the causal relationship in our ST network is no longer applicable/valid. Therefore, we apply a mask layer for the predictive value to convert the potential features related to the future to zero, so as to ensure that our causal self-attention module can be implemented correctly

$$\texttt{Mask}(X) = \begin{cases} X_{(i,j)} = X_{(i,j)}, & \text{if } i \leq j \\ X_{(i,j)} = -\texttt{inf}, & \text{otherwise.} \end{cases} \tag{13}$$

Then, (12) can be revised as

$$\texttt{Attention}(Q, K, V) = \texttt{softmax}\left(\frac{\texttt{Mask}\left(QK^T\right)}{d_k}\right)V. \tag{14}$$

Multihead attention can linearly integrate the weight values of multiple causal self-attention layers. By applying a matrix to fuse the attention values, we can get the final output values $\text{Att} \in \mathbb{R}^l$ for each time step. Then we multiply X_l by Att to form X_{Att}.

IV. Performance Evaluation

In this section, in order to examine and verify the performance of GCN framework on crowd moving flow prediction, we first present the experiment setup and real-world data sets, and then introduce the selected most well-known state-of-the-art methods for the evaluation. Moreover, we summarize the results of different models and present a hyperparameter study as well, a detailed discussion is also provided.

A. Implementation Details

We implement the proposed model with Pytorch framework on the server with NVIDIA GTX 2080 Ti GPU and 64-GB CPU memory. The data size of crowd flow images is $6\times16\times16\times2$ for both data sets, where 6 is the length previous time slot l used for prediction and each time interval is 1 h, 16×16 is the size of the cell regions, and 2 is the number of channels representing inflow and outflow. The original size of flow OD image is $6 \times 16 \times 16 \times 256$. After dimensionality reduction by encoding, the input data size of OD image is $6 \times 16 \times 16 \times K$ ($K = 2, 8, 16, 32, 64$). We set the learning rate and batch size as 0.001 and 32, and choose Adam as the optimizer. The input data size of ST Block are in–out flow $6 \times 2 \times 16 \times 16$ and OD $6 \times N \times 16 \times 16$, respectively.

B. Evaluation Metrics

Given the prediction results $\hat{x}_i$ and the ground truth x_i, we use two most widely used metrics to evaluate model performance for crowd flow prediction: 1) root mean square error ($\text{RMSE} = \sqrt{(1/Z)\sum_{i=1}^{Z}(\hat{x}_i - x_i)^2}$); and 2) mean absolute error ($\text{MAE} = (1/Z)\sum_{i=1}^{Z}|\hat{x}_i - x_i|$), where Z is the total number of test samples.

TABLE I
Data Sets Description

Dataset	TaxiNYC	BikeNYC
Start time	1/1/2015	1/1/2015
End time	12/31/2015	12/31/2015
Time interval	1 hour	1 hour
Grid (Region) Size	(16, 16)	(16, 16)
number of time intervals	8760	8760

C. Data Sets

As shown in Table I, we conduct experiments based on two real-world trajectory data sets NYC-Taxi and NYC-Bike [7]. In order to guarantee the fairness of performance comparison, we strictly follow the data format settings in [7] in the evaluation.

D. State-of-the-Art Baselines

We evaluate MG-ASTN with other state-of-the-art baselines including: 1) ConvLSTM [13] combines the CNN and LSTM to model the spatial and temporal features, which is widely used in various prediction tasks; 2) spatiotemporal residual network (ST-ResNet) [14] uses residual convolution unit for different temporal properties (closeness, period, and trend) modeling crowd flows; 3) STDN [4] is a unified framework to model the dynamic similarity between locations and periodicity shift for traffic flow prediction; 4) GEML [15] is a multitask learning framework using grid embedding that predicts both in–out flow and OD flow simultaneously; 5) MDL [6] is a state-of-the-art multitask learning framework for predicting both the in–out flow and OD flow in ST networks; and 6) MT-ASTN [7] is the most recent and related state-of-the-art multitask learning approach which predicts in–out flow and OD flow simultaneously.

The baseline models are implemented based on the original papers or we use the publicly available code. If we meet the difficulty during the implementation, we will directly use the prediction results in [7] to guarantee the comparison fairness.

To further evaluate whether the key components employed in our framework are useful to the target problems, we also compare the fully functional version MG-ASTN with the variants—MG-ASTN without MGCNs (w/o MGCNs), and MG-ASTN without dual stream networks(w/o Fusion). In addition, we compared the performance on different similarity functions.

E. Experiment Performance

1) Performance Comparison: The flow prediction results on two data sets are shown in Tables II and III.

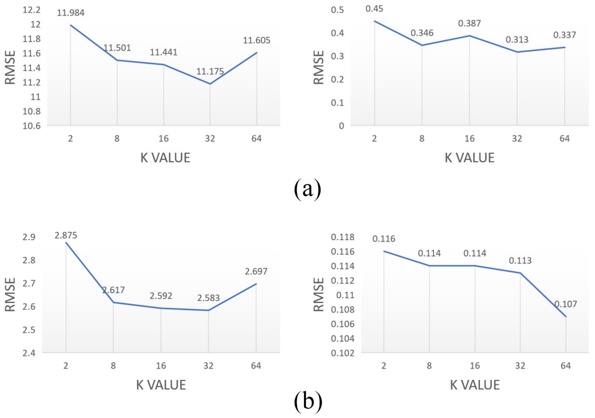
First, we can observe that our proposed model MG-ASTN achieves the best performance regarding all the metrics on both data sets for in–out flow prediction, reducing the RMSE better than the second best baseline MT-ASTN by RMSE reducing 8.1% and 13.7% on NYC-Taxi and NYC-Bike, respectively. Moreover, the performance of MG-ASTN on OD flow prediction beyond other baselines in most cases.

TABLE II
IN–OUT FLOW PREDICTION ON TWO DATA SETS

Model	in-out flow			
	NYC-Taxi		NYC-Bike	
	RMSE	MAE	RMSE	MAE
ConvLSTM	15.179	7.028	3.597	1.674
ST-ResNet	11.558	5.314	2.755	1.264
STDN	21.169	8.637	6.491	1.794
GEML	22.073	10.449	6.344	2.828
MDL	21.492	11.750	8.715	4.250
MT-ASTN	12.299	6.417	2.995	1.413
MG-ASTN	**11.175**	**4.996**	**2.583**	**1.187**
w/o MGCNs	11.253	5.027	2.730	1.252

TABLE III
OD FLOW PREDICTION ON TWO DATA SETS

Model	OD flow			
	NYC-Taxi		NYC-Bike	
	RMSE	MAE	RMSE	MAE
ConvLSTM	0.551	0.320	0.106	0.019
ST-ResNet	0.315	0.080	0.122	0.016
STDN	0.159	0.074	0.127	0.021
GEML	0.670	0.136	0.147	0.014
MDL	0.153	0.095	0.154	0.041
MT-ASTN	0.087	0.030	0.074	0.011
MG-ASTN	**0.313**	**0.076**	**0.113**	**0.016**
w/o MGCNs	0.325	0.079	0.115	0.017



(a)

(b)

Fig. 5. Change of RMSE with K value of AE on OD flow prediction in MG-ASTN. (a) NYC-Taxi in–out flow and OD flow. (b) NYC-Bike in–out flow and OD flow.

Specifically, when our model predicts in–out flow relative to MT-ASTN, the RMSE error on the NYC-Taxi data set was reduced by 1.124 while was reduced by 0.412 on the BYC-Bike. Although in predicting OD fLow in NYC-Taxi and NYC-Bike, the error is 0.226 and 0.039 larger than MT-ASTN, compared to the improvement of in–out flow, these errors are acceptable that help customers understand current traffic conditions (the minimum unit for vehicles or bike is 1).

In addition, "w/o MGCNs" represent the results when not using MGCNs. We can also see that without MGCNs, the performance became worse, which presents the necessity of constructing such modules on predicting the dynamic in–out flow and OD flow for efficient feature extraction.

2) Hyperparameter K Analysis: We evaluate the effectiveness of the hyperparameter $K = 2$, 8, 16, 32, 64, as illustrated in Fig. 5.

TABLE IV
PERFORMANCE ON MG-ASTN

Function	NYC_Taxi				NYC_Bike			
	Flow		OD		Flow		OD	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Spearman	11.481	5.657	0.299	0.074	2.602	1.239	0.105	0.014
Cosine	11.531	5.220	0.300	0.074	2.641	1.246	0.105	0.015
Kendall	11.556	5.371	0.299	0.072	2.700	1.262	0.105	0.014
Euclidean	11.550	5.242	0.299	0.071	2.597	1.216	0.105	0.014
Manhattan	11.807	5.271	0.298	0.071	2.692	1.290	0.106	0.014
Pearson	**11.242**	**5.126**	**0.299**	**0.071**	**2.561**	**1.210**	**0.105**	**0.014**

TABLE V
PERFORMANCE ON MODULE W/O FUSION

Function	NYC_Taxi				NYC_Bike			
	Flow		OD		Flow		OD	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Spearman	14.945	7.509	0.300	0.071	2.937	1.566	0.107	0.014
Cosine	14.001	6.759	0.298	0.071	2.943	1.429	0.107	0.014
Kendall	14.414	6.982	0.298	0.072	2.851	1.352	0.106	0.014
Euclidean	13.902	6.804	0.299	0.072	2.997	1.470	0.107	0.014
Manhattan	14.197	6.959	0.299	0.070	2.943	1.482	0.106	0.014
Pearson	**13.602**	**6.717**	**0.297**	**0.073**	**2.873**	**1.414**	**0.105**	**0.014**

We can observe that the performance of MG-ASTN achieved the best performance on in–out flow and OD flow for NYC-Taxi when $K = 32$. And for NYC-Bike flow prediction, when MG-ASTN get the lowest RMSE at $K = 32$ and $K = 64$ (very close to $K = 32$) for in–out flow prediction and OD flow prediction, respectively. We can observe that a larger K will enable the model capture more global correlations at the cost of increased model complexity and noise, which would help us to make a tradeoff on parameter setting. In addition, since the design of MG-ASTN is flexible for block switching—we can use $K = 32$ for bike in–out flow prediction while $K = 64$ for OD flow prediction though the change of RMSE is minor.

3) Similarity Function: In order to better understand correlation coefficients in GCNs, we compare the performance of using different similarity functions as shown in Tables IV and V. More specifically, we select five other most popular functions in addition to the Pearson correlation: 1) *Spearman*'s rank [16]; 2) *Kendall*'s tau [17] correlations are nonparametric measure of correlation between two variables; 3) *Cosine* similarity [18] is a measure of similarity between two nonzero vectors of an inner product space; 4) *Euclidean* distance [19] is a measure of distance between two points in Euclidean space; 5) *Manhattan* distance [20] is a distance metric that measures the distance between two points in a grid based on the sum of the absolute differences of their coordinates; and 6) *Pearson* correlation [17] analysis is the most commonly used method for correlation analysis.

In Table IV, we can see that under different correlation coefficient conditions, the model using Pearson coefficient has the same results as other functions in OD flow, while the calculation results in in–out flow have been significantly improved. When using Pearson coefficients, we can obtain more stable models and better in–out flow prediction results.

In addition, we also tested the performance of model w/o fusion under different similarity functions (Table V). Where "w/o fusion" represents the model did not fuse the features of in–out flow and OD flow, but instead predicted in–out flow and OD flow separately using two separate paths. Here, we present results similar to Table IV, where we can achieve the best performance when using Pearson coefficient calculations.

Therefore, the comparison results on both data sets demonstrate that MG-ASTN are robust in real-world flow prediction. It also reveals the great potential of graph representations exploring complex correlations between spatial and temporal dependencies for urban flow analysis.

V. Related Work

In this section, we focus on reviewing mainstream schemes on deep learning-based approaches for urban flow (in–out flow and OD flow) prediction in ST network. For more comprehensive surveys on forecasting traffic flows, we refer to the following publications [21], [22].

By leveraging deep neural networks, Zhang et al. [23] presented DeepST model for crowd flow forecasting by modeling both spatial near and distant dependencies. To improve the prediction efficiency, a deep ST-ResNet is proposed [8], [14] which uses three different residual networks to separately model closeness, period, and trend. While these studies only take temporal sequential or spatial dependencies into account. To address the issue, STDN [4] employs flow gating mechanism and periodically shifted to handle dynamic spatial similarity and temporal periodic similarity jointly. Such models do not consider the interaction between spatial correlation and temporal correlation and thus are less practical in real-world scenarios [24]. Recently, researchers have extended traditional CNN and RNN structure to graph-based models for crowd flow prediction [25]. Zhang et al. [1] proposed ST-GDN preserving both local and global regionwise dependencies, via a hierarchically structured graph neural architecture. A multiscale self-attention network is also deployed to explore the multiresolution traffic transitional information. What is more, Ali et al. [26] designed DHSTNet that uses GCN considering spatio-temporal dependency and external factors of the dynamic crowd flows. However, such frameworks simply concatenate features of each neighboring node without considering the variation in different time steps, thus the separate ST correlations [27].

Many studies on region-level OD flow prediction have been proposed to facilitate ride hailing services [28]. Chu et al. [29] developed a multiscale convolutional LSTM network to predict the future travel demand and OD flows, which introduced OD tensor to avoid losing any geographical information. Inspired by recent advances in GCNs, Bai et al. [5] proposed STG2Seq to capture spatial and temporal relationships by utilizing multiple GCN layers to form a gated graph convolutional module (GGCM) while at the same time deploying an attention module to model the dynamic temporal and channelwise information. What is more, Wang et al. [15] proposed GEML, a grid-embedding-based multitask framework that models the spatial mobility patterns in different areas and captures temporal trends of passenger demands.

To simultaneously forecast in–out flow and OD flow, MT-ASTN [7] employs a shared-private framework decomposing the data into task-common features and task-specific features. A semantic graph is used in MT-ASTN to capture the ST correlations of two tasks based on OD flow matrix (global mobility patterns). MDL [6] also takes in–out flows and transitions between regions into account, but it simply concatenated the features of different tasks without well considering the mutual influence of them. Several approaches like [30] and [31] have been proposed to jointly model ST features of in–out flow and OD flow. However, these methods have mainly focused on single-task prediction, and have not fully considered the spatial correlations of crowd flows, such as region similarities based on historical data.

To address these limitations, we propose a novel framework that leverages multiple GCNs. To the best of our knowledge, our approach is the first to simultaneously explore the mutual influence of both in–out flow and OD/transition flow prediction tasks. Additionally, we apply a cross-channel attention mechanism with a 3-D TCN to address the heterogeneity of the two MGCNs-based ST streams, improving the accuracy of the final crowd flow prediction.

VI. Conclusion

In this article, we explore the effectiveness of multiple GCNs on urban crowd moving pattern/flow prediction problem. We simultaneously model the ST feature using the GCN-based framework—MG-ASTN with attentive ST networks and capture the correlations between in–out flow and OD flow. We evaluate different models on two real-world data sets (collected by real on-board sensors), which shows that GCN-based models could outperform other state-of-the-art baselines. In the future, we will consider more flexible geometric models like dynamic graph neural networks to extend the application scenario of the proposed methods.

References

[1] X. Zhang et al., "Traffic flow forecasting with spatial-temporal graph diffusion network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2020, pp. 15008–15015.

[2] M. X. Hoang, Y. Zheng, and A. K. Singh, "FCCF: Forecasting citywide crowd flows based on big data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2016, pp. 1–10.

[3] H. Guo, D. Zhang, L. Jiang, K.-W. Poon, and K. Lu, "ASTCN: An attentive spatial temporal convolutional network for flow prediction," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3215–3225, Mar. 2022.

[4] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5668–5675.

[5] L. Bai, L. Yao, S. Kanhere, X. Wang, and Q. Sheng, "STG2Seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting," in *Proc. 28th Int. Joint Conf. Artif. Intell., (IJCAI)*, 2019, pp. 1981–1987.

[6] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 468–478, Mar. 2020.

[7] S. Wang, H. Miao, H. Chen, and Z. Huang, "Multi-task adversarial spatial-temporal networks for crowd flow prediction," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manag.*, 2020, pp. 1555–1564.

[8] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artif. Intell.*, vol. 259, pp. 147–166, Jun. 2018.

[9] S. Woo, J. Park, and J.-Y. Lee, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[10] X. Geng et al., "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3656–3663.

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

[12] A. Vaswani et al., “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[13] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[14] 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.

[15] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, “Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling,” in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1227–1235.

[16] C. Liu, T. Cao, and L. Zhou, “Learning to rank complex network node based on the self-supervised graph convolution model,” *Knowl.-Based Syst.*, vol. 251, Sep. 2022, Art. no. 109220.

[17] K. Qin et al., “Using graph convolutional network to characterize individuals with major depressive disorder across multiple imaging sites,” *eBioMedicine*, vol. 78, Apr. 2022, Art. no. 103977.

[18] W. Li, X. Liu, Z. Liu, F. Du, and Q. Zou “Skeleton-based action recognition using multi-scale and multi-stream improved graph convolutional network,” *IEEE Access*, vol. 8, pp. 144529–144542, 2020.

[19] L. Maiani and M. Testa, “Final state interactions from euclidean correlation functions,” *Phys. Lett. B*, vol. 245, nos. 3–4, pp. 585–590, 1990.

[20] W.-Z. Nie, M.-J. Ren, A.-A. Liu, Z. Mao, and J. Nie, “M-GCN: Multi-branch graph convolution network for 2D image-based on 3D model retrieval,” *IEEE Trans. Multimedia*, vol. 23, pp. 1962–1976, Jul. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9133153

[21] P. Xie, T. Li, J. Liu, S. Du, X. Yang, and J. Zhang, “Urban flow prediction from spatiotemporal data using machine learning: A survey,” *Inf. Fusion*, vol. 59, pp. 1–12, Jul. 2020.

[22] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, “A comprehensive survey on traffic prediction,” 2020, *arXiv:2004.08555*.

[23] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, “DNN-based prediction model for spatio-temporal data,” in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2016, pp. 1–4.

[24] L. Kuang, X. Yan, X. Tan, S. Li, and X. Yang, “Predicting taxi demand based on 3D convolutional neural network and multi-task learning,” *Remote Sens.*, vol. 11, no. 11, p. 1265, 2019.

[25] Y. Chen and X. M. Chen, “A novel reinforced dynamic graph convolutional network model with data imputation for network-wide traffic flow prediction,” *Transp. Res. Part C, Emerg. Technol.*, vol. 143, Oct. 2022, Art. no. 103820.

[26] A. Ali, Y. Zhu, and M. Zakarya, “Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction,” *Neural Netw.*, vol. 145, pp. 233–247, Jan. 2022.

[27] C. Song, Y. Lin, S. Guo, and H. Wan, “Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 914–921.

[28] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, and L. Lin, “Contextualized spatial-temporal network for taxi origin-destination demand prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3875–3887, Oct. 2019.

[29] K.-F. Chu, A. Y. Lam, and V. O. Li, “Deep multi-scale convolutional LSTM network for travel demand and origin-destination predictions,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3219–3232, Aug. 2020.

[30] J. Feng, Z. Lin, T. Xia, F. Sun, D. Guo, and Y. Li, “A sequential convolution network for population flow prediction with explicitly correlation modelling,” in *Proc. IJCAI*, 2020, pp. 1331–1337.

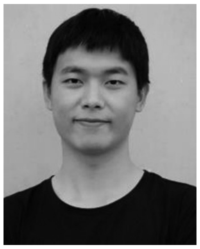
[31] T. Xia et al., “3DGCN: 3-dimensional dynamic graph convolutional network for citywide crowd flow prediction,” *ACM Trans. Knowl. Disc. Data*, vol. 15, no. 6, pp. 1–21, 2021.

Cheng Luo received the master’s degree in computer science from the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, in 2023.

His research interests include machine learning, mobile computing, and computer vision.

Rusheng Cai received the master’s degree from the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, in 2022.

His research interests include data mining, neural networks, and machine learning.



Haizhou Guo received the master’s degree from the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, in 2021.

His research interests include machine learning, neural networks, and spatiotemporal data mining.

Siting Luo will receive the qualification for admission to the bachelor’s degree in software engineering from Xidian University, Xi’an, China, in 2024.

Her research interests include big data analytics and deep learning.

Rui Mao received the B.S. and M.S. degrees in computer science from the University of Science and Technology of China, Hefei, China, in 1997 and 2000, respectively, and the M.S. degree in statistics and the Ph.D. degree in computer science from The University of Texas at Austin, Austin, TX, USA, in 2006 and 2007, respectively.

After three years work with Oracle USA Corporation, he joined Shenzhen University, Shenzhen, China, in 2010, where he is currently an Associate Dean of the College of Computer Science and Software Engineering, and the Executive Director of the Shenzhen Institute of Computing Sciences. His research mainly focuses on universal data processing.

Landu Jiang received the B.Eng. degree in information security engineering from Shanghai Jiao Tong University, Shanghai, China, in 2010, the master’s degree in computer science with a minor in construction management from the University of Nebraska-Lincoln, Lincoln, NE, USA, in 2012, and the Ph.D. degree in computer science from McGill University, Montreal, QC, Canada, in 2018.

He is currently a Research Associate Professor with Shenzhen University, Shenzhen, China. His research interests include ubiquitous computing, machine learning applications, mobile computing, computer vision, cyber–physical systems, and green energy solutions.

Dian Zhang (Member, IEEE) received the Ph.D. degree in computer science and engineering from The Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2010.

After that, she worked as a Research Assistant Professor with Fok Ying Tung Graduate School, HKUST. She also worked as an Associate Professor with Lingnan university, Hong Kong. She is currently a Professor with Shenzhen University, Shenzhen, China. Her research interests include big data analytics and mobile computing.