Towards Fine-grained Flow Forecasting: A Graph Attention Approach for Bike Sharing Systems

Suining He
Department of Computer Science & Engineering
The University of Connecticut
suining.he@uconn.edu

Kang G. Shin
Department of Electrical Engineering & Computer Science
The University of Michigan–Ann Arbor
kgshin@umich.edu

ABSTRACT

As a healthy, efficient and green alternative to motorized urban travel, bike sharing has been increasingly popular, leading to wide deployment and use of bikes instead of cars. Accurate bike-flow prediction at the individual station level is essential for bike sharing service. Due to the spatial and temporal complexities of traffic networks and the lack of data-driven design for bike stations, existing methods cannot predict the fine-grained bike flows to/from each station.

To remedy this problem, we propose a novel data-driven spatio-temporal ***G***raph attention convolutional neural network for ***Bike s***tation-level flow prediction (**`GBikes`**). We develop data-driven and spatio-temporal designs, and model bike stations (nodes) and inter-station bike rides (edges) as a graph. In particular, we design a novel graph attention convolutional neural network (GACNN) with attention mechanisms capturing and differentiating station-to-station correlations. Multi-level temporal closeness, spatial distances and other external factors (e.g., weather and points of interest) are jointly considered for comprehensive learning and accurate prediction of bike flows at each station. Extensive experiments upon a total of over 11 million trips collected from three large-scale bike-sharing systems in New York City, Chicago, and Los Angeles have corroborated `GBikes`'s significant improvement of accuracy, robustness and effectiveness over prior work.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**.

KEYWORDS

Bike sharing; station-level; flow forecasting; smart city

ACM Reference Format:
Suining He and Kang G. Shin. 2020. Towards Fine-grained Flow Forecasting: A Graph Attention Approach for Bike Sharing Systems . In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380097

1 INTRODUCTION

As one of the most popular forms of urban shared economy and smart cities, bike sharing has been changing the metropolitan transportation and people's daily lives in a significant way. Powered by recent advances in urban computing and big data, it provides an efficient, green and healthy alternative to motorized modalities, enabling the first/last-mile connectivities within a city. 35 million bike-sharing trips in the US are reported to have taken place in 2017 alone, while the global bike-sharing market is predicted to post a compound annual growth rate of close to 21% during the period 2018–2022 [31].

Considering the economic/social significance and rapid growth of bike sharing, it is essential to accurately predict the number of bike pick-ups/drop-offs at each station (dock). It also enables responsive demand-supply balancing [34], city route planning [1], station relocation [24], and fine-grained mobility analytics [6] improving the service providers' or operators' profitability and enhancing the public welfare. Despite the various approaches proposed thus far, station-level bike-flow prediction or traffic forecasting remains to be challenging due to following issues and concerns:

(1) *Complex and dynamic bike-flow patterns*: The large degrees of freedom in the first/last-mile city connectivity make bike pick-ups/drop-offs at each station highly complex and dynamic over time. In many previous studies, neighboring stations are often clustered into groups or aggregated within discretized zones (as illustrated by (a) and (b) in Fig. 1), leading to coarser granularity and making it difficult to balance each station.
(2) *Data-driven studies and designs of bike-sharing stations*: Despite the numerous time series and machine learning models studied so far, few of them have taken comprehensive data-driven approaches for deriving model parameters, components and insights for enhanced learning of rides flows.
(3) *Spatio-temporal correlations between bike stations*: Drop-offs and pick-ups (in/out flows, or $(\mathbf{I}, \mathbf{O})$) can be greatly influenced by station-to-station correlations. For example, a bike is more likely to be picked up (dropped off) at a station with more bike (dock) availability, which also depends on that of neighboring stations. Bike stations, like a *network graph*, are "linked" by riders' trips, establishing spatio-temporal inter-station correlations and particular neighbors' "attentions" (as Fig. 1(c)) which are required to predict bike flows in and out of each station.

To address these concerns, we propose `GBikes`, data-driven bike-flow prediction at each station based on *spatio-temporal graph attention convolutional neural network*. In particular, this paper makes the following contributions:

- **Comprehensive data-driven designs for bike-sharing station networks**: We have conducted comprehensive data analytics for bike station networks to design/derive data-driven components and parameters. As illustrated in Fig. 2,

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380097

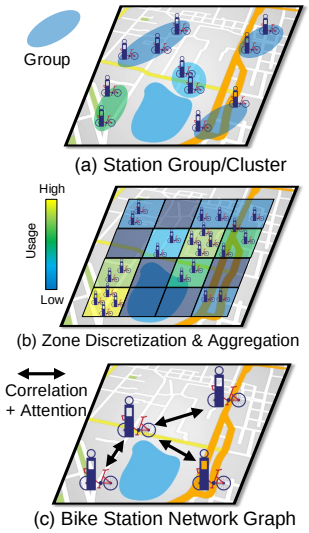

Figure 1: Difference of our formulation with previous studies.

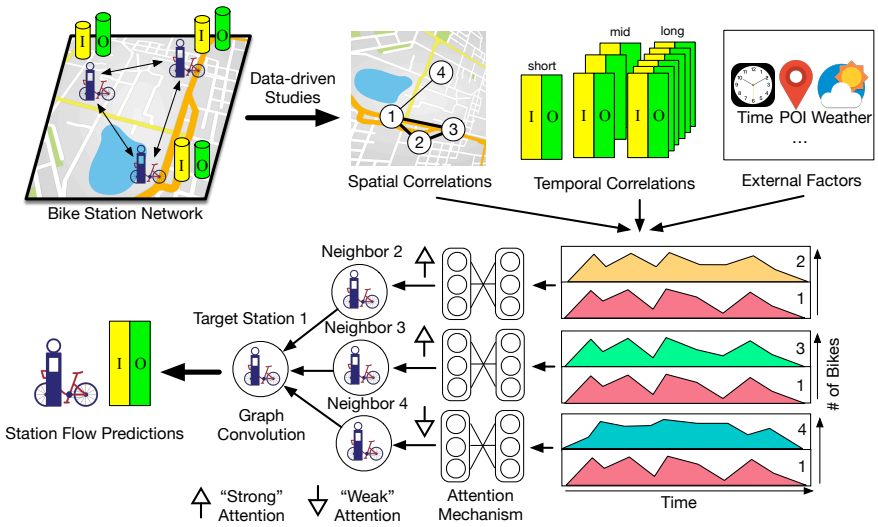

Figure 2: Illustration of comprehensive data-driven designs (upper) and graph attention (lower) in `GBikes` for a bike station network graph.

we have studied spatio-temporal factors, such as spatial station-to-station connections, multi-level temporal (**I**, **O**) trip correlations, points-of-interest (POIs) and other external factors, and then derived the corresponding component designs for `GBikes`.

- **A novel spatio-temporal graph attention convolutional neural network for fine-grained station bike-flow prediction**: Based on the data-driven designs, we propose a novel spatio-temporal design based on graph attention convolutional neural network (GACNN). Specifically, we formalize the bike station network (stations as nodes and trips as edges) into a *graph* with *attention* upon each station's neighborhood structure, as illustrated in Fig. 2. By incorporating spatio-temporal and multi-level features as well as comprehensive external factors, `GBikes` captures the complex bike-flow patterns. Station neighbors with stronger correlations are further identified and discriminated by our attention mechanism, leading to fine-grained correlation modeling and accurate bike-flow prediction.
- **Extensive experimentation and model validation**: We have conducted extensive data analytics and experimental studies on over 1.13×10^7 bike trips from three metropolitan bike-sharing systems in New York City (NYC), Chicago and Los Angeles (LA). `GBikes` is shown to outperform state-of-the-arts in terms of prediction accuracy (often by more than 20% in error reduction), effectiveness (fine-grained prediction with short time intervals) and robustness given environmental variation.

Note that our data-driven design and prediction model can be extended to applications in station-less bike-sharing [1] systems and studies of other emerging mobility/transportation networks [14, 39], including human mobility flows [22, 42], ride-sharing [10, 11, 19, 26, 35] and public transportation systems [36], and other graph network applications [37].

The rest of this paper is organized as follows. We first review the related work in Sec. 2. Then, we present the problem statement, datasets and frameworks in Sec. 3, followed by the data-driven studies and designs in Sec. 4. We then show the core model formulation of `GBikes` in Sec. 5, and the experimental results in Sec. 6. Finally, we conclude in Sec. 7.

2 RELATED WORK

Driven by increasing connectivity and exploding data in ubiquitous computing, smart transportation [17, 33, 36], including the recent bike sharing [9], has recently attracted significant attention [21, 22]. Various conventional time-series and statistical feature learning analyses have been explored for bike traffic prediction [4, 7]. The authors of [16] studied different feature learning algorithms for prediction of bike demands at a station without considering station-to-station correlations. Predicting the aggregated bike flows of stations by grouping them into clusters has been studied [5, 21] (Fig. 1(a)), which cannot support fine-grained prediction and re-balancing [16]. Since many cities have already been aware of irregular drop-offs and road congestion caused by station-less bike sharing, and have thus enforced geo-fenced deployment, we focus on the station-based model thanks to its better social acceptance.

By discretizing a city map into grids or zones [40], image processing techniques like deep residual network [43] and fusion of CNN (Convolutional Neural Network) with LSTM (Long Short-Term Memory) or RNN (Recurrent Neural Network) [27] have been considered to predict aggregated flows for each zone (Fig. 1(b)). However, the image-based formulation may not be easily extended to fine-grained prediction of flows at each individual station.

With advances of geometric signal processing [2], deep graph learning for the non-Euclidean data has been proposed and studied [37, 38]. Graph data in many real-world applications [15, 37], with variable numbers of both un-ordered nodes and neighbors for each node, makes conventional operations like convolutions difficult to apply. To enable graph convolution, various theoretical foundations have been established, including those on spectral

graph theory [3], spatial-based aggregation [8] and pooling modules [12]. Despite the differences in notations and approximations, their basic idea all tries to propagate and aggregate the neighbor feature information of nodes in a graph iteratively until convergence.

The graph convolutional neural network [18] has attracted attention in formatting datasets as networks (say, knowledge graphs and social networks). Recently, they have been extended to urban traffic, investigating speed prediction for road segments and vehicle flows in [4, 23, 41]. While others only considered correlating zones or locations with their mutual geographical distances, GBikes investigates comprehensive spatio-temporal features via data analytics to enable more fine-grained model designs, differentiating correlations of nearby stations and determining multiple levels of temporal correlations. It designs spatio-temporal *graph attention mechanisms* which efficiently capture inter-station flows, without relying on sophisticated sequence matching via LSTM/RNN [30] and complicated image convolutions.

In the neighborhood aggregation process, conventional graph convolution assigns a weight upon two neighboring nodes based on only their degrees. Unlike the above, graph attention in GBikes, as illustrated in Fig. 2, introduces an additional network structure between neighboring nodes, and thus more important/correlated neighboring station nodes (say, stations 2 & 3 for target station 1) are assigned with "stronger attentions" and larger weights than others (say, station 4 in the example). This way, the stations which are more correlated spatio-temporally can be further differentiated, yielding better flow prediction.

3 PROBLEM, DATA & FRAMEWORK

We first present the important concepts and core problem of GBikes in Sec. 3.1. Then, we show the datasets for our data-driven designs and experimental evaluation in Sec. 3.2, followed by an overview of the data-driven forecasting framework of GBikes in Sec. 3.3.

3.1 Important Concepts & Core Problem

Definition 1. *(Bike station network): The bike-sharing system under consideration consists of N stations. Each station i is associated with location coordinates* $[lat_i, lon_i]$*. Each trip, as a link, corresponds to a user's bike ride from one station to another within a certain amount of time. With stations as nodes and trips as edges, a graph can be formed to characterize the bike station network.*

Two bike stations are considered to be *adjacent* or *neighbors* if there are trips between them. The *adjacency* matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ is then formed as a weight function representing their correlations (detailed in Secs. 4 & 5).

Definition 2. *(Station-level bike flows): Let* $\tau(i,j)$ *be the total number of rides with a start station i and destination j. Note that* $\tau(i,j) \neq \tau(j,i)$*. Then, the aggregated flows in and out of station i are, respectively, denoted as*

$$I_i = \sum_{j=1}^{N} \tau(j,i), \quad \textit{and} \quad O_i = \sum_{j=1}^{N} \tau(i,j). \tag{1}$$

For ease of flow studies, the time domain can be discretized into slots of certain fixed length, which can be task-dependent, balancing between granularity and efficiency. When initiating each forecasting, we let k be the latest time interval with known historical flows and $k+1$ be the following interval.

Formally, let $\mathbf{h}_i^{(k)} = \left(I_i^{(k)}, O_i^{(k)}\right)$ represent the numbers of aggregated bike drop-offs (in) and pick-ups (out) at a station i in the time interval k ($\mathbf{h}_i^{(k)} \in \mathbb{R}^2$), and $\mathbf{h}^{(k)} = \left[\mathbf{h}_1^{(k)}, \ldots, \mathbf{h}_N^{(k)}\right]$ be the flows of all stations. Each set of flows $\mathbf{h}^{(k)}$, as the features of all stations, is then formulated into an $N\times 2$ matrix for later model input. Let $\mathbf{E}^{(k)}$ be the set of other environment or *external* factors (say, weather) related to the bike flows.

Definition 3. *(Station-level bike-flow prediction): Given w sets of historical flows*

$$\mathbf{H} = \left\{\mathbf{h}^{(k-w)}, \mathbf{h}^{(k-w+1)}, \ldots, \mathbf{h}^{(k)}\right\}, \tag{2}$$

at all stations, station-to-station correlations $\mathbf{A}$*, and other external factors* $\mathbf{E}^{(k)}$ *at time interval k, we want to predict the station-level bike flows* $\widehat{\mathbf{h}}^{(k+1)}$ *at the interval (k+1) by a prediction method* $\mathbb{F}$*, i.e.,*

$$\widehat{\mathbf{h}}^{(k+1)} = \mathbb{F}\left(\mathbf{H}, \mathbf{A}, \mathbf{E}^{(k)}\right), \quad \widehat{\mathbf{h}}^{(k+1)} \in \mathbb{R}^{N\times 2}. \tag{3}$$

3.2 Datasets for Analytics & Evaluation

We conduct extensive station network studies and model evaluations based on the following three open datasets:

- **Citi Bike, NYC**: which consists of a total of N=502 stations and 7,628,418 trips in 2015Q3Q4 and 2016Q1Q2.
- **Divvy, Chicago**: which consists of a total of N=607 stations and 3,214,965 trips in 2018Q2-Q4.
- **Metro Bike, LA**: which consists of a total of N=135 stations and 447,408 trips in 2017Q3Q4 and 2018Q1-Q4.

The trip datasets of $\tau(i,j)$'s include start/destination stations (GPS coordinates), and related pick-up/drop-off timestamps (and trip duration). Regarding each dataset, we have conducted data cleaning to filter out those with abnormal trip duration (say, negative readings or more than 24 hours) or missing pick-up/drop-off locations. For each city, we also include the points of interests (POIs), the city map (from the Open Street Map (OSM)), external factors like events, time and weather (detailed in Sec. 4).

3.3 System Framework of GBikes

As shown in Fig. 3, the system framework of GBikes consists of data-driven studies and designs (Sec. 4), and the in/out flow prediction (Sec. 5), for both training and testing the GBikes model. In real-world deployment by the bike-sharing service providers, the entire system of GBikes can be run on a server or cluster.

In the model training and offline learning, the bike-sharing service providers first provide the historical bike-sharing records, station locations, city map (with points of interests) and other external factors (including weather). GBikes pre-processes the above based on the data-driven designs, and the studies provide the following inputs to the in/out flow prediction $\mathbb{F}(\cdot)$: the historical bike trips and in/out flows $\mathbf{H}$ parsed in different levels of temporal closeness (say, Levels 1 to J; Sec. 4.2) from the target interval $(k+1)$ to be predicted (with the corresponding correlations $\mathbf{A}$ between stations; Sec. 5.1), and the external factors $\mathbf{E}$ (Sec. 4.3).

The inputs are, respectively, fed to the following two different components: a set of total J Graph Attention Convolutional Neural

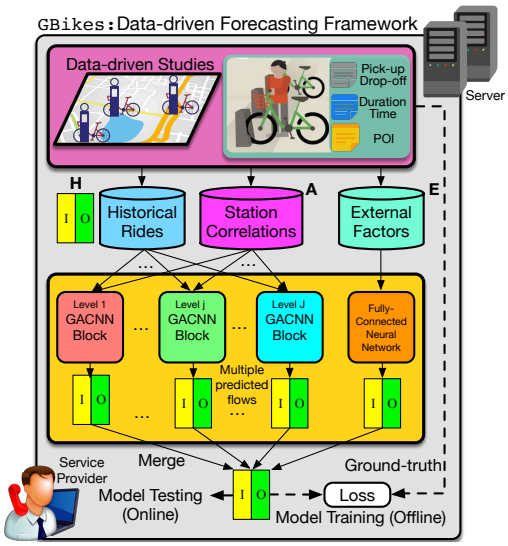

Figure 3: System framework & information flow of GBikes.

Network (GACNN) blocks regarding multi-level bike flows (Sec. 5.2), and a fully-connected neural network regarding external factor **E** (Sec. 5.3). Finally, total $(J+1)$ sets of predictions are merged into the final predicted flows. The prediction values are compared against the ground-truth in/out flows in the target interval, and the core model is trained by minimizing the loss.

In model testing and online deployment, the station-level bike flows are predicted given a batch of historical rides (parsed in multiple levels as above) and returned to the service provider, enabling other advanced/high-level applications like station-level bike balancing [20] or anomaly detection. In a practical deployment, the model can be updated, given new bike-flow records and external factors via online learning [20] or model fine-tuning [33].

4 DATA-DRIVEN STUDIES & DESIGNS

In this section, we focus a representative data-driven design on the Citi Bike system in NYC while presenting some variations in other two systems. Fig. 4 shows the spatial distributions of NYC bike stations with their total historical demands (aggregated within the first week of 2015Q3). One can see most of the demands (warmer colors) taking place at the stations in Central Business District of Manhattan. Since the co-occurrence of those station bike demands is dynamic and complex, our studies need to derive important designs and data structures of the station network connection, benefiting the subsequent traffic forecasting.

Given the complexity of bike trips linking stations, we further incorporate the following data-driven designs within GBikes's model: *station-to-station distances* (Sec. 4.1), *levels of temporal closeness* (Sec. 4.2), and *points-of-interest (POIs) & other factors* (Sec. 4.3). Note that the parameters derived from our data-driven studies are based on the historical (training) data which is separate from the test data.

4.1 Station-to-station Distances

A bike station network caters for the first/last-mile urban commute. Since a rider is likely to pick up or drop off a bike from the nearest station, the station-to-station distances have the greatest effect upon the bike flows.

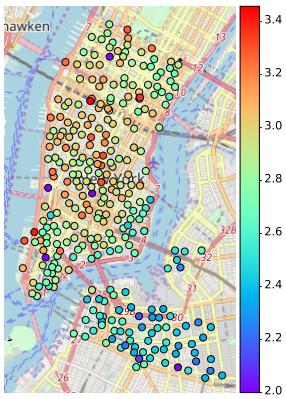

Figure 4: Distributions of NYC station demands ($\log_{10}(\cdot)$).

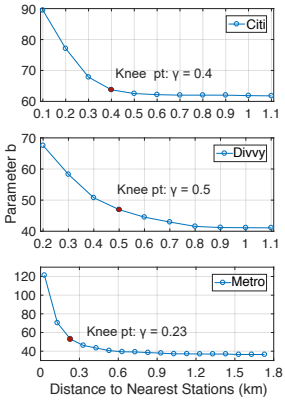

Figure 5: Station usage vs. distance to nearest neighbors.

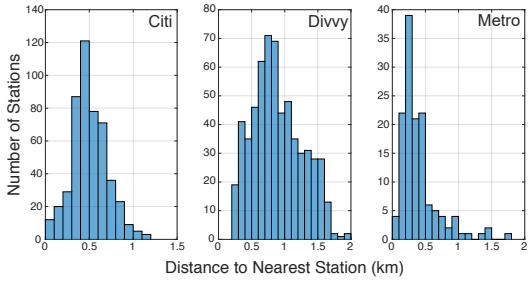

Figure 6: Histograms of stations' distances to their nearest peers (NYC Citi Bike, Chicago Divvy & LA Metro).

Observations: Specifically, in a metropolitan area, we consider the shortest road/street path between each pair of stations, which is derived from its map from OSM (we find and use the street centerlines of the three cities from OSM in this study).

To evaluate station-to-station correlations, we conduct negative binomial regression (NBR) [13] on the station usage (total pick-ups and drop-offs) against different shortest path distances d (km) to the nearest peers. NBR finds the set of parameters b's ($b > 0$) that maximize the log-likelihood of

$$\ln z = b_0 + b \cdot d. \tag{4}$$

Fig. 5 shows that use of a station's bikes is more correlated with that of its neighbors', and as distance grows, the positive coefficient b between stations begins to decrease for all systems due to fewer distant trips. Also, Fig. 6 shows the histograms of stations' distances (the shortest road paths on the map) to their nearest neighbors, exhibiting the last/first-mile connectivity of the bike station networks. We can also observe Divvy has much sparser bike station network than other two systems. Therefore, despite far more stations in total, the closely related neighbors for Divvy are not significantly more than other systems in NYC and LA (which will be reflected by the number of attention heads in Sec. 6.2).

Designs: To better differentiate the *neighbors* $\mathcal{N}_i$ for each station i, we select the knee point d in Fig. 5 (where a curve "turns", or formally, where a curve is best approximated by a pair of lines) *w.r.t.* each curve, and decide i and j to be *close neighbors* if $d(i,j) \le \gamma$, and *distant neighbors* otherwise. Since a station has trips both starting from and returning to itself, we also consider $i \in \mathcal{N}_i$ in GBikes's formulation.

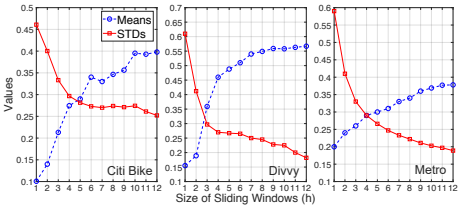

Figure 7: Station usage correlations vs. levels of temporal closeness.

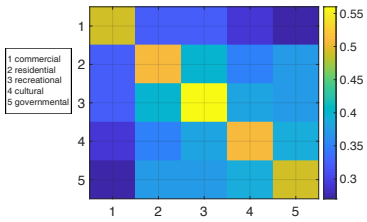

Figure 8: Matrix of mean correlations between different POI types (NYC Citi Bike).

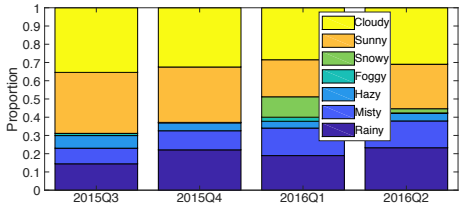

Figure 9: Proportions of weather conditions (NYC Citi Bike).

4.2 Levels of Temporal Closeness

Recall that bike-flow prediction of GBikes takes into account the historical bike trip records **H** at each station. Due to dynamic variations in people's commute purposes, relations between station bike flows may vary with time. We characterize in GBikes the variations into *multi-level temporal closeness* from the target time interval to be predicted.

Observations: We show in Fig. 7 the station-to-station correlations vs. levels of temporal closeness. For each station pair, we find the *Pearson correlation* of station bike pick-ups within a certain sliding time window, and plot the means and standard deviations (STDs) of all pairs' correlations. Note that the correlation values between some stations can be negative. We can see lower and more dynamic short-term (say, less than 3 hours) correlations due to individual pick-up patterns, and larger and more consistent long-term (say, greater than 6 hours) patterns resulting from the general trend of commute flows and the spatial functionality of the city regions.

We have studied the distributions of travel time between stations. Since most bike rides are done within a certain period of time (due to the nature of the first/last-mile connectivity), the size of short-term partition caters for the majority of bike trips, say, 90% in the cumulative distribution of travel time for each bike-sharing system (Citi: 0.46 h; Divvy: 0.43 h; Metro: 0.70 h).

Designs: In this work, we propose using multiple levels of temporal closeness to accommodate above. In our prototype studies, we hence incorporate the *short-*, *mid-* and *long*-term levels of temporal correlations (with the numbers of historical intervals or the window sizes respectively denoted as w_A, w_B and w_C) in GBikes. In other words, we set $J = 3$ in Fig. 3 (Sec. 3.3).

Specifically, based on the above time-domain studies (each interval is 15 min here), we set the number of intervals (w_A, w_B, w_C) as (3, 16, 24) for Citi, (3, 12, 16) for Divvy and (5, 12, 20) for Metro. For each interval $(k+1)$ to predict and each $w \in \{w_A, w_B, w_C\}$, we find the historical records

$$\mathbf{h}^{(w)} \triangleq \left\{ \left(\sum_{t=1}^{w} I_i^{(k-t+1)}, \sum_{t=1}^{w} O_i^{(k-t+1)} \right), \forall i \right\}, \tag{5}$$

which is formed to be an N×2 matrix as input.

Regarding selection of interval/slot size, 15 min discretization is chosen for fine-grained and practical prediction of real-world bike-sharing systems, and is the minimum time-granularity of weather data available to us (NOAA open data). However, our scheme can be applied with shorter intervals (say, 5 min) if and when such data is available. Furthermore, existing studies select even larger time intervals (say, 1 hour in [43]), while randomness and complexity in a shorter slot poses more challenges upon bike-flow prediction. We show later in Sec. 6 the better experimental performance of GBikes than other schemes under different interval sizes.

4.3 POIs & Other External Factors

Points-of-Interest (POIs) Data: The bike-sharing riders usually have frequent travel patterns (habits) between certain city functional regions [25] due to their specific commute purposes and ride preferences, particularly during morning/evening rush hours.

We use the points-of-interest (POI) in each bike station's neighborhood to accommodate these patterns. A total of 19,867 POIs in NYC are collected via NYC Open Data Portal, 4,329 POIs in Chicago, and 5,948 POIs in LA based on OSM. Taking NYC as a typical example, we show in Fig. 8 the matrix of mean correlations (sliding window of 6 hours) among different types of the POI neighborhoods. We classify the POIs into five major groups: *commercial*, *residential*, *recreational*, *cultural*, and *governmental* (indexed by 1 to 5 in Fig. 8). Then, we summarize and find the majority types of POIs near each station (within a circle of radius 0.4 km centered at the station). We can see clearly lower correlations between stations in residential areas and others, due to reverse directions of inter-station bike flows during daily commute.

Meteorological & Event Data: We also take into account the meteorological (obtained from NOAA database [29]) and event factors, including the categorical (7 types of weather conditions) and non-categorical ones (temperature, sky visibility, wind speed, wind direction, humidity, air pressure, day of a week, hour of a day and public holiday or not). Taking NYC as an example, we have shown in Fig. 9 the proportions of weather conditions in the four different seasons (2015Q3–2016Q2). Such temporally diverse patterns are considered within GBikes formulation to enhance the prediction accuracy.

Designs: We process each type of the categorical data (say, sunny or not) by one-hot encoding (say, 1 if sunny and 0 otherwise), and other non-categorical ones into [0,1] by max-min normalization. We combine meteorological, time/event and POIs (5 types in a non-categorical form; number of each type within a certain distance from the station) into a vector of external factors (further embedding is discussed in Sec. 5.3). Table 1 further summarizes **E** with their dimensions and detailed descriptions in GBikes.

5 SPATIO-TEMPORAL GRAPH ATTENTION CONVOLUTION

A spatial and temporal design is essential to accommodate aforementioned factors. Since we model the bike stations as the network graph, a graph neural network model is proposed to capture the

Table 1: External factors E studied in GBikes.

Factors	Data
Weather Conditions (7-D)	cloudy/sunny/foggy/hazy/misty/rainy /snowy or not
Meteorological Metrics (6-D)	temperature, sky visibility, wind speed, wind direction, humidity, pressure
Event/Time (3-D)	day of a week, hour of a day, holiday or not
Neighborhood POIs (5-D)	commercial, residential, recreational, cultural, governmental

station-to-station correlations, which are further differentiated by our graph attention mechanism.

We first present the design of spatial and temporal closeness in Sec. 5.1, and then show the core graph attention convolution in Sec. 5.2, followed by the core network architecture in Sec. 5.3.

5.1 Design of Spatial & Temporal Closeness

In a bike station network, closer stations likely have stronger links and mutual effects than those more distant stations, which are characterized within the adjacency matrix $\mathbf{A}$ of our graph model.

Definition 4. *(Spatial closeness): To differentiate the close and distant neighbors (Sec. 4.1), we define spatial closeness between stations i and j in terms of mutual distance as*

$$A^{(dist)}(i,j) \triangleq \begin{cases} (1+d(i,j))^{-1}, & \text{if} \quad d(i,j) \le \gamma; \\ (1+d(i,j))^{-2}, & \text{otherwise.} \end{cases} \tag{6}$$

where $d(i,j)$ (unit: km) is derived based on the shortest path distance on the OSM map, and γ is the decision boundary derived in Sec. 4.1 between close and distant neighbors.

In other words, $A^{(\text{dist})}(i,j)$ decreases given larger $d(i,j)$ between stations i and j, and close and distant neighbors in Sec. 4.1 are differentiated in GBikes formulation by Eq. (6).

Besides the closeness in mutual distance, the adjacency matrix also takes into account the temporal correlations between stations. Specifically, let $F^{(t)}(i,j)$ $(i \neq j)$ be the total number of trips happening between stations i and j in interval t, *i.e.*,

$$F^{(t)}(i,j) = \tau^{(t)}(i,j) + \tau^{(t)}(j,i). \tag{7}$$

Definition 5. *(Temporal closeness): Based on the cosine similarity, we define the temporal closeness of flows $A_w^{(temp)}(i,j)$ for $\forall i,j \in \{1,\ldots,N\}$, in the recent w windows as*

$$A_w^{(temp)}(i,j) \triangleq \frac{\sum_{n=1}^{N} (F_w(i,n) \cdot F_w(j,n))}{\sqrt{\sum_{n=1}^{N} (F_w(i,n))^2} \cdot \sqrt{\sum_{n=1}^{N} (F_w(j,n))^2}}, \tag{8}$$

where the aggregated flow $F_w(i,n)$ from stations i to n in a sliding window of w by the current interval k is given by

$$F_w(i,n) = \sum_{t=k-w+1}^{k} F^{(t)}(i,n). \tag{9}$$

In other words, two stations are considered to be more temporally correlated if they have more similar traffic "pulses" within a window of w, as characterized by Eq. (8).

For different levels of temporal closeness w's (Sec. 4), we find the corresponding $\mathbf{A}_w^{(\text{temp})}$. Finally, the closeness between stations representing strengths of their mutual connectivities, formed as the weight (adjacency) matrix $\mathbf{A}_w$, is given by

$$\mathbf{A}_w \triangleq \mathbf{A}^{(\text{dist})} + \mathbf{A}_w^{(\text{temp})}, \quad w \in \{w_A, w_B, w_C\}, \tag{10}$$

where each element $A_w(i,j)$ is then normalized *w.r.t.* each station i before being fed to GBikes's core.

5.2 Graph Attention Convolution in GBikes

We introduce graph convolution and present core layer of graph attention convolutional neural network (GACNN).

Basic Graph Convolution: Based on the spectral graph theory [2], the operation of spectral convolutions on graphs [18] is defined as the multiplication of an input signal $\mathbf{x} \in \mathbb{R}^N$ with a *graph filter* g_θ in the Fourier domain, *i.e.*,

$$g_\theta \star \mathbf{x} = \sum_{p=1}^{P} \theta_p \mathbf{U}\mathbf{\Lambda}^p\mathbf{U}^T\mathbf{x} = \sum_{p=1}^{P} \theta_p \mathbf{L}^p\mathbf{x}, \tag{11}$$

where $\mathbf{U}$ is the matrix consisting of eigenvectors from the graph Laplacian $\mathbf{L}$, *i.e.*, $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$. Here, $\mathbf{D}$ is the degree matrix, $\mathbf{A}$ is the adjacency matrix, $\mathbf{\Lambda}$ is a diagonal matrix of $\mathbf{L}$'s eigenvalues, $\mathbf{L}^p$ represents its p-th power, and $\mathbf{U}^T\mathbf{x}$ is the graph Fourier transform.

The work in [18] further simplified the graph convolution operation by the first-order polynomial approximation, *i.e.*,

$$g_\theta \star \mathbf{x} \approx \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{x}\theta, \tag{12}$$

which then serves as the graph convolution operation.

Given above, by formulating the station flows $\mathbf{h}$ as input signal $\mathbf{x}$ in Eq. (12) (each pair of pick-up/drop-off values as a feature vector of a station), we further leverage the graph convolution, as a *graph filter* [18] for prediction. The convolved signal matrix $\mathbf{z}^{(l)} \in \mathbb{R}^{N \times D'}$ from the l-th graph convolution layer is then given by

$$\mathbf{z}^{(l)} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_w\mathbf{D}^{-\frac{1}{2}}\mathbf{h}^{(l)}\mathbf{W}^{(l)}, \tag{13}$$

where $\mathbf{W}^{(l)} \in \mathbb{R}^{D \times D'}$ is the weight matrix of the neurons at the layer l, and D (D') is the filter's input (output) feature dimension. In each graph convolution layer, the input signals are first fed to the graph filter in order to aggregate the neighborhood information of each node. Similar to 2-D image convolution, connections between a station node and its neighbors are captured within the graph convolution [37].

Graph Attention & Core GACNN Layer: In order to better learn the complexity of station bike flows, we further design *graph attention mechanisms* [32] upon $\mathbf{z}^{(l)}$ to *differentiate the more closely correlated stations*. Let $\mathbf{\Omega} \in \mathbb{R}^{D' \times D}$ be the weight matrix applied upon each station node. We define the attention coefficient, as a weighting function with *ELU* (exponential linear unit) activation (denoted as $\sigma(\cdot)$), between a station i and one of its neighbors j's ($j \in \mathcal{N}_i$) as

$$e(i,j) \triangleq \sigma\left(\vec{\mathbf{a}}^T\left[\mathbf{\Omega}\mathbf{h}_i || \mathbf{\Omega}\mathbf{h}_j\right]\right), \quad \mathbf{h}_i, \mathbf{h}_j \in \mathbb{R}^D, \tag{14}$$

where $\mathbf{\Omega}\mathbf{h}_i, \mathbf{\Omega}\mathbf{h}_j \in \mathbb{R}^{D'}$, and $\vec{\mathbf{a}} \in \mathbb{R}^{2D'}$ is the attention weight vector of a single-layer feed forward neural network, and $||$ is the concatenation operation of two input vectors. Here the *ELU* activation function is formally given by

$$\sigma(x) \triangleq \begin{cases} \lambda(\exp(x)-1), & \text{if} \quad x < 0; \\ x, & \text{otherwise.} \end{cases} \tag{15}$$

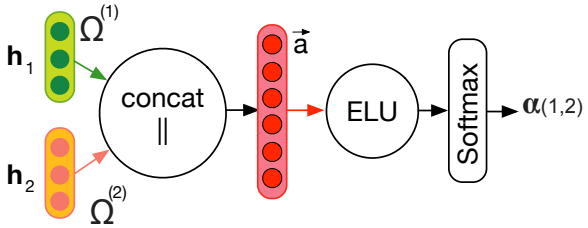

Figure 10: Illustration of attention mechanism structure between stations 1 & 2. Their weighted inputs by Ω are concatenated and fed via a single layer with weight vector $\vec{a}$.

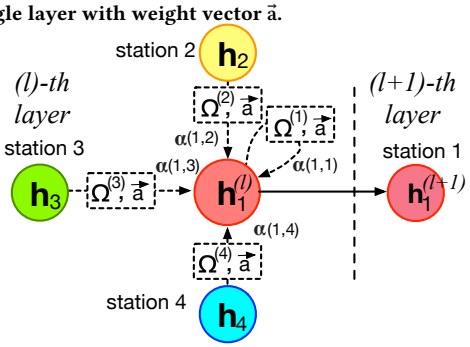

Figure 11: Illustration of attention propagation in station 1's neighbors. Information from neighbors and the station itself is combined to form the new features for station 1.

We set $\lambda = 1.0$ by default in our model.

Then, the attention coefficient $e(i, j)$ is passed through a `softmax` function with normalization, and we get

$$\alpha(i,j) \triangleq \texttt{softmax}(e(i,j)) = \frac{\exp(e(i,j))}{\sum_{n\in\mathcal{N}_i}\exp(e(i,n))}, \tag{16}$$

which ensures that the neighborhood attention coefficients of a station i sum up to one. We further illustrate the attention structure between neighboring stations 1 and 2 in Fig. 10.

For each link from a neighboring station $j \in \mathcal{N}_i$ to i, we consider M attention *heads* or mechanisms, each of which is weighted by an attention coefficient $\alpha^{(m)}(i, j)$. At the processing layer of `GBikes`, the features from the M attention heads (with weight $\mathbf{\Omega}^{(m)} \in \mathbb{R}^{D'\times D}$ for each head) are *concatenated* to obtain the output to layer $(l + 1)$, denoted as $\mathbf{h}^{(l+1)}$, *i.e.*,

$$\mathbf{h}^{(l+1)} \triangleq \Big\Vert_{m=1}^{M} \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha^{(m)}(i,j)\mathbf{z}^{(l)}\mathbf{\Omega}^{(m)}\right), \tag{17}$$

where we also apply the *ELU* activation as $\sigma(\cdot)$.

Unlike the input layer, at the prediction layer of `GBikes`, the outputs from all attention mechanisms of all neighboring stations in the previous layer l are *averaged* (and fed to *ELU* activation) into

$$\mathbf{h}^{(l+1)} \triangleq \sigma\left(\frac{1}{M}\sum_{m=1}^{M}\sum_{j\in\mathcal{N}_i}\alpha^{(m)}(i,j)\mathbf{z}^{(l)}\mathbf{\Omega}^{(m)}\right), \tag{18}$$

which is more sensitive to signals between layers than concatenation in Eq. (17) [32]. Fig. 11 illustrates the attention heads among four stations (nodes) with input features $\mathbf{h}_1$ to $\mathbf{h}_4$. For station 1, attentions from itself and its neighbors are concatenated (Eq. (17)) or averaged (Eq. (18)), and new features $\mathbf{h}'_1$ are returned for the next layer's processing.

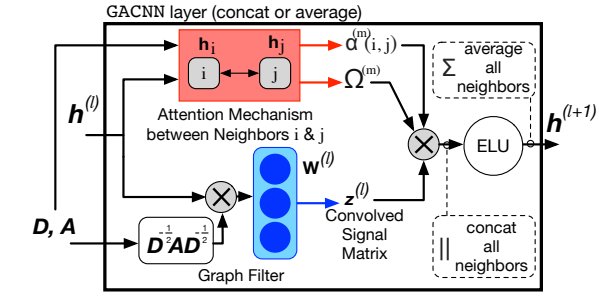

Figure 12: Illustration of the GACNN layer, where the inputs are propagated through both graph filter and attention mechanism. Concatenation or averaging operation is applied at the final output.

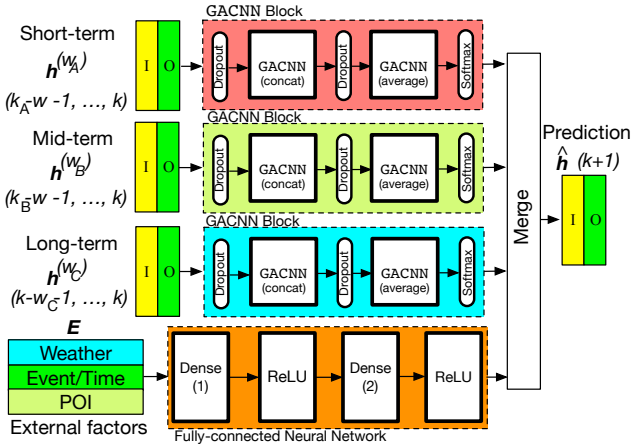

Figure 13: Station-level flow prediction architecture in `GBikes`.

Via the above attention mechanism structure, `GBikes` is enabled with better focus upon the station neighbors which are more closely related, thus deriving more important correlations to enhance prediction accuracy.

Finally, we summarize the structure of GACNN layer in Fig. 12. Inputs are respectively fed to graph convolution filter and attention mechanism, whose results are later merged.

5.3 Network Architecture of Station-level Traffic Forecast

We present in Fig. 13 the network architecture and information flows for traffic forecast. Specifically, `GBikes` consists of two processing components, *i.e.*, the *spatio-temporal* and *external* learners. Via bike station network studies, we first derive the bike flows, $\mathbf{h}^{(w_A)}$, $\mathbf{h}^{(w_B)}$ and $\mathbf{h}^{(w_C)}$, in three levels of temporal closeness, as well as external factors $\mathbf{E}$.

Spatio-temporal learner by *GACNN Blocks*: Three blocks of GACNNs respectively learn each level of in/out flows. Each block consists of:

- *GACNN*: As illustrated in Fig. 12, both GACNN layers have graph filter (Eq. (13)) accompanied by the graph attention mechanisms (Eqs. (17) or (18)). Each attention coefficient $\alpha(i, j)$ is calculated via Eqs. (14) and (16), as in Fig. 10. The first GACNN layer applies the concatenation operation in Eq. (17), while the second applies averaging as Eq. (18).
- *Dropout & softmax*: Dropout layers are placed between input and GACNN (`concat`), and between the two GACNN layers, to

coarsen the graphs into high-level sub-structures [37]. This way, more compact representation of the station network can be derived. The softmax layer connects the block output for predictions.

The layer-to-layer information propagation is as follows: (1) In each block, the input first goes through the dropout layer, followed by GACNN (`concat`). In GACNN (`concat`), the flow input $\mathbf{h}$ after dropout is fed to the graph filter as in Fig. 12, and the output, a convolved signal matrix $\mathbf{z}$, is fed to Eq. (17) with M attention heads' coefficients concatenated into a vector. (2) The result after another dropout layer is then fed to GACNN (`average`), where the convolved signal $\mathbf{z}$ from graph filter is averaged by all attention mechanisms of all neighboring stations as in Eq. (18). The output is finally fed to the `softmax` layer and a set of predictions is returned *w.r.t.* each GACNN block.

Then, we obtain three sets of predictions in total regarding different levels of temporal closeness.

External learner by *Fully-connected Neural Network*: External factors $\mathbf{E}$ are fed to a fully-connected neural network, with two dense layers interleaved with two *ReLU* layers. The first layer serves as an embedding one to extract features from the external factors [33]. The network also returns the set of prediction upon the in/out flows. Note that the external learner is general to accommodate other factors if available.

Four sets of in/out flows (each is an $N\times 2$ matrix) are returned (three ($\widehat{\mathbf{h}}^A, \widehat{\mathbf{h}}^B, \widehat{\mathbf{h}}^C$) from spatio-temporal and one $\widehat{\mathbf{h}}^E$ from external) and merged into final predictions by averaging.

6 EXPERIMENTAL EVALUATION

Given above, we first present the experimental settings in Sec. 6.1, followed by the evaluation results in Sec. 6.2.

6.1 Experimental Settings

Based on the datasets in Sec. 3.2, we compare `GBikes` with the following baseline methods and state-of-the-arts:

• *HA* and *SHA*: Historical Average [7] and Seasonal Historical Average, which leverage the averages of the historical records belonging to the same periods of all or the same seasons. For example, we average all numbers of the rides during 8:00 – 8:15 of all recorded Mondays to predict that of 8:00 – 8:15 on a targeted Monday.

• *ARIMA*: predicts the trip series based on Auto Regressive Integrated Moving Average. We empirically set the size of sliding window to 12.

• *ANN*: leverages the Artificial Neural Network for trip series regression (trained upon a window of 12) and prediction.

• *RNN* [30]: leverages the Recurrent Neural Network for trip time-series prediction.

• *LSTM*: learns (upon a window of 12) and predicts the trip series with the Long Short-Term Memory neural network.

• *STCNN*: discretizes the map into grids, learns the bike trip heatmap distributions with Spatio-Temporal Convolutional Neural Network [27], and outputs each station's flows by the average of all stations within a grid.

• *GC*: predicts the trips with the conventional Graph Convolutional neural network [23], which only considers the link correlations between stations.

Table 2: Prediction RMSEs of all schemes in different systems.

Schemes	Citi	Divvy	Metro
HA	7.69±2.61	7.41±1.48	5.58±0.64
SHA	5.22±1.92	4.87±1.62	3.23±0.61
ARIMA	3.85±2.38	3.70±1.35	3.28±0.65
ANN	4.76±2.46	5.13±1.84	3.06±0.75
LSTM	3.76±0.53	3.67±1.83	3.20±0.63
RNN	3.37±0.98	4.05±2.23	2.92±1.42
STCNN	3.25±1.07	2.38±0.58	2.21±0.40
GC	2.45±0.95	2.12±0.46	2.03±0.41
MGN	2.43±1.57	2.13±0.44	1.97±0.45
GBikes	**1.74±0.94**	**1.69±0.45**	**1.47±0.39**

Table 3: Prediction RMSEs of all schemes during rush hours.

Rush Hours	Schemes	Citi	Divvy	Metro
Morning	STCNN	2.72±1.13	2.44±0.82	2.32±0.83
	GC	2.73±1.03	2.30±0.61	2.18±0.83
	MGN	2.53±1.43	2.23±0.86	2.16±0.88
	GBikes	**1.78±1.01**	**1.85±0.62**	**1.49±0.85**
Evening	STCNN	3.07±1.44	3.10±0.66	3.09±0.60
	GC	2.99±0.94	2.90±0.60	2.17±0.66
	MGN	2.88±0.98	2.96±0.62	2.22±0.58
	GBikes	**2.30±0.70**	**2.33±0.61**	**1.53±0.79**

• *MGN*: predicts the trips with Multi-Graph Neural network [4], which considers some correlations between stations without comprehensive data-driven designs and graph attention.

We have implemented `GBikes` and other schemes in Python and Tensorflow, and the models are trained and evaluated upon a desktop server with Intel i7-8700K 3.70 GHz, 16GB RAM, Nvidia GTX 1080Ti and Windows 10. Computationally, the time complexity of a single graph filter takes $O(|E|DD')$ [18], and that of a single attention head in `GBikes` takes $O(NDD' + |E|D')$ [32]. The number of edges $|E|$ in our studies is overall linear in N, the number of stations, due to the sparsity of the bike station network (Fig. 6). Training time of `GBikes` in our studies is around: 4.1 hours for NYC, 4.9 hours for Chicago, and 1.1 hours for LA, based on the above machine we used.

Unless otherwise stated, we use the following default experimental and parameter settings. In Eq. (6), γ is 0.4, 0.5 and 0.23 for Citi, Divvy and Metro, respectively. Number of heads M is set to 8 for Citi, 9 for Divvy, and 7 for Metro. We have 96 intervals per day for all three datasets. For each dataset, we use the first 60 days' trips for model training, the following 30 days for model validation and sensitivity analysis, and the rest for model testing. The number of epochs is 200 and that of the graph filter for each GACNN is 3. Dropout rate is set to 0.5. For the graph filters of the two GACNN layers, *i.e.*, `concat` and `average`, input dimension D is 2 and 8; output dimension D' is 8 and 2. Output dimensions for Dense (1) and (2) in Fig. 13 are 10 and $2N$, respectively. We adopt the *Adam* optimizer for model training (learning rate as 0.01).

Regarding the POI features, for each station we find a POI feature vector, each element of which represents the number of POIs of a certain type within the radius. As the radius increases, the numbers increase, and we observe when radius reaches 0.4km, the average percentage of non-zero elements of all vectors rises right above 75% and the increase begins to slow down. Thus, we choose 0.4km, which suffices to provide informative POI vectors. We evaluate the

Table 4: Prediction RMSE of GBikes under design variations.

Variations	Citi	Divvy	Metro
No POI	1.94±0.46	1.65±0.86	1.54±0.56
No weather	1.81±0.51	1.64±0.53	1.56±0.79
No $\mathbf{E}$	2.14±0.34	1.78±0.76	1.74±0.45
No $\mathbf{A}_{w}^{(temp)}$	2.05±0.69	1.98±0.82	1.96±0.58
No attention	2.18±0.61	2.12±0.62	2.06±0.52
Complete	1.66±0.62	1.44±0.61	1.10±0.38

accuracies of all schemes based on *RMSE* (root-mean-square error), *i.e.*, RMSE $= \sqrt{\frac{1}{W}\sum_i (h_i - \widehat{h}_i)}$, where h_i and $\widehat{h}_i$ are ground-truth and prediction, and W is the total number of all predicted values.

6.2 Evaluation Results

We first present the overall performance of GBikes and other comparison schemes, and then provide the sensitivity studies under different settings, followed by a comprehensive case study with dynamic flow predictions.

Overall performance: Table 2 shows the RMSEs of all different schemes at the three systems. Without considering the correlations between stations, conventional methods for time series analysis result in higher rates of error. STCNN takes into account regions and hence achieves better accuracy than ANN, RNN and LSTM, but its convolution cannot provide fine-grained bike flows for each station. GC and MGN consider the graph of stations and their connectivity, but fail to comprehensively capture their spatio-temporal correlations and relative importance.

Compared to the above schemes, GBikes achieves better overall performance thanks to its comprehensive data studies and spatio-temporal attention designs. As Citi Bike in NYC has more stations and trips than the other two systems, higher prediction errors are expected than in the other two cities. Stations of Metro Bike in LA (Fig. 6) are distributed over different towns/districts with simpler traffic network structures, and thus better prediction accuracies can be observed.

We also show in Table 3 the prediction results of GBikes, MGN, GC and STCNN *w.r.t.* the morning and evening rush hours, *i.e.*, 7:00 – 10:00 AM and 5:00 – 8:00 PM (local time). Due to larger bike volumes during rush hours, all schemes are shown to have higher errors than in Table 2. Nonetheless, thanks to its attention mechanisms, multi-level temporal closeness and correlation studies, GBikes achieves better prediction accuracy than other schemes.

We can also infer the reliability of GBikes upon different station usages from the data during rush hours in the analysis. Since the bike-flow patterns (including correlations of station usages) during rush hours are likely to be different from the overall patterns shown in Table 2, the overall robust performance of GBikes shown in Table 3 implies the robustness/applicability to scenarios with different flow patterns.

Sensitivity analysis: Table 4 shows the performance variations of GBikes by removing some architecture components, *i.e.*, the RMSE variations of different settings. Removal of any component causes notable accuracy degradation, demonstrating the importance of each part.

We further show in Fig. 14 the RMSEs vs. numbers of graph attention heads (M in Eqs. (17) & (18)) for the three bike-sharing systems. For each bike-sharing system in Fig. 14, we increase M from 5 to 12 and evaluate the influence. We can see that too few or too many attention heads may not achieve high accuracies. Too few heads make it difficult for GBikes to capture complex station-to-station correlations, while too many of them introduce noise in the model learning. As the station network of Metro tends to have smaller distance between stations with high correlation (Fig. 5), fewer attention heads are needed than Citi and Divvy. More interestingly, due to a much sparser bike station network (Fig. 6), Divvy does not require notably more attention heads than Citi and Metro.

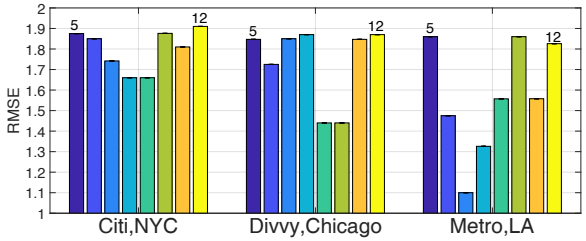

Figure 14: RMSEs vs. the number of attention heads (5 to 12).

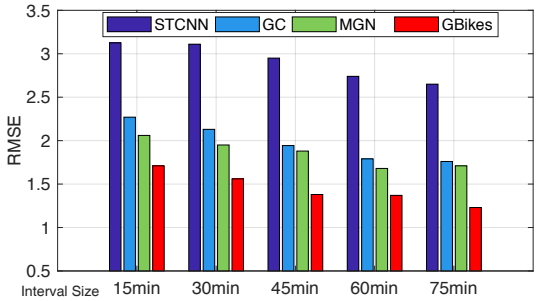

Figure 15: RMSEs vs. the sizes of each time interval (Citi, NYC).

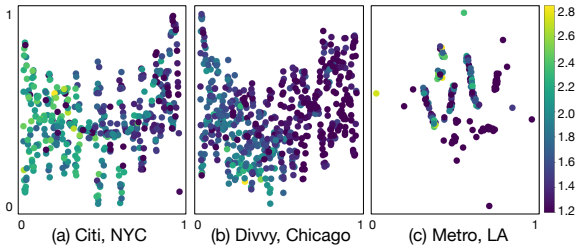

Figure 16: Visualization of the layer activation after GACNN (concat). The color denotes the logarithm of station usage.

We also show in Fig. 15 the RMSEs vs. the size of each time interval. We compare GBikes with GC, MGN and STCNN based on the validation/sensitivity dataset of Citi, NYC to illustrate the effect of the time interval. We vary the size of time interval from 15 min to 75 min. As we can see from the results, the accuracy generally increases with larger time intervals, mainly because a smaller time interval may experience more complicated users' pick-up/drop-off behaviors than a larger one, making it more challenging for real-time flow prediction. Considering the importance of timeliness and proactiveness for other subsequent applications (say, station re-balancing or anomaly detection), we focus on the small time interval, *i.e.*, 15 min, in our experimental evaluation.

We visualize the layer activation values (high-dimensional data) after one GACNN (concat) for the three datasets in Fig. 16 through t-SNE (t-Distributed Stochastic Neighbor Embedding) [28]. Each dot represents a bike station, and we have 502, 607 and 135 dots, respectively, in (a), (b) and (c). Each dot's high-dimension activation

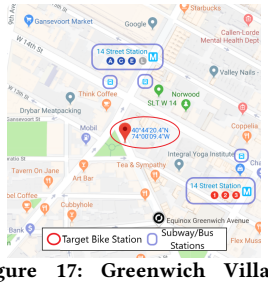

Figure 17: Greenwich Village (40.7390°N, -74.0026°W).

Figure 18: Union Square Park (40.7345°N, -73.9907°W).

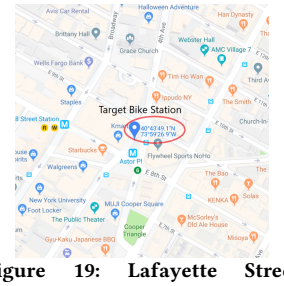

Figure 19: Lafayette Street (40.7303°N, -73.9908°W).

Figure 20: Battery Park City (40.7153 °N, -74.0166°W).

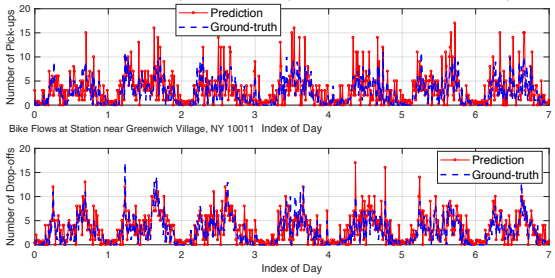

Figure 21: Bike flows and predictions for Greenwich (Fig. 17).

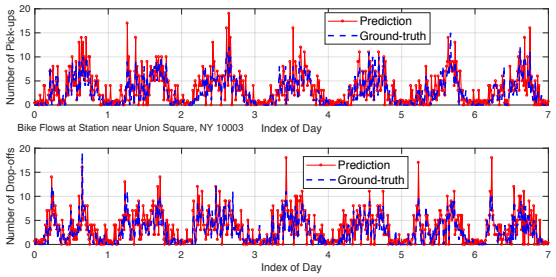

Figure 22: Bike flows and predictions for Union Square (Fig. 18).

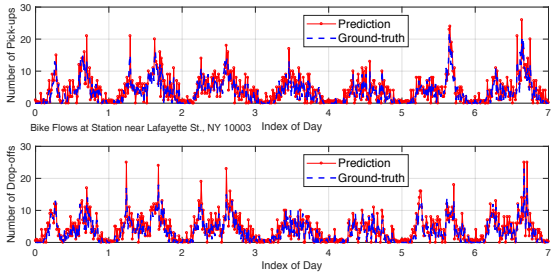

Figure 23: Bike flows and predictions for Lafayette (Fig. 19).

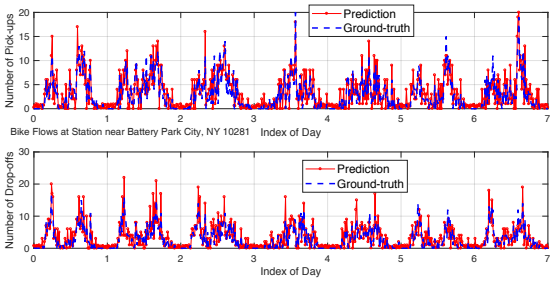

Figure 24: Bike flows and predictions for Battery Park (Fig. 20).

value is mapped to 2-D space (with normalized 2-D coordinates), and warmer color indicates more station bike usage (logarithm of total in/out flows). We set perplexity of t-SNE as 50, and other parameters by their default in `sklearn` [28].

We can see that after one GACNN layer `GBikes` has started to discriminate stations with and without (lighter and darker colors) heavy bike usage, correlating each group closely and enabling high flow prediction accuracy in GACNN (`average`). Since Metro's stations in LA (Fig. 16(c)) are distributed in multiple cities/towns under LA county, more distinct clusters can be observed there than NYC Citi and Chicago Divvy.

Case studies: We further conduct the case studies on the dataset of Citi, NYC, and show in Figs. 17–24 the neighborhood maps and the ground-truth bike flows (in/out) of four most popular stations (in terms of total pick-ups/drop-offs) and the predictions by `GBikes`. The study is based on one-week trip records (Sunday to Saturday). Note that each data point corresponds to the aggregated bike pick-ups or drop-offs of that station in default 15 min interval. With the resembled trends and accurate predictions, we show in general the `GBikes` model captures the dynamics of the bike flows.

In particular, for the station near Greenwich Village within the lower Manhattan (Figs. 17 & 21), since there are multiple subway/bus stations nearby, we can notice multiple remarkable peaks which are close to each other in time domain. Regarding the station near the Union Square Park (Figs. 18 & 22), the crowds near the points of interests introduce more fluctuations and small peaks/valleys in the bike flows. From the station near the Lafayette Street of the lower Manhattan (Figs. 19 & 23), we can observe more bike usage during the afternoon time due to neighboring commercial, business and tourism activities. At the station near the ferry terminal around the Battery Park City (Figs. 20 & 24), the intermittent peak patterns in the bike flows are likely due to the frequency of ferries nearby. Thanks to its comprehensive data-driven studies and spatio-temporal designs, fine-grained accuracy can be achieved and the predictions can effectively capture the neighborhood patterns.

7 CONCLUSION

We have conducted extensive data-driven and experimental studies of flow prediction for bike-sharing stations using graph attention convolutional neural networks. State-of-the-arts often focus on prediction for a group of stations, without comprehensive data-driven designs for bike stations and their correlations. In contrast, we formalize the network of stations into a graph. We provide comprehensive spatio-temporal designs, taking into account spatial correlation and temporal closeness of stations and their bike flows. Graph attention mechanisms are also designed to better capture the inherent station-to-station correlations. Extensive experimental studies upon three metropolitan bike-sharing stations in NYC,

Chicago and LA have corroborated the effectiveness, robustness and accuracy of `GBikes` in fine-grained bike-flow prediction.

REFERENCES

[1] Jie Bao, Tianfu He, Sijie Ruan, Yanhua Li, and Yu Zheng. 2017. Planning Bike Lanes Based on Sharing-Bikes' Trajectories. In *Proc. ACM KDD*. 1377–1386.

[2] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine* 34, 4 (July 2017), 18–42.

[3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. In *Proc. ICLR*.

[4] Di Chai, Leye Wang, and Qiang Yang. 2018. Bike Flow Prediction with Multi-graph Convolutional Networks. In *Proc. ACM SIGSPATIAL*. 397–400.

[5] Longbiao Chen, Daqing Zhang, Leye Wang, Dingqi Yang, and *et al.* 2016. Dynamic Cluster-based Over-demand Prediction in Bike Sharing Systems. In *Proc. ACM UbiComp*. 841–852.

[6] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *Proc. WWW*. 1459–1468.

[7] Jon Froehlich, Joachim Neumann, and Nuria Oliver. 2009. Sensing and Predicting the Pulse of the City Through Shared Bicycling. In *Proc. IJCAI*. 1420–1426.

[8] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proc. NIPS*. 1024–1034.

[9] Suining He and Kang G. Shin. 2018. (Re)Configuring Bike Station Network via Crowdsourced Information Fusion and Joint Optimization. In *Proc. ACM MobiHoc*. 1–10.

[10] Suining He and Kang G. Shin. 2019. Spatio-Temporal Adaptive Pricing for Balancing Mobility-on-Demand Networks. *ACM Trans. Intell. Syst. Technol. (TIST)* 10, 4, Article Article 39 (July 2019), 28 pages.

[11] Suining He and Kang G. Shin. 2019. Spatio-Temporal Capsule-based Reinforcement Learning for Mobility-on-Demand Network Coordination. In *Proc. WWW*. 2806–2813.

[12] Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163* (2015).

[13] Joseph M Hilbe. 2011. *Negative Binomial Regression*. Cambridge University Press.

[14] H. Hu, G. Li, Z. Bao, Y. Cui, and J. Feng. 2016. Crowdsourcing-based real-time urban traffic speed estimation: From trends to speeds. In *Proc. IEEE ICDE*. 883–894.

[15] Jilin Hu, Chenjuan Guo, Bin Yang, and Christian S. Jensen. 2019. Stochastic Weight Completion for Road Networks using Graph Convolutional Networks. In *Proc. IEEE ICDE*. 1274–1285.

[16] Pierre Hulot, Daniel Aloise, and Sanjay Dominik Jena. 2018. Towards Station-Level Demand Prediction for Effective Rebalancing in Bike-Sharing Systems. In *Proc. ACM KDD*. 378–386.

[17] Shenggong Ji, Yu Zheng, Zhaoyuan Wang, and Tianrui Li. 2019. A Deep Reinforcement Learning-Enabled Dynamic Redeployment System for Mobile Ambulances. *Proc. ACM IMWUT* 3, 1, Article 15 (March 2019), 20 pages.

[18] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proc. ICLR*.

[19] Minne Li, Zhiwei Qin, Yan Jiao, Yaodong Yang, Jun Wang, Chenxi Wang, Guobin Wu, and Jieping Ye. 2019. Efficient Ridesharing Order Dispatching with Mean Field Multi-Agent Reinforcement Learning. In *Proc. WWW*. 983–994.

[20] Yexin Li, Yu Zheng, and Qiang Yang. 2018. Dynamic Bike Reposition: A Spatio-Temporal Reinforcement Learning Approach. In *Proc. ACM KDD*. 1724–1733.

[21] Yexin Li, Yu Zheng, Huichu Zhang, and Lei Chen. 2015. Traffic Prediction in a Bike-sharing System. In *Proc. ACM SIGSPATIAL*. 33:1–33:10.

[22] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. 2018. GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction.. In *Proc. IJCAI*. 3428–3434.

[23] Lei Lin, Zhengbing He, and Srinivas Peeta. 2018. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. *Transportation Research Part C: Emerging Technologies* 97 (2018), 258 – 276.

[24] J. Liu, Q. Li, M. Qu, W. Chen, J. Yang, H. Xiong, H. Zhong, and Y. Fu. 2015. Station Site Optimization in Bike Sharing Systems. In *Proc. IEEE ICDM*. 883–888.

[25] Junming Liu, Leilei Sun, Qiao Li, Jingci Ming, Yanchi Liu, and Hui Xiong. 2017. Functional Zone Based Hierarchical Demand Prediction For Bike System Expansion. In *Proc. ACM KDD*. 957–966.

[26] S. Ma, Y. Zheng, and O. Wolfson. 2013. T-share: A large-scale dynamic taxi ridesharing service. In *Proc. IEEE ICDE*. 410–421.

[27] Xiaolei Ma, Zhuang Dai, Zhengbing He, Jihui Ma, Yong Wang, and Yunpeng Wang. 2017. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* 17, 4 (2017), 818.

[28] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[29] NOAA. 2019. National Oceanic and Atmospheric Administration, website: https://www.noaa.gov/.

[30] Yan Pan, Ray Chen Zheng, Jiaxi Zhang, and Xin Yao. 2019. Predicting bike sharing demand using recurrent neural networks. *Procedia Computer Science* 147 (2019), 562 – 566.

[31] Technavio. 2018. Global Bike-sharing market 2018–2022, https://www.technavio.com/report/global-bike-sharing-market-analysis-share-2018?utm_source=t9&utm_medium=bw_wk52&utm_campaign=businesswire.

[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proc. ICLR*.

[33] D. Wang, W. Cao, J. Li, and J. Ye. 2017. DeepSD: Supply-Demand Prediction for Online Car-Hailing Services Using Deep Neural Networks. In *Proc. IEEE ICDE*. 243–254.

[34] Shuai Wang, Tian He, Desheng Zhang, Yuanchao Shu, Yunhuai Liu, Yu Gu, Cong Liu, Haengju Lee, and Sang H. Son. 2018. BRAVO: Improving the Rebalancing Operation in Bike Sharing with Rebalancing Range Prediction. *Proc. ACM IMWUT* 2, 1, Article 44 (March 2018), 22 pages.

[35] Z. Wang, Z. Qin, X. Tang, J. Ye, and H. Zhu. 2018. Deep Reinforcement Learning with Knowledge Transfer for Online Rides Order Dispatching. In *Proc. IEEE ICDM*. 617–626.

[36] G. Wu, Y. Li, J. Bao, Y. Zheng, J. Ye, and J. Luo. 2018. Human-Centric Urban Transit Evaluation and Planning. In *Proc. IEEE ICDM*. 547–556.

[37] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. 2019. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596* (2019).

[38] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proc. AAAI*. 7444 – 7452.

[39] C. Yang, C. Zhang, X. Chen, J. Ye, and J. Han. 2018. Did You Enjoy the Ride? Understanding Passenger Experience via Heterogeneous Network Embedding. In *Proc. IEEE ICDE*. 1392–1403.

[40] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. 2019. Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction. In *Proc. AAAI*.

[41] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proc. IJCAI*. 3634–3640.

[42] Chao Zhang, Keyang Zhang, Quan Yuan, Haoruo Peng, Yu Zheng, Tim Hanratty, Shaowen Wang, and Jiawei Han. 2017. Regions, Periods, Activities: Uncovering Urban Dynamics via Cross-Modal Representation Learning. In *Proc. WWW*. 361–370.

[43] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction.. In *Proc. AAAI*. 1655–1661.