

# Dynamic Flow Distribution Prediction for Urban Dockless E-Scooter Sharing Reconfiguration

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

Thanks to recent progresses in mobile payment, IoT, electric motors, batteries and location-based services, Dockless E-scooter Sharing (DES) has become a popular means of last-mile commute for a growing number of (smart) cities. As e-scooters are getting deployed dynamically and flexibly across city regions that expand and/or shrink, with subsequent social, commercial and environmental evaluation, accurate prediction of the distribution of e-scooters given reconfigured regions becomes essential for the city planners and service providers.

To meet this need, we propose GCscoot, a novel dynamic flow distribution prediction for reconfiguring urban DES systems. Based on the real-world datasets with reconfiguration, we analyze the mobility features of the e-scooter distribution and flow dynamics for the data-driven designs. To adapt to dynamic reconfiguration of DES deployment, we propose a novel spatio-temporal graph capsule neural network within GCscoot to predict the future dockless e-scooter flows given the reconfigured regions. GCscoot preprocesses the historical spatial e-scooter distributions into flow graph structures, where discretized city regions are considered as nodes and their mutual flows as edges. Given data-driven designs regarding distance, ride flows and region connectivity, the dynamic region-to-region correlations embedded within the temporal flow graphs are captured through the graph capsule neural network which accurately predicts the DES flows. We have conducted extensive empirical studies upon three different e-scooter datasets (>2.8 million rides in total) in populous US cities including Austin TX, Louisville KY and Minneapolis MN. The evaluation results have corroborated the accuracy and effectiveness of GCscoot in predicting dynamic distribution of dockless e-scooters' mobility.

## CCS CONCEPTS

• Information systems → Spatial-temporal systems;

## KEYWORDS

Dockless e-scooter; reconfiguration; distribution prediction

## ACM Reference Format:

Suining He and Kang G. Shin. 2020. Dynamic Flow Distribution Prediction for Urban Dockless E-Scooter Sharing Reconfiguration. In *Proceedings of*

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

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

© 2020 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-7023-3/20/04.

<https://doi.org/10.1145/3366423.3380101>

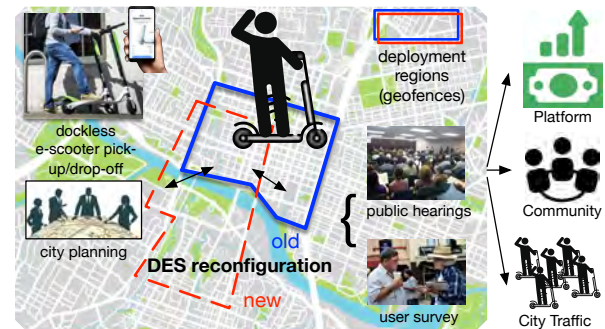


Figure 1: Illustration of dockless e-scooters & DES reconfiguration.

*The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3366423.3380101>

## 1 INTRODUCTION

Powered by the rapid growth of on-demand and sharing economy, dockless electric-scooter sharing (DES) systems have been proliferating in many metropolitan areas worldwide. As illustrated in Fig. 1, built upon mobile payment, Internet-of-Things (IoT) and location-based services, DES does not, in general, require fixed docking stations for users to receive or return the e-scooters. With motorized and dockless features, DES provides another faster and easier first/last-mile connectivity of the city [4, 32] beyond conventional bike sharing.

Due to the increasing commercial potential (DES platforms like Bird and Lime have raised USD\$1.48 billion by April 2019 [4]) and growing social acceptance, many city planners as well as service providers are considering expanding their deployment coverage. For example, the expansion program in Washington D.C. is expected to increase 50% of the e-scooter deployment in 2019 [1]. On the other hand, e-scooter geofences may shrink in some regions of the city given new administrative decisions. Such expansion and shrinkage, or *reconfiguration* as shown in Fig. 1, is done region-by-region followed by official evaluations (say, public hearings and user survey). Therefore, how to accurately forecast the e-scooter distributions in to-be-reconfigured regions is essential for the predictive and precautionary decisions of city planners and DES service providers.

Such a proactive and accurate forecast also provides the initial clues of selecting regions to enhance expected platform revenues and mitigate potential alternation of local traffic environments. Public expenditure due to labor-intensive site surveys can be reduced/eliminated. Furthermore, accurate knowledge of e-scooter proliferation helps balance the demands and supplies of the e-scooters, and prevent under-served customers and over-congested side walks due to excessive parking, which has become substantial

impediments to commercial profitability and social welfare of many DES systems and their communities.

Despite the progresses made in existing DES deployment, there still remain several technical challenges and concerns before a satisfactory reconfiguration decision can be made. Due to sparsity and scarcity, or even absence, of initial trials in newly-expanded regions, future e-scooter distributions may not be simply modeled by existing time-series prediction based on historical records. Furthermore, due to its dockless, easy-maneuvering and last-mile nature, e-scooters traveling across city regions form complex mutual dependencies, commute connectivities and mobility correlations, making it even more difficult to predict post reconfiguration effects. Introducing or removing certain deployment regions may lead to sophisticated effects upon the mobility patterns of neighborhoods. Finally, few studies have conducted data analytics on the spatial and temporal distributions of existing metropolitan DES systems. Designs of models/components largely rely upon the deployment insights, which are essential for a data-driven model study.

To address the above concerns, we propose GCScoot, a novel dynamic e-scooter flow prediction scheme based on spatio-temporal graph capsule neural network for reconfiguring the urban e-scooter sharing systems. Specifically, based on the extensive data analytics upon DES datasets, we identify several design features for the urban DES systems. Using these results, we design a novel spatio-temporal graph capsule convolutional neural network, called STGCapNet, for dynamic flow distribution of urban DES reconfiguration. Taking into account the reconfigured city regions and their mutual correlations, GCScoot comprehensively captures the dynamic impacts of reconfigured city regions upon the e-scooter distributions, and the graph capsule convolution accurately predicts the DES flows.

The contributions of GCScoot are summarized as follows.

- *Data-driven analytics and designs for dockless e-scooter reconfiguration:* In order to adapt to expansion dynamics, we have conducted extensive data-driven studies using the real-world datasets from the DES systems. We have studied the mobility features of DES systems given reconfigured deployment, and provided comprehensive data-driven designs for the following flow prediction. To the best of our knowledge, this is the *first* that investigates, identifies and formulates the dockless e-scooter sharing flow distribution prediction problem given the reconfigured deployment.
- *Spatio-temporal graph capsule neural network for reconfigured DES distribution forecasting:* We propose a data-driven design for DES distribution prediction based on novel spatio-temporal graph capsule neural network called STGCapNet. The proposed multi-scale feature extraction based on multi-layer graph convolution comprehensively retrieves the correlations among the reconfigured regions. By mining the connectivities between the existing, new and removed regions, STGCapNet learns the reconfigured DES mobility. Further routings between the graph convolutions and capsules capture the geographical properties via vectorized representations [31], thus leading to high accuracy in dynamic flow prediction.
- *Extensive data-driven and experimental studies:* Based on the above analytics and network formulation, we have conducted extensive experimental studies upon 2.8 million rides from three different e-scooter datasets in populous US cities including Austin

TX, Louisville KY and Minneapolis MN. Our results have corroborated the accuracy and effectiveness of GCScoot in dynamic distribution prediction of dockless e-scooter mobility.

The rest of the paper is organized as follows. We first discuss the related work in Sec. 2, followed by an overview of the concepts, problem formulation and data sets in Sec. 3. Then, we present in Sec. 4 the data-driven analytics and designs for DES reconfiguration, followed by the core dynamic prediction framework in Sec. 5. We then experimentally evaluate the performance of GCScoot in Sec. 6, and finally conclude the paper in Sec. 7.

## 2 RELATED WORK

We briefly discuss the related work as follows.

*Smart transportation:* Recent advances of big data and deep learning have redefined many research problems towards smarter transportation and the resultant sharing economy [38]. Unlike prior smart and shared mobility studies [11, 36], our work focuses on rapidly proliferating dockless e-scooter systems, deriving important insights and data-driven designs for practical reconfiguration deployment of e-scooters. Despite the prototype studies upon the e-scooter datasets, GCScoot can be adapted and extended to other on-demand modalities with reconfigured deployment, including e-vehicle station relocation [8], car-sharing [17, 21], and bike-sharing [15, 18, 19].

*Traffic flow prediction:* Conventional statistical analysis and machine learning tools have been used for traffic flow forecast. By modeling the traffic speed distributions as images, Ma *et al.* [26] investigated the application of convolutional neural network. Zhang *et al.* [42] proposed the spatio-temporal residual neural network for bike mobility prediction. With the road network graphs, a deep learning design for traffic series prediction was discussed by Yu *et al.* [40]. For station-based bike sharing, Chai *et al.* [9] designed a multi-graph convolutional neural network, which is followed by Geng *et al.* [14] upon ride sharing services. Yao *et al.* [39] and Wang *et al.* [35] respectively proposed a new meta-learning approach and a region transfer method to transfer knowledge from multiple cities. With meta-learning, Pan *et al.* [30] further studied the diversity of spatial and temporal correlations.

Unlike these studies, GCScoot takes into account the dynamic reconfiguration of scooter deployment regions, and provides highly adaptive and accurate flow predictions for e-scooter sharing. We propose novel data-driven designs for DES systems, and show that our spatio-temporal graph capsule neural network adapts to the dynamic reconfiguration. DES service providers can leverage the adaptability and accuracy of GCScoot predictions for more proactive reconfiguration decisions and evaluations.

*Dockless vehicle mobility analytics:* Thanks to location sensing and IoTs, many single-track vehicle sharing systems like bikes and e-scooters have rendered parking docks obsolete. Pan *et al.* [28] studied a reinforcement learning algorithm in order to balance the dockless bike sharing system. Liu *et al.* [23] leveraged the factor analysis and convolutional neural network to transfer the knowledge between two cities, which is further followed by [24] regarding domain adaptation. Smith *et al.* [32] studied the mobility benefits of e-scooter sharing in Chicago.

GCScoot differs from these in the following perspectives. We focus on the dynamic flow analytics for the emerging dockless

e-scooter sharing due to its higher mobility and more urban impact beyond the bike sharing. To address the pressing reconfiguration concerns of the city planners, we have conducted the pioneering studies and identified the reconfiguration problems for DES deployment. Furthermore, through experimental studies upon the three urban DES datasets, we have shown that the proposed network within GCScoot outperforms the existing schemes in adapting to the dynamic city regions through the novel joint graph convolution and capsule learning.

### 3 CONCEPTS, PROBLEM FORMULATION & DATA SETS

We first briefly introduce the important concepts related to GCScoot's formulation in Sec. 3.1, and then present the problem formulation in Sec. 3.2. Finally, we describe the datasets for our data analytics in Sec. 3.3.

#### 3.1 Important Concepts

Presented below are the definitions of important concepts.

**Definition 1 (City Regions & Time Intervals).** Following the previous works [39, 42], the entire map of a city is discretized into a set of total  $N$  regions (say, rectangle grid in our case), yielding a finite geographical set for computational convenience. Each region is represented by the coordinate of its center, i.e.,  $\mathbf{r}_n = [lat_n, lon_n]$  ( $n \in \{1, \dots, N\}$ ). Similar to the discretization of city regions, the time domain is discretized into equal intervals (of 30 min in our prototype studies), each of which is labeled with  $k$ .

**Definition 2 (DES Flow).** Each trip represents a user's scooter ride at a certain time from a city region to another. Specifically, a set of trips starting from region  $i$  to region  $j$  can be represented as  $\tau(i, j) = \{i, j, (t_i, t_j)'s\}$ , where  $(t_i, t_j)$  is the pick-up/drop-off timestamps of each trip in  $\tau(i, j)$ . We further denote the DES flow at region  $i$  in an interval  $k$  as  $\mathbf{F}_i = (P_i, D_i)$ , where  $P_i$  and  $D_i$  are the respective DES pick-ups and drop-offs there.

To better characterize the correlations, dependencies and connectivities of different regions in a city, we model the urban DES systems into a network graph structure. Based on the above regions and flows, we introduce the DES network graph.

**Definition 3 (DES Network Graph).** Given  $T(i, j)$  trips between regions  $i$  and  $j$  ( $T(i, j) > 0$ ,  $T(i, j) = T(j, i)$ , and  $i, j \in \{1, \dots, N\}$ ), we form the link or network connectivity of the two regions. Considering  $N$  regions as vertices  $\mathbf{V}$  and their mutual connectivities (mutual flows) as edges  $\mathbf{E} = \{T(i, j)'s\}$ ,  $\forall i, j \in \{1, \dots, N\}$ , we form the DES network graph as  $\mathbb{G} = (\mathbf{V}, \mathbf{E})$ .

In practice, the DES network undergoes dynamic reconfiguration due to evolving user demands, city urbanization and traffic alternation. Thus we have:

**Definition 4 (DES Network Reconfiguration).** Given the periodic alternation (expansion or shrinkage) of DES deployment, we have two stages before and after each reconfiguration, i.e., two sets of  $N$  and  $N'$  pick-up/drop-off regions denoted as  $\mathbf{V}$  and  $\mathbf{V}'$ , respectively. The reconfigured regions are, therefore, given by  $(\mathbf{V} \cup \mathbf{V}') \setminus (\mathbf{V} \cap \mathbf{V}')$ .

The dynamic mobility of the DES users may lead to variations in either  $\mathbf{V}$  or  $\mathbf{E}$ . To characterize the dynamically evolving DES networks, we have

**Definition 5 (Spatio-Temporal DES Network Graph).** At time interval  $k$ , given the deployed regions  $\mathbf{V}^{(k)}$  with pick-ups/drop-offs and

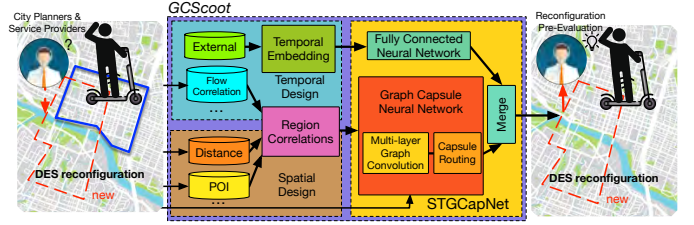


Figure 2: Illustration of the system framework in GCScoot.

the mutual flows of the regions  $\mathbf{E}^{(k)}$ , we denote the spatio-temporal DES network graph at the time interval  $k$  as  $\mathbb{G}^{(k)} = (\mathbf{V}^{(k)}, \mathbf{E}^{(k)})$ . Similarly, we denote the DES flows at the region  $i$  at the interval  $k$  as  $\mathbf{F}_i^{(k)} = (P_i^{(k)}, D_i^{(k)})$ .

#### 3.2 Problem Formulation & System Overview

Based on the concepts introduced above, we formally present the problem formulation as follows.

**Definition 6 (Dynamic Flow Prediction for Reconfigured DES Systems).** Given the spatio-temporal DES network graphs in the past  $w$  time intervals, i.e.,  $\{\mathbb{G}^{(k-w+1)}, \mathbb{G}^{(k-w+2)}, \dots, \mathbb{G}^{(k)}\}$  of the scooter pick-ups and drop-offs, as well as the reconfigured regions  $\mathbf{V}^{(k+1)}$  at the target time interval  $k+1$ , we want to proactively predict the dynamic flows  $\mathbf{F}^{(k+1)} = \{\mathbf{F}_i^{(k+1)}\}$  ( $i \in \{1, \dots, N\}$ ) in the  $\mathbb{G}^{(k+1)}$ .

Based on the case studies and pilot programs of DES, the reconfigured regions  $\mathbf{V}^{(k+1)}$  can be the result of negotiation between the DES service providers and the city. The information can be collected through public hearings, site survey and market analysis [41].

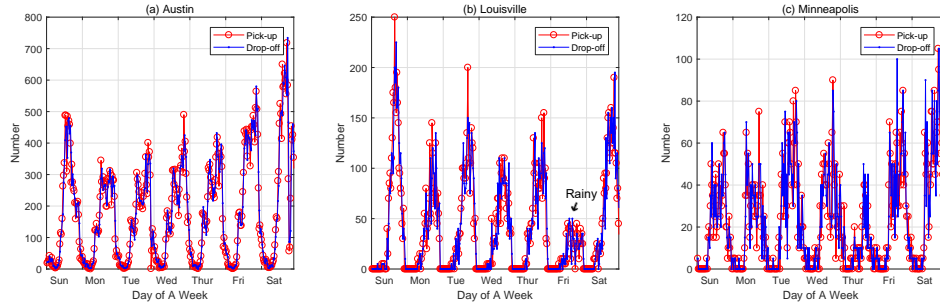
In order to solve the above problem, we propose GCScoot, whose system framework as well as information flow are illustrated in Fig. 2. Given the deployment trips of the DES system (including regions before and after the reconfiguration), analytics are conducted to pre-process the data, deriving the spatial and temporal correlations characterizing the DES deployment.

Specifically, spatial designs like road networks, regional points-of-interest (POIs) and region-to-region distances are collected. Temporal designs including historical flows and other external factors (like time and weather) are also retrieved from the DES deployment. We embed the external factors for the fully connected neural network to learn the temporal dynamics, while region correlations including flow, distance and POIs are jointly considered in the graph capsule neural network. Multi-layer graph convolution and capsule routing are applied to capture dynamic flow patterns. Predicted e-scooter flows for the DES reconfiguration via both networks are merged and returned. Combining the above factors, STGCapNet learns the correlations between reconfigured regions and provides accurate hints for the reconfiguration pre-evaluation.

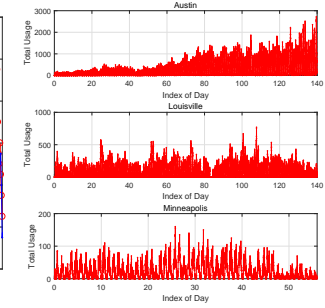
#### 3.3 Data Sets for Analytics & Evaluation

We have conducted our data analytics and experimental evaluation based on the following three datasets:

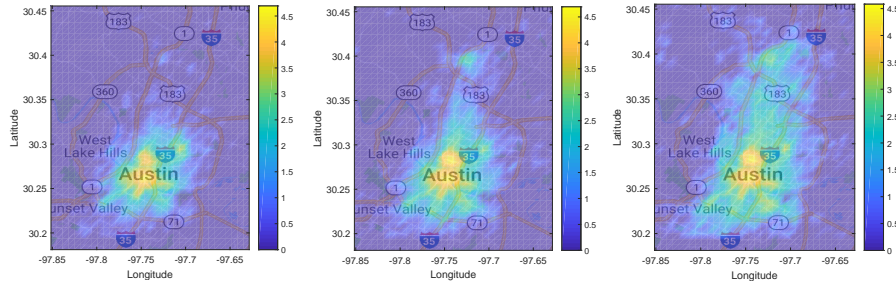
- **Austin, TX** (May, 2018 – January, 2019): In total, 2,430,806 DES trips have been recorded, with the pick-up/drop-off coordinates and timestamps, covering the bounding box of  $[-97.9^\circ\text{W}$ ,



**Figure 3: Dynamic flows of pick-ups/drop-offs of a week (Sunday to Saturday) in the three datasets: (a) Austin (Aug., 2018); (b) Louisville (Oct., 2018); (c) Minneapolis (Oct., 2018).**



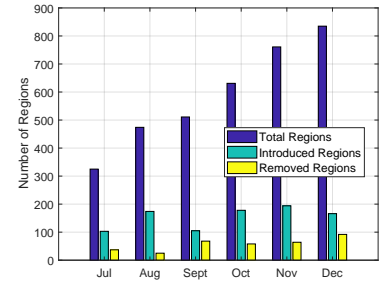
**Figure 4: Usage dynamics at all regions in the three datasets.**



**Figure 5: Pick-up distributions in August, 2018 (Austin).**

**Figure 6: Pick-up distributions in October, 2018 (Austin).**

**Figure 7: Pick-up distributions in December, 2018 (Austin).**



**Figure 8: Statistics of the DES deployment regions.**

-97.58°W, 30.2°N, 30.499°N]. Outliers have been removed when trip distance falls out of [0.1, 500] miles or a trip lasts for more than 24 hours.

- **Louisville, KY** (August, 2018 – May, 2019): The dataset contains in total 193, 937 trips with pick-up/drop-off coordinates and timestamps, covering a geographic bounding box of [-85.903°W, -85.486°W, 38.081°N, 38.340°N]. Outliers have been removed when trips were less than 0 miles or greater than 25 miles.
- **Minneapolis, MN** (July, 2018 – November, 2018): This dataset provides totally 225, 543 trips with pick-up/drop-off coordinates as well as timestamps, covering a bounding box of [-93.38°W, -93.08°W, 44.89°N, 45.02°N]. Outliers have been removed when trips last for over 7 hours and were less than 0 miles or exceeded 24 miles.

We further show the dynamic flows of the DES pick-ups and drop-offs of a week in each city in Fig. 3. We can observe more DES flows (pick-ups/drop-offs) in Austin than the other two cities, as well as high and dynamic volume of daily rides during weekends. Besides the above datasets, we also retrieve the city map (street centerlines) and obtain POI data from the OpenStreet Map [27].

## 4 DES RECONFIGURATION ANALYTICS

Given the above datasets from the DES systems, we conduct the reconfiguration analytics upon the DES networks. We first overview the DES reconfiguration in Sec. 4.1, and then present the data-driven studies of spatial and temporal factors in Sec. 4.2.

### 4.1 Overview of Urban DES Reconfiguration

We have conducted the following deployment studies and data analytics on the DES reconfiguration.

**Urban DES deployment and reconfiguration:** E-scooters and their sharing economy have recently been shown to be more competitive with the dominant car-commute lifestyle in the United States than bicycles. Besides high mobility and dockless features, the sizes of most sharing e-scooters deployed are physically smaller than conventional shared bicycles, hence easier to maneuver and less space to park. Let us consider the system records of both dockless e-scooter and bike sharing (same period) in Austin, TX as examples. Due to faster speed (up to 15 mph) and easier maneuvering, the DES systems enjoy an 83.87% shorter trip time duration and 33.14% wider spectrum of travel distance than the bike sharing (Austin B-Cycle) [3]. Such high mobility features as well as more random pick-up/drop-off behaviors also make the prediction more difficult than conventional dock-based or dockless bike sharing forecast schemes [10, 23].

On the other hand, such features also lead to an urban controversy regarding safety, sidewalk/lane sharing and parking. In many pilot studies (like in San Francisco), approaches that demonstrated a highest level of commitment have been discussed before deployment in order to address known challenges and concerns, ranging from public safety and user education to equitable access and collaboration with the city and its diverse communities. Based on feedbacks from city supervisors and citizens at public hearings, many DES service providers expand or shrink their service areas.

Throughout the pilot program and subsequent deployment, while DES vehicles are usually provided in pre-determined regions, additional service regions may be introduced to cater for the expanded demands. For example, during dynamic reconfiguration, the service providers in Austin may likely increase DES vehicles at the city regions other than the initially licensed ones [2]. Therefore, we have observed additional deployment regions within Austin,

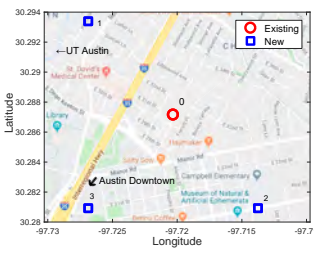


Figure 9: Centers of existing & new deployment regions (south of Cherrywood, Austin).

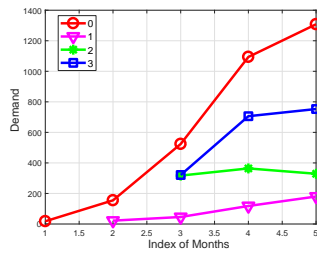


Figure 10: Monthly demand in- & new deployment regions (near Cherrywood, Austin).

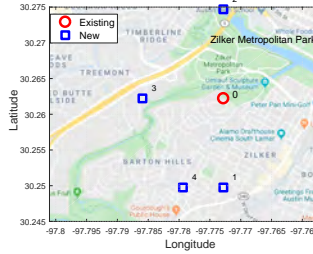


Figure 11: Centers of existing & new deployment regions (near Zilker Metropolitan Park, Austin).

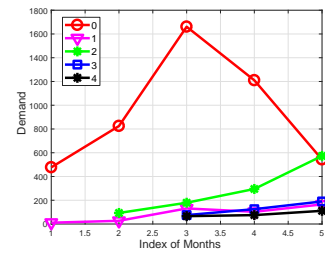


Figure 12: Monthly demand decreases with neighborhood expansion (ids: locations in Fig. 11).

leading to an expansion as shown in Fig. 4 (parts of the datasets are illustrated). Similar deployment expansions have been observed in Louisville and Minneapolis.

On the other hand, the DES deployment may be restricted to certain areas of a city. One can thus observe deployment shrinkage as shown in Fig. 4 (particularly for Louisville and Minneapolis). For example, according to the Dockless Mobility Program by Austin Transportation Department, dockless e-scooters are not allowed within areas like parks, off-street parking lots/garages. The Department may dynamically reduce the deployment of DES within a specific area, resulting in DES mobility alternation. The aforementioned dynamics in Fig. 4 given deployment reconfiguration, along with mobility routines in Fig. 3, make it very challenging to predict the entire reconfigured flow accurately.

**Reconfigured deployment regions:** Taking Austin as an example, we show in Figs. 5, 6 and 7 the geographic pick-up locations of deployment regions in August, October and December of 2018, respectively. Specifically, we plot the heatmap of pick-ups (in  $\log_{10}(\cdot)$ ) w.r.t. each of these months, where the warmer colors indicate more and denser e-scooter pick-ups. We further show the statistics of the reconfigured regions (each region takes  $0.2496\text{km}^2$ ) from July to December 2018 in Fig. 8. We can observe that the reconfiguration happens and the deployment regions are expanding or shrinking at different parts of the city. Clearly, it is very challenging to adaptively predict the e-scooter flows for dynamic reconfiguration, especially for those regions without prior deployment knowledge of the DES flows.

The reconfigured regions, either introduced or removed, may influence their neighbor regions significantly. We also show in Figs. 9–12 the mutual influence of the region usage. Specifically, we show in Figs. 9 and 10 the regions (center locations) to the east of University of Texas, Austin. The demands at region 0 which existed before the reconfiguration benefit from the introduction of other regions, since the expansions attract more neighborhood users (including the university students). On the other hand, in Figs. 11 and 12 we have illustrated a negative effect of expansions on the demands near the metropolitan park. With more options of pick-up/drop-off locations, we can observe a decrease in demand at region 0.

We can observe that the introduction of a region during the reconfiguration may decrease or increase the usage of its neighborhood, which may make it very challenging to predict DES flows accurately. To address this difficulty, we need to construct comprehensive network structures regarding the region-to-region correlations, which

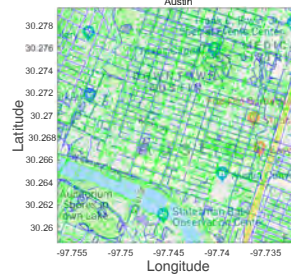


Figure 13: Road network, Austin, TX (downtown).

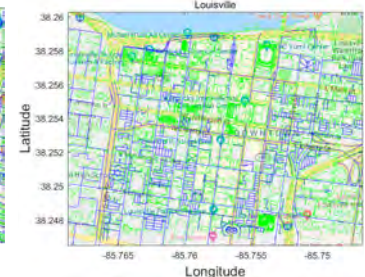


Figure 14: Road network, Louisville, KY (downtown).

can effectively and efficiently capture the relationship between the reconfigured regions.

## 4.2 Analytics on Spatial & Temporal Factors

We analyze the spatial factors related to the DES deployment as follows.

**Distance:** We correlate the regions based on their mutual closeness. Due to diverse terrains and buildings in urban and metropolitan areas, the travel distances between the two regions can be greater than the geographic ones measured along the surface of the earth. We show parts of the urban road networks (blue lines: road segments; green polygons: buildings) in downtown areas of Austin, Louisville and Minneapolis in Figs. 13, 14 and 15, respectively.

To reflect the terrain characteristics, we define the shortest path distance (unit: km) between regions  $i$  and  $j$  as  $sp(r_i, r_j)$ . Based on the street centerline obtained from the open data portal of the local governments [5–7], we can obtain the distance between two city regions based on the shortest paths on the map. We aggregate the lengths of the road segments along the shortest paths between the centers of the two regions, and form the corresponding distance between them.

We first show the correlations between regions versus the distances. Taking Austin and Louisville as two examples, we show in Fig. 19 the usage correlations of regions vs. their mutual shortest path distances. While more distant regions generally have lower correlations, we can still observe some high correlated regions due to similar commute or entertainment purposes there.

**POI factors:** The functionality of city regions affects the DES reconfiguration. To reflect this, we have collected the points-of-interest (POIs) information from the Open Street Map (OSM). Each POI is associated with specific attributes: name, addresses, GPS coordinate and the corresponding category. We have collected the

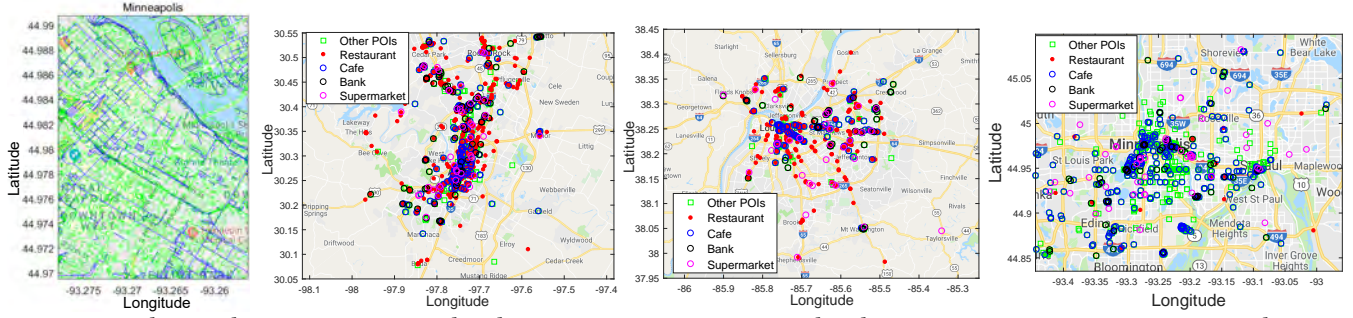


Figure 15: Road network, Figure 16: POI distributions in Minneapolis, MN. Figure 17: POI distributions in Austin, TX. Figure 18: POIs in Minneapolis, MN, Louisville, KY.

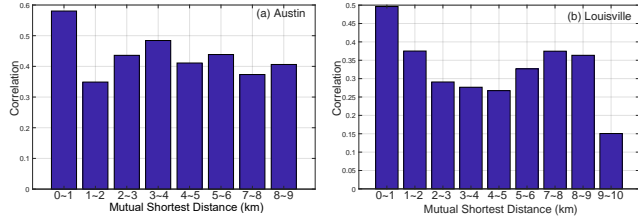


Figure 19: Usage correlations vs. distances: (a) Austin; (b) Louisville.

following POI categories: bank, bar, bike parking, cafe, car rental, cinema, clinic, fast food, hospital, kindergarten, library, park, pharmacy, post office, pub, restaurant, school and supermarket.

We take into account the following categories retrieved from OSM. In total, we have collected 2,072, 1,014 and 2,204 POIs for Austin, Louisville and Minneapolis, which are, respectively, as illustrated in Figs. 16, 17 and 18. We can observe the variations of POI types and location distributions across the three cities. We also show in Figs. 20, 21 and 22 the POI similarity matrices for all the regions *w.r.t.* each of the three cities.

**Temporal & external factors:** Recall that Fig. 3 shows the temporal dynamics of DES pick-ups and drops-offs over the time (one week). Clearly, the DES flows are shown to experience burst during morning and late afternoon rush hours, mainly because many DES users ride the dockless e-scooters for commute. On the other hand, we further consider the effect of weather conditions upon the DES deployment. E-scooter usage can be influenced significantly by the weather. For example, the sudden drops in both pick-ups/drop-offs on Friday in Fig. 3(b) can be the result of consecutive rainy conditions illustrated in Fig. 23. In fact, the service providers and city planners may suspend services during tough weather (say, the dockless mobility service may be suspended temporarily or throughout the remainder of the program during unsafe winter riding conditions in Minneapolis). Therefore, we further take into account the time (including days of week and public holidays) as well as meteorological factors as external factors within GCScoot formulation.

## 5 GCSCOOT: DYNAMIC LEARNING & FLOW PREDICTION

Based on the above data analytics, we propose the design of GCScoot in order to accommodate the complexity in DES reconfiguration. Specifically, we first present the spatial and temporal designs in Sec. 5.1. We then provide the graph capsule designs in Sec. 5.2, followed by the architecture of STGCapNet in Sec. 5.3.

### 5.1 Spatial and Temporal Designs

We present the spatial and temporal designs in GCScoot as follows.

**Distance:** Since the regions closer in the geographic space are more likely to be correlated, we form the spatial correlations which account for mutual distances. Specifically, we have the *spatial distance correlations* between regions  $i$  and  $j$  in terms of shortest path distance as

$$A_D(i, j) \triangleq \frac{1}{1 + \text{sp}(\mathbf{r}_i, \mathbf{r}_j)}. \quad (1)$$

**POI:** To reflect the correlations due to city functionality, we measure the *PoI closeness* between the feature vectors  $\mathbf{P}_i$  and  $\mathbf{P}_j$  of the two regions based on the cosine similarity, which is formally given by

$$A_P(i, j) \triangleq \cos(\mathbf{P}_i, \mathbf{P}_j) = \frac{\mathbf{P}_i \cdot \mathbf{P}_j}{\|\mathbf{P}_i\| \cdot \|\mathbf{P}_j\|}. \quad (2)$$

Each dimension of  $\mathbf{P}_i$  corresponds to the number of POIs of a category within region  $i$ .

**Temporal correlation:** Some pairs of regions may have correlated DES flows due to the users' commute between them. Let  $T_i^{(t)}$  be the set of DES rides starting from region  $i$  at an interval  $t$ , *i.e.*,

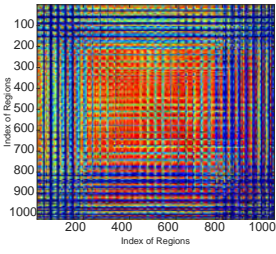
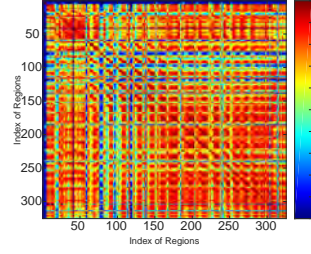
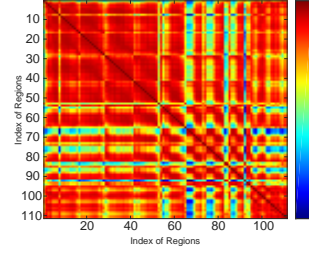
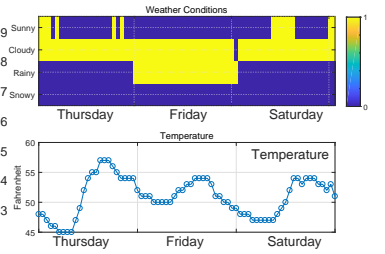
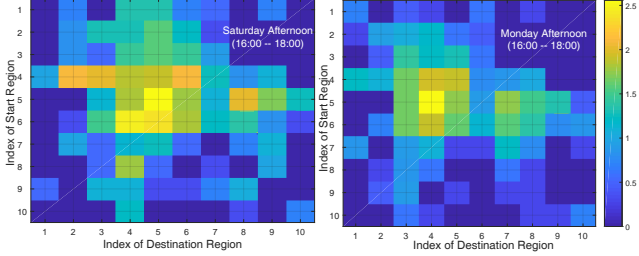
$$|T_i^{(t)}| = \sum_{j=1}^N |\tau(i, j)^{(t)}|. \quad (3)$$

To measure the flow correlations, we further define the *temporal correlations* of DES flows between regions  $i$  and  $j$  in the most recent  $w$  time intervals, which is formally given by

$$A_C^*(i, j) \triangleq \frac{\sum_{t=k-w}^{t=k} |T_i^{(t)}| \cdot |T_j^{(t)}|}{\sqrt{\sum_{t=k-w}^{t=k} |T_i^{(t)}|^2} \cdot \sqrt{\sum_{t=k-w}^{t=k} |T_j^{(t)}|^2}}. \quad (4)$$

**Connectivity:** We have observed that the DES users usually travel frequently among regions due to their commute routes and preferences. During rush hours we observe more frequent travels between work and residential areas, while recreational and residential areas are more likely connected during weekends. We show in Fig. 24 the volumes of flows (in  $\log_{10}(\cdot)$ ) from the start regions (vertical axis) to the destinations (10 selected regions in Austin), and we can observe more diverse DES flows during weekend due to the broader riding purposes.

Consideration of Eq. (4) only cannot comprehensively reflect the directional dependency between regions. So, we further integrate


**Figure 20: Region-to-region POI similarities, Austin.**

**Figure 21: Region-to-region POI similarities, Louisville.**

**Figure 22: Region-to-region POI similarities, Minneapolis.**

**Figure 23: Weekend weather conditions for Fig. 3(b) in Louisville.**

**Figure 24: Connectivities of regions on Saturday & Monday (Austin).**

the connectivity among regions within our formulation. Specifically, we first define the proportion of e-scooter rides  $\tau^{(k)}(i, j)$  from regions  $r_i$  to  $r_j$  in the time interval  $k$  as

$$a^{(k)}(i, j) \triangleq \frac{\tau^{(k)}(i, j)}{\sum_{l=1, l \neq i}^N \tau^{(k)}(i, l)}. \quad (5)$$

Then, we design a vector representing the relative flow proportion, *i.e.*,

$$\tilde{\mathbf{u}}(i, j) \triangleq \left[ a^{(k)}(i, j), 1 - a^{(k)}(i, j) \right], \quad (6)$$

where  $a^{(k)}(i, j)$  increases (and  $1 - a^{(k)}(i, j)$  decreases) if more rides in  $i$  head to  $j$ .

Considering the DES network graph  $\mathbb{G}$ , we adapt the designs of first-order proximity in the network embedding [33], and design a *connectivity metric*  $h(i, j)$  for DES rides between regions  $i$  and  $j$  as

$$h(i, j) \triangleq \frac{1}{1 + \exp(-\tilde{\mathbf{u}}(i, j) \cdot \tilde{\mathbf{u}}(j, i))}, \quad (7)$$

where the dot product of the two vectors,  $\tilde{\mathbf{u}}(i, j) \cdot \tilde{\mathbf{u}}(j, i)$ , increases if two regions have more DES rides heading towards each other.

As  $h(i, j) \in (0, 1)$ , we adjust the  $\mathbf{A}_C^*(i, j)$  by

$$\mathbf{A}_C(i, j) = \mathbf{A}_C^*(i, j) \cdot h(i, j). \quad (8)$$

In other words, regions  $i$  and  $j$  are considered more correlated in flows if they have more similar flow dynamics and stronger mutual flow connectivities with each other.

Finally, we have the following adjacency matrix characterizing the structures of spatio-temporal DES network graph,

$$\mathbf{A} = [\mathbf{A}_D, \mathbf{A}_P, \mathbf{A}_C]. \quad (9)$$

**Reconfiguration masking:** Due to dynamic reconfiguration, the DES deployment regions can be activated (introduced) or deactivated (removed) over the time domain, forming the network graph  $\mathbb{G}^{(k)}$ . To adaptively reflect this in GCScoot's formulation, for each interval  $k$ , we apply a mask operation  $m^{(k)}(\cdot)$  upon each matrix  $\mathbf{A}_\star$  ( $\star \in \{D, P, C\}$ ), where  $\mathbf{A}'_\star(i, :) = \mathbf{0}$  ( $\mathbf{A}'_\star(:, i) = \mathbf{0}$ ) if the region  $i$  is

deactivated,  $\mathbf{A}'_\star(i, :) = \mathbf{A}_\star(i, :) \cdot \mathbf{1}$  ( $\mathbf{A}'_\star(:, i) = \mathbf{A}_\star(:, i) \cdot \mathbf{1}$ ) otherwise. From the reconfiguration plan of the government or city planners, we can also obtain the reconfigured regions as  $\mathbf{V}^{(k+1)}$ . Therefore, we have the resultant masked correlations as

$$\mathbf{A}'_D = m^{(k)}(\mathbf{A}_D), \quad \mathbf{A}'_P = m^{(k)}(\mathbf{A}_P), \quad \mathbf{A}'_C = m^{(k)}(\mathbf{A}_C). \quad (10)$$

## 5.2 Graph Capsule Designs

The core framework of the graph capsule network STGCapsNet within GCScoot includes the following two major components: 1) *multi-layer graph convolutions*: which consists of multiple graph convolution layers capturing multi-scale graph features; 2) *capsule routing*: which consists of primary and routing capsules to further derive fine-grained graph features.

**Graph convolution:** First, we design a multi-scale region feature extraction with different layers, where the extracted features are represented in the form of capsules. Specifically, to extract the features at the city regions, we apply the graph convolution [20]. The formulation framework of the graph convolution is applied upon each region as well as the peers with trips from/to it, which returns the new representation  $\mathbf{Z}^{(l+1)} \in \mathbb{R}^{N \times d'}$  of the region features given the inputs  $\mathbf{Z}^{(l)} \in \mathbb{R}^{N \times d}$ , *i.e.*,

$$\mathbf{Z}^{(l+1)} \triangleq \sigma \left( f(\mathbf{A})\mathbf{Z}^{(l)}\mathbf{W}^{(l)} \right), \quad (11)$$

where  $\mathbf{W}^{(l)} \in \mathbb{R}^{d \times d'}$  is a trainable weight matrix as a channel filter.

Since multiple correlation matrices are applied as shown in Eq. (9), we integrate them within the convolution as follows. Let  $\mathbf{W}_{ij}^{(l)} \in \mathbb{R}^{d \times d'}$  be the trainable weight matrix within the graph convolution layer  $l \in \{1, \dots, L\}$ , and  $\sigma(\cdot)$  be the non-linear activation function (we adopt ReLU in our prototype). We define the channel filter from all the channels in the  $l$ -th layer to the  $j$ -th channel in the  $(l+1)$ -th layer.

$$\mathbf{Z}_j^{(l+1)} \triangleq \sigma \left( \sum_{\mathbf{A}' \in m^{(k)}(\mathbf{A}_\star)} \sum_i \tilde{\mathbf{A}}\mathbf{Z}_i^l \mathbf{W}_{ij}^{(l)} \right), \quad (12)$$

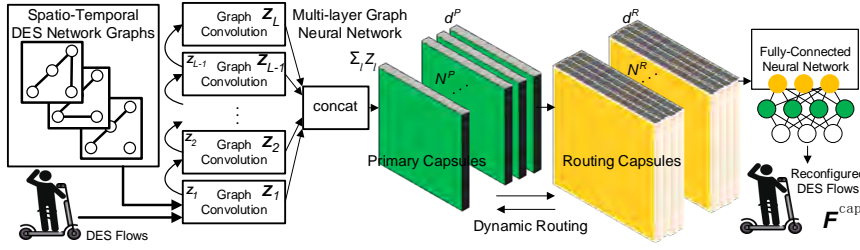
where we adopt the symmetric normalized Laplacian  $f(\cdot)$  [20], and have

$$\tilde{\mathbf{A}} = f(\mathbf{A}') = \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{A}' \tilde{\mathbf{D}}^{-\frac{1}{2}}, \quad (13)$$

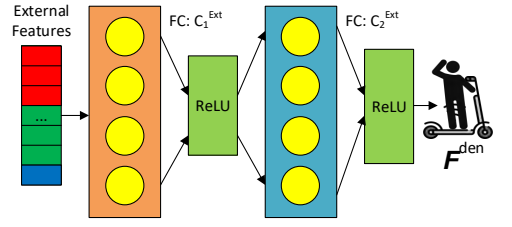
and the masked correlation matrices and the degree matrix are

$$\mathbf{A}' \in \{\mathbf{A}'_D, \mathbf{A}'_P, \mathbf{A}'_C\}, \quad \tilde{\mathbf{D}}_{ii} = \sum_j \mathbf{A}'(i, j). \quad (14)$$

**Capsule routing:** To handle the complexity of the urban DES network, we introduce the capsule structures within GCScoot which can better capture the spatial and temporal DES flow dynamics.



**Figure 25: Graph capsule neural network for DES networks (before).** Input: spatio-temporal DES network graphs and DES flows; Output: predictions of reconfigured DES flows  $F^{\text{cap}}$ .



**Figure 26: Dense neural network for external factor processing.**

The conventional neural network, including convolutional neural network, usually encodes the structural properties (say, geographical locations, directions and connections) in a scalar form. They have been identified to exhibit poor efficiency in preserving the structural properties of the input object [31]. To address this problem, the capsule network [31, 34, 37] has been proposed to extend the scalar into a vector such that the structural information can be preserved more efficiently for better computation and feature extraction. The features within the capsule network are represented with capsules, which are a structured group of neurons forming a vector-like representation for the inputs.

Specifically, the region features extracted from all the  $L$  graph convolution layers are concatenated into a tensor of higher dimension, *i.e.*,  $[Z_1, Z_2, \dots, Z_{L-1}, Z_L]$ , and fed to the primary capsules. Each layer of graph convolution represents the probability that the entity represented by the capsule is present in the current input. Let  $W^c$  be an  $N^P \times N^R$  weight matrix,  $W_{ij}^c$  be the weight of capsules  $i$  in PC and  $j$  in RC, and  $e_{ij}$  be the coupling coefficients that are determined through the dynamic routing process.

The dynamic routing process determines the likelihood, denoted as  $b_{ij}$ , that a preceding capsule  $i$  ( $i \in \{1, \dots, N^P\}$ ) in primary capsules (PC) should be coupled with a succeeding peer  $j$  ( $j \in \{1, \dots, N^R\}$ ) in the routing capsules (RC). For the succeeding RC, the input  $s_j$  to a capsule  $j$  there is formally given by

$$s_j = \sum_i e_{ij} \hat{u}_{j|i}. \quad (15)$$

where the coupling coefficient  $e_{ij}$  is given by a routing softmax function applied between the primary capsules and the routing capsules, *i.e.*,

$$e_{ij} = \frac{\exp(b_{ij})}{\sum_z \exp(b_{iz})}. \quad (16)$$

and the prediction vector  $\hat{u}_{j|i}$  represents the link between a capsule  $i$  in PC and  $j$  in RC, *i.e.*,

$$\hat{u}_{j|i} = W_{ij}^c u_i. \quad (17)$$

The resultant vector output, denoted as  $v_j$ , from the capsule  $j$  is then given by a squashing function to differentiate the long and short vector inputs, *i.e.*,

$$v_j = \frac{\|\mathbf{q}_j\|^2}{1 + \|\mathbf{q}_j\|^2} \cdot \frac{\mathbf{q}_j}{\|\mathbf{q}_j\|}, \quad (18)$$

via which the long vectors gets mapped towards ones while the short ones are shrunk towards zeros. The results are further used to update  $b_{ij}$  into  $b'_{ij}$  in the next iteration, *i.e.*,

$$b'_{ij} = b_{ij} + \hat{u}_{j|i} \cdot v_j. \quad (19)$$

Through iterations with the Eqs. (15)–(19), the graph capsule network learns the structured features within the input DES networks.

### 5.3 Architecture Summary

We further summarize the core framework in GCScoot, namely STGCapNet, as follows.

**Graph capsule neural network for DES networks:** We summarize the architecture of the multi-layer graph capsule neural network as in Fig. 25, where  $d^P$  and  $d^R$  are the capsule dimensions in the primary and routing capsules. Multiple graph convolution neural networks first extract the multi-scale features from the input spatio-temporal DES network graphs. This way, GCScoot obtains the initial activation with the spatio-temporal DES network graphs, and preserve the features of the sub-components of the graphs. The inputs at the first layer of graph convolution are the spatio-temporal DES network graphs represented by  $A$  and  $N \times 2$  flow matrix  $F^{(k)}$ , which serves as  $Z^0$ . Lower output  $Z^{(l-1)}$  is fed to the upper layer  $l$ . The outputs from all the graph convolutions are further concatenated and fed to the primary capsules. The results are then processed by the routing capsules, with dynamic routing with the preceding primary capsules. At the last stage, the fully connected neural network (we adopt in our prototype two dense layers with dimensions of  $C_1^{\text{Den}}$  and  $C_2^{\text{Den}}$ ) processes the outputs from the routing capsules and map them back to the reconfigured DES flows  $F^{\text{cap}} \in \mathbb{R}^{N' \times 2}$ .

**Dense network for external factors:** The external factors including weather conditions (we adopt 5 typical dimensions in our prototype, *i.e.*, temperature in Fahrenheit, sunny or not, rainy or not, cloudy or not, snowy or not, as shown in Fig. 23), day of a week and hour of a day are concatenated together into a vector and fed to a multi-layer fully-connected neural network in Fig. 26 in order to integrate external factors related to DES mobility. Two fully-connected neural networks (with output dimensions of  $C_1^{\text{Ext}}$  and  $C_2^{\text{Ext}}$ ) with ReLU activation function between them are adopted here. This component returns  $F^{\text{den}} \in \mathbb{R}^{N' \times 2}$ .

Finally, given the predictions of reconfigured DES flows  $F^{\text{cap}}$  and  $F^{\text{den}}$  from the graph capsule neural network as well as the dense network, GCScoot merges and averages the predicted flows at target interval  $k+1$  as in Fig. 2, and returned the final results for the city planners and service providers. The proposed model can be dynamically updated over the time given new deployment data.

## 6 EXPERIMENTAL EVALUATION

We first present the experimental settings in Sec. 6.1 and then the experimental results in Sec. 6.2.



## 6.1 Experimental Settings

We compare GCScoot with the following traditional and state-of-the-art methods:

- HA and SHA: which estimate the DES flows via the historical average and seasonal historical average. For example, HA (or SHA) predicts the flow volume at 10:00am – 11:00am of a Monday by averaging the flows of all Mondays (or all Mondays in the same season).
- LSTM: which estimates the DES flows based on the long-short-term memory.
- RNN: which predicts the time series of the DES flows based on recurrent neural network [29].
- STCNN: which models the DES flows via spatio-temporal convolutional neural network [26].
- MGCN: which estimates the flows through multi-graph convolutional neural network [9].
- MTL: which adaptively predicts the DES flows via spatio-temporal convolutional neural network and meta-learning [12, 39].
- FA+CNN: which adapts the domains of the previous DES flows with factor analysis and predicts the transformed traffics with convolutional neural network [23].
- DANN: which predicts and adapts to the dynamic flows via convolutional neural network with domain adaptation [13, 24].
- MSGN: which adaptively learns and forecasts the traffic flows with the multi-scale graph neural network [25].

Our experimentation has been done on a desktop server with Intel i7-8700K 3.70 GHz, 32GB RAM and Nvidia GTX 1080Ti (11 GB GDDR5). All algorithms are implemented through Python 3.6.5 with Tensorflow/Keras/PyTorch. As for the quantity offsets of e-scooters after reconfiguration, we adjust the predictions based on the number of deployed e-scooters based on the government statistics.

Unless otherwise stated, we use the following parameters by default. We adopt a temporal discretization interval of 30min as it is the minimum interval for the meteorological datasets, and set  $w = 12$  for Eq. (4). In the map preprocessing, we observe that a large grid eases prediction and computation but lowers the granularity, while a small grid introduces higher degree of correlations and computational overheads. To balance these, like the discretization in [22, 39], we evaluate the grid settings and discretize the maps of Austin, Louisville and Minneapolis into  $32 \times 32$ ,  $25 \times 13$  and  $11 \times 11$  grid maps, respectively. The map discretization takes into account the shape of the city area as well as coverage of DES deployment, and ensures that the average DES pick-ups/drop-offs of all deployment grids (with nonzero usage) are at least 100. We adopt  $L = 5$  graph convolution layers. The output dimensions ( $d'$ ) for each layer are set to  $\{100, 600, 100, 600, 100\}$  for  $Z_1$  to  $Z_5$ , and the important network parameters in STGCapNet are set as

$$\{N^P, d^P, N^R, d^R, C_1^{\text{Den}}, C_2^{\text{Den}}, C_1^{\text{Ext}}, C_2^{\text{Ext}}\} = \{10, 60, 10, 60, 16, 2, 16, 2\}.$$

We set the default number of dynamic routing in STGCapNet to 4, and the number of epochs to 200. Adam optimizer is used with a learning rate of 0.01. For the CNN in STCNN, FA+CNN and MTL, we set the number of filters to 64. The number of steps and dimensions of hidden states in LSTM are 6 and 12, respectively.

For our experimental settings, we consider a significant reconfiguration happens if 10% of deployment regions in the 7 days onward have changed (introduced or removed). In total, we have identified

**Table 1: Overall performance of all schemes in the three datasets.**

Schemes	Austin		Louisville		Minneapolis	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
HA	8.29	7.29	7.92	6.34	6.19	4.88
SHA	8.09	6.11	7.13	5.83	5.69	4.09
LSTM	7.60	5.90	7.05	5.57	4.99	3.52
RNN	6.84	5.28	6.83	5.18	4.63	3.03
STCNN	6.51	4.98	6.22	4.75	4.11	2.90
MGCN	6.14	4.86	4.81	4.45	3.67	2.39
MTL	4.71	4.51	4.43	4.09	3.52	2.28
FA+CNN	4.84	4.45	4.17	4.01	3.45	2.19
DANN	4.64	4.15	3.88	2.69	2.64	1.81
MSGN	4.07	3.87	3.12	2.54	1.76	1.12
GCScoot	<b>2.69</b>	<b>2.05</b>	<b>2.28</b>	<b>1.83</b>	<b>1.38</b>	<b>0.96</b>

**Table 2: RMSE regarding the new and existing regions during reconfiguration periods of the three cities.**

Schemes	Austin		Louisville		Minneapolis	
	New	Existing	New	Existing	New	Existing
GCScoot	3.88	2.04	3.45	1.88	2.18	1.31
MSGN	5.90	3.87	5.13	2.89	3.43	1.52
DANN	6.11	4.44	6.54	2.73	3.52	2.19
MTL	7.30	4.26	6.11	3.87	5.29	3.45
FA+CNN	7.51	4.82	6.34	3.99	5.18	3.14

**Table 3: RMSE on the rush hours & weekends at the three cities.**

Schemes	Austin		Louisville		Minneapolis	
	RH	Weekends	RH	Weekends	RH	Weekends
GCScoot	3.12	2.63	2.92	3.03	1.81	2.04
MSGN	5.83	5.57	5.26	5.89	3.35	3.74
DANN	5.54	5.32	6.55	6.73	3.66	3.46
MTL	5.21	5.74	6.22	6.83	5.08	4.72
FA+CNN	6.11	5.90	6.18	6.22	5.84	5.14

21 reconfigurations in Austin, 22 in Louisville and 9 in Minneapolis in the datasets. For each city, we conduct the model sensitivity studies upon the first 30 days' samples (first 80% for initial model training and 20% for validation), and use the rest for overall prediction performance comparison. The training time of GCScoot for Austin, Louisville and Minneapolis is around 5h, 1.1h and 0.5h based on our machine. For each dataset, we train GCScoot and other models with domain adaptation or transfer (MSGN, DANN, FA+CNN and MTL) based on the samples of 7 days before the reconfiguration happens, and test the models upon those samples between this reconfiguration and the next. For other schemes without adaptation, we use the historical records for flow prediction.

We evaluate the performance of all schemes based on the root mean square error (RMSE),

$$\text{RMSE} = \sqrt{\frac{1}{KN} \sum_{i=1}^N \sum_{k=1}^K \left( F_i^{(k)} - \widehat{F}_i^{(k)} \right)^2}, \quad (20)$$

and mean absolute error (MAE),

$$\text{MAE} = \frac{1}{KN} \sum_{i=1}^N \sum_{k=1}^K \left| F_i^{(k)} - \widehat{F}_i^{(k)} \right|, \quad (21)$$

where  $N$  and  $K$  are the numbers of regions and time intervals, and  $F_i^{(k)}$  and  $\widehat{F}_i^{(k)}$  are the actual and predicted DES flows.

## 6.2 Experimental Results

Presented below are our experimental results.

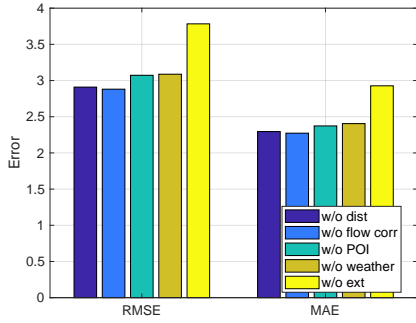


Figure 27: Performance of GCScout w/ and w/o the components & designs (Austin).

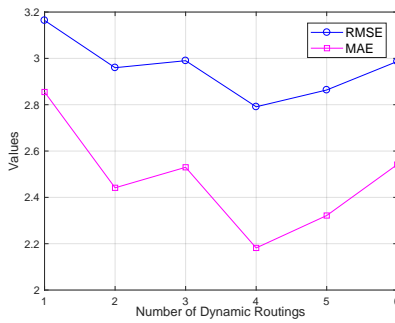


Figure 28: Performance of GCScout vs. numbers of dynamic routings (Austin).

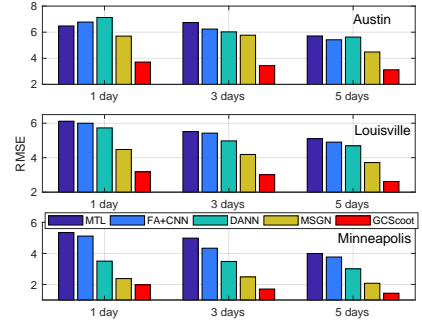


Figure 29: Performance of the schemes vs. the numbers of training samples.

*Prediction Performance Comparison:* We first present in Table 1 the experimental results on the performance on each of the datasets. Without proper feature adaptation, conventional schemes like HA, SHA, LSTM, RNN, STCNN and MGCN cannot achieve high prediction accuracy given reconfigured DES deployment. Compared to FA+CNN, DANN and MSGN, GCScout achieves higher accuracy due to its more comprehensive modeling of the DES flows and fine-grained feature extraction via the graph capsule network. We also observe the performance variations of the schemes for different datasets. All the schemes have experienced higher errors in Austin than other two datasets, due to higher volumes of usage and more complex DES usage.

By focusing on the new and existing ( $V \cap V'$ ) regions after the reconfigurations, we further show in Table 2 the performance on the three datasets. Specifically, we show the RMSE of the schemes GCScout, MSGN, DANN and FA+CNN. Forecast of dynamic flows regarding the new regions is overall more challenging due to the absence of historical data. GCScout is shown to outperform the other schemes in predicting the flows from both the new and existing regions, which validates GCScout’s high adaptability to the DES reconfiguration.

By focusing on the morning/evening rush hours (RH: 08:00 am - 10:00am; 05:00 pm - 07:00 pm) and weekends (Saturdays and Sundays), we show in Table 3 the performance of GCScout and the other four schemes for the three datasets. Note that mobility prediction during rush hours can be challenging due to high and dense traffic volumes, while diverse travel purposes at the weekends render the accurate forecast rather difficult. We can see that GCScout outperforms the other state-of-the-arts in predicting the flows due to its higher adaptivity to the DES traffic flows.

*Model Sensitivity:* After presenting the overall performance upon the three datasets, we evaluate the model sensitivity of GCScout to variations in spatial/temporal/external factors, dynamic routings, and number of samples (in terms of days) for model training. This sensitivity has been studied on the first 30 days’ samples (total 1,440 time intervals) for each dataset, and we show the results upon the validation data (Sec. 6.1).

Taking Austin dataset as an example, we first show in Fig. 27 the performance (RMSE and MAE) of GCScout with and without each of the spatial and temporal correlations as well as the external/temporal factors. Specifically, we show the RMSEs and MAEs of GCScout without distances (w/o dist), flow correlations (w/o flow corr), POIs (w/o POIs), weather (w/o weather) and fusion with

dense network processing the external factors (w/o ext). By incorporating the aforementioned factors, GCScout is shown to be able to adaptively predict the reconfigured DES flows.

We further present in Fig. 28 the performance (RMSE) of GCScout vs. the number of dynamic routings. More dynamic routings generally help GCScout capture more DES flow correlations among the regions, and hence better accuracy. The improvement via more routings and iterations begins to converge after introducing more routings. Therefore, we set the default number of dynamic routings to 4 in our experimental studies.

Finally, we show in Fig. 29 the performance (RMSEs) of GCScout, MSGN, DANN, MTL and FA+CNN given only 1, 3 and 5 days of the training samples with respect to each of the three datasets. Fewer training data imposes more challenges upon all the schemes, which also reflect the common practice when only a few pilot studies have been conducted before reconfiguration is made. Via more comprehensive learning based on the graph capsule neural network, GCScout still outperforms all the baselines. This way, the city planners and service providers may be able to conduct proactive flow studies in the DES initialization.

While our experimental studies focus on spatial, temporal and external factors including meteorological data, GCScout is general enough to accommodate other factors (including satellite images [23], local traffic [16] and demographic information) to enhance accuracy further.

## 7 CONCLUSION

In this paper, we have studied the dynamic mobility patterns of dockless e-scooter sharing (DES) systems due to the deployment reconfiguration (expansion and shrinkage of the covered regions). We have proposed a novel system framework called GCScout for dynamic distribution prediction of DES reconfiguration. Via data-driven studies upon the DES data, we have analyzed various spatial and temporal factors related to the DES flows, including e-scooter flow dynamics, distances and region connectivities. Taking the analysis results into account, we have proposed a novel spatio-temporal graph capsule neural network, which comprehensively and adaptively forecasts the resultant e-scooter flows given the altered deployment regions. We have conducted extensive experimental evaluation upon three different e-scooter datasets in three populous US cities, showing that GCScout outperforms state-of-the-arts and is effective and accurate in forecasting e-scooter mobility and flows after reconfiguration.

## REFERENCES

- [1] 2018. D.C. to expand the number of scooters and dockless bikes allowed in 2019. (2018). <https://dc.curbed.com/2018/11/9/18076394/dc-scooters-dockless-bikes-transportation-cycling-ddot>.
- [2] 2019. Austin Microbilty. (2019). <https://austintexas.gov/micromobility>.
- [3] 2019. Dataset of Austin B-Cycle Trips. (2019). <https://data.austintexas.gov/Transportation-and-Mobility/Austin-B-Cycle-Trips/tyfh-5r8s>.
- [4] 2019. Electric Scooter Sharing Market in US and Europe 2019-2024. (2019). <https://mobilityforesights.com/product/scooter-sharing-market-report/>.
- [5] 2019. Street Centerline, Austin. (2019). <https://data.austintexas.gov/Locations-and-Maps/Street-Centerline/m5w3-uea6>.
- [6] 2019. Street Centerline, Louisville. (2019). <https://data.louisvilleky.gov/dataset/street-centerline>.
- [7] 2019. Street Centerline, Minneapolis. (2019). [http://opendata.minneapolismn.gov/datasets/e68d01d782c04d88876bbd51e1c40702\\_0](http://opendata.minneapolismn.gov/datasets/e68d01d782c04d88876bbd51e1c40702_0).
- [8] Georg Brandstätter, Michael Kahr, and Markus Leitner. 2017. Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand. *Transportation Research Part B: Methodological* 104 (2017), 17 – 35.
- [9] Di Chai, Leye Wang, and Qiang Yang. 2018. Bike Flow Prediction with Multi-graph Convolutional Networks. In *Proc. ACM SIGSPATIAL*. 397–400.
- [10] Y. Duan and J. Wu. 2019. Optimizing Rebalance Scheme for Dock-Less Bike Sharing Systems with Adaptive User Incentive. In *Proc. IEEE MDM*. 176–181.
- [11] Zhixuan Fang, Longbo Huang, and Adam Wierman. 2017. Prices and Subsidies in the Sharing Economy. In *Proc. WWW*. 53–62.
- [12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic Meta-learning for Fast Adaptation of Deep Networks. In *Proc. ICML*. 1126–1135.
- [13] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. In *Proc. ICML*. 1180–1189.
- [14] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal Multi-Graph Convolution Network for Ride-hailing Demand Forecasting. In *Proc. AAAI*.
- [15] 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.
- [16] 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.
- [17] Suining He and Kang G. Shin. 2019. Spatio-Temporal Capsule-based Reinforcement Learning for Mobility-on-Demand Network Coordination. In *Proc. WWW*. 2806–2813.
- [18] Tianfu He, Jie Bao, Ruiyuan Li, Sijie Ruan, Yanhua Li, Chao Tian, and Yu Zheng. 2018. Detecting Vehicle Illegal Parking Events Using Sharing Bikes' Trajectories. In *Proc. ACM KDD*. 340–349.
- [19] J. Hu, Z. Yang, Y. Shu, P. Cheng, and J. Chen. 2017. Data-Driven Utilization-Aware Trip Advisor for Bike-Sharing Systems. In *Proc. IEEE ICDM*. 167–176.
- [20] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proc. ICLR*.
- [21] 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.
- [22] Yuxuan Liang, Kun Ouyang, Lin Jing, Sijie Ruan, Ye Liu, Junbo Zhang, David S. Rosenblum, and Yu Zheng. 2019. UrbanFM: Inferring Fine-Grained Urban Flows. In *Proc. ACM KDD*. 3132–3142.
- [23] Zhaoyang Liu, Yanyan Shen, and Yanmin Zhu. 2018. Inferring Dockless Shared Bike Distribution in New Cities. In *Proc. WSDM*. 378–386.
- [24] Zhaoyang Liu, Yanyan Shen, and Yanmin Zhu. 2018. Where Will Dockless Shared Bikes Be Stacked?: – Parking Hotspots Detection in a New City. In *Proc. ACM KDD*. 566–575.
- [25] Man Luo, Hongkai Wen, Yi Luo, Bowen Du, Konstantin Klemmer, and Hongming Zhu. 2019. Dynamic Demand Prediction for Expanding Electric Vehicle Sharing Systems: A Graph Sequence Learning Approach. *arXiv preprint arXiv:1903.04051* (2019).
- [26] 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.
- [27] OpenStreetMap. 2019. (2019). [www.openstreetmap.org](http://www.openstreetmap.org).
- [28] Ling Pan, Qingpeng Cai, Zhixuan Fang, Pingzhong Tang, and Longbo Huang. 2019. Rebalancing Dockless Bike Sharing Systems. In *Proc. AAAI*.
- [29] Yan Pan, Ray Chen Zheng, Jiayi Zhang, and Xin Yao. 2019. Predicting bike sharing demand using recurrent neural networks. *Procedia Computer Science* 147 (2019), 562 – 566.
- [30] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. 2019. Urban Traffic Prediction from Spatio-Temporal Data Using Deep Meta Learning. In *Proc. ACM KDD*. 1720–1730.
- [31] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Proc. NIPS*. 3856–3866.
- [32] C Scott Smith and Joseph P Schwieterman. 2018. E-Scooter Scenarios: Evaluating the Potential Mobility Benefits of Shared Dockless Scooters in Chicago. (2018).
- [33] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *Proc. WWW*. 1067–1077.
- [34] Saurabh Verma and Zhi-Li Zhang. 2018. Graph capsule convolutional neural networks. In *Proc. Joint ICML and IJCAI Workshop on Computational Biology*.
- [35] Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. 2019. Cross-city transfer learning for deep spatio-temporal prediction. In *Proc. IJCAI*. 1893 – 1899.
- [36] Shuai Wang, Tian He, Desheng Zhang, Yunhui Liu, and Sang H. Son. 2019. Towards Efficient Sharing: A Usage Balancing Mechanism for Bike Sharing Systems. In *Proc. WWW*. 2011–2021.
- [37] Zhang Xinyi and Lihui Chen. 2019. Capsule graph neural network. In *Proc. ICLR*.
- [38] Zidong Yang, Ji Hu, Yuanchao Shu, Peng Cheng, Jiming Chen, and Thomas Moscibroda. 2016. Mobility Modeling and Prediction in Bike-Sharing Systems. In *Proc. ACM MobiSys*. 165–178.
- [39] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. 2019. Learning from Multiple Cities: A Meta-Learning Approach for Spatial-Temporal Prediction. In *Proc. WWW*. 2181–2191.
- [40] 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.
- [41] Jiawei Zhang, Xiao Pan, Moyin Li, and Philip S. Yu. 2016. Bicycle-sharing Systems Expansion: Station Re-deployment Through Crowd Planning. In *Proc. ACM SIGSPATIAL*. Article 2, 10 pages.
- [42] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction.. In *Proc. AAAI*. 1655–1661.