

Distribution Prediction for Reconfiguring Urban Dockless E-Scooter Sharing Systems

Suining He^{ID}, Member, IEEE and Kang G. Shin^{ID}, Life Fellow, IEEE

Abstract—Dockless E-scooter Sharing (DES) has become a popular means of last-mile commute for many smart cities. As e-scooters are getting deployed dynamically and flexibly across city regions that expand and/or shrink, accurate prediction of the e-scooter distribution given the reconfigured regions becomes essential for city planning. We present **GCScoot**, a novel flow distribution prediction approach for reconfiguring urban DES systems. Based on real-world datasets with reconfiguration, we analyze e-scooter distribution features and flow dynamics for the data-driven designs. We propose a novel spatio-temporal graph capsule neural network within **GCScoot** to predict future dockless e-scooter flows given the reconfigured regions. **GCScoot** pre-processes historical spatial e-scooter distributions into flow graph structures, where discretized city regions are considered as nodes and inter-region flows as edges. To facilitate initial training, we cluster the regions and generate virtual data for new deployment regions based on their peers in the same cluster. Given above designs, the region-to-region correlations embedded within the temporal flow graphs are captured via the multi-graph capsule convolutional neural network which accurately predicts the DES flows. Extensive studies upon four e-scooter datasets (total > 3.4 million rides) in four populous US cities have corroborated accuracy and effectiveness of **GCScoot** in predicting the e-scooter distributions.

Index Terms—Dockless e-scooter, reconfiguration, distribution prediction

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 (IoTs) and location-based services, DES does not, in general, require fixed docking stations for users to receive or return the e-scooters. With dockless and motorized (electric motors and batteries) features, DES provides another faster and easier first/last-mile connectivity of the city [1], [2] beyond the conventional bike sharing. According to the National Association of City Transportation Officials, over 85,000 e-scooters were deployed in around 100 cities in US as of 2018.¹

Due to the increasing commercial potential (DES platforms like Bird and Lime have raised USD\$1.48 billion by April 2019 [1]) and growing socio-economic 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 percent of the e-scooter deployment in 2019 [3]. Spin, the DES supplier acquired by Ford at the end of 2018, has announced in February 2020 its expansion into Europe, including their first international fleet in Cologne, Germany [4]. 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, town hall meetings 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 alteration of local traffic environments. City 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 e-scooters, and prevent/reduce underserved customers and over-congested side walks due to excessive parking of e-scooters, 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:

- Suining He is with the Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269 USA.
E-mail: suining.he@uconn.edu.
- Kang G. Shin is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA.
E-mail: kgshin@umich.edu.

Manuscript received 22 July 2020; revised 9 Jan. 2021; accepted 9 Feb. 2021.
Date of publication 25 Feb. 2021; date of current version 7 Nov. 2022.

(Corresponding author: Suining He.)

Recommended for acceptance by M. A. Cheema.

Digital Object Identifier no. 10.1109/TKDE.2021.3062074

- (a) 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.
- (b) Due to its dockless, easy-maneuvering and last-mile nature, e-scooters traveling across city regions form



Fig. 1. Illustration of dockless e-scooters & DES reconfiguration.

complex inter-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.

- (c) Few studies have conducted data analytics on the spatial and temporal distributions of existing metropolitan DES systems. However, models/components should be carefully designed based on the deployment insights, which are essential for any data-driven model study.

To address the above challenges, 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 four 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 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 main contributions of GCScoot are summarized as follows:

- 1) *Data-driven analytics & designs for dockless e-scooter reconfiguration* (Section 4): 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* study that investigates, identifies and formulates the dockless e-scooter sharing flow distribution prediction problem given the reconfigured deployment.
- 2) *Spatio-temporal graph capsule neural network for reconfigured DES distribution forecasting* (Section 5): We propose a data-driven design for DES distribution prediction based on a 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 features. Dynamic routings between the graph convolutions and capsules capture the geographical properties via vectorized representations [5], thus leading to high accuracy in the dynamic flow prediction. We have further designed multi-graph mechanism to enhance the GCScoot's accuracy based on multiple temporal DES network graphs in the historical records.

- 3) *Region clustering & virtual data generation for efficient GCScoot initialization* (Section 6): To further enhance GCScoot's adaptability to regions without historical DES deployment, we design an efficient region clustering and virtual trip data generation for model initialization. By clustering the DES regions before reconfiguration, we find the similar ones in terms of region-to-region POIs (points-of-interest) similarities and generate the initial and short-term trip distribution data as the training inputs of the new regions. This way, the model initialization of GCScoot can be facilitated with enhanced training and prediction efficiency.
- 4) *Extensive data-driven & experimental studies* (Section 7): Based on the above analytics and network formulation, we have conducted extensive experimental studies upon over 3.4 million rides from different e-scooter datasets in four populous US cities including Austin TX, Louisville KY, Minneapolis MN and Chicago IL. Our results have corroborated the accuracy, effectiveness and robustness of GCScoot in predicting the dynamic distribution of dockless e-scooter mobility under various experimental settings.

The rest of the paper is organized as follows. We first discuss the related work in Section 2, followed by an overview of the concepts, problem formulation and data sets in Section 3. Then, we present in Section 4 the data-driven analytics and designs for DES reconfiguration, followed by the core dynamic prediction framework in Section 5. Afterwards, the module integration and model initialization/cold-start details are presented in Section 6. We then experimentally evaluate the performance of GCScoot in Section 7, discuss its deployment in Section 8, and finally conclude the paper in Section 9.

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 [6], [7]. Unlike prior smart and shared mobility studies [8], [9], [10], our work focuses on rapidly proliferating dockless e-scooter systems, and derives important insights and data-driven designs for reconfiguring the deployment of e-scooters. Despite its prototype studies upon the e-scooter datasets, GCScoot can be adapted and extended to other on-demand modalities with reconfigured deployment [11], [12], including e-vehicle station relocation [13], car-sharing [14],

[15], bike-sharing [16], [17], [18], [19], [20], [21] and urban planning [22], enhancing their service quality for the emerging smart connected community.

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.* [23] investigated the application of convolutional neural network. Zhang *et al.* [24] proposed the spatio-temporal residual neural network for bike mobility prediction. With the road network graphs, a spatio-temporal graph convolutional network design for traffic series prediction was discussed by Yu *et al.* [25], and graph neural network studies have also been conducted by Wang *et al.* [26]. For station-based bike sharing, Chai *et al.* [27] designed a multi-graph convolutional neural network, which is followed by Geng *et al.* [28] upon ride sharing services. Wu *et al.* [29] proposed a graph wavenet with self-adaptive adjacency matrix and graph convolution with dilated causal convolution, and validated its performance with traffic speed datasets. Yao *et al.* [30] and Wang *et al.* [31], respectively, proposed a new meta-learning approach and a region transfer method to transfer knowledge from multiple cities. With meta-learning, Pan *et al.* [32] also 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-based service and IoTs, many single-track vehicle sharing systems like bikes and e-scooters have rendered parking docks obsolete. Pan *et al.* [33] studied a reinforcement learning algorithm in order to balance the dockless bike sharing system. Liu *et al.* [34] leveraged the factor analysis and convolutional neural network to transfer the knowledge between two cities, which is further followed by [35] regarding domain adaptation. Smith *et al.* [2] 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 four 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.

A preliminary/early version of this work was presented [36]. Besides motivating, elaborating and discussing more on the core formulation and deployment (Sections 1, 2 and 8), this version adds significantly more to the conference version from the following three major perspectives:

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on April 11, 2024 at 01:06:59 UTC from IEEE Xplore. Restrictions apply.

- *Multi-Graph Mechanism* (Section 5): We further enhance the accuracy of GCScoot's prediction by introducing the multi-graph mechanism within GCScoot. Specifically, the design characterizes and identifies the temporal region-to-region correlations through multiple DES network graphs in the past time intervals, and the performance improvement is validated through our experimental evaluation (Section 7).
- *Adaptive Cold Start Mechanism based on Clustered Regions* (Section 6): We also add a novel region clustering and data generation to enhance GCScoot's adaptability to cold-start, i.e., facilitating the model initialization for the regions without historical DES trip data. We have generated the virtual DES flows for those regions based on their correlations with others with historical trips. This enables GCScoot to better capture the potential trips in the initial testing phase.
- *Further Data Analytics & Experimental Studies* (Sections 3, 4 and 7): We have also conducted more data-driven and experimental studies with additional data (including DES deployment from Chicago IL) and settings regarding different design components, and validated the effectiveness and accuracy of our proposed scheme.

3 CONCEPTS, PROBLEM FORMULATION & DATA SETS

We first briefly introduce the important concepts related to GCScoot's formulation in Section 3.1, and then present the problem formulation in Section 3.2. Finally, we describe the datasets for our data analytics in Section 3.3. The important symbols used in this paper are listed in Table 1.

3.1 Important Concepts

Presented below are the definitions of important concepts.

Definition 1 (City Regions & Time Intervals). Like prior work [24], [30], the entire map of a city is discretized into a set of total N regions (say, rectangular grids in our case), yielding a finite geographical set for computational convenience. Each region is represented by the coordinate of its center, i.e., $\mathbf{r}_n = [\text{lat}_n, \text{lon}_n]$ ($n \in \{1, \dots, N\}$). 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 DFS 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

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on April 11, 2024 at 01:06:59 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
Major Symbols in GCScoot Formulation

Notations	Definitions
\mathbf{r}_n	Geographic location (center) of region n in the city map
$\tau(i, j)$	Trips from regions i to j
$T(i, j)$	Number of trips between regions i and j
\mathbf{F}_i	DES flow at region i
\mathbf{G}, \mathbf{V} and \mathbf{E}	DES network graph, region vertices and their trip edges
$\mathbf{A}_D(i, j), \mathbf{A}_P(i, j), \mathbf{A}_C(i, j)$	Spatial, point of interest & temporal correlations of \mathbf{r}_i and \mathbf{r}_j
$\mathbf{A}'_D(i, j), \mathbf{A}'_P(i, j), \mathbf{A}'_C(i, j)$	Masked correlations of \mathbf{r}_i and \mathbf{r}_j given region introduction/removal
$a^{(k)}(i, j)$	Proportion of e-scooter rides from regions i to j
$\vec{\mathbf{u}}(i, j)$	Relative flow proportion between regions i and j
$h(i, j)$	Connectivity metric between regions i and j
$\mathbf{Z}^{(l)}$	Representation of region features at the layer l
$\mathbf{W}^{(l)}$	Trainable weight matrix at the layer l
$\tilde{\mathbf{D}}$	Diagonal matrix consisting of degrees of the input graph
\mathbf{s}_j	Input vector to capsule j
\mathbf{e}_{ij}	Coupling coefficient between primary and routing capsules
\mathbf{v}_j	Output vector from capsule j
b_{ij}	Likelihood that capsules i and j are coupled

$i, j \in \{1, \dots, N\}$), 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, i.e.,

$$\mathbf{E} = \{T(i, j)'s\}, \quad \forall i, j \in \{1, \dots, N\}, \quad (1)$$

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 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

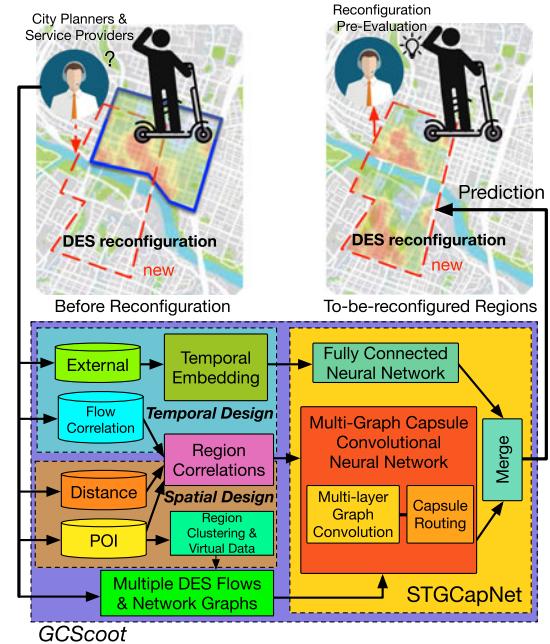


Fig. 2. Illustration of the system framework in GCScoot.

$$\mathbf{F}_i^{(k)} = (P_i^{(k)}, D_i^{(k)}). \quad (2)$$

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.,

$$\{\mathbb{G}^{(k-w+1)}, \mathbb{G}^{(k-w+2)}, \dots, \mathbb{G}^{(k)}\}, \quad (3)$$

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

$$\mathbf{F}^{(k+1)} = \{\mathbf{F}_i^{(k+1)}\}, \quad (i \in \{1, \dots, N\}), \quad (4)$$

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, town hall meetings, site survey and market analysis [37].

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

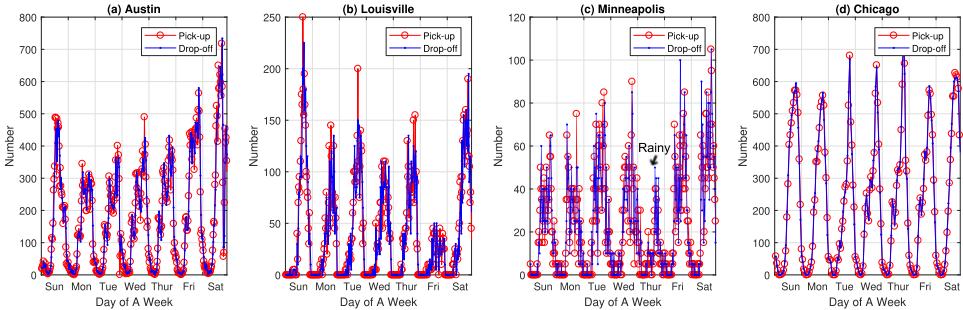


Fig. 3. Dynamic flows of a week: (a) Austin (Aug., 2018); (b) Louisville (Oct., 2018); (c) Minneapolis (Oct., 2018); (d) Chicago (Sept. 2019).

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. We also prepare the region clustering and generate cold-start (virtual) trip data for model training.

Multi-layer graph convolution and capsule routing, with multi-graph mechanism processing multiple input graphs and flows, are applied to capture dynamic flow patterns. To enhance the adaptability in the model cold-start, regions are clustered based on the POI similarities, and virtual trip data is then generated for the new regions based on their assigned clusters. Predicted e-scooter flows for the DES reconfiguration via both networks are merged and returned. Combining the above factors, STGCNet 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 extensive data analytics and experimental evaluation based on the following four datasets from populous US cities:

- *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°W, -97.58°W, 30.2°N, 30.499°N].² 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].³ 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 coordinations as well as timestamps, covering a bounding box of [-93.38°W, -93.08°W, 44.89°N, 45.02°N].⁴ Outliers have been removed when trips last for over 7 hours and were less than 0 miles or exceeded 24 miles.

2. <https://data.austintexas.gov/Transportation-and-Mobility/Shared-Micromobility-Vehicle-Trips/7d8e-dm7r>, Accessed: May-2020.

3. <https://data.louisvilleky.gov/dataset/dockless-vehicles>, Accessed: May-2020.

4. <http://www.minneapolismn.gov/publicworks/trans/WCMSP-212816>, Accessed: May-2020.

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on April 11, 2024 at 01:06:59 UTC from IEEE Xplore. Restrictions apply.

- *Chicago, IL* (June, 2019 – Sept, 2019): This dataset provides a total of 568,073 pilot scooter trips with pick-up/drop-off coordinations as well as timestamps, covering a bounding box of [-87.81°W, -87.63°W, 41.83°N, 41.96°N].⁵ Outliers have been removed when the trips have pick-up/drop-off locations missing, or exceed 8 hours.

We further show the dynamic flows of the DES pick-ups and drop-offs during a week in each city in Fig. 3. We can observe more DES flows (pick-ups/drop-offs) in Chicago and Austin than Louisville and Minneapolis, as well as high and dynamic volume of daily rides during weekends. Regarding the Chicago deployment, only hour-level granularity is provided by the Chicago open data portal following the related privacy rules. Besides the above datasets, we also retrieve the city map (street centerlines) and obtain the POI data from the OpenStreet Map [38].

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 Section 4.1, and then present the data-driven studies of spatial and temporal factors in Section 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 & 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 percent shorter trip time duration and 33.14 percent wider spectrum of travel distance than the bike sharing (Austin B-Cycle) [39]. 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 [34], [40].

5. https://www.chicago.gov/city/en/depts/cdot/supp_info/escooter-share-pilot-project.html, Accessed: May-2020.

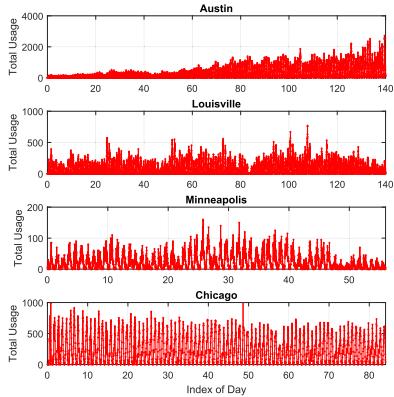


Fig. 4. Usage dynamics at all regions in the four datasets.

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 [41]), 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 the 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 are likely to increase DES vehicles at the city regions other than the initially licensed ones [42]. Therefore, we have observed additional deployment regions within Austin, 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. Compared to other three cities, Chicago experiences reconfiguration with minor shrinkage.

On the other hand, the DES deployment may be restricted to certain areas of a city. Deployment shrinkage may happen due to regulation by the government in city planning decisions. For example, according to the Dockless Mobility Program by Austin Transportation Department, dockless e-scooters are not allowed within parks, off-street parking lots or garages. The Transportation Department may dynamically reduce the deployment of DES within a specific area, thus changing the DES mobility.

To summarize, 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 km^2) from July to December 2018 in Fig. 8. We can observe that the reconfiguration

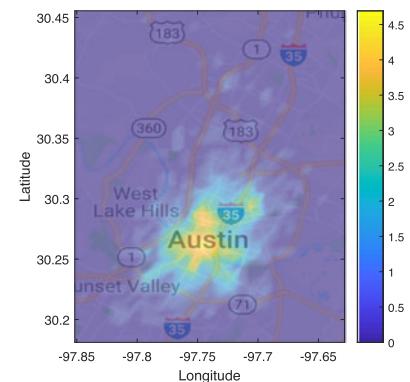


Fig. 5. Pick-up distributions in August, 2018 (Austin).

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, 10, 11, and 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

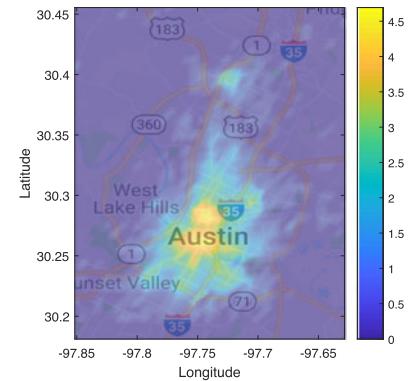


Fig. 6. Pick-up distributions in October, 2018 (Austin).

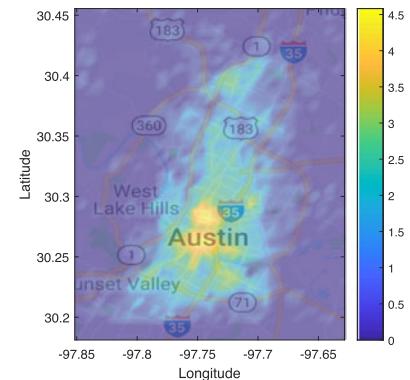


Fig. 7. Pick-up distributions in December, 2018 (Austin).

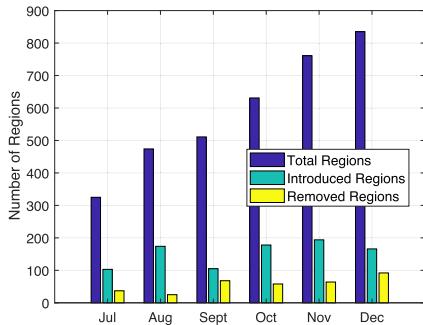


Fig. 8. Statistics of the DES deployment regions (Austin).

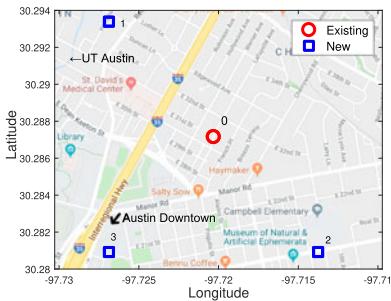


Fig. 9. Centers of existing & new deployment regions (south of Cherrywood, Austin).

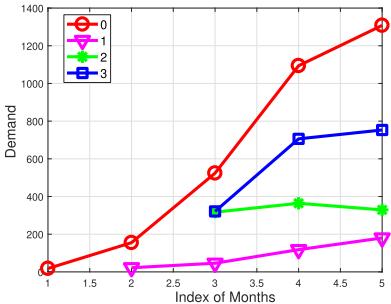


Fig. 10. Monthly demand increases with neighborhood expansion (ids: locations in Fig. 9).

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 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.

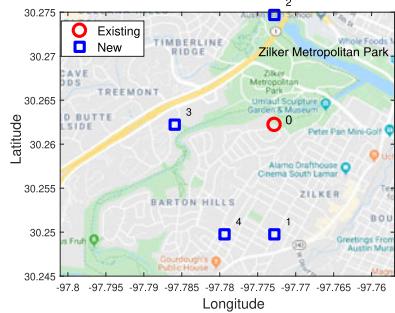


Fig. 11. Centers of existing & new deployment regions (near Zilker Metropolitan Park, Austin).

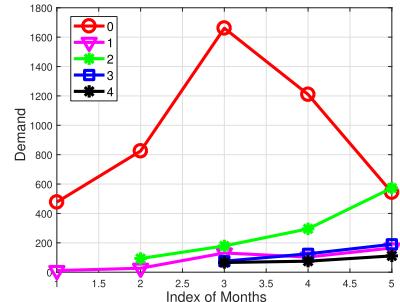


Fig. 12. Monthly demand decreases with neighborhood expansion (ids: locations in Fig. 11).

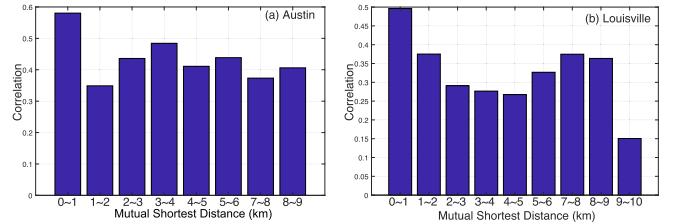


Fig. 13. Usage correlations versus distances: (a) Austin; (b) Louisville.

To reflect the terrain characteristics in our collected road network data, we define the shortest path distance (unit: km) between regions i and j as $\text{sp}(\mathbf{r}_i, \mathbf{r}_j)$. Based on the street centerline obtained from the open data portal of the local governments [43], [44], [45], [46], 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. 13 the mean usage correlations of regions versus their mutual shortest path distances. In particular, we find all the pairs of city regions whose mutual shortest path distance is within a certain range (say, 1~2 km), calculate their mutual DES usage correlations based on Pearson correlation (sliding window size is 1 day within a week), and find the mean of all correlation values between all pairs. While more distant regions generally have lower correlations, we can still observe some highly 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

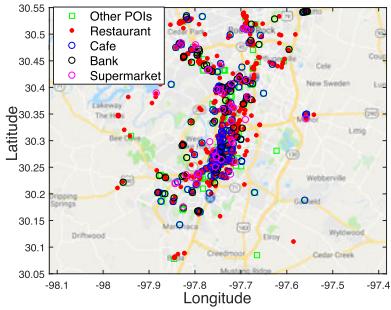


Fig. 14. POIs in Austin, TX.

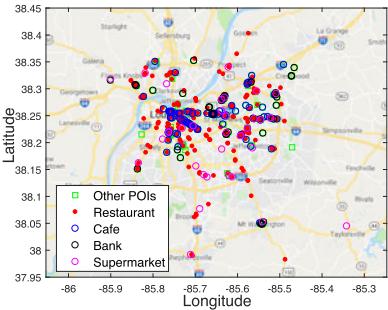


Fig. 15. POIs in Louisville, KY.

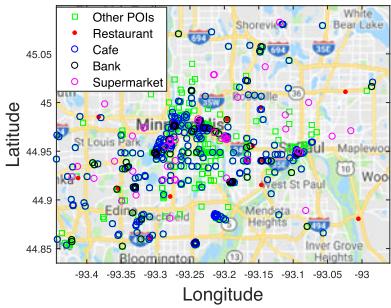


Fig. 16. POIs in Minneapolis, MN.

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 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, 2,204 and 15,026 POIs for Austin, Louisville, Minneapolis and Chicago, which are, respectively, as illustrated in Figs. 14, 15, 16 and 17. We can observe the variations of POI types and location distributions across the four cities. We also show in Figs. 18, 19, 20 and 21 the POI similarity matrices for all the regions *w.r.t.* each of the four cities we have studied. Here we discretize the city maps (with geographic bounding boxes defined in Section 3.3) of Austin, Louisville, Minneapolis and Chicago into 32×32 , 25×13 , 11×11 and 15×15 grids. This discretization takes into account the prediction granularity and the computational efficiency. Besides the prediction modeling, we will leverage the POI similarities for the region clustering and virtual data generation.

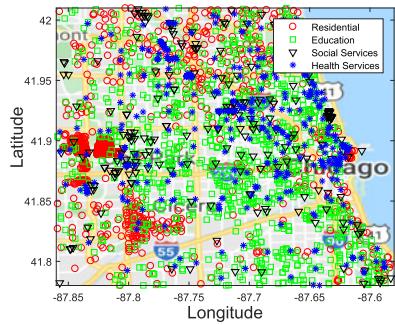


Fig. 17. POIs in Chicago, IL.

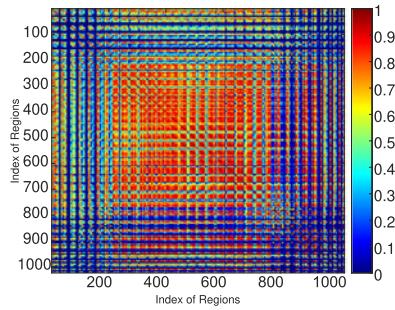


Fig. 18. Region-to-region POI similarities, Austin.

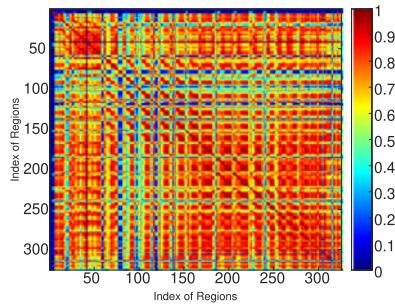


Fig. 19. Region-to-region POI similarities, Louisville.

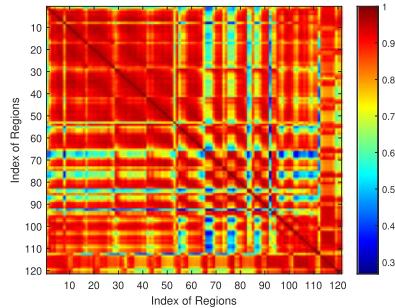


Fig. 20. Region-to-region POI similarities, Minneapolis.

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

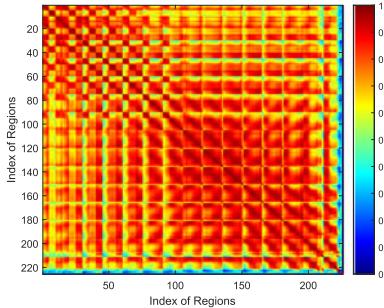


Fig. 21. Region-to-region POI similarities, Chicago.

Friday in Fig. 3b can be the result of consecutive rainy conditions illustrated in Fig. 22. 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 Section 5.1. We then provide the core graph capsule designs of GCScoot in Section 5.2.

5.1 Spatial and Temporal Designs

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

Spatial 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

$$\mathbf{A}_D(i, j) \triangleq \frac{1}{1 + \text{sp}(\mathbf{r}_i, \mathbf{r}_j)}. \quad (5)$$

Spatial POI. To reflect the correlations due to city functionality, we measure the *PoI closeness* between the feature

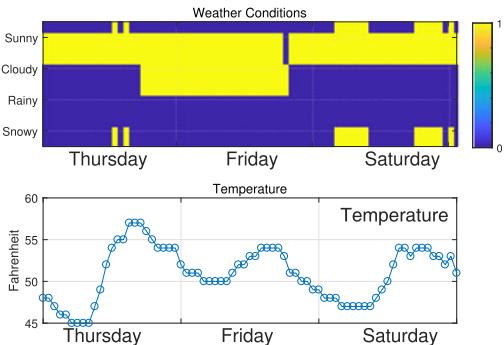


Fig. 22. Weekend weather conditions for Fig. 3b in Louisville.

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on April 11, 2024 at 01:06:59 UTC from IEEE Xplore. Restrictions apply.

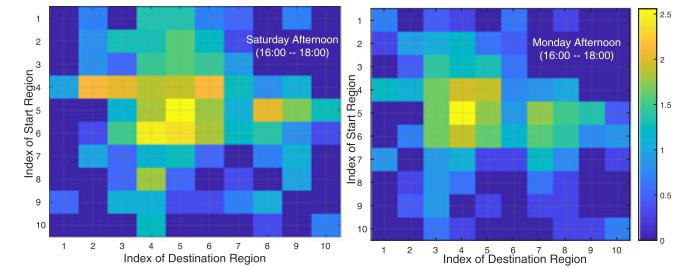


Fig. 23. Illustration of connectivities of city regions on Saturday & Monday (Austin).

vectors \mathbf{P}_i and \mathbf{P}_j of the two regions based on the cosine similarity, which is formally given by

$$\mathbf{A}_P(i, j) \triangleq \cos(\mathbf{P}_i, \mathbf{P}_j) = \frac{\mathbf{P}_i \cdot \mathbf{P}_j}{\|\mathbf{P}_i\| \cdot \|\mathbf{P}_j\|}. \quad (6)$$

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' similar commute routines between them. Let $T_i^{(t)}$ be the set of DES rides starting from region i at an interval t , i.e.,

$$|T_i^{(t)}| = \sum_{j=1}^N |\tau(i, j)^{(t)}|, \quad (7)$$

which characterizes the overall DES trend starting from i . 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

$$\mathbf{A}_C^*(i, j) \triangleq \frac{\sum_{t=k-w}^{t=k} |T_i^{(t)}| \cdot |T_j^{(t)}|}{\sqrt{\sum_{t=k-w}^{t=k} |T_i^{(t)}|^2} \cdot \sqrt{\sum_{t=k-w}^{t=k} |T_j^{(t)}|^2}}. \quad (8)$$

In other words, $\mathbf{A}_C^*(i, j)$ increases if the regions i and j have more similar and concurrent trends of rides.

Spatio-Temporal 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. 23 the flow volumes (in $\log_{10}(\cdot)$) from the start regions (vertical axis) to the destinations (10 selected regions in Austin with the maximum DES usage), and we can observe more diverse DES flows during weekends due to broader riding purposes.

Consideration of Eq. (8) only cannot comprehensively reflect the directional dependency between regions. So, we further integrate the connectivity among regions within our formulation. Specifically, we first define the proportion of e-scooter rides $\tau^{(k)}(i, j)$ from regions \mathbf{r}_i to \mathbf{r}_j in the time interval k as

$$a^{(k)}(i, j) \triangleq \frac{\tau^{(k)}(i, j)}{\sum_{l=1, l \neq i}^N \tau^{(k)}(i, l)}. \quad (9)$$

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

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on April 11, 2024 at 01:06:59 UTC from IEEE Xplore. Restrictions apply.

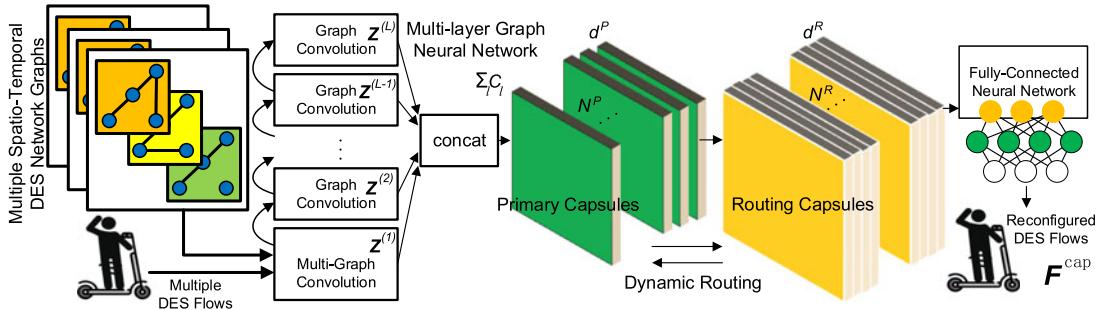


Fig. 24. Graph capsule neural network for DES networks (before). Input: multiple spatio-temporal DES network graphs and the corresponding DES flows; Output: predictions of reconfigured DES flows F^{cap} .

$$\vec{u}(i, j) \triangleq [a^{(k)}(i, j), 1 - a^{(k)}(i, j)], \quad (10)$$

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 [47], and design a *connectivity metric* $h(i, j)$ for DES rides between regions i and j as

$$h(i, j) \triangleq \frac{1}{1 + \exp(-\vec{u}(i, j) \cdot \vec{u}(j, i))}, \quad (11)$$

where the dot product of the two vectors, $\vec{u}(i, j) \cdot \vec{u}(j, i)$, 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

$$\mathbf{A}_C(i, j) = \mathbf{A}_C^*(i, j) \cdot h(i, j). \quad (12)$$

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. We adopt $h(i, j) \in (0, 1)$ as a weight parameter imposed upon the adjacency/correlation between regions to adjust their relative importance.

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

$$\mathbf{A} = [\mathbf{A}_D, \mathbf{A}_P, \mathbf{A}_C]. \quad (13)$$

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 input matrix \mathbf{A}_* ($*$ denotes D , P or C in Eq. (13)), where

$$\mathbf{A}'_*(i, :) \triangleq \begin{cases} \mathbf{0}, & \text{if region } i \text{ is deactivated;} \\ \mathbf{A}_*(i, :) \cdot \mathbf{1}, & \text{otherwise.} \end{cases} \quad (14)$$

And $\mathbf{A}'(:, i)$ also applies due to symmetry.

From the government or city planners, we can also obtain the newly reconfigured regions as $\mathbf{V}^{(k+1)}$. Then, based on $\mathbf{V}^{(k+1)}$ and Eq. (14), we activate the regions in the expansion and deactivate the removed ones, and have the resultant masked correlations as

$$\mathbf{A}'_D = m^{(k)}(\mathbf{A}_D), \quad \mathbf{A}'_P = m^{(k)}(\mathbf{A}_P), \quad \mathbf{A}'_C = m^{(k)}(\mathbf{A}_C). \quad (15)$$

These masked correlations will be fed to the graph capsule network in Section 5.2.

5.2 Graph Capsule Designs

The core framework of the graph capsule network STGCNet within GCScoot includes the following three major designs: 1) *multi-layer graph convolutions*: which consist of multiple graph convolution layers capturing multi-scale graph features; 2) *multi-graph convolution for temporal modeling*: which incorporates multiple DES network graph in the past intervals; 3) *capsule routing*: which consists of primary and routing capsules to further derive fine-grained graph features. The core structure is illustrated in Fig. 24.

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 [48]. 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.,

$$\mathbf{Z}^{(l+1)} \triangleq \sigma(f(\mathbf{A})\mathbf{Z}^{(l)}\mathbf{W}^{(l)}), \quad (16)$$

where $f(\cdot)$ is the $N \times N$ matrix generated via aggregation operation, and $\mathbf{W}^{(l)} \in \mathbb{R}^{d \times d'}$ is a trainable weight matrix as a channel filter.

Since multiple correlation matrices are applied as shown in Eq. (13), 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, i.e.,

$$\mathbf{Z}_j^{(l+1)} \triangleq \sigma \left(\sum_{\tilde{\mathbf{A}}_* \in \{\tilde{\mathbf{A}}'_D, \tilde{\mathbf{A}}'_P, \tilde{\mathbf{A}}'_C\}} \sum_i \tilde{\mathbf{A}}_* \mathbf{Z}_i^{(l)} \mathbf{W}_{ij}^{(l)} \right), \quad (17)$$

where the symmetric normalized Laplacian $f(\cdot)$ [48] is applied upon each of the masked correlations $\mathbf{A}'_D, \mathbf{A}'_P, \mathbf{A}'_C$ in Eq. (15) as

$$\tilde{\mathbf{A}}_* = f(\mathbf{A}'_*) = \tilde{\mathbf{D}}_*^{-\frac{1}{2}} \mathbf{A}'_* \tilde{\mathbf{D}}_*^{-\frac{1}{2}}, \quad (18)$$

and the degree matrix is

$$\tilde{\mathbf{D}}_*(i, i) = \sum_j \mathbf{A}'_*(i, j). \quad (19)$$

Multi-Graph Convolution for Temporal Modeling. To further enhance the accuracy and robustness of GCScoot, we have designed a multi-graph mechanism within the graph neural network structure of GCScoot. Multiple DES network graphs in the past W' intervals are jointly considered to further learn the temporal dependency for the target interval estimation.

Specifically, we further incorporate the multiple DES network graphs in the sliding time window as the input at the first layer, which models the dependency as

$$\mathbf{Z}^{(1)} = \sigma \left(\bigsqcup_{w \in \{1, \dots, W'\}} \sum_{\tilde{\mathbf{A}}_* \in \{\tilde{\mathbf{A}}'_D, \tilde{\mathbf{A}}'_P, \tilde{\mathbf{A}}'_G\}} \tilde{\mathbf{A}}_* \mathbf{Z}^{(0)} \mathbf{W}^{(0)} \right), \quad (20)$$

where \bigsqcup denotes the aggregation operation for the multiple input graphs, and we adopt summation operation. Here $\mathbf{Z}^{(0)} \in \mathcal{R}^{N \times 2}$ represents the input DES flow features, the initial input of graph feature representation to the entire GCScoot. Multi-graph convolution helps GCScoot better model the temporal relationships between the graphs compared to previous single settings [36]. Our experimental studies in Section 7 will further demonstrate the performance improvements.

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

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 [5]. To address this problem, the capsule network [5], [49], [50] 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.,

$$[\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \dots, \mathbf{Z}^{(L-1)}, \mathbf{Z}^{(L)}], \quad (21)$$

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 \mathbf{W}^c be an $N^P \times N^R$ weight matrix, \mathbf{W}_{ij}^c be the weight parameter of the link between 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 \mathbf{s}_j to a capsule j there is formally given by

$$\mathbf{s}_j = \sum_i e_{ij} \hat{\mathbf{u}}_{j|i}. \quad (22)$$

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

$$e_{ij} = \frac{\exp(b_{ij})}{\sum_z \exp(b_{iz})}, \quad (23)$$

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

$$\hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}^c \mathbf{u}_i. \quad (24)$$

The resultant vector output, denoted as \mathbf{v}_j , from the capsule j is then given by a squashing function to differentiate the long and short vector inputs, i.e.,

$$\mathbf{v}_j = \frac{\|\mathbf{q}_j\|^2}{1 + \|\mathbf{q}_j\|^2} \cdot \frac{\mathbf{q}_j}{\|\mathbf{q}_j\|}, \quad (25)$$

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.,

$$b'_{ij} = b_{ij} + \hat{\mathbf{u}}_{j|i} \cdot \mathbf{v}_j. \quad (26)$$

Through iterations with the Eqs. (22)–(26), the graph capsule network learns the structured features within the input DES networks.

6 MODULE INTEGRATION & COLD-START

Given the core models, we now present the module integration in Section 6.1, followed by the model cold-start based on the clustered regions and virtual data generation in Section 6.2.

6.1 GCScoot Module Integration

The important modules in the GCScoot framework are integrated as follows.

Graph Capsule Neural Network for DES Networks. We summarize the architecture of the multi-layer graph capsule neural network as in Fig. 24, 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 \mathbf{A} and $N \times 2$ flow matrix $\mathbf{F}^{(k)}$, which serves as $\mathbf{Z}^{(0)}$. The lower output $\mathbf{Z}^{(l-1)}$ is fed to the upper layer l .

As illustrated in Fig. 24, we stack multiple graph convolution layers in order to enhance the receptive field of the

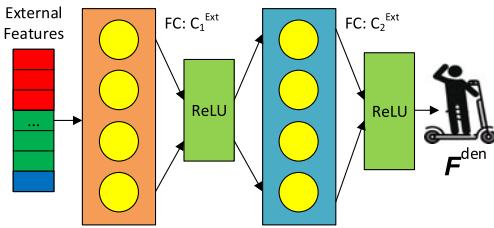


Fig. 25. Dense neural network for external factor processing.

graph convolution operations and better capture the latent spatio-temporal correlations.

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^{Den} and C_2^{Den}) processes the outputs from the routing capsules and maps them back to the reconfigured DES flows $\mathbf{F}^{\text{cap}} \in \mathbb{R}^{N' \times 2}$.

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. 22), day of a week and hour of a day are concatenated together into a vector. The vector is then fed to a multi-layer fully-connected neural network in Fig. 25 in order to integrate external factors related to DES mobility. Two fully-connected neural networks (with output dimensions of C_1^{Ext} and C_2^{Ext}) with ReLU activation function between them are adopted here. This component returns $\mathbf{F}^{\text{den}} \in \mathbb{R}^{N' \times 2}$.

Finally, given the predictions of reconfigured DES flows \mathbf{F}^{cap} and \mathbf{F}^{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 returns the final results for the city planners and service providers. The proposed model can be dynamically updated over the time with new deployment data if available.

6.2 Region Clustering & Virtual Data Generation

For regions without any historical datasets, how to enhance the training and prediction accuracy remains a critical issue. To further mitigate the cold-start influence and enhance the model adaptability, we have designed an efficient and effective model cold-start approach with the short-term virtual trip data generated from the regions with similar neighborhood trip patterns.

Specifically, the designs of dataset generation and model cold-start consist of the following steps:

- 1) *Region clustering:* Geographically close regions with the similar neighborhood spatial features tend to have similar DES flows. So, we consider finding the spatially similar old regions for each new one as the basis for virtual trip generation. We leverage the region-to-region POI similarities (Eq. (6)) as the spatial correlations between the new regions and those old ones, and cluster the regions into multiple region groups. Then, we generate the virtual trips of a new region based upon its peer old regions with historical datasets in the same cluster.

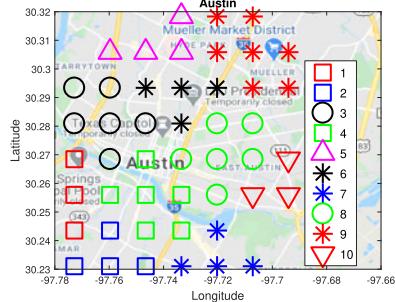


Fig. 26. Clustered DES regions in Austin, TX (by Aug, 2018).

In particular, the POI similarities of the regions are fed to the affinity propagation clustering algorithm [51] which does not need specification of the number of clusters like the conventional k-means clustering [52]. We further illustrate the clustered regions in Austin (August, 2018) and Chicago (July, 2019) in Figs. 26 and 27, where 10 and 12 clusters of regions have been respectively discovered and generated.

- 2) *Virtual data generation:* We take into account the spatial correlations, since geographically close regions tend to have similar flow patterns. For each new region i at time interval k , each flow record of the virtual time series (pick-ups or drop-offs) is generated based on those of the N' ($N' \ll N$) disjoint regions with the shortest geographic distances from it within the cluster, i.e.,

$$\tilde{\mathbf{F}}_i^{(k)} = \sum_{j=1}^{N'} \omega_j \mathbf{F}_j^{(k)}, \quad k \in \{1, \dots, K\}, \quad (27)$$

where K is the number of time intervals in a time window before the reconfiguration, and the normalized weight ω_j is given by

$$\omega_j = \left(\frac{1}{\text{dist}(\mathbf{r}_i, \mathbf{r}_j)} \right) / \left(\sum_{j=1}^{N'} \frac{1}{\text{dist}(\mathbf{r}_i, \mathbf{r}_j)} \right), \quad (28)$$

where $\text{dist}(\mathbf{r}_i, \mathbf{r}_j)$ is the geographic distance between region i and its neighbor j (unit: km). In this paper, we consider N' as the number of other regions in the assigned cluster with the minimum value of 5. We adopt the virtual trips in a short-term period (say, 12 hours; $K = 12$ for Chicago or $K = 24$ for

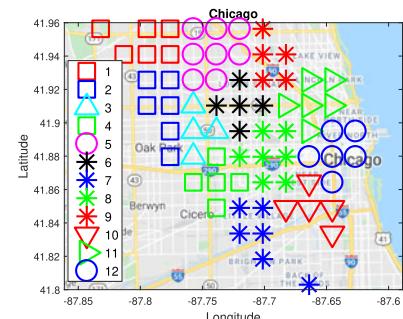


Fig. 27. Clustered DES regions in Chicago, IL (by July, 2019).

- other datasets) before the reconfiguration for the model initialization of GCScoot, considering the data recency and computational efficiency.
- 3) *Model training initialization:* Given the short-term virtual trips, we combine them with the historical trips, and further train the GCScoot model before the reconfiguration. We activate the elements in \mathbf{A} (Eq. (14)) when training GCScoot upon the generated data in order to enable the graph convolution to characterize the region-to-region correlations.

The DES service provider can in practice conduct the above processing before each reconfiguration. When the reconfigured regions are decided and the real historical trips are fed to replace the virtual ones, GCScoot can be further updated with the new data for the following forecast.

7 EXPERIMENTAL EVALUATION

We first describe the experimental settings in Section 7.1 and then present the experimental results in Section 7.2.

7.1 Experimental Settings

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

- 1) 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).
- 2) LSTM: which estimates the DES flows based on the long short-term memory.
- 3) RNN: which predicts the time series of the DES flows based on recurrent neural network [53].
- 4) STCNN: which models the DES flows via spatio-temporal convolutional neural network [23].
- 5) MGNC: which estimates the flows through multi-graph convolutional neural network [27].
- 6) GWN: which leverages the graph wavenet for the traffic flow prediction [29].
- 7) STGCN: which leverages the spatial and temporal graph convolutional neural network [25].
- 8) MTL: which adaptively predicts the DES flows via spatio-temporal convolutional neural network and meta-learning [30], [54].
- 9) FA+CNN: which adapts the domains of the previous DES flows with factor analysis and predicts the transformed traffics with convolutional neural network [34].
- 10) DANN: which predicts and adapts to the dynamic flows via convolutional neural network with domain adaptation [35], [55].
- 11) MSGN: which adaptively learns and forecasts the traffic flows with the multi-scale graph neural network [56].

We also compare the performance of the current version (labeled as GCScoot-2.0) with the GCScoot in the conference version [36] (denoted as GCScoot-1.0), to demonstrate the difference and improvement. Unless otherwise stated, we use GCScoot to represent the latest version.

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. For Austin, Louisville and Minneapolis, we adopt a temporal discretization interval of 30 min as it is the minimum interval for the meteorological datasets, and set $w = 12$ for Eq. (8) which is equal to 6 hours. For Chicago, we are given 60 min granularity processed by the Chicago Department of Transportation (DOT), and hence 12-hour trips are used for Eq. (8). 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 [30], [57], we evaluate the grid settings and discretize the maps of Austin, Louisville, Minneapolis and Chicago into 32×32 , 25×13 , 11×11 and 15×15 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, and leverage $W' = 6$ latest consecutive DES network graphs to incorporate the temporal dependency. 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 STGCNet are set as

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

We set the default number of dynamic routing in STGCNet 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 at least 10 percent of deployment regions in the 7 days onward have changed (introduced or removed). In total, we have identified 21 reconfigurations in Austin, 22 in Louisville, 9 in Minneapolis, 4 in Chicago in the four datasets. For each city, for GCScoot we conduct the model sensitivity studies upon the first 30 days' samples (first 80 percent for initial model training and 20 percent for validation), and use the rest for overall prediction performance comparison. 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 comprehensively evaluate the performance of all schemes based on the root mean square error (RMSE)

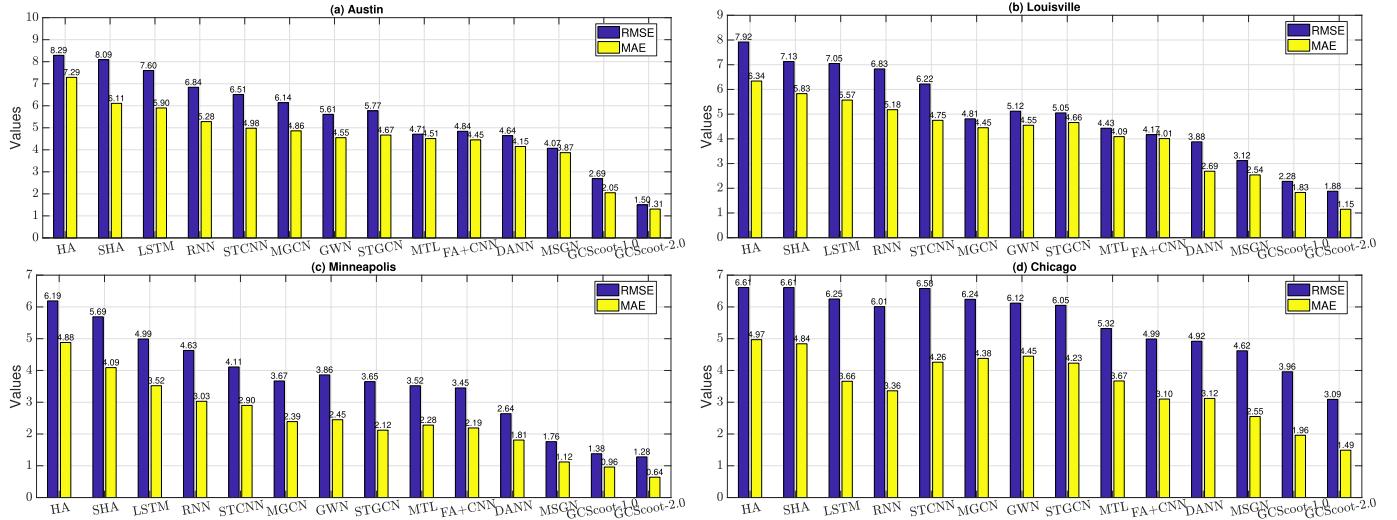


Fig. 28. Overall performance of all schemes in the four datasets: (a) Austin; (b) Louisville; (c) Minneapolis; (d) Chicago.

$$\text{RMSE} = \sqrt{\frac{1}{KN} \sum_{i=1}^N \sum_{k=1}^K (\mathbf{F}_i^{(k)} - \hat{\mathbf{F}}_i^{(k)})^2}, \quad (29)$$

mean absolute error (MAE)

$$\text{MAE} = \frac{1}{KN} \sum_{i=1}^N \sum_{k=1}^K |\mathbf{F}_i^{(k)} - \hat{\mathbf{F}}_i^{(k)}|, \quad (30)$$

and error rate (ER) (which follows [58]; used in sensitivity studies)

$$\text{ER} = \frac{1}{K} \sum_{k=1}^K \frac{\sum_{i=1}^N |\mathbf{F}_i^{(k)} - \hat{\mathbf{F}}_i^{(k)}|}{\sum_{i=1}^N \mathbf{F}_i^{(k)}}, \quad (31)$$

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

7.2 Experimental Results

Prediction Performance Comparison. We present the overall prediction results and evaluate the performance variations under different data settings.

Overall Prediction Performance. We first present in Fig. 28 the experimental results of performance on each of the four datasets. Without proper spatial feature adaptation, conventional time-series schemes like HA, SHA, LSTM and RNN cannot accommodate the spatial correlations between regions as well as their reconfiguration, resulting in lower prediction accuracy. STCNN, MGNC, GWN and STGCN focus on the influence of spatial features upon the temporal flows. However, these approaches have not considered the dynamically reconfigured regions in their settings, and hence cannot adapt well to the reconfigured regions. FA+CNN and DANN focus on the domain studies, and MSGN considers the region-to-region connections. GCScoot achieves higher accuracy than the above three approaches, thanks 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 these schemes for different datasets. All of the schemes suffer

higher errors in Chicago and Austin than other two datasets, due to higher volumes of usage and more complex DES usage. GCScoot-2.0 makes at least 63.14, 39.74, 27.27 and 33.12 percent improvements over the state-of-the-arts *w.r.t.* the four datasets. Compared to GCScoot-1.0, the proposed new designs and components are shown to improve the accuracy by 7.25 to 44.24 percent in all the four datasets we have evaluated.

Predictions on New & Existing Regions. By focusing on the new and existing ($\mathbf{V} \cap \mathbf{V}'$) regions after the reconfigurations, we further show in Table 2 the performance on the four datasets. Specifically, we show the RMSE of the schemes GCScoot-2.0, GCScoot-1.0, MSGN, DANN and FA+CNN. Forecasting dynamic flows regarding the new regions is more challenging due to the absence of historical data. Both GCScoot-2.0 and GCScoot-1.0 are shown to outperform the other schemes in predicting the flows from both the new and existing regions, demonstrating GCScoot's high adaptability to the DES reconfiguration. Thanks to the proposed new multi-graph and virtual data designs, GCScoot-2.0 improves the accuracy over GCScoot-1.0 for the new regions by 10.90 percent on average in the datasets we have studied, and outperforms the state-of-the-arts by at least 23.19 percent based on the results.

Predictions on Rush Hours and Weekends. 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 GCScoot-2.0, GCScoot-1.0 and the other four schemes for the four 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 both GCScoot-2.0 and GCScoot-1.0 outperform the other state-of-the-arts in predicting the flows due to their higher adaptivity to the DES traffic flows. On average, GCScoot-2.0 outperforms GCScoot-1.0 by 14.72 percent in terms of predicting rush-hour flows, and by 17.47 percent in terms of weekend distributions.

Model Sensitivity Analysis. After presenting the overall performance upon the four datasets, we evaluate the model sensitivity of GCScoot (we focus on GCScoot-2.0 in the

TABLE 2
RMSE Regarding the New and Existing Regions During Reconfiguration Periods of the Four Cities

Schemes	Austin		Louisville		Minneapolis		Chicago	
	New	Existing	New	Existing	New	Existing	New	Existing
GCScoot-2.0	3.04	1.73	3.29	1.71	2.14	1.26	3.71	2.75
GCScoot-1.0	3.88	2.04	3.45	1.88	2.18	1.31	4.39	2.98
MSGN	5.90	3.87	5.13	2.89	3.43	1.52	4.83	3.23
DANN	6.11	4.44	6.54	2.73	3.52	2.19	5.27	3.81
MTL	7.30	4.26	6.11	3.87	5.29	3.45	6.13	4.50
FA+CNN	7.51	4.82	6.34	3.99	5.18	3.14	6.92	4.73

TABLE 3
RMSE on the Rush Hours & Weekends at the Four Cities

Schemes	Austin		Louisville		Minneapolis		Chicago	
	RH	Weekends	RH	Weekends	RH	Weekends	RH	Weekends
GCScoot-2.0	2.84	2.22	2.85	2.45	1.52	1.42	3.74	4.61
GCScoot-1.0	3.12	2.63	2.92	3.03	1.81	2.04	5.46	4.40
MSGN	5.83	5.57	5.26	5.89	3.35	3.74	5.64	5.77
DANN	5.54	5.32	6.55	6.73	3.66	3.46	5.17	7.33
MTL	5.21	5.74	6.22	6.83	5.08	4.72	6.21	6.13
FA+CNN	6.11	5.90	6.18	6.22	5.84	5.14	7.32	8.19

following due to its better performance) to variations in spatial/temporal/external factors/components, dynamic routings, grid sizes, multi-graph mechanism and number of samples (in terms of days) for model training. This sensitivity has been studied on the first 30 days' samples (total 720 time intervals for Chicago and 1,440 for others) for each dataset, and we show the results upon the validation data (Section 7.1).

Important Modules. Taking Austin dataset as an example, we first show in Fig. 29 GCScoot's performance (RMSE and MAE) 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 GCScoot (w/ all) and its variations: without distances (w/o dist), flow correlations (w/o flow corr), POIs (w/o POIs), weather (w/o weather), fusion with dense network processing the external factors (w/o ext), multi-graph mechanism (w/o multi)

and virtual data generation (w/o virtual). By incorporating the aforementioned factors, GCScoot is shown to be able to adaptively predict the reconfigured DES flows.

Dynamic Routing. We also present in Fig. 30 the performance (RMSE and MAE) of GCScoot versus the number of dynamic routings. More dynamic routings generally help GCScoot 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.

Map Discretization. We evaluate GCScoot's performance (RMSE, MAE and ER) while varying the grid sizes and map discretization in Fig. 31. Specifically, we evaluate the map discretization of 16×16 , 32×32 , 48×48 and 64×64 . We can observe that as the number of grids increases, the errors of GCScoot tend to grow, mainly because the matrix

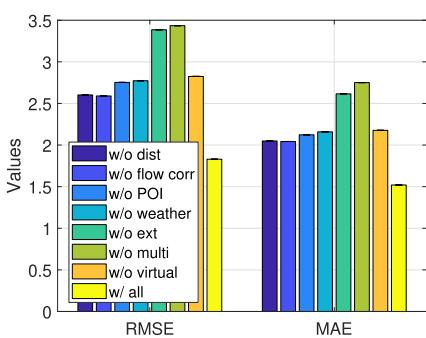


Fig. 29. Performance of GCScoot w/ and w/o the components & designs (Austin).

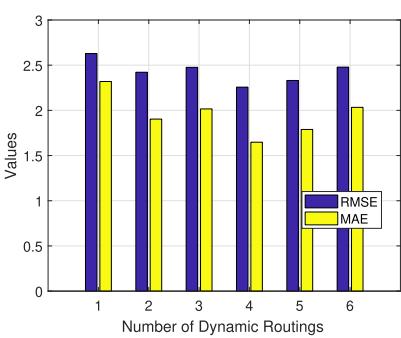


Fig. 30. Performance of GCScoot versus numbers of dynamic routings (Austin).

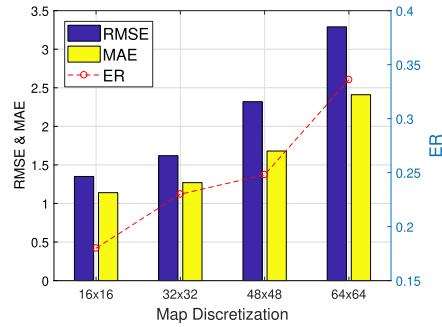


Fig. 31. Performance of GCScoot versus the number of grids (Austin).

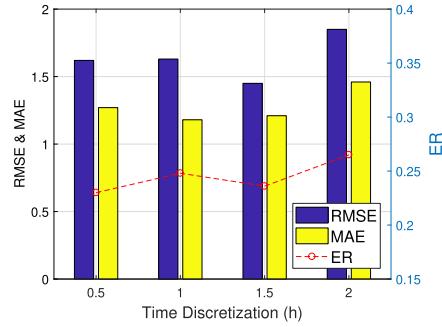


Fig. 32. Performance of GCScoot versus time discretization (Austin).

representing the region-to-region correlation grows significantly with the number of grids, making it difficult for the model to learn the flow features further. To enable efficient computation and maintain the granularity of the estimation, we choose 32×32 for Austin.

Time Discretization. We have further conducted experimental studies on the Austin dataset regarding the effect of time discretization in Fig. 32. We set each interval as 0.5, 1, 1.5 and 2 h, and show GCScoot's performance in terms of RMSE, MAE and ER. Given larger time discretization, a slight increase of errors can be observed, while, overall, GCScoot is less sensitive to the time discretization than the map discretization in Fig. 31.

Multi-Graph Settings. Taking Austin and Chicago as examples, we further show the performance (RMSE as well as ER) of GCScoot's multi-graph mechanism versus the number of flows (W' in Eq. (20)) in Fig. 33. We can observe that the accuracy of GCScoot increases with respect to the number of flows, mainly because more flow features can be learned from more historical data. However, the improvement diminishes

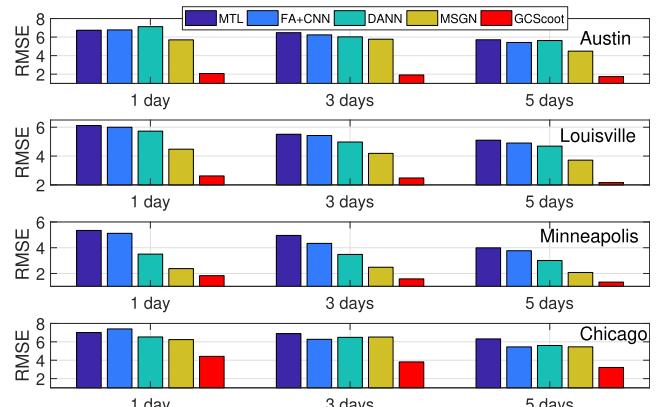


Fig. 34. Performance of the schemes versus the numbers of training samples in the four datasets.

after a few more time intervals (say, 6 or 7 in our provided cases) are fed, which is likely due to more random trip patterns included in a longer period of time. Therefore, we set $W' = 6$ consecutive time intervals by default in our experimental studies.

Number of Training Samples. We show in Fig. 34 the performance (RMSEs) of GCScoot, MSGN, DANN, MTL and FA+ CNN given only 1, 3 and 5 days of the training samples with respect to each of the four datasets. Fewer training data poses more challenges to all the schemes, which also represents the common practice when only a few pilot studies have been done before reconfiguration. Via more comprehensive feature learning based on the graph capsule neural network, GCScoot 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.

Adaptive Cold-Start Mechanism. We have conducted further studies regarding the performance (RMSE and MAE) of GCScoot while varying the length of virtual trips (4, 8, 12 and 16 h) in the cold-start mechanism. One can see from Fig. 35 that the RMSE and MAE of GCScoot tend to decrease as more time intervals of virtual trips are involved. However, as more intervals are included, the long-term virtual trips may not necessarily characterize the short-term dynamics due to the last-mile nature of the DES flows. Based on this trade-off, we set 12 hours in our default settings.

Effect of Reconfiguration Ratios. We have also parsed the dataset and shown the performance (RMSE and MAE) of GCScoot in Fig. 36 under different reconfiguration ratios

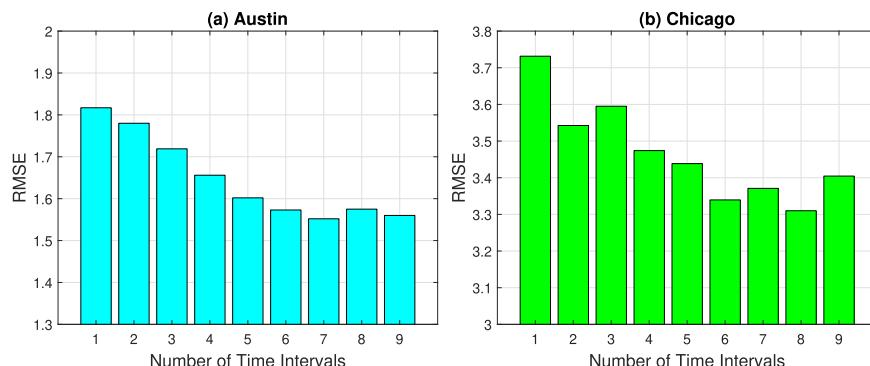


Fig. 33. Performance of GCScoot versus numbers of flows: (a) Austin; (b) Chicago.

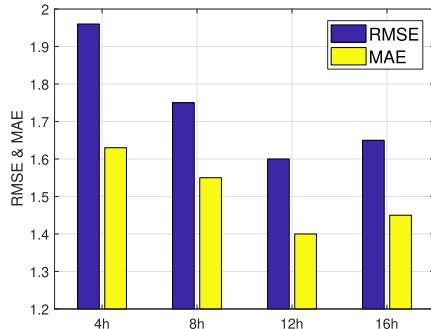


Fig. 35. Sensitivity of adaptive cold-start mechanism (Austin).

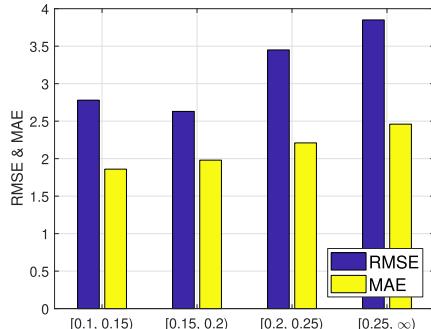


Fig. 36. Effect of reconfiguration ratios (Austin).

(Section 7.1). In particular, we studied the RMSE and MAE in the reconfigured regions in terms of following ratios: $[0.1, 0.15]$, $[0.15, 0.2]$, $[0.2, 0.25]$ and $[0.25, \infty)$ in Austin. One can see that the prediction error generally increases as the reconfiguration ratio rises. In practice, we have observed that more than 81 percent of the reconfiguration operations have ratios of less than 0.25, and hence GCScoot can maintain overall reasonable prediction accuracy in real-world deployment.

Training Time Comparison. We briefly summarize and discuss the training time and computational efficiency (based on our machine in Section 7.1). The training time of GCScoot for Austin, Louisville, Minneapolis and Chicago is around 5, 1.1, 0.5 and 0.68 h. Compared to GCScoot in these four cities, MTL takes around 4, 2.5, 0.8 and 0.8 h; FA+ CNN takes around 3.5, 0.8, 0.2 and 0.3 h due to the simplicity of the model; DANN takes around 5.1, 0.8, 0.5 and 0.5 h; MSGN takes around 5.5, 1, 0.5 and 0.8 h. Due to larger DES volumes, all schemes generally need a longer training time for the Austin and Louisville datasets than the other two. Further model efficiency enhancement and training acceleration of GCScoot will be part of our future work.

Comparison With Bike Sharing. We have briefly compared the DES and the bike sharing system (B-Cycle) in Austin. We conduct the experimental studies upon the DES and bike sharing datasets collected in May 2019. The spatial and temporal discretizations of bike sharing flows follow the same settings as DES (Section 7.1). From Fig. 37, we can see that predicting DES flows is overall more challenging compared with bike sharing, due to the higher degree of freedom in maneuvering (Section 4.1) and user behavioral difference (say, preferred escooters for leisure use [59], [60]). Further investigation on user behaviors will be considered in our future work.

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on April 11, 2024 at 01:06:59 UTC from IEEE Xplore. Restrictions apply.

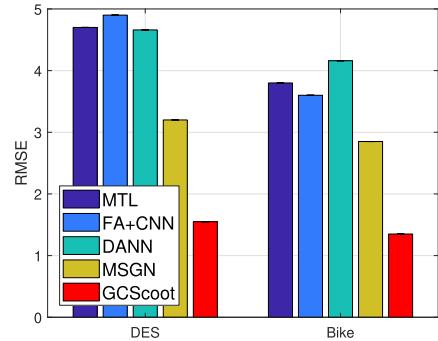


Fig. 37. Performance comparison on DES & bike sharing (Austin).

8 DEPLOYMENT DISCUSSION

In this section, we further provide some discussion regarding the deployment, and provide future research direction:

Inclusion of Other Data Sources. Due to the resource constraints and early stage of DES deployment, there exist many other factors influencing the DES distribution [61], including demographic distributions, which have not been discussed in this work [11]. With the increasing community acceptance and deployments, the related DES studies are expected to be supported by more available data sources. While our experimental studies focus on spatial, temporal and external factors including meteorological data, GCScoot is general enough to accommodate other factors (including satellite images [34], local traffic [62] and demographic information) to enhance the accuracy further.

Urban Environment Impact & Abnormal City Events. While our current studies consider how the external factors like POI region correlations, weather and time events like holidays influence scooter traffic, it would be interesting to study the (positive/negative) effects of DES deployment upon the urban environment [2], including pollutant levels of road network [63], traffics of other transportation systems, resultant congestion levels of sidewalks and vehicle parking availability [41]. Furthermore, abnormal city events, including illegal parking, crimes or city shutdown during pandemics like COVID-19,⁶ could significantly impact DES deployment, which requires multi-disciplinary efforts as related sources become available [64].

9 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 GCScoot 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, with multi-graph designs fusing the flows of consecutive time intervals, which comprehensively and adaptively forecasts

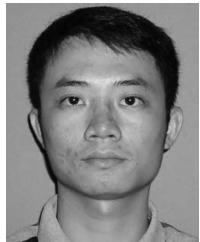
6. <https://www.theverge.com/2020/3/20/21188119/electric-scooter-coronavirus-bird-lime-spin-suspend-bikes>, Accessed: May-2020.

the resultant e-scooter flows given the altered deployment regions. To enhance GCScoot's adaptability towards the new regions, we have further designed an efficient region clustering and model cold-start approach. We have conducted extensive experimental evaluation upon four different e-scooter datasets in four populous US cities which consist of total more than 3 million trips, showing that GCScoot outperforms state-of-the-arts and is effective and accurate in forecasting e-scooter mobility and flows after reconfiguration.

REFERENCES

- [1] Electric Scooter Sharing Market in US and Europe 2019–2024, 2019. [Online]. Available: <https://mobilityforesights.com/product/e-scooter-sharing-market-report/>
- [2] C. S. Smith and J. P. Schwieterman, "E-Scooter scenarios: Evaluating the potential mobility benefits of shared dockless scooters in chicago," *Final Report of Chaddick Institute for Metropolitan Development Chicago*, 2018.
- [3] A. Giambrone, "D.C. to expand the number of scooters and dockless bikes allowed in 2019," 2018. [Online]. Available: <https://dc-curbed.com/2018/11/9/18076394/dc-scooters-dockless-bikes-transportation-cycling-ddot>
- [4] C. Hampel, "Ford's e-scooter sharing service Spin expands to Europe," 2020. [Online]. Available: <https://www.electrive.com/2020/02/27/fords-e-kick-scooter-sharing-service-spin-expands-to-europe/>
- [5] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3856–3866.
- [6] Z. Yang, J. Hu, Y. Shu, P. Cheng, J. Chen, and T. Moscibroda, "Mobility modeling and prediction in bike-sharing systems," in *Proc. Annu. Int. Conf. Mobile Syst. Appl. Services*, 2016, pp. 165–178.
- [7] L. Lin, J. Li, F. Chen, J. Ye, and J. Huai, "Road traffic speed prediction: A probabilistic model fusing multi-source data," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1310–1323, Jul. 2018.
- [8] Z. Fang, L. Huang, and A. Wierman, "Prices and subsidies in the sharing economy," in *Proc. Int. Conf. World Wide Web*, 2017, pp. 53–62.
- [9] S. Wang, T. He, D. Zhang, Y. Liu, and S. H. Son, "Towards efficient sharing: A usage balancing mechanism for bike sharing systems," in *Proc. Int. Conf. World Wide Web*, 2019, pp. 2011–2021.
- [10] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 468–478, Mar. 2020.
- [11] S. He and K. G. Shin, "Information fusion for (re)configuring bike station networks with crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, early access, Apr. 27, 2020, doi: [10.1109/TKDE.2020.2991000](https://doi.org/10.1109/TKDE.2020.2991000).
- [12] M. Liu *et al.*, "Analysis of E-scooter trips and their temporal usage patterns," *ITE J.*, vol. 89, no. 6, pp. 44–49, 2019.
- [13] G. Brandstätter, M. Kahr, and M. Leitner, "Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand," *Transp. Res. B: Methodol.*, vol. 104, pp. 17–35, 2017.
- [14] M. Li *et al.*, "Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning," in *Proc. Int. Conf. World Wide Web*, 2019, pp. 983–994.
- [15] S. He and K. G. Shin, "Spatio-temporal capsule-based reinforcement learning for mobility-on-demand network coordination," in *Proc. Int. Conf. World Wide Web*, 2019, pp. 2806–2813.
- [16] J. Hu, Z. Yang, Y. Shu, P. Cheng, and J. Chen, "Data-driven utilization-aware trip advisor for bike-sharing systems," in *Proc. IEEE Int. Conf. Data Mining*, 2017, pp. 167–176.
- [17] S. He and K. G. Shin, "(Re)Configuring bike station network via crowdsourced information fusion and joint optimization," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2018, pp. 1–10.
- [18] T. He *et al.*, "Detecting vehicle illegal parking events using sharing bikes' trajectories," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 340–349.
- [19] S. He and K. G. Shin, "Towards fine-grained flow forecasting: A graph attention approach for bike sharing systems," in *Proc. Int. Conf. World Wide Web*, 2020, pp. 88–98.
- [20] S. He and K. G. Shin, "Spatio-temporal capsule-based reinforcement learning for mobility-on-demand coordination," *IEEE Trans. Knowl. Data Eng.*, early access, May 04, 2020, doi: [10.1109/TKDE.2020.2992565](https://doi.org/10.1109/TKDE.2020.2992565).
- [21] L. Lin, Z. He, and S. Peeta, "Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach," *Transp. Res. C: Emerg. Technol.*, vol. 97, pp. 258–276, 2018.
- [22] X. Kong, M. Li, T. Tang, K. Tian, L. Moreira-Matias, and F. Xia, "Shared subway shuttle bus route planning based on transport data analytics," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1507–1520, Oct. 2018.
- [23] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, 2017, Art. no. 818.
- [24] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.
- [25] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3634–3640.
- [26] X. Wang *et al.*, "Traffic flow prediction via spatial temporal graph neural network," in *Proc. Int. Conf. World Wide Web*, 2020, pp. 1082–1092.
- [27] D. Chai, L. Wang, and Q. Yang, "Bike flow prediction with multi-graph convolutional networks," in *Proc. ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.*, 2018, pp. 397–400.
- [28] X. Geng *et al.*, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3656–3663.
- [29] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 1907–1913.
- [30] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities: A meta-learning approach for spatial-temporal prediction," in *Proc. Int. Conf. World Wide Web*, 2019, pp. 2181–2191.
- [31] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang, "Cross-city transfer learning for deep spatio-temporal prediction," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 1893–1899.
- [32] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1720–1730.
- [33] L. Pan, Q. Cai, Z. Fang, P. Tang, and L. Huang, "A deep reinforcement learning framework for rebalancing dockless bike sharing systems," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 1393–1400, 2019.
- [34] Z. Liu, Y. Shen, and Y. Zhu, "Inferring dockless shared bike distribution in new cities," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2018, pp. 378–386.
- [35] Z. Liu, Y. Shen, and Y. Zhu, "Where will dockless shared bikes be stacked? — Parking hotspots detection in a new city," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 566–575.
- [36] S. He and K. G. Shin, "Dynamic flow distribution prediction for urban dockless E-scooter sharing reconfiguration," in *Proc. Int. Conf. World Wide Web*, 2020, pp. 133–143.
- [37] J. Zhang, X. Pan, M. Li, and P. S. Yu, "Bicycle-sharing systems expansion: Station re-deployment through crowd planning," in *Proc. ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.*, 2016, pp. 2:1–2:10.
- [38] OpenStreetMap, 2019. [Online]. Available: www.openstreetmap.org
- [39] Dataset of Austin B-Cycle Trips, 2019. [Online]. Available: <https://data.austintexas.gov/Transportation-and-Mobility/Austin-B-Cycle-Trips/tfyh-5r8s>
- [40] Y. Duan and J. Wu, "Optimizing rebalance scheme for dock-less bike sharing systems with adaptive user incentive," in *Proc. 20th IEEE Int. Conf. Mobile Data Manage.*, 2019, pp. 176–181.
- [41] SFMTA: Powered Scooter Share Permit and Pilot Program at San Francisco, 2020. [Online]. Available: <https://www.sfmta.com/projects/powerd-scooter-share-permit-and-pilot-program>
- [42] Austin Micromobility, 2019. [Online]. Available: <https://austintexas.gov/micromobility>
- [43] Street Centerline, Austin, 2019. [Online]. Available: <https://data.austintexas.gov/Locations-and-Maps/Street-Centerline/m5w3-uea6>
- [44] Street Centerline, Louisville, 2019. [Online]. Available: <https://data.louisvilleky.gov/dataset/street-centerline>
- [45] Street Centerline, Minneapolis, 2019. [Online]. Available: http://opendata.minneapolismn.gov/datasets/e68d01d782c04d88876bb51e1c40702_0

- [46] Street Centerline, Chicago, 2020. [Online]. Available: <https://data.cityofchicago.org/Transportation/Street-Center-Lines/6imu-mäu>
- [47] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. Int. Conf. World Wide Web*, 2015, pp. 1067–1077.
- [48] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.
- [49] S. Verma and Z.-L. Zhang, "Graph capsule convolutional neural networks," in *Proc. Joint ICML IJCAI Workshop Comput. Biol.*, 2018.
- [50] X. Zhang and L. Chen, "Capsule graph neural network," in *Proc. Int. Conf. Learn. Representations*, 2019.
- [51] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.
- [52] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.
- [53] Y. Pan, R. C. Zheng, J. Zhang, and X. Yao, "Predicting bike sharing demand using recurrent neural networks," *Procedia Comput. Sci.*, vol. 147, pp. 562–566, 2019.
- [54] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.
- [55] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.
- [56] M. Luo, B. Du, K. Klemmer, H. Zhu, H. Ferhatosmanoglu, and H. Wen, "D3P: Data-Driven demand prediction for fast expanding electric vehicle sharing systems," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 1, 2020, Art. no. 21, doi: [10.1145/3381005](https://doi.org/10.1145/3381005).
- [57] Y. Liang et al., "UrbanFM: Inferring fine-grained urban flows," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 3132–3142.
- [58] Y. Li, Y. Zheng, H. Zhang, and L. Chen, "Traffic prediction in a bike-sharing system," in *Proc. ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.*, 2015, pp. 33:1–33:10.
- [59] G. McKenzie, "Spatiotemporal comparative analysis of scooter-share and bike-share usage patterns in Washington, DC," *J. Transport Geogr.*, vol. 78, pp. 19–28, 2019.
- [60] R. Zhu, X. Zhang, D. Kondor, P. Santi, and C. Ratti, "Understanding spatio-temporal heterogeneity of bike-sharing and scooter-sharing mobility," *Comput. Environ. Urban Syst.*, vol. 81, 2020, Art. no. 101483.
- [61] K. Schneider, "Wheels for all: Ensuring equitable access to dockless mobility in Los Angeles," *UCLA: Institute of Transportation Studies*, 2019.
- [62] S. He and K. G. Shin, "Spatio-temporal adaptive pricing for balancing mobility-on-demand networks," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 4, Jul. 2019, Art. no. 39.
- [63] J. Hollingsworth, B. Copeland, and J. X. Johnson, "Are E-scooters polluters? The environmental impacts of shared dockless electric scooters," *Environ. Res. Lett.*, vol. 14, no. 8, 2019, Art. no. 084031.
- [64] C. Huang, C. Zhang, J. Zhao, X. Wu, D. Yin, and N. Chawla, "MiST: A multiview and multimodal spatial-temporal learning framework for citywide abnormal event forecasting," in *Proc. Int. Conf. World Wide Web*, 2019, pp. 717–728.



Suining He (Member, IEEE) received the PhD degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2016. He is currently working as an assistant professor with the Department of Computer Science and Engineering, University of Connecticut (UConn), Storrs, Connecticut. Before joining UConn, he worked as a postdoctoral research fellow with the Real-Time Computing Lab (RTCL), Department of Electrical Engineering and Computer Science, University of

Michigan, Ann Arbor, Michigan from 2016 to 2019. He received the UConn Research Excellence Program (REP) Award and the 17th IEEE International Conference on Mobile Ad-Hoc and Smart Systems (MASS) Best Paper Runner-Up Award in 2020. He is a Google PhD fellow, 2015. His research interests include smart transportation, urban data science, and mobile computing. He is an ACM and ASEE member.



Kang G. Shin (Life Fellow, IEEE) received the BS degree in electronics engineering from Seoul National University, Seoul, South Korea, in 1970, and the MS and PhD degrees in electrical engineering from Cornell University, Ithaca, New York, in 1976 and 1978, respectively. He is the Kevin and Nancy O'Connor professor of computer science and founding director of the Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Michigan. At Michigan, he has supervised the completion of 85 PhDs and also chaired the Computer Science and Engineering Division at Michigan for three years starting 1991. From 1978 to 1982, he was on the faculty of Rensselaer Polytechnic Institute, Troy, New York. His current research focuses on QoS-sensitive computing and networks as well as on embedded real-time and cyber-physical systems. He has authored/coauthored more than 980 technical articles and about 60 patents or invention disclosures. He has coauthored (with C. M. Krishna) a textbook "Real-Time Systems," McGraw Hill, 1997. He has received numerous awards, including 2019 Caspar Bowden Award for Outstanding Research in Privacy Enhancing Technologies, and Best Paper Awards from the 2011 ACM International Conference on Mobile Computing and Networking (MobiCom'2011), the 2011 IEEE International Conference on Autonomic Computing, the 2010 & 2000 USENIX Annual Technical Conference, the 2003 IEEE IWQoS, and the 1996 IEEE Real-Time Technology and Application Symposium. He also won the 2003 IEEE Communications Society William R. Bennett Prize Paper Award and the 1987 Outstanding IEEE Transactions on Automatic Control Paper Award. He has also received several institutional awards, including the Research Excellence Award in 1989, Outstanding Achievement Award in 1999, Service Excellence Award in 2000, Distinguished Faculty Achievement Award in 2001, and Stephen Attwood Award in 2004 from the University of Michigan (the highest honor bestowed to Michigan Engineering faculty); a Distinguished Alumni Award of the College of Engineering, Seoul National University in 2002; 2003 IEEE RTC Technical Achievement Award; and 2006 Ho-Am Prize in Engineering. He has held visiting positions with the U.S. Airforce Flight Dynamics Laboratory, AT&T Bell Laboratories, Computer Science Division within the Department of Electrical Engineering and Computer Science, UC Berkeley, and International Computer Science Institute, Berkeley, California, IBM T. J. Watson Research Center, Carnegie Mellon University, HP Research Laboratories, Hong Kong University of Science and Technology, Ewha Womans University in Korea, and Ecole Polytechnique Federale de Lausanne (EPFL) in Switzerland. He is a fellow of the ACM, and overseas member of the Korean Academy of Engineering, served as the general co-chair for 2009 ACM Annual International Conference on Mobile Computing and Networking (MobiCom'09), was the general chair for 2008 IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'08), the 3rd ACM/USENIX International Conference on Mobile Systems, Applications, and Services (MobiSys'05) and 2000 IEEE Real-Time Technology and Applications Symposium (RTAS'00), the program chair of the 1986 IEEE Real-Time Systems Symposium (RTSS), the general chair of the 1987 RTSS, a program co-chair for the 1992 International Conference on Parallel Processing, and served numerous technical program committees. He also chaired the IEEE Technical Committee on Real-Time Systems during 1991–93, an editor of *IEEE Transactions on Parallel and Distributed Systems*, and an area editor of *International Journal of Time-Critical Computing Systems*, *Computer Networks*, and *ACM Transactions on Embedded Systems*. He has also served or is serving on numerous government committees, such as the US NSF Cyber-Physical Systems Executive Committee and the Korean Government R&D Strategy Advisory Committee. He was a co-founder of two startups and is serving as an executive advisor for Samsung Research.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csl.