Origin-destination Flow Prediction with Vehicle Trajectory Data and Semi-supervised Recurrent Neural Network

Tao Huang
DiDi Research America
Mountain View, CA, USA
taohuang@didiglobal.com

Yintai Ma
Northwestern University
Evanston, IL, USA
yintaima2020@u.northwestern.edu

Zhiwei (Tony) Qin
DiDi Research America
Mountain View, CA, USA
qinzhiwei@didiglobal.com

Jianfeng Zheng
DiDi Chuxing
Beijing, China
zhenjianfeng@didiglobal.com

Henry X. Liu
DiDi Chuxing
Beijing, China
henryliu@didiglobal.com

Hongtu Zhu
DiDi Chuxing
Beijing, China
zhuhongtu@didiglobal.com

Jieping Ye
DiDi Chuxing
Beijing, China
yejieping@didiglobal.com

***Abstract*—Origin-Destination (OD) flow data is an important instrument for traffic study and management. So far traditional ways like surveys or detectors are costly and only give limited availability of OD flows. Various statistical and stochastic models for OD flow estimation and prediction based on limited link volume data or automatic vehicle identification (AVI) data have been developed. However, smartphone-generated trajectory data has not been as much leveraged in this field, though the usage of smartphones in traveling is emerging in recent years. In this paper, we propose a semi-supervised deep learning based model that appropriately combines both AVI and smartphone trajectory data during training and is able to generate predictions of OD flows in an urban network solely based on the smartphone trajectory data at inference time. Our model can provide OD estimation and prediction services on larger spatial areas beyond the limited spatial coverage of AVI data. Tests of our model using real data have shown promising results, compared with an AVI input-dependent Kalman filter model. Potentially, our model can easily be embedded to a trajectory collecting platform and generate continuous real-time OD flow predictions online.**

***Index Terms*—Semisupervised learning, Transportation, Recurrent neural networks, Spatial-temporal data**

I. Introduction and Review

Origin-Destination (OD) data are important for transportation applications. It shows the traffic flow volume between an origin (O) node to a destination (D) node at a specific time t. In most cases, for a pre-defined set of nodes, S, OD flow of a specific time can be expressed as a matrix of $|S| \times |S|$ dimension, therefore sometimes it is also referred to as OD matrices. The prediction of OD flow could be used to gain insights for traffic patterns, to assist in improving traffic management, and ultimately help in real-time traffic signal control system or infrastructure planning in a large-scale transportation network, etc.

Conventional methods for OD data collection rely on home interviews or surveys which generally are very costly, e.g., national household travel surveys in US. With emerging techniques like camera detectors, such OD flow data can be extracted from automatic vehicle identification (AVI) data. Accordingly, the process highly relies upon proper operation of all these detectors.[1] Besides, due to the cost of detector installation and maintenance, the availability of measured OD data is limited. Therefore, even though detectors/sensors become cheaper and serve with improving quality nowadays, and they become available in more and more cities, we might still suffer from data instability and coverage issues.

This problem can now be mitigated by tracking vehicle trajectories generated by vastly used GPS-enabled devices, such as smartphones. Trajectory data collected through phone application usages are maintained well in industry. They can potentially become a stable source of input with coverage over a considerable range of areas. In this work, we seek to predict OD flow using these trajectory data collected from phones with AVI data serving as partial supervision signals during training. At inference time, OD flow prediction does not depend on the working conditions of detectors. Further, through a semi-supervised framework developed, we are able to expand the prediction ability to OD pairs without direct AVI measurement.

OD flow estimation and prediction has long been an important open question in the literature. Using link volume data to fit partially collected survey data or AVI data with various models[2] have been popular for OD estimation [1]–[7], as link volume data can be retrieved from devices like *induction loop* underneath the roads. Upon the estimation, future OD flow can be modeled as linear combinations of historical OD flows with noises, thus leveraging tools from time series analysis like ARIMA/ARIMA-GARCH (see more

[1]Detectors are subject to physical breakdown, signal transmission breakout, etc.; high-resolution cameras are subject to unpleasant weather condition, blind-spot etc.

[2]Mostly based on *traffic assignment* theory

978-1-7281-0858-2/19/$31.00 ©2019 IEEE

details in [8]), Kalman Filter (KF), etc [9]–[17]. Numerous coefficients in these models, such as assignment matrices, are either pre-defined, which is still hard in practice, or solved in an optimization, which adds the scaling difficulty. Besides, a "shallow" linear/affine model might not capture the spatio-temporal dynamics of OD flows as it is complicated in nature.

As deep learning (DL) evolving fast in recent years, neural networks (NN) are used to build the estimation- and prediction models [18]–[20]. When geological grid/block level data that are analogous to images/graphs are available, traffic properties, such as local demand (hot/cold zones), congestion index or travel speed, form matrices or tensors, where index positions in data convey geological closeness or connectivity. In such cases, methods like convolutional neural network (CNN) can be useful for prediction models, such as grid in-out flow [21], [22], grid traffic condition [23], or travel demand [24]. With graph convolutional network (GCN) [25], graph-structured data, such as link flows of controlled-access highways (e.g. New Jersey Turnpike), can be learned efficiently to build OD flow prediction models [26]. Success of these methods depends on the data availability, format, specific problem abstraction and definition, and prediction scope.

On the data side, GPS-enabled devices enable researchers to get subsamples of real trajectories. Researchers can proactively send probe vehicles with GPS, or collect data from roadside sensors or detectors. Rapid growing usage of mobile phones, especially smartphones, in recent years makes the collection of trajectory data much cheaper and more sustainable, which is critical for building an online prediction tool. Probe trajectory data was incorporated into state-space models to provide extra source of constraints/measurements [27], [28]. In [29], car data from taxis were used to derive a-priori matrices for estimation and analyze route choices. Using split rates estimated from floating car data and remote traffic sensors, [30] modified the static OD demands to obtain the time-varying OD demands. In [31], probe vehicle data were used to estimate link flows, which were further used with historical OD matrices to estimate dynamic OD matrices in a bi-level model. Based on the observed link/probe ratios, [32] built scaled probe OD matrices as a-priori and applied them to link flow frameworks to solve for OD matrices. Using phone data, [33] developed an algorithm to estimate population's travel demand using phone location data. Tower-to-tower transient OD matrices were generated from records of mobile calls to make OD estimation in [34].

Our contributions come in two-fold. First, we develop a semi-supervised learning framework for OD flow predictions beyond the usual coverage of AVI data, which is limited by the physical availability and reliability of the detection devices. By appropriately combining AVI data and smartphone-generated trajectory data, our model is able to make predictions for as many OD pairs as the trajectory data covers. Second, in contrast to KF-based methods, our model depends only on the trajectory data for the entire inference time period, while AVI data are used only during training as partial supervision signals. This is desirable for practical implementation as 1) from coverage perspective, most smartphones nowadays has GPS capability and they are widely used; 2) from stability perspective, the data is maintained in reliable platforms.

The rest of the paper is organized as follows. In Section II, we discuss our motivation and methodology; In Section III, we present the models we develop in detail; In Section IV, we describe how we collected the real-world data for our experiments, explain the settings of our models and present the numerical results of the experiments carried; In Section V, we analyze several critical factors and parameters in our proposed models; Finally, in Section VI, we draw the conclusions for this work and discuss possible future works.

II. Methodology

The trajectory data are anonymous vehicle trajectories collected through the use of phone applications, such as a ride-hailing app. With aforementioned limitation of AVI data and advantages of phone trajectory data, we explore building an OD flow prediction model with only input from these trajectory data (when serving) for a more promising usage scenario.

There are several considerations in developing our models. First, over a large spatial area, all OD flows would "intertwine" together and interact with each other and evolve in a complicated way, as described in [35]. At each moment, each OD flow is composed of flows of a number of different paths, and each path is composed of a number of links (an arc, or a segment of a road). Each link flow is the downstream of upstream links from last moment, and will distribute to downstream links at next moment. Moreover, OD pairs can share part of paths and links.

To capture such spatial-temporal patterns among OD flow to a large extent, we leverage a deep recurrent neural network (RNN) like Long Short-Term Memory (LSTM) [36], [37]. The main advantage of DL is the ability to model highly nonlinear dynamics and patterns. As a comparison, linear models like KF models capture linear dynamics, which might not be sufficient in this case. The problem and data we consider here is point-to-point OD flows for a list of OD pairs, and thus data matrices do not convey spatial/geographical meaning as in congestion heatmap or other grid level data. Therefore, we might not use CNN to exploit the spatial patterns. Besides, we do not involve link level volumes, and OD pairs mostly form a fully-connected directed graph, therefore GCN might be less useful in this case.

Second, each trajectory is an actual travel event (unlike simple route query records), generated by actual travel willingness and action at the moment, hence it represents an aforementioned "survey" but could only be more accurate. The cohort of such trajectory becomes a sub-sampling of overall OD flows. Moreover, many ride-hailing travels are generated routinely, and therefore reflect the temporal patterns of overall flows. Our baseline KF model, to some extent, linearly upscales the subsamples to overall flow via a penetration ratio matrix, like

the traditional way [27], [33]. In contrast, we use RNNs to "learn" the mapping back to the overall volumes.[3]

Third, AVI data can only provide the OD flow counts where detectors like high-resolution cameras are equipped. For uncovered traffic nodes, we only have phone trajectory data. To leverage the most of the large coverage of phone trajectory, we expand the neural network to a semi-supervised learning framework by using a special objective inspired by *label propagation* (LP) [38].

LP is a semi-supervised learning algorithm that assigns labels to unlabeled data points probabilistically in a node graph (G, V) where part of nodes are labeled and others are not. For any two nodes of the graph, (i, j), their similarity/closeness is represented as *edge weight* w_{ij}. Label of one node can propagate to its neighbors. The probability of jumping from j to i, or j propagating its label to i, is

$$p_{ij} = \mathbb{P}(j \to i) = \frac{w_{ij}}{\sum_k w_{kj}} \tag{1}$$

Let z_{ic} be the probability that node i has label c, then in one propagation step, z_{ic} update itself as

$$z_{ic} \leftarrow \sum_k p_{ik} z_{kc}, \text{ or } \mathbf{Z} \leftarrow \mathbf{PZ} \tag{2}$$

in matrix form, where $\mathbf{P}$ and $\mathbf{Z}$ are matrices formed by all p_{ij}s and z_{ic}s respectively. In this propagation, an unlabeled point is more likely to get its label from a "similar" labeled point than those "dissimilar" labeled points. In [38], repeatedly executing this propagation, together with renormalization of $\mathbf{Z}$ and clamping the labeled data, guarantees convergence to a solution where every node is labeled.

As a conclusion for this section, to exploit the advantages of trajectory data over AVI data, we build a semi-supervised learning model, based on RNN and an LP-inspired special objective, to enable prediction of OD flow for OD pairs with or without ground truth. We only use trajectory implied flows as input to our model while AVI data are used only for training.

Some of the main symbols and notations used in this work can be seen in Table I.

TABLE I: Notations

Symbol	Definitions
S	a pre-defined traffic nodes set of interest
N	a pre-defined list of OD pairs, each of which composed of (O)rigin node and (D)estination node from S. Without loss of generality, we can index it as $N = \{1, \cdots, n\}$
$\mathcal{T}$	time frame, set of time interval t considered
$\mathbf{x}_t$	$\mathbf{x}_t = (x_{1t}, \cdots, x_{nt})^T$, smartphone collected trajectory data OD flow counts at time t for all OD pairs in N
$\mathbf{X}$	$\{\mathbf{x}_t\}_{t\in\mathcal{T}}$, smartphone trajectory data OD flow dataset, a matrix
N_s,N_u	the set of (s)upervised and (u)nlabeled OD pairs. $N_u = N \setminus N_s$. In a supervised framework, $N_u = \varnothing$
$\mathbf{y}_t$	$\mathbf{y}_t = (y_{1t}, \cdots, y_{nt})^T$, vector of the overall OD flow counts from AVI data at time t for all OD pairs in N
$\mathbf{Y}$	$\{\mathbf{y}_t\}_{t\in\mathcal{T}}$, ground truth
W	window length of time intervals for making predictions
$\mathbf{C}$	a rank correlation between rows of $\mathbf{X}$. Each entry c_{ij} is the correlation of trajectory OD flow between ith and jth OD pair along time dimension
$\mathbf{R}$	magnitude ratios between ground truths. Each entry r_{ij} represents $\max_{t\in\mathcal{T}} y_{it} / \max_{t\in\mathcal{T}} y_{jt}$
$\boldsymbol{\theta}$	neural network parameters
$\mathcal{L}, \mathcal{L}^R$	symbol for a general loss function, specific meaning can be inferred from context
$h(\cdot,\cdot)$	loss term used in semi-supervised model objective
ϕ	kernel function used in $\mathbf{R}$ matrix estimate model learning
$\hat{\cdot}$	hat symbols represent estimates of the variable underneath the "hat"
$\mathbf{A}_{i,:}, \mathbf{A}_{:,j}$	for any matrix $\mathbf{A}$, the ith row/jth column
$\mathbf{A}_{B,:}, \mathbf{A}_{:,B}$	for any matrix $\mathbf{A}$, the sub-matrix composed of rows with row index in a set B/sub-matrix composed of columns with column index in a set B

III. Models

In this section, we first introduce a supervised learning (SL) model, where OD pairs that we predict all have available ground truth. Then, we augment this model with an LP-inspired objective to make predictions in a semi-supervised setting, where more OD pairs are covered even though they have no corresponding ground truth during training.

A. Supervised Model

We assume there is a function parameterized by $\boldsymbol{\theta}$, $f(\cdot;\boldsymbol{\theta})$ that maps a sequence of input $\mathbf{x}_t$s, such as $\{\mathbf{x}_{t-W+1}, \cdots, \mathbf{x}_t\}$, to a prediction of ground truth at $t+1$,

$$\hat{\mathbf{y}}_{t+1} = f(\mathbf{x}_{t-W+1}, \cdots, \mathbf{x}_t; \boldsymbol{\theta})$$

[3]For example, a fully-connected layer with vanilla linear activation is a linear upscaling

f only uses trajectory flow data as input, and thus avoids the dependence on the ground truth feed at inference. We use an RNN based network to approximate this mapping, and learn it in a fully supervised learning frame ($N = N_s$). $\boldsymbol{\theta}$ is learned by minimizing

$$L_{sl}(\boldsymbol{\theta}) = \sum_t \mathcal{L}\left(f(\mathbf{x}_{t-W+1}, \cdots, \mathbf{x}_t; \boldsymbol{\theta}), \mathbf{y}_{t+1}\right), \tag{3}$$

where $\mathcal{L}(\cdot,\cdot)$ can be any regular loss term such as the mean square error (MSE).

B. Semi-supervised Model

Now we consider predicting more OD pairs which do not have available ground truth from AVI data or other sources. The goal of this semi-supervised learning (SSL) model is to relax the requirement for ground truth data for each OD pair intended to predict, and thus leave room for partially available AVI data or flawed (on part of OD pairs) AVI data. In this setting, the set of "unlabeled" OD pairs is N_u and $N = N_s \bigcup N_u$. The phone-based trajectory flow data are available for OD pairs of N_u, so that we can still infer the OD flows for these pairs.

The neural network model structure is shown in Fig. 1. We augment the proposed supervised model with a special loss term inspired by LP to form our SSL model. This loss term leverages information from two matrices, $\mathbf{C}$ and $\mathbf{R}$, to make the prediction for OD pairs of N_u. Assume a neural network

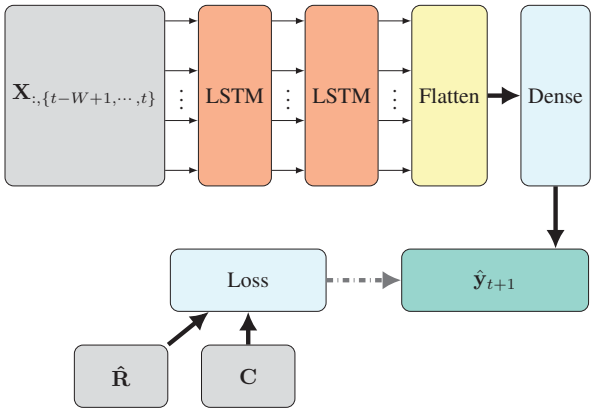

Fig. 1: Model architecture of the semi-supervised model. Trajectory data flow volumes form the input tensor. Multiple layers of LSTM form the learning core for spatial-temporal patterns of OD flows. The outputs of RNN module are fed to fully-connected layers to generate prediction for next time interval. Estimates $\hat{\mathbf{R}}$ and correlations $\mathbf{C}$ are involved in the special LP loss during training process.

$f(\cdot;\boldsymbol{\theta})$ makes predictions using sequence of $\{\mathbf{x}_t\}$, then the LP loss term is expressed as follows:

$$\hat{\mathbf{y}}_{t+1} = f(\mathbf{x}_{t-W+1},\cdots,\mathbf{x}_t;\boldsymbol{\theta}) \tag{4}$$

$$L_{lp}(\boldsymbol{\theta}) = \sum_t \sum_{i\in N_u, j\in N_s} c_{ij} h\left(\hat{y}_{i(t+1)}, \hat{y}_{j(t+1)}; \hat{r}_{ij}\right), \tag{5}$$

where c_{ij} is the Kendall's τ correlation between $\mathbf{X}_{i,:}$ and $\mathbf{X}_{j,:}$ computed from the training input set, and $\hat{r}_{ij}$ is the (i,j) entry of the estimated ground truth magnitude ratios $\hat{\mathbf{R}}$, which is the scale ratio of ith and jth OD pair flow with respect to the maximum flow volume. $h(\cdot,\cdot;r)$ is a loss which we will address later.

Combining the individual components (3) and (5) with a weight λ_{lp}, we obtain the overall objective for our semi-supervised model:

$$L_{ssl}(\boldsymbol{\theta}) = L_{sl}(\boldsymbol{\theta}) + \lambda_{lp} L_{lp}(\boldsymbol{\theta}) \tag{6a}$$

$$= \sum_t \mathcal{L}(\hat{\mathbf{y}}_{N_s,t+1}, \mathbf{y}_{N_s,t+1}) \tag{6b}$$

$$+ \lambda_{lp} \sum_t \sum_{\substack{i\in N_u \\ j\in N_s}} c_{ij} h\left(\hat{y}_{i(t+1)}, \hat{y}_{j(t+1)}; \hat{r}_{ij}\right) \tag{6c}$$

where sub-array $\mathbf{y}_{N_s,t+1}$ stands for the entries with indices in N_s. Notice that in (6), only ground truth data of OD pairs in N_s are involved in training process, used in term (6b), as in this setting we only have such supervision signals. Besides, predictions for OD pairs in N_s and N_u are made jointly in this model. Correlation c_{ij}s are directly derived from trajectory data and $\hat{r}_{ij}$s are estimated using trajectory data via a pre-trained model. More details of the semi-supervised model are discussed in the following subsections.

1) Objective Intuition: In (5), $\hat{\mathbf{R}}$ is our estimate of the magnitude ratios between any two OD pairs. Each element, $\hat{r}_{ij}$ is an estimate of $\max_{t\in\mathcal{T}} y_{it} / \max_{t\in\mathcal{T}} y_{jt}$. For an (i,j) pair, if both index are in N_s, certainly we can calculate the ratio from the observed ground truth, and if i or j is in N_u, then we need to estimate it. With an appropriately estimated $\hat{\mathbf{R}}$, we can use $\hat{r}_{ij}\mathbf{Y}_{j,:}$ as an estimator for $\mathbf{Y}_{i,:}$, which assumes for OD pair i and j, they share similar relative trend along the time dimension and only differs in scale. For an (i,j) where $i \in N_u$ and $j \in N_s$, $h(\hat{y}_{it}, \hat{y}_{jt}; \hat{r}_{ij})$ can take the following form [4] using any regular loss $\mathcal{L}$ such as squared loss,

$$h(\hat{y}_{it}, \hat{y}_{jt}; \hat{r}_{ij}) = \mathcal{L}(\hat{y}_{it}, (\hat{r}_{ij}\hat{y}_{jt})). \tag{7}$$

Any estimate which makes $\hat{y}_{it}/\hat{y}_{jt} \neq \hat{r}_{ij}$ will be penalized by h, therefore realizes "propagation". N_s and N_u OD pairs are linked by this loss term h.

In the original LP, the link is done by (2). In (7), if $\mathcal{L}$ is the squared error,

$$h(\hat{y}_{it}, \hat{y}_{jt}; \hat{r}_{ij}) = \mathcal{L}(\hat{y}_{it}, (\hat{r}_{ij}\hat{y}_{jt})) = \left(\hat{y}_{it} - \hat{r}_{ij}\hat{y}_{jt}\right)^2,$$

then the prediction minimizing L_{lp} should have

$$\hat{y}_{it} = \frac{\sum_{j\in N_s} c_{ij}(\hat{r}_{ij}\hat{y}_{jt})}{\sum_{j\in N_s} c_{ij}}, \text{ for all } i \in N_u, t \in \mathcal{T}.$$

If we see c_{ij} as the edge weights, and analogous to (1), see jumping probability $p_{ij} = c_{ij}/\sum_j c_{ij}$, and $\hat{r}_{ij}\hat{y}_{jt}$ as the estimate of y_{it} using j as "propagation source", denoted as $\tilde{y}^i_{jt}$, then we have

$$\hat{y}_{it} = \sum_{j\in N_s} p_{ij}\tilde{y}^i_{jt}, \text{ for } i \in N_u, t \in \mathcal{T}. \tag{8}$$

This is essentially label propagation (2) with extra dimension t and continuous value instead of discrete class label. Correlation $\mathbf{C}$ is more of a "soft" gate that controls the portion that each prediction for N_u OD pairs would inherit "propagation" from any prediction for N_s OD pairs. Note that we need all elements of $\mathbf{C}$ to be nonnegative for p_{ij}s to be meaningful probabilities. Correlation indeed can be negative. However, in our problem, as all OD pairs tend to have similar temporal patterns (morning/afternoon peak, off-peak, etc.) in nature, most correlations are positive — only less than 2% of $\mathbf{C}$ elements in our experiment is negative and the smallest value is only -0.036.

LP has its root in graph models. Therefore, to some extent, our approach still connects to graph models though we did not explicitly use graph based models like GCN.

2) Correlation: Kendall's τ [39] is a commonly used correlation. Like Spearman's ρ, τ is a rank correlation. Therefore it shows the similarity of the orderings of two data and is invariant of any monotone mapping of data, which does not always hold for moment correlation such as Pearson's r, though the three correlations are equivalent for Gaussian variables [40].

In this paper, we focus more on the similarity between the time series trends of the trajectory flow volumes of OD pairs than their similarity on the actual values. Therefore, we

[4] or a reverse form $h(\hat{y}_{it}, \hat{y}_{jt}; \hat{r}_{ji}) = \mathcal{L}\left((\hat{r}_{ji}\hat{y}_{it}), \hat{y}_{jt}\right)$, but notice that we need to use $\hat{r}_{ji}$ here for (i,j) entry. One can even use $\mathcal{L}\left((\hat{r}_{ji}\hat{y}_{it}), \hat{y}_{jt}\right) + \mathcal{L}(\hat{y}_{it}, (\hat{r}_{ij}\hat{y}_{jt}))$ to achieve a symmetry.

prefer the rank correlation as the proxy of similarity in LP implementation.

3) $\mathbf{R}$ *Matrix:* $\mathbf{R}$ matrix conveys the magnitude ratios between OD pairs. Geographical distances are important to estimate the ratios, but the trajectory flow volumes are also indicative — the implicit subsamples reflect the overall volume and ratio. Therefore we bring a model to take in all these factors to estimate $\mathbf{R}$ matrix, especially the ratios between pairs in N_s and N_u:

$$\hat{r}_{ij} = g(\max_t x_{it}, \max_t x_{jt}, dist_{ij}, \frac{\max_t x_{it}}{\max_t x_{jt}}, loc_i, loc_j),$$

where $g(\cdot)$ is the mapping needs to be learned, $dist_{ij}$ is the averaged distance between the origin and destination of OD pair i and j, and loc_i is the latitude and longitude of origin and destination of pair i. Of course, if more information is available, it is beneficial to expand the model to include more attributes of OD pairs.

The function $g(\cdot)$ can be modeled with a mainstream supervised model. In our context, among Random Forest, a neural network-based model and Xgboost [41], we have chosen Xgboost for its best performance. Overall, from our experience of real data, a tree based model fits better for this ratio estimate model as it is less demanding to the appropriate preprocessing of the meta-data input.

As we train the estimate model using OD pairs in N_s, for $i, j \in N_s$, both $\hat{r}_{ij}$ and $\hat{r}_{ji}$ will be in the objective. A "vanilla" loss like $\sum_{i\in N_s}\sum_{j\in N_s}(\hat{r}_{ij} - r_{ij})^2$ is not preferable since the reciprocal of r_{ij} is r_{ji} and the loss implicitly penalizes proportional error to the larger ratio more compared to its reciprocal. Our solution is to transform the ratios first with a kernel ϕ and then penalize on the kernel outputs.

For a measure of deviation/error, $\mathcal{L}^R(\cdot,\cdot)$, the kernel ϕ should be a function "symmetric" to proportional error of a value a and a^{-1}, or more specifically, for any value a and a multiplier δ, ϕ should have

$$\mathcal{L}^R(\phi(a), \phi(\delta a)) = \mathcal{L}^R\left(\phi(1/a), \phi(1/(a\delta))\right). \tag{9}$$

There are plenty of choices for such ϕ, e.g. *natural log*, $\phi(a) = \ln(a)$, or *reverse odds*, $\phi(a) = a/(1+a)$, satisfies (9) on Minkowski distances. In implementation, we use natural log and squared error loss in our experiment realization,[5] therefore the objective used for $\mathbf{R}$ matrix estimate model is

$$\sum_{i\in N_s}\sum_{\substack{j\in N_s\\ j\neq i}} \mathcal{L}^R\left(\phi(\hat{r}_{ij}), \phi(r_{ij})\right) = \sum_{i\in N_s}\sum_{\substack{j\in N_s\\ j\neq i}} \left(\ln(\hat{r}_{ij}) - \ln(r_{ij})\right)^2. \tag{10}$$

In our implementation, the ratio estimation model is pre-trained, as the ratio estimates need to be embedded to the training of the main SSL model. Throughout the training of the ratio estimate model, only the ground truth data of OD pair in N_s are used, for calculating r_{ij}s in (10).

[5]Both suggested kernels have similar performance under squared error loss, as most ratios are not "extreme" and natural log and reverse odds only differs distinctly in very extreme ranges.

IV. Experiment Setup and Results

A. Experiment Setup

Our experiment is conducted in a rectangular area of a major city of China, on a network composed of 14 nodes with camera detectors, shown in Fig. 2. The AVI data comes from these traffic detectors,[6] and the trajectory data is collected from a popular ride-hailing app. Each trajectory is a sequence of the geographic locations (including longitude and latitude) of a vehicle (loc) with timestamps (ts). To get the OD flow count, for each trajectory $(\text{loc}_1, \text{ts}_1) \to (\text{loc}_2, \text{ts}_2) \to \cdots \to (\text{loc}_m, \text{ts}_m)$, if there exists a loc_i in the origin list, and loc_j with $j > i$ in the destination list of loc_i,[7] then we add 1 to the OD flow count of $(\text{loc}_1, \text{loc}_j)$ at time ts_i. With a pre-defined and properly stored OD pair list, the parsing takes $O(Ml_{\max}|N|^2)$ time and is parallelizable, where M is the number of trajectories, $l_{\max}$ is the maximum length of a trajectory. For AVI data grouped via the encoded plate number, they are parsed similarly to get the overall OD counts time series $\mathbf{Y}$, except that vehicles are only witnessed at these traffic nodes, while app trajectories are more "continuous". Each time interval is 30min long.

1) Basic Settings: 182 OD pairs exist among the 14 nodes, which form set N. We assume the first 11 nodes are AVI data available and the rest 3 are not. Any OD pair involves a node without available AVI data do not have data for supervision, therefore $|N_s| = 110$ and $|N_u| = 72$. The dataset includes AVI and trajectory data collected from Sep. 20, 2017 to Nov. 1, 2017. The former 3/4 data forms the training set, and the rest 1/4 forms the testing data. For both SL and SSL model, we use a time window of 24h ($W = 48$) to make predictions for next interval.

2) Evaluation: MAE/MAPE are main metrics used to measure the model performances, but each metric has its limitation. The flow values vary from about 10-ish before dawn to over 1000 at peak hours. In low volume hours, a small absolute error can cause a large percentage error; yet in peak hours, the situation is reversed. For traffic management perspective, low volume periods are less concerned, compared to high volume periods during a day. To address above issues, we look at 3 time ranges:

- main hours, 6am to 10pm;
- peak hours, morning/afternoon peak, 6:30am to 9:30am, 4:30pm to 7:30pm;
- 24 hours, all day

3) Model and Parameters:

a) KF Baseline: As discussed in Section I, time series models such as ARIMA and KF are widely used for OD flow prediction [10]–[17]. Therefore, as a comparison, we also construct a KF-based baseline model, with ARIMA used for penetration ratio. To the best of our knowledge, there is no

[6]The plate and other vehicle information are encrypted

[7]$(\text{loc}_i, \text{loc}_j)$ in the OD pair list

Fig. 2: Experiment area map.

semi-supervised framework that is comparable to ours. We construct our KF model with following prediction evolution:

$$\begin{aligned} \mathbf{x}_t &= \mathbf{A}_t \mathbf{x}_{t-1} + \mathbf{u}_{t-1}, \\ \mathbf{y}_t &= \mathbf{\Gamma}_t \mathbf{x}_t + \mathbf{v}_{t-1}, \end{aligned} \tag{11}$$

where the $\{\mathbf{u}_t\}_{t\in\mathcal{T}}$ and $\{\mathbf{v}_t\}_{t\in\mathcal{T}}$ are white noises with distribution $\mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_1)$ and $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_2)$; $\mathbf{A}_t$ is a transition matrix and $\mathbf{\Gamma}_t$ is a diagonal matrix with $\mathbf{\Gamma}_t = \text{diag}(\boldsymbol{\gamma}_t)^{-1}$. Here $\boldsymbol{\gamma}_t$ stands for the penetration ratio of trajectory flow, i.e., $\mathbf{x}_t/\mathbf{y}_t$. The actual data implied penetration ratio is not suitable for direct use for its large volatilities. Instead, each entry, $\gamma_{i,t}$, is modeled by an individual ARIMA(1,1,1) model, fitted by the true ratios of training data and predicted with time series up to $t-1$. *Simple moving average* (SMA) of length 2 window is used to the predictions for further smoothing. These methods help improve the baseline performance.

We assume that noises $\mathbf{u}_t$ and $\mathbf{v}_t$ are i.i.d. and follow the standard normal distribution, i.e. we have $\mathbf{Q}_1 = \mathbf{Q}_2 = \mathbf{I}$. The matrix $\mathbf{A} = 0.9\mathbf{I}$. The state covariance is initialized as $0.01\mathbf{I}$ in baseline model.

b) Semi-supervised model: The main neural network is composed of two LSTM layers with 128 units, and a dense layer with 64 units before the final layer making an $\mathbb{R}^{182}$ output. Inputs are fed in as a 3-mode tensor with each layer of dimension 48×182. RNN outputs are flattened before sent to dense layers. The dense layer uses scaled exponential linear unit (SELU) activation as it produces smoother result in low-volume period than rectified linear unit (RELU). *Batch normalizations* are added to layers and demonstrate enhancement to the performance, probably because the data differs a lot between different OD pairs and different times. The weight of the label propagation loss is $\lambda_{lp} = 0.1$ in the objective (6). The choices here are made based on the data and experience. Part of parameters are tuned in a reasonable range.

For the $\mathbf{R}$ estimation model, we use an XGBoost [41] with 100 regression trees and max depth of 50. We used grid search over a small set of parameters for the XGboost model, and the presented parameters here are the set that performs the best.

c) Supervised model: The network structure is the same as our semi-supervised model, except that the last layer has 110 units and objective is MSE as (3). $(\mathbf{X}_{N_s,:}, \mathbf{Y}_{N_s,:})$ are used for training. Prediction is made for OD pairs of N_s.

Remark 1: To align the models, only $\mathbf{Y}_{N_s,:}$ are used for training for each model. $\mathbf{Y}_{N_u,:}$ is only used for testing performance. Baseline and SL model can only predict the OD pairs in N_s while the SSL model predicts for all OD pairs.

Remark 2: In testing/inference stage, compared to SL and SSL model, which can run with only $\{\mathbf{x}_t\}$ input, KF baseline still needs **continuous** input of ground truth to keep making prediction adjustment. Of course, one can make forecasts dynamically by using predicted $\hat{\mathbf{y}}_t$s from previous stage, but the process will either disastrously blow up or converge to $\mathbf{0}$, depending on whether $\mathbf{A}_t \succeq \mathbf{I}$ or $\mathbf{A}_t \prec \mathbf{I}$. This inherent drawback of KF model makes its deployment subjects to the vulnerabilities described previously about the AVI data.

B. Results

1) $\mathbf{R}$ Estimation: There are $2|N_s||N_u| + |N_u|^2 = 21024$ ratios to be estimated as they involved one or more OD pairs from N_u. Estimated values, compared with true values, are plotted in Fig. 3. The MSE of log-ratio is 0.078 and the MAE of log-ratio is 0.219. Factor importance analysis of Xgboost is plotted in Fig. 4. It shows that the distance and trajectory OD flow ratio between two pairs are the primarily dominating factors. Compared to them, the rest features are far less important.

2) Prediction Models: We use *KF* to represent the KF baseline model, *SL-k* and *SSL-k* to represent the SL model and SSL model with k nodes set as ground truth unavailable (in this experiment, $k = 3$) respectively. In Table II, we compare the prediction performance between the models. In comparison with the baseline, we also show the KF model without ARIMA for penetration ratio prediction and smooth (using actual data implied $\boldsymbol{\gamma}_t$ instead). The huge increase in MAPE and MAE demonstrates the necessity of ARIMA for the baseline KF model. Test results of a selection of OD pairs, from the pairs of best 5 and worst 5 performance (in MAPE sense) in N_s and N_u, are plotted in Fig. 5, 6, 7 and 8.

During test, the baseline model rolls forward with continuous $(\mathbf{x}_t, \mathbf{y}_t)$s input, while our model only uses trajectory data for prediction. Yet, for the OD pairs in N_s, the semi-supervised model reaches a similar performance with the baseline, with slightly weaker MAPE on 75% quantile. In peak hours, it even outperforms the baseline on MAE, though baseline still looks better on MAPE. Detailed empirical distribution function (EDF) of respective MAPE and MAE over these OD pairs is outlined in Fig. 10 for the full percentile information. As to the OD pairs of N_u, it is intuitive that the results are generally worse than the supervised OD pairs, as they do not have any supervision of its own during the training. Still, on the set N_u, our model achieved 32.8% median MAPE on main hours and 33.6% median MAPE on peak hours.

The discrepancy between MAPE and MAE can be explained in two folds: 1) for two predictions, the one with smaller MAE might still have a larger MAPE;[8] 2) the OD pairs where

[8]For two absolute errors, the smaller one might still cause a larger percentage error, if the underlying truth is very small

TABLE II: Comparison of percentile MAE/MAPE over OD pairs, three time range resp.

Metric	Model	Peak Hours			Main Hour			24H		
		25%	50%	75%	25%	50%	75%	25%	50%	75%
MAPE	KF w/ ARIMA	42.3%	62.2%	83.3%	45.6%	69.6%	89.5%	48.4%	64.9%	77.9%
	KF	19.7%	22.1%	30.4%	17.4%	20.3%	27.8%	21.0%	25.9%	33.4%
	SL-3	21.9%	26.8%	33.8%	21.2%	25.8%	30.5%	28.2%	32.3%	40.6%
	SSL-3 N_s	18.2%	23.4%	35.3%	18.9%	21.8%	30.2%	27.3%	31.2%	38.9%
	SSL-3 N_u	26.2%	33.6%	42.5%	25.7%	32.8%	40.8%	34.5%	40.4%	46.6%
MAE	KF w/ ARIMA	50.3	103.7	182.4	56.1	118.7	213.3	42.6	92.6	155.6
	KF	24.1	44.2	65.6	20.5	41.1	60.5	17.0	33.6	48.3
	SL-3	25.9	44.3	62.3	27.7	44.1	62.9	22.5	36.3	49.6
	SSL-3 N_s	23.7	41.4	54.7	24.1	42.2	56.0	20.8	36.3	46.4
	SSL-3 N_u	54.3	65.7	111.9	51.4	69.9	116.7	41.8	54.4	87.4

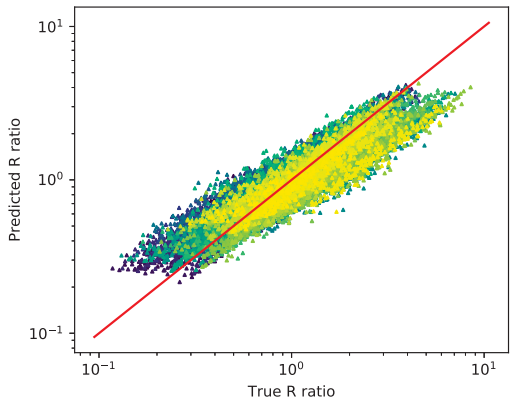

Fig. 3: 2D plot of $\mathbf{R}$ estimation model output and true ratios. Log-scale axes are used to show ratios of all range clearly and symmetrically. Overall, most points concentrated around the line $y = x$. Viridis colormap here is purely for ease of illustration.

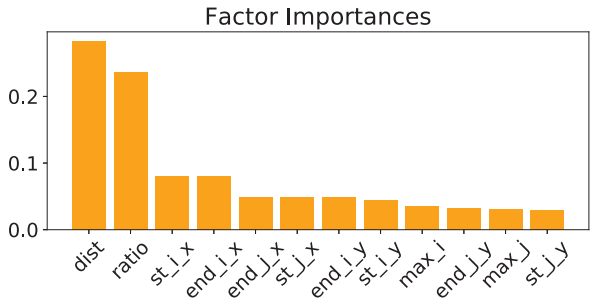

Fig. 4: The importance analysis of factors in the xgboost model. `ratio` is the ratio of trajectory implied OD counts between OD pair i and j; `max_i` is the maximum trajectory volume on pair i; `dist` is the averaged distance between OD pair i and j; `st_i_x` and `st_i_y` stands for the longitude and latitude of the starting node of OD pair i respectively; `end_i_x` and `end_i_y` stands for the longitude and latitude of the destination node of OD pair i respectively.

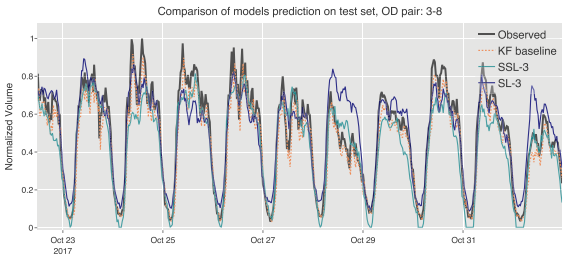

Fig. 5: An example of best performance OD pairs in N_s. All values normalized using the maximum volume of the *observation* during the test period.

semi-supervised model outperforms baseline on MAE might correspond to the lower end or higher end in the MAPE EDF plot (where SSL-3 actually outperforms baseline in peak hours, see Fig. 10).

SSL-3 is also slightly better than SL-3 on N_s, which shows that the LP loss term L_{lp} does not worsen the prediction for supervised part. In fact, it might even help — input $\mathbf{X}_{N_u,:}$ might help correct the prediction of N_s, or LP loss term might help alleviate over-fitting of the supervised part.

As stated previously, the issue of volume fluctuation between peak/off-peak hours has a huge impact on our measures, and therefore we look at three different time ranges. This issue is more obvious in the plot of percentage error vs. observed OD flow, shown in Fig. 9. The percentage error always shows a reverse pattern to the observed OD flow — when OD flow is very low, the percentage error raises to very high; when OD flow raises during main hours, the percentage error drops to around 20% level.

V. Performance Analysis

A. Proportion of no ground truth nodes

Fig. 11 and 12 show the prediction performance with respect to the number of nodes without ground truth. We set $k = 2 \sim 9$ nodes out of all 14 masked as "unlabeled", and then repeat the whole training and prediction process with multiple runs. For any k, say $k = 4$, for each run, the "unlabeled" nodes are randomly chosen initially in the run. Median MAPE plots of main and peak hours are shown here.

From the mean metric lines, the OD pairs in N_s have better performance than OD pairs in N_u. The prediction performance trends keep relatively "stable" as the size k changes, except for slightly uprising trends in supervised MAPE.

Compared to mean metric, the standard deviations (SD) of metrics fluctuate more often and more drastically. Overall, the

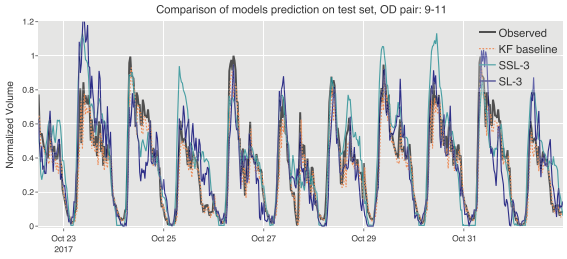

Fig. 6: An example of worst performance OD pairs in N_s. Values normalized the same way as Fig. 5.

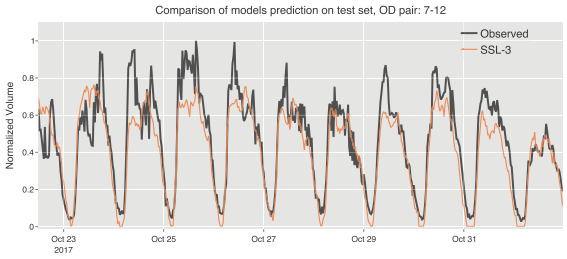

Fig. 7: An example of best performance OD pairs in N_u. Values normalized the same way as Fig. 5.

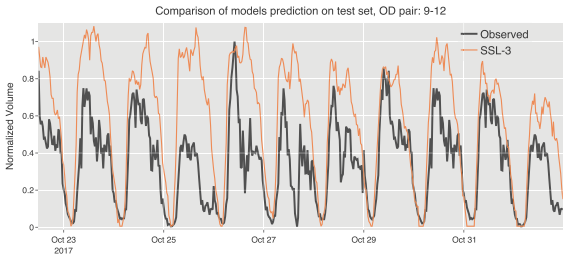

Fig. 8: An example of worst performance OD pairs in N_u. Values normalized the same way as Fig. 5.

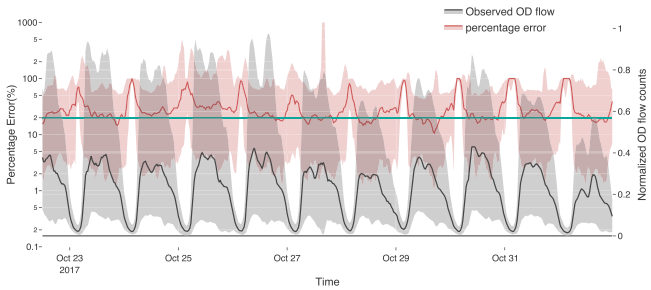

Fig. 9: Percentage error of SSL-3 OD flow prediction overlaid with actual observed OD flow counts during test period. The red area represents the 5 to 95 percentile of prediction percentage error among all OD pairs at each time interval, with the solid line showing the median; The gray area represents the observed OD flow counts value. All lines get smoothed via moving average with window length 3. The observed OD flow is normalized using the maximum of the percentile values during the test period.

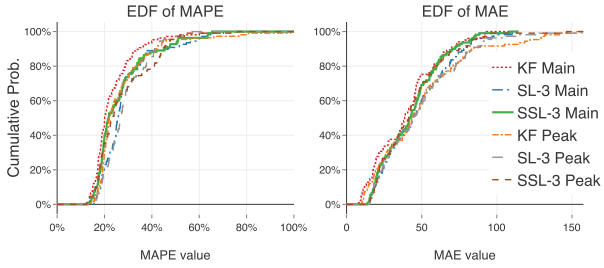

Fig. 10: EDF of MAPE/MAE on main hours and peak hours for baseline and SSL model. The comparison is based on the OD pairs in N_s.

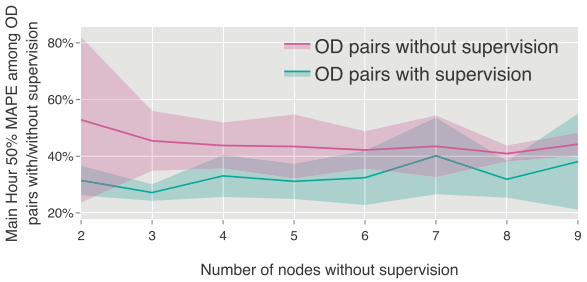

Fig. 11: Median MAPE of OD pairs with/without supervision on *main hours* vs. number of nodes without supervision. Mean metric is the solid line and the bands show the range within 1 SD among the samples.

band of median MAPE of N_u OD pairs tends to shrink while the band of N_s OD pairs tends to increase, as size k expands. The reason is the following: OD pairs with relative small volume tend to have larger MAPE even with small MAE. When size k is small, e.g. 2, it is likely to choose a set of all low volume nodes or all high volume nodes and therefore have holistically higher or lower MAPE. When the set grows, this situation is much less likely. This is why at 2, the SD of MAPE is larger than the rest. This also partially explains why the SD of supervised median MAPE increases. [9]

Overall, the experiments demonstrate a more robust performance than expectation as the supervision coverage decreases, which is a good sign for further experiments and real-world application. Of course, more case studies on different cities and seasons are required to give more insight and more confident conclusions.

B. Correlation

For all the experiments we have done, for each OD pair in N_u, we compute the correlations of the ground truth between itself and each OD pair in N_s in that experiment, and take the average correlation. We know that among different experiments, even the same OD pair in N_u will have different performance since inputs are changed, e.g. the split of N to

[9]Another reason is that the models underfit as supervision becomes inadequate.

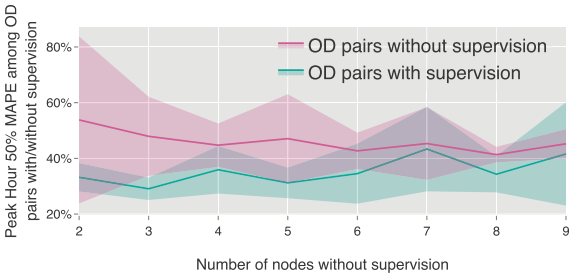

Fig. 12: Median MAPE of OD pairs with/without supervision on *peak hours* vs. number of nodes without supervision.

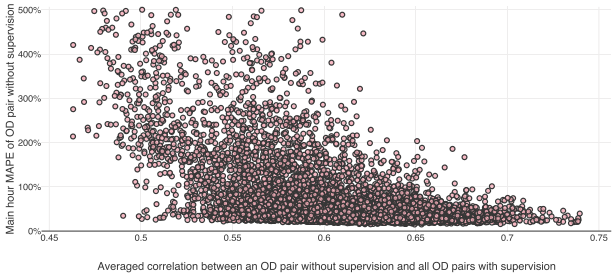

Fig. 13: The relationship between the MAPE of OD pairs in N_u and averaged correlation. Each point represents the main hour MAPE of an N_u OD pair in an experiment vs. the averaged correlation between itself and all OD pairs in N_s in that experiment.

N_s and N_u initially. However, separating the correlations out might still reveal insights between this factor and the final prediction performance. Fig. 13 shows that a larger correlation between an OD pair in N_u and all supervised OD pair tends to lead a better performance on main hour MAPE.

This result suggests using more OD pairs that share similar temporal pattern (which may lead to a high averaged correlation) with the N_u OD pairs as supervisions when utilizing our framework.

C. Magnitude ratio $\mathbf{R}$

Similar to previous subsection, we also plot the main hour MAPE vs. the averaged traffic ratio between N_u OD pairs and N_s OD pairs in Fig. 14. Generally, the points scattered to a "crescent" shape. If the averaged traffic ratio is greater than 1, the MAPE increases as the averaged traffic ratio increases; on the contrary, the MAPE decreases as the traffic ratio increases. The slope is steeper for the latter case, for the reason that a small averaged traffic ratio normally indicates an low-volume OD pair, which in return tends to have higher MAPEs as we explained previously.

This observation is intuitive. When an OD pair in N_u has volumes of much higher scale/lower scale than all OD pairs in N_s, it is hard to correctly "propagate" the volume of this (unlabeled) OD pair solely based on the supervisions. Therefore, when designing the network, it is beneficial to include both high volume OD pairs and low volume OD pairs

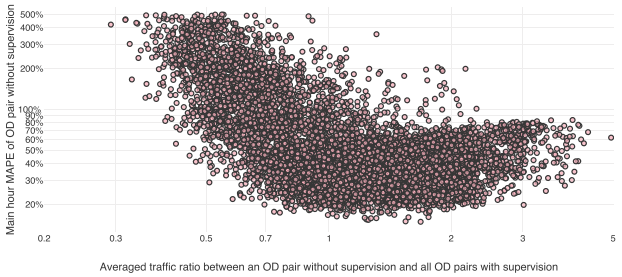

Fig. 14: The relationship between the MAPE of OD pairs and averaged traffic ratio. Each point represents the main hour MAPE of an N_u OD pair in an experiment vs. the averaged (ground truth implied) traffic ratio between itself and all OD pairs in N_s in that experiment. Both x and y axes are logarithmic.

(of course, relative to the volume range of OD pairs in N_u) in N_s, so that the averaged traffic ratio is not "extreme".

VI. Conclusions

In this work, we build a DL model based on RNN to learn the evolution pattern of OD flow time series and the mapping from subsample trajectory OD flow to overall OD flow on a real and complex road network with real data. Continuous real-time predictions can be made all at once for the OD pairs involved. A magnitude ratio $\mathbf{R}$ estimation model is built, which further enables a new loss inspired by LP for the proposed semi-supervised model, which learn and predict for OD pairs without any supervision signals. This innovation mitigates the issue of lacking part of supervisions, and further leverages the high coverage of phone trajectories.

The experiment demonstrates promising results. On half hour intervals, our model obtains median MAPE of 21.8% for main hours and 23.4% for peak hours for OD pairs with supervisions, and the corresponding median MAE is around 42 for main hours and peak hours, which is 1.4 on average per min. For the OD pairs without supervision, the model obtains median MAPE of around 33% for main and peak hours. Several important factors are analyzed to give insights for utilizing our models.

All of our results, unlike the baseline KF model, do not rely on continuous AVI data input. The model purely uses app recorded trajectories to make prediction during inference, even though the trajectories only contribute a small portion of all travels. The results encourage learning and prediction on traffic nodes not limited to where detectors are available. It also shows potential of usage in industry to develop a real-time online prediction system, as GPS-enabled smartphones are widely used and can continuously generate big data of trajectory with wide spatial coverage.

There could be several future works. First, on data side, the raw AVI data are still subject to various aforementioned adverse factors. Before new technologies emerge, how to remediate the data is an important area. Second, on the model

side, we demonstrated a framework and one specific realization based on this framework that works for this example. The framework leaves spaces for improvement and adaptations. For example, we can explore better hyper-parameters, structures and types of the neural network; We can replace part of layers with *attention*, or create a more transferable model with pre-trained embedded *encoders*. For the $\mathbf{R}$ estimate, the available features are limited to our dataset. When more features, especially more details of the road network, are available, it is worthwhile to see if an improved model can be developed.

REFERENCES

[1] H. J. Zuylen and L. G.Willumsen, "The most likely trip matrix estimated from traffic counts," *Transportation Research Part B*, vol. 14, no. 3, pp. 281–293, 1980.

[2] M. G.H.Bell, "The estimation of origin-destination matrices by constrained generalized least squares," *Transportation Research Part B*, vol. 25, no. 1, pp. 13–22, 1991.

[3] H. Yang, Y. Iida, and T. Sasaki, "The equilibrium-based origin-destination matrix estimation problem," *Transportation Research Part B*, vol. 28, no. 2, pp. 23–33, 1994.

[4] H. Yang, "Heuristic algorithms for the bilevel origin-destination matrix estimation problem," *Transportation Research Part B: Methodological*, vol. 29, no. 4, pp. 231–242, 1995.

[5] X. Zhou and H. S. Mahmassani, "Dynamic origin-destination demand estimation using automatic vehicle identification data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 105–114, 2006.

[6] "Trip matrix and path flow reconstruction and estimation based on plate scanning and link observations," *Transportation Research Part B: Methodological*, vol. 42, no. 5, pp. 455 – 481, 2008.

[7] "Norm approximation method for handling traffic count inconsistencies in path flow estimator," *Transportation Research Part B: Methodological*, vol. 43, no. 8, pp. 852 – 872, 2009.

[8] J. Durbin and S. J. Koopman, *Time series analysis by state space methods*. Oxford University Press, 2012, vol. 38.

[9] K. Ashok and M. Ben-Akiva, "Dynamic origin-destination matrix estimation and prediction for real-time traffic management systems. transportation and traffic theory," *Proceedings of the 12th ISTTT. Elsevier, Amsterdam*, pp. 465–484, 1993.

[10] M. Bierlaire and F. Crittin, "An efficient algorithm for real-time estimation and prediction of dynamic od tables," *Operations Research*, vol. 52, no. 1, pp. 116–127, 2004.

[11] C. Chen, J. Hu, Q. Meng, and Y. Zhang, "Short-time traffic flow prediction with arima-garch model," in *Intelligent vehicles symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 607–612.

[12] L. Zhang, J. Ma, and J. Sun, "Examples of validating an adaptive kalman filter model for short-term traffic flow prediction," in *CICTP 2012: Multimodal Transportation Systems—Convenient, Safe, Cost-Effective, Efficient*, 2012, pp. 912–922.

[13] L. R. L. Ojeda, A. Y. Kibangou, and C. C. De Wit, "Adaptive kalman filtering for multi-step ahead traffic flow prediction," in *2013 American Control Conference (ACC 2013)*, 2013.

[14] J. Guo, W. Huang, and B. M. Williams, "Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, 2014.

[15] Z. Lu, W. Rao, Y.-J. Wu, L. Guo, and J. Xia, "A kalman filter approach to dynamic od flow estimation for urban road networks using multi-sensor data," *Journal of Advanced Transportation*, vol. 49, no. 2, pp. 210–227, 2015.

[16] L. Lv, M. Chen, Y. Liu, and X. Yu, "A plane moving average algorithm for short-term traffic flow prediction," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2015, pp. 357–369.

[17] D.-w. Xu, Y.-d. Wang, L.-m. Jia, Y. Qin, and H.-h. Dong, "Real-time road traffic state prediction based on arima and kalman filter," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 2, pp. 287–302, 2017.

[18] L. Mussone, S. Grant-Muller, and H. CHEN, "A neural network approach to motorway od matrix estimation from loop counts," *Journal of Transportation Systems Engineering and Information Technology*, vol. 10, no. 1, pp. 88–98, Feb. 2010.

[19] J. Liu, W. Wang, X. Gong, X. Que, and H. Yang, "A hybrid model based on kalman filter and neutral network for traffic prediction," in *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, vol. 2. IEEE, 2012, pp. 533–536.

[20] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[21] H. Yao, X. Tang, H. Wei, G. Zheng, Y. Yu, and Z. Li, "Modeling spatial-temporal dynamics for traffic prediction," *arXiv preprint arXiv:1803.01254*, 2018.

[22] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *31st AAAI Conference on Artificial Intelligence*, 2017.

[23] X. Cheng, R. Zhang, J. Zhou, and W. Xu, "Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[24] D. Wang, Y. Yang, and S. Ning, "Deepstcl: A deep spatio-temporal convlstm for travel demand prediction," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

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

[26] X. Xiong, K. Ozbay, L. Jin, and C. Feng, "Dynamic prediction of origin-destination flows using fusion line graph convolutional networks," *arXiv preprint arXiv:1905.00406*, 2019.

[27] C. Antoniou, M. Ben-Akiva, and H. Koutsopoulos, "Incorporating automated vehicle identification data into origin-destination estimation," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1882, pp. 37–44, Jan. 2004.

[28] S. M. Eisenman and G. F. List, "Using probe data to estimate od matrices," *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems*, pp. 291–296, 2004.

[29] R. Ásmundsdóttir, "Dynamic od matrix estimation using floating car data," Master's thesis, Delft University of Technology, Mar. 2008.

[30] "Estimation of time-varying od demands incorporating fcd and rtms data," *Journal of Transportation Systems Engineering and Information Technology*, vol. 10, no. 1, pp. 72 – 80, 2010.

[31] P. Cao, T. Miwa, T. Yamamoto, and T. Morikawa, "Bilevel generalized least squares estimation of dynamic origin-destination matrix for urban network with probe vehicle data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2333, pp. 66–73, 2013.

[32] X. Yang, Y. Lu, and W. Hao, "Origin-destination estimation using probe vehicle trajectory and link counts," *Journal of Advanced Transportation*, p. 18, 2017.

[33] F. Calabrese, G. Di Lorenzo, L. Liu, and C. Ratti, "Estimating origin-destination flows using mobile phone location data," *IEEE Pervasive Computing*, vol. 10, no. 4, pp. 36–44, 2011.

[34] "Development of origin-destination matrices using mobile phone call data," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 63–74, 2014.

[35] Y. Sheffi, *Urban transportation networks*. Prentice-Hall, Englewood Cliffs, NJ, 1985.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[37] F. A. Gers, J. A. Schmidhuber, and F. A. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[38] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Carnegie Mellon University, Tech. Rep., 2002.

[39] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[40] W. H. Kruskal, "Ordinal measures of association," *Journal of the American Statistical Association*, vol. 53, no. 284, pp. 814–861, 1958.

[41] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.