Transportation Research Part C 129 (2021) 103241

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Day-to-day dynamic origin–destination flow estimation using connected vehicle trajectories and automatic vehicle identification data

Yumin Cao [a], Keshuang Tang [a,*], Jian Sun [a], Yangbeibei Ji [b]

[a] Key Laboratory of Road and Traffic Engineering, Ministry of Education & College of Transportation Engineering, Tongji University, No. 4800, Cao'an Road, Shanghai 201804, China

[b] School of Management, Shanghai University, No. 99, Shangda Road, Shanghai 200444, China

ARTICLE INFO

Keywords:
Dynamic OD estimation
Connected vehicle
Automatic vehicle identification data
Day-to-day traffic modeling
Self-supervised learning

ABSTRACT

Dynamic vehicular origin–destination (OD) flow is a fundamental component of traffic network modeling and its estimation has long been studied. Although ideal observing conditions and behavioral assumptions are often indispensable for estimation, day-to-day traffic recurrences and variations are seldom utilized to improve the estimation performance. In this paper, we propose a new method to recover day-to-day dynamic OD flows using both connected vehicle (CV) trajectories and automatic vehicle identification (AVI) observations. The method involves two modules: the first module provides reliable prior OD flows given limited observations, while the second module seeks the optimal estimates based on the prior OD flows. In the first module, linear projection is extended to consider temporal and spatial variation of the CV penetration rate, and non-negative Tucker decomposition (NTD) is adopted to address the data sparsity issue caused by the low CV penetration rate. In the second module, a self-supervised learning model called the latency-constrained autoencoder (LCAE) is established to search for the optimal OD flows according to the priors with given robust latent features. To avoid local minima and ensure consistency between estimates, a novel algorithm called adaptive sub-sample correction (ASC) is proposed and integrated into the optimization process of LCAE, which can iteratively correct the most inconsistent samples based on the day-to-day traffic flow characteristics. The proposed method is examined on an empirical urban arterial network, a calibrated simulation network, and a synthetic large-scale grid network. Our results indicated that the proposed method requires very few AVI detectors and CV trajectories to achieve competitive estimation performance against two benchmark models. Furthermore, general robustness to several factors with respect to observing conditions and data quality was investigated, and satisfactory scalability was also demonstrated in terms of both estimation accuracy and computational cost.

1. Introduction

Dynamic origin–destination (OD) flow reveals the time-dependent travel demand on road networks. It serves as the fundamental input for dynamic traffic assignment (DTA) models as well as for network optimization programs (Arsava et al., 2018; Peeta &

* Corresponding author.
E-mail address: tang@tongji.edu.cn (K. Tang).

https://doi.org/10.1016/j.trc.2021.103241
Received 4 June 2020; Received in revised form 21 May 2021; Accepted 30 May 2021
Available online 9 June 2021

Ziliaskopoulos, 2001). The dynamic OD flow fluctuates from day-to-day due to variations and stochasticity in trip patterns. Thus, active traffic network management also requires accurate estimation of day-to-day dynamic OD flows to handle the uncertainty of traffic demand.

Despite the extensive studies across decades, obtaining accurate time-dependent OD flows given network observations remains challenging due to the observability issue in traffic networks (Castillo et al., 2008a). The observations from the network are much less than the unknown OD flows, and thus, models may not produce a unique solution. Under such circumstances, the existing models often start with a prior OD estimate and solve an optimization program to identify the solution that is most consistent with available observations and assumptions. Then, the objective of such program is to minimize the deviation between estimated and observed (or prior) variables while maintaining network flow conservation described by DTA process. However, a reliable prior may not always exist, especially at central business districts or under rapid urbanization in many developing countries. Besides, DTA models are usually established based on departure time and route choice behavior assumptions to approach user equilibrium status, which could largely deviate from the realistic situation (Yildirimoglu & Kahraman, 2017; Zhu & Levinson, 2015). Furthermore, most current studies have focused on within-day situations while considering deterministic OD flows, such as estimation of the morning peak period; they have not considered day-to-day recurrence and variation of OD flows. Only a few studies have dealt with day-to-day OD flows, but they have mainly estimated the mean, variance and covariance assuming certain OD demand distribution, e.g. multivariate normal distribution (Ma & Qian, 2018b; Shao et al., 2014).

In recent years, connected vehicles, such as vehicles of DiDi and Uber that equipped with GPS unit or drivers that use navigation service on their mobile phone, have emerged as a promising mobile data source because they can provide detailed and accurate traffic flow information. Meanwhile, vehicle re-identification systems have also been rapidly deployed in many countries. The main components of these systems are automatic vehicle identification (AVI) detectors that could uniquely identify each vehicle, including radio frequency identification device (RFID)-based detectors, blue-tooth detectors, and license plate recognition (LPR) devices. Among them, the LPR cameras are mostly used in China, and the data could be accessed by the supplier as well as law enforcement apartment. These data sources can directly provide OD and path-related observations. Owing to the availability of these day-to-day continuous and multi-source heterogeneous observations, external priors and unrealistic assumptions may no longer be required (Ma & Qian, 2018b; Yang et al., 2018). Nevertheless, according to our literature review, few efforts have been devoted to fully utilize the day-to-day observations of these emerging sensors to address the external prior OD usage issue and assumptions in DTA modeling. Therefore, in this paper, we eliminate historical priors and behavioral assumptions and infer OD flows using both CV trajectories and AVI observations via a purely data-driven method.

Early attempts on OD flow estimation mainly focused on the modeling framework based on fixed link counts, including the entropy minimization model (Van Zuylen & Willumsen, 1980), the maximum likelihood estimator (Spiess, 1987), Bayesian inference method (Maher, 1983), and generalized least squares (GLS) models (Bell, 1991; Cascetta, 1984). Among them, the GLS-based models have been most frequently extended and tested, including bi-level and single-level GLS models. The bi-level models consider the effects of congestion, in which the upper level minimizes deviation terms in the form of least square error and the lower level performs DTA based on inference of the equilibrium states (Yang et al., 1991; Yang et al., 1992). To better describe the DTA process, simulators are often incorporated and models are then solved by the stochastic perturbation simultaneous approximation algorithm (Lu et al., 2015). However, this bi-level structure usually leads to non-convexity; therefore, single-level models have been proposed based on relaxation techniques (Lu et al., 2013; Nie & Zhang, 2010). Several recent studies have also shed light on data-driven approaches. Ma and Qian (2018a) used high-granular traffic count and speed data to estimate multi-year 24/7 dynamic OD demands, where a GLS model is established given estimated assignment ratio; Krishnakumari et al. (2020) also uses count and speed data to estimate OD matrix under the mild assumption of proportional flow on shortest paths. Both studies showed satisfactory results and demonstrated that data-driven approaches are promising and efficient. Generally, with only aggregated traffic counts available, reliable priors and effective assumptions are indispensable to fill the observability gap. However, obtaining reliable initial OD matrices is generally difficult and labor-intensive, and the aforementioned assumptions could possibly deviate from realistic conditions.

In terms of AVI observations, several researchers have derived travel times and traffic counts from AVI detectors and have conducted OD flow estimation and prediction by integrating link counts with these observations (Dixon & Rilett, 2002; Zhou & Mahmassani, 2006). In addition to traffic counts, these detectors can reproduce partial paths of vehicles and thus provide further flow and travel time constraints to facilitate path and OD flow estimation. Following this direction, Bayesian methods have been adopted to recover paths (Castillo et al., 2008b; Castillo et al., 2008c; Mo et al., 2020), and state-space models, especially particle filtering (Feng et al., 2015; Rao et al., 2018; Yang & Sun, 2015), have been also introduced to probabilistically reconstruct the path of each individual vehicle. Subsequently, path and OD flows can be obtained by aggregation. Despite the promising results, these works require a large AVI coverage rate (e.g., 40–80%), which is rare occasion in a realistic network, especially large ones. In addition, reducing the uncertainties in the exact travel origin and destination is difficult using AVI-based OD flow estimation methods. Although travel paths can be effectively recovered between AVI detectors, the path from the origin to the first detected location (or last detected location to destination) can hardly be recognized, and route choice assumption is still necessary to address this issue.

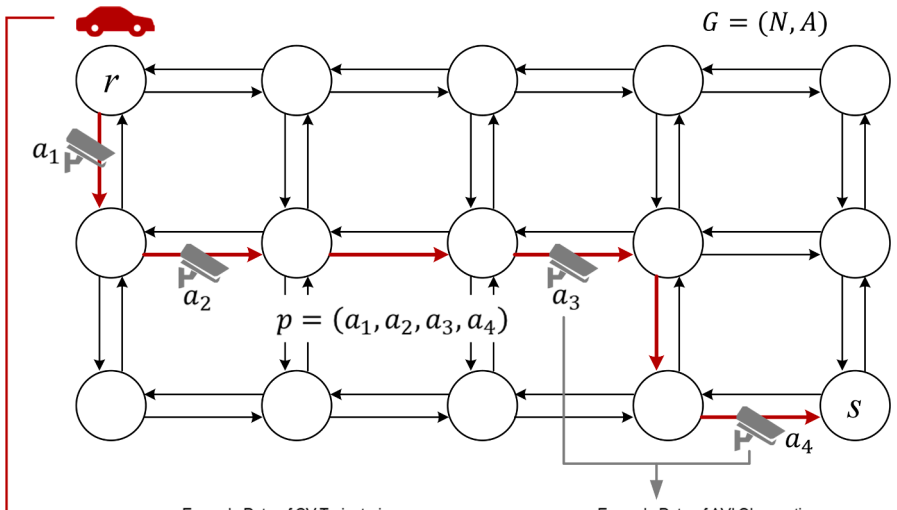
CVs have recently facilitated many tasks in traffic modeling including the OD flow estimation. Compared with fixed detectors, which use indirect variables to estimate OD flow and suffer from the observability issue, CVs can nearly cover the entire network and provide direct OD flow samples. With such high-coverage, time-continuous OD samples, the focus of modeling shifts from reducing the uncertainties of unobservable OD flow to measuring the reliability of sampled OD flow (i.e., CV OD flow). Following this direction, the quantity and quality requirements of probe data for estimating population OD flows have been discussed and examined by some early studies based on several toy network examples (Eisenman & List, 2004; Van Aerde et al., 1993), and penetration rates of 10–30% have been regarded as sufficient. Moreover, several studies have investigated the route choices and trip distributions of probe vehicles and

Table 1
List of Notations.

Network variables	
A	The set of all links
A^o	The set of links installed with AVI detectors
RS	The set of all OD pairs
K_{rs}	The set of recognized paths between OD pair rs
D	The set of days in the study period
I	The set of time intervals
P	The set of paired links
Traffic variables	
X_{irs}	The flow of OD pair rs during time interval i
X^v_{irs}	The CV flow of OD pair rs during time interval i
Q^o_{ia}	The flow observed by AVI on link a during time interval i
Q^o_{ip}	The flow observed by AVIs on paired links p during time interval i
Q^v_{ia}	The CV flow on link a during time interval i
Q^v_{ip}	The CV flow on paired links p during time interval i
π_{ia}	The penetration rate of CV on link a during time interval i
π_{ip}	The penetration rate of CV on paired links p during time interval i
ϵ_a	The missing detection rate of AVI on link a
ϵ_p	The missing detection rate of AVI on paired links p
Model variables	
$\mathscr{X}$	Target tensor for Tucker decomposition ($\mathscr{X} \in \mathbb{R}^{D\times I\times K}$)
F, G, H	Dimensions of the core tensor in Tucker decomposition
$\mathscr{C}$	Core tensor of Tucker decomposition ($\mathscr{C} \in \mathbb{R}^{F\times G\times H}$)
$\mathscr{B}$	Binary tensor for selecting entries to calculate loss ($\mathscr{B} \in \mathbb{R}^{D\times I\times K}$)
$\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$	Factor matrices of Tucker decomposition ($\boldsymbol{U} \in \mathbb{R}^{D\times F}, \boldsymbol{V} \in \mathbb{R}^{I\times G}, \boldsymbol{W} \in \mathbb{R}^{K\times H}$)
$\lambda, \lambda^{'}$	Weight of regularization term in optimization objectives
$\mathscr{R}_{\boldsymbol{\theta}}$	Encoder network parameterized by $\boldsymbol{\theta}$
$\mathscr{M}_{\boldsymbol{\omega}}$	Decoder network parameterized by $\boldsymbol{\omega}$
$\mathscr{L}$	Optimization objective (or loss function) of LCAE
$\boldsymbol{v}_o, \boldsymbol{v}_e, \boldsymbol{v}_t$	One-hot input vector, embedded vector and target vector in vector embedding
$\boldsymbol{\Lambda}, \boldsymbol{\Psi}$	Weight matrices for vector embedding
Z	Sample set for training LCAE
BS	Batch size for mini-batch gradient descent
T	Number of training epochs for neural network
$\boldsymbol{Z}, \boldsymbol{M}$	Prior OD flow sample and latent vector sample in the sample set Z
$\boldsymbol{z}_j$	Sub-sample of z-th sample in sample set Z
δ	The step size (or learning rate) of gradient descent
α, β, γ	Parameters of *Beta* distribution and value sampled from $\mathrm{Beta}(\alpha, \beta)$
$DTA(\cdot)$	Generalized function of DTA
$LID(\cdot)$	Locally Intrinsic Dimensionality (LID) calculator
$f(\cdot, \cdot)$	Generalized measure of deviation

have demonstrated the feasibility of using projected probe OD as prior OD for estimation (Ásmundsdóttir, 2008; Ásmundsdóttir et al., 2010). Based on these insights, bi-level GLS models with exogenous DTA simulators have been employed to further incorporate probe vehicles or floating car trajectories (Cao et al., 2013; Carrese et al., 2017). In another benchmark study, Yang et al. (2017) formulated two single-level GLS models based on both probe vehicle trajectories and link counts. The route choices of probe vehicles were used to compute traffic assignment fractions, and the relationship between OD and link penetration rates was established; thus, there were few assumptions made in this model. Generally, these studies projected the CV OD flow as an estimate or prior according to presumed or derived penetration rates. However, few studies have explicitly considered the error of projected OD flow in the model. Thus, optimization models may be trapped in local minima and the solution may be inaccurate. Furthermore, these existing studies often assumed penetration rates of more than 10%, which is considered to be rare in the current CV market (Tan et al., 2019; Yao et al., 2019), and the estimation performance rapidly deteriorated with a decrease in penetration rate.

To summarize, with the availability of detailed and continuous observations, recent AVI-based and CV-based studies require less historical data and assumptions compared with link count-based studies. The superiority in estimation accuracy has also been demonstrated under certain observation conditions, which often deviates from the currently prevailing market status. However, two



Example Data of CV Trajectories

Vehicle ID	Timestamp	Longitude	Latitude
151233	2021/2/20 0:00:47	121.219096	31.276463
151233	2021/2/20 0:00:50	121.218704	31.276688
151233	2021/2/20 0:00:53	121.217980	31.277174

Example Data of AVI Observation

Detector ID	Timestamp	Vehicle ID
4441	2021/2/20 0:00:54	151233
4441	2021/2/20 0:01:05	148765
4441	2021/2/20 0:15:26	150523

Fig. 1. Illustration of research background and notations.

research problems still need to be addressed. The first problem concerns obtaining a reliable prior estimate, when there are limited available AVI detectors or CVs. Several studies have recognized the minimum level of AVI coverage rate or CV penetration rate; however, only a few have explicitly dealt with the accompanying problem of data sparsity. The second research problem concerns ensuring an optimal estimate according to the prior. Most existing methods rely on the DTA process to establish constraints, such as link flow conservation and travel time consistency. In this way, estimation errors are prone to increase because of either unrealistic assumption regarding user behavior or improper simplification of the DTA process.

In view of the first problem, we translated the limited observation problem into the reliability of the CV OD flow projection and problem of data sparsity imputation. Linear projection is extended based on fusion of AVI observations and CV trajectories to reduce the bias in the prevailing simple scaling, and a low-rank approximation method facilitated by the multi-dimensional tensor is adopted to deal with the sparsity problem in projected OD flow. To deal with the second research problem, we propose to robustly reconstruct prior OD flows via self-supervised learning. Based on the day-to-day traffic flow characteristics, we developed an adaptive correction algorithm to dynamically adjust the objective surface during optimization. Thus, the local minima could be avoided and the final estimates could be obtained without any theoretical assumptions. The proposed methodology is comprehensively examined on an empirical dataset from an urban arterial, a simulation dataset from a regional network, and a synthetic large-scale grid network. The proposed method exhibited satisfactory estimation accuracy, robustness to several influencing factors, and good scalability.

In general, the main contributions of this paper are three-fold:

(1) A novel methodology for estimating day-to-day dynamic OD flow estimation fusing AVI observations and CV trajectories is proposed. Within this methodology, the characteristics of both data sources are effectively utilized.
(2) Linear projection is extended to deal with variations in CV penetration rates, and non-negative Tucker decomposition (NTD) is applied to impute sparsity values in projected OD flows. Reliable prior OD flows are provided through the two steps even under limited observing conditions.
(3) A self-supervised learning model called the latency-constrained autoencoder (LCAE) is established to search for the optimal solution based on the estimated prior. Meanwhile, an adaptive sub-sample correction (ASC) algorithm is proposed to incorporate day-to-day traffic flow characteristics to facilitate the optimization process.

2. Methodology

2.1. Background and notations

Table 1 presents all the variables and notations used in this paper. Considering a road network G specified by link set A and OD pair set RS, and an analysis period consisting of multiple consecutive days represented by D, for each OD pair $rs \in RS$, a path set K_{rs} exists; for each day $d \in D$, a number of identical time intervals are split and denoted by I. Here, a special note should be given to the day-to-day context of this paper, as the evolutionary dynamics is out of concern and the focus is placed on the recurrence and variation of

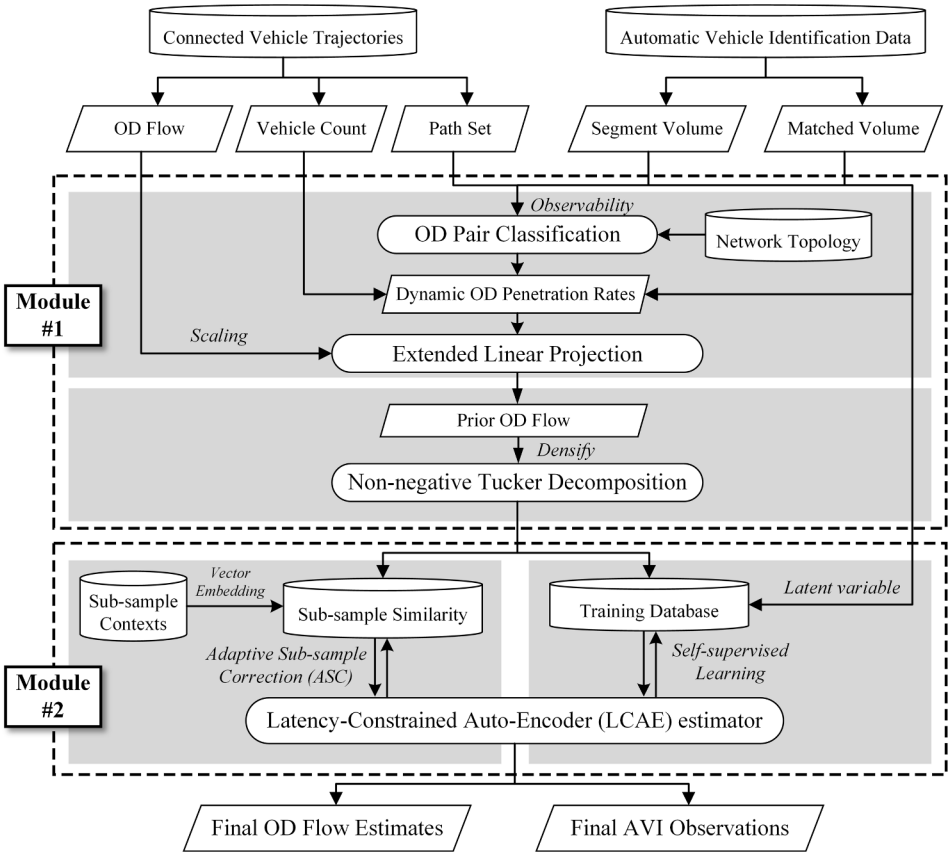


Fig. 2. General Framework.

observations and OD flow. Among the network, a number of links, represented by A^o, are installed with AVI detectors, and a portion of vehicles are regarded as CVs that can accurately transmit the travel information with high granularity (e.g., 1–5 s). Note that here we distinguish CVs from traditional probe vehicles or floating cars by its transmission resolution and accuracy. Thus, the location and map matching errors are also not considered. For several links $a_1, a_2, \cdots, a_p \in A^o$, if at least one vehicle traversed all these links in a single trip, then these links ordered in traversing sequence form paired link $p = (a_1, a_2, \cdots, a_p)$ belongs to the paired link set P. In other words, any AVI sequence that is at least traversed by one vehicle path forms a paired link p. The illustration of the background and several notations are presented in Fig. 1. Hereafter, the superscripts $\cdot^o$ and $\cdot^v$ indicate that the variable is observed by AVI detectors and CVs, respectively. The splash symbol $\tilde{\cdot}$ and hat symbol $\hat{\cdot}$ indicate the prior and estimated variables, respectively.

2.2. General framework

This study aims at estimating dynamic OD flows in a day-to-day context based on continuous AVI observations and CV trajectories. This section introduces the general framework of the proposed method.

As is presented in Fig. 2, the proposed method consists of two major modules. The first module, called the prior estimation module, focuses on obtaining a preliminary estimate of OD flow, i.e., prior OD flow. As mentioned in the previous section, using linearly projected CV OD flow has been previously justified to be feasible. This projection process is also applied and extended in this study by considering the temporal variation and different observability conditions of OD pairs. Then, the projected OD flows are expressed by a 3-mode tensor. Furthermore, the NTD method is applied to approximate the sparse entries considering multiple dependencies (e.g., time-of-day dependency).

The second module, called the prior correction module, focuses on optimizing or correcting the prior OD flow and output the final estimates. The main process is to search for a better estimate by minimizing the deviation terms and simultaneously correcting the priors. A self-supervised neural network, referred to as the latent-constrained autoencoder (LCAE), is introduced to approximate the prior OD flow in a parametric manner given robust latent variables. Meanwhile, an ASC algorithm is integrated into the training process of LCAE to adaptively interpolate the most inconsistent prior OD flows by similar ones. By vector embedding, discrete information (e.g., time-of-day, OD pair) is transformed into continuous vectors, and the similarities are measured. The method converges through an iterative process and provides the final OD flow estimates.

2.3. Module 1: Prior estimation module

2.3.1. Extended linear projection

In this step, the CV OD flows are projected to obtain prior OD flows via the penetration rates estimated from a fusion of both data sources. Linear data projection is a widely applied methodology for unbiased estimation of traffic data (Wong & Wong, 2019). Under the OD estimation problem, directly scaling CV OD flows or sampled OD flows by the observed or estimated penetration rates also belongs to this category. Numerous existing studies have adopted this strategy to obtain a general estimate of the population OD flow (Eisenman & List, 2004; Van Aerde et al., 1993; Yang et al., 2017). However, most current studies simply use network or period (e.g., one hour) average penetration rates as the scaling factor, as presented in Eq. (1), by which the spatial and temporal variability is largely neglected and systematic distortion may be introduced.

$$\widetilde{X_{irs}} = \frac{|A^o| \cdot X^v_{irs}}{\sum_{a \in A^o} \pi_{ia}} \tag{1}$$

Thus, to deal with the issue, we extended the basic procedure of linear OD flow projection in Eq. (1) to obtain a more reliable prior OD flow that could reveal the general travel pattern of road networks. By dividing the study period (e.g., a month) into short time intervals (e.g., 10–15 min), the temporal variability of the penetration rates could be better emphasized. In terms of the spatial variability, local paths recovered by AVI detectors and sampled paths from CVs could favor better description. Nevertheless, AVI detectors could not successfully identify every passing vehicle, leading to missing detection, and the missing detection rate is defined as the number of identified vehicles over the number of passed vehicles. Here, the CVs and AVI detectors are assumed to share the same encryption rules for each unique vehicle. In other words, we could recognize that if a CV is detected by the AVI or not by matching the unique vehicle identity. In addition, as the missing detection is often caused by system error and is less affected by the vehicle characteristics, each vehicle on link $a \in A^o$ is assumed to have the same probability of being missed. Subsequently, the missing detection at different links are mutually independent. Then, the missing detection of link a follows the binomial distribution with probability ϵ_a, which could be estimated using the fraction of passing CVs that are not detected among all CVs, as presented in Eq. (2). Afterward, the missing detection rate of paired link p could also be recovered using Eq. (3). Using the estimated missing detection rate, the link flow as well as partial path flow on a paired link p could be recovered. Notably, the missing detection rate is estimated from the aggregated data of the entire analysis period because the error often come from a systematic problem and the missing detection rate is generally stable across a period of time (e.g., a few weeks).

$$\epsilon_a = 1 - \frac{Q^{ov}_a}{Q^v_a}, a \in A^o \tag{2}$$

$$\epsilon_p = 1 - \prod_{a \in p} (1 - \epsilon_a), p \in P \tag{3}$$

According to the observability of AVI detectors, the OD pair set K could be split into three subsets—the undetected OD pair set RS_0, the detected OD pair set RS_1, and the matched OD pair set RS_2. Then, we have $RS = RS_0 \cup RS_1 \cup RS_2$. Hereafter, we refer to the matched OD pairs as OD pairs whose paths cover more than one AVI detector, detected OD pairs as OD pairs whose paths cover only one AVI detector, and undetected OD pairs as OD pairs whose paths do not cover any AVI detector. Notably, the OD pair classification criterion applies to any path between an OD pair. For example, at least one path between OD pair that belongs to RS_2 is observed by more than one AVI detector. For each OD pair subset, the penetration rates are calculated separately. The penetration rates on link a in each short time interval (e.g., 10 min), i.e., π_{ia}, are estimated using Eq. (4), and the penetration rates on paired link p, i.e.,π_{ip}, are estimated using Eq. (5).

$$\pi_{ia} = \frac{Q^v_{ia}}{Q^o_{ia}}, a \in A^o \tag{4}$$

$$\pi_{ip} = \frac{Q^v_{ip}}{Q^o_{ip}}, p \in P \tag{5}$$

Afterward, the penetration rates of different OD pairs could be approximated. Generally, the penetration rate of paths between the OD could be used as the approximation of as penetration rate of the OD pair. Therefore, for matched OD pairs with path-level observation, the penetration rate is approximated by the penetration rate on the recognized local paths, as presented in Eq. (6). For the detected OD pairs, the penetration rate is approximated using the link penetration rate it covers (path size equals to one), as presented in Eq. (7). As for the undetected OD pairs, there is no available information to estimate the penetration rate, we use the link volume-weighted network average penetration rate (the total number of CV over the total number of vehicles on all observed links) as the estimate, as presented in Eq. (8).

$$\pi_{irs} = \frac{\sum_{p \in P_{rs}} \pi_{ip} \cdot Q^o_{ip}}{|P_{rs}| \cdot \sum_{p \in P_{rs}} Q^o_{ip}}, rs \in RS_2 \tag{6}$$

$$\pi_{irs} = \pi_{ia}, a \in P_{rs} \cap A^o, rs \in RS_1 \tag{7}$$

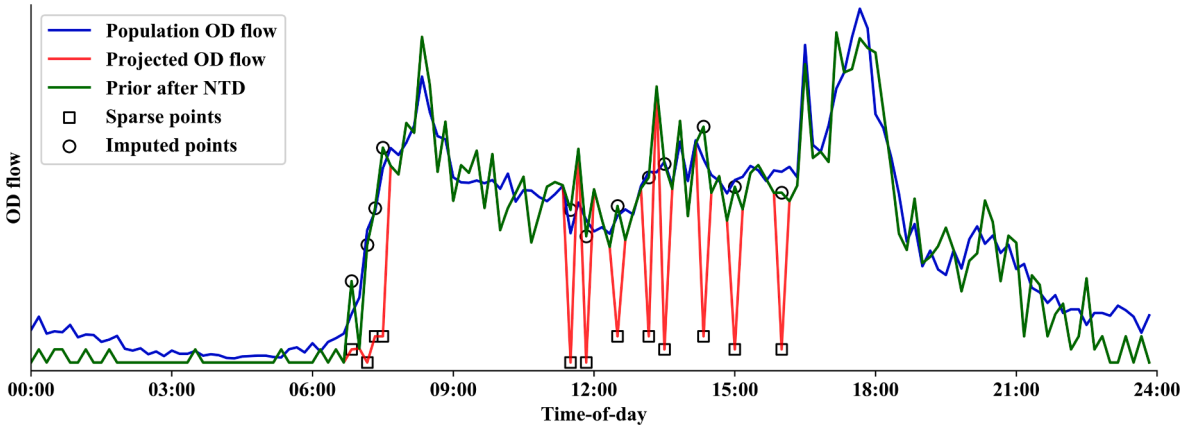


Fig. 3. Illustration of the sparsity issue.

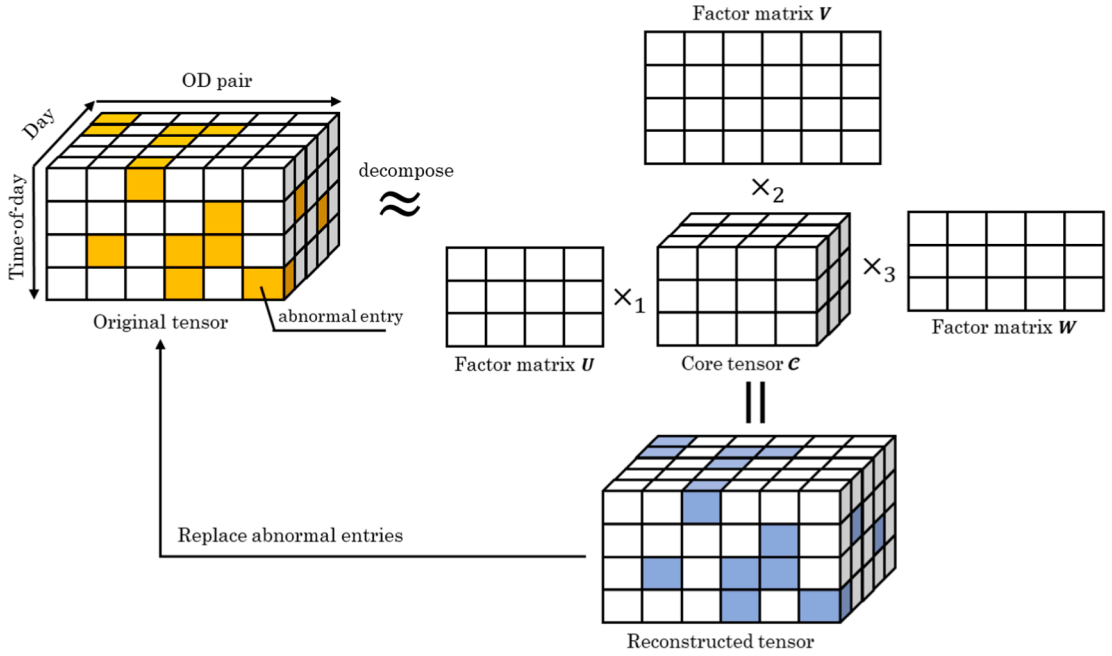


Fig. 4. Process of NTD.

$$\pi_{irs} = \frac{\sum_{a \in A^o} \pi_{ia} \cdot Q^o_{ia}}{|A^o| \cdot \sum_{a \in A^o} Q^o_{ia}}, rs \in RS_0 \tag{8}$$

Then, the prior OD flow can be separately projected using Eq. (9). Notably, if the approximated penetration rate equals to zero, the corresponding prior OD flow is set to be zero as well.

$$\widetilde{X_{irs}} = \frac{X^v_{irs}}{\pi_{irs}}, rs \in RS_0, RS_1, RS_2 \tag{9}$$

2.3.2. Non-negative Tucker decomposition (NTD)

Although the linear projection process is extended to consider the variation in penetration rates, the projected OD flows would probably face severe sparsity problems as CV OD flows, i.e., the numerator of Eq. (9), are often sparse. This sparsity has been indicated in several real-world case studies because CVs often cover less than 10% of the population vehicle (Tan et al., 2019; Yao et al., 2019; Zheng & Liu, 2017). From the perspective of OD pairs, this sparsity is caused by the stochasticity in CV attendance on OD pairs, while the population OD flow dynamics is often considerably more stable. For example, as illustrated in Fig. 3, the population OD flow smoothly transitioned from the morning peak period to mid-day off-peak period, whereas the CV OD flows during the mid-day off-peak

period (e.g., 10:00–16:00 in Fig. 2) encountered obvious sparse values (values that are considerably smaller than the true values and close to zero). This sparsity could be regarded as data corruption and could be imputed by generic time-series methods.

In recent years, high-order tensors have been identified as an efficient and effective data structure in OD estimation problems as it could express multi-dimensional dependencies in a compact and delicate data structure (Tang et al., 2020). Based on such data structure, the multi-dimensional dependencies can be mined and analyzed. One of the famous method that utilized high-order tensors is the low-rank approximation method, including CANDECOMP/PARAFAC decomposition, Tucker decomposition and several nonnegative variants (Kolda & Bader, 2009), have been extensively applied to spatiotemporal vehicle trajectory analysis and traffic data pattern mining because of their efficiency and effectiveness in capturing complex correlations (Chen et al., 2018; Naveh & Kim, 2019; Zhang et al., 2019). The shared core idea of these approaches is that the target data set is correlated in multiple dimensions, and thus abnormalities in the dataset could be imputed by simultaneously looking at all the dimensions. Here, the nonnegative version of Tucker decomposition (Kim & Choi, 2007) is applied to impute the abnormal entries in the projected OD flows.

First, the projected OD flow is transformed into a three-order tensor ($\mathscr{X}$) to express the multi-dimensional dependencies. The three modes (or dimensions) of the constructed tensor correspond to day, time-of-day, and OD pair. For instance, the entry (d, i, rs) in tensor $\mathscr{X}$ is the projected OD flow of OD pair rs on the i-th interval of the d-th day. Thus, the dimension of the constructed tensor $\mathscr{X}$ is $\mathbb{R}^{|D|\times|I|\times|RS|}$, as presented in the left-side tensor in Fig. 4.

Afterward, the sparse entries in the projected OD flow tensor could be approximated using NTD. As illustrated in Fig. 4, The NTD method decomposes the original three-order tensor into a core three-order tensor ($\mathscr{C}$) multiplied by three factor matrices $(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})$ along each dimension of the tensor. For a non-full rank matrix, certain vectors can be represented by the linear combination of a set of linearly independent vectors. In the context of tensors, the rank-n refers to the rank of n-th dimension of the tensor and could be viewed as the number of linearly independent components on this dimension. Hence, the NTD receives pre-specified rank-n of the core tensor, which is often lower than the full rank of each dimension, and uses these independent components to re-form the original tensor (Kolda & Bader, 2009). Therefore, the reconstructed tensor can deal with noises and missing values.

This decomposition and recombination process could be transformed into an optimization program presented in Eq. (10). the core tensor and factor matrices can be obtained by minimizing the least square error plus a L2 regularization term. $\|\hat{\mathrm{A}} \cdot\|_F^2$ refers to the square of the Frobenius norm of a matrix or tensor, and $\times_n$ refers to the mode-n product, i.e., matrix product between the dimension n of a tensor and a matrix. By forcing the sparse entries in tensor $\mathscr{X}$ as zero in tensor $\mathscr{B}$, and non-sparse entries as one, the NTD will minimize only the reconstruction error of non-sparse entries and produce estimates for sparse entries. Notably, the term error may often refer to the deviation between estimated and true value. Here, to recover the sparse entries, the non-sparse entries, though not perfectly true value, are viewed as true values during the NTD process.

$$\min_{\mathscr{C},\boldsymbol{U},\boldsymbol{V},\boldsymbol{W}} \|\mathscr{B} \cdot (\mathscr{X} - \mathscr{C} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{W})\|_F^2 + \frac{\lambda}{2}\left(\|\mathscr{C}\|_F^2 + \|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2 + \|\boldsymbol{W}\|_F^2\right) \tag{10}$$

This special program could be solved using high-order orthogonal iteration (HOOI) initialization and multiplicative updating for a number of iterations. The HOOI algorithm is the singular value decomposition method generalized from matrices for high-order tensors, and the multiplicative updating method is an efficient and compact updating method for core tensors and factor matrices. Readers interested in the detailed derivation of the HOOI algorithm and updating formulation could refer to De Lathauwer et al. (2000) and Kim and Choi (2007). After approximation using NTD, the projected OD flow is then updated to address the sparsity issue and obtain reliable prior OD flows.

2.4. Module 2: Prior correction module

2.4.1. Latency-constrained autoencoder (LCAE)

Considering the output of the first module as the input, the second module aims to optimize or correct prior OD flows. In fact, a prior OD flow can be viewed as the addition of a true OD flow X_{irs} and its random noises ε_{irs}. Then, the most straightforward method to recover the true OD flow from the prior OD flow is to recognize the robust components in prior OD flows and remove the noises.

Representation learning, or feature learning, is a prominent field in the machine learning community. It can largely facilitate extraction of robust features and denoising of prior OD flows. Representation learning involves identifying and disentangling the underlying explanatory factors hidden in data, especially complex but highly structured ones (LeCun et al., 2015). Probabilistic graphical models (PGMs) as well as several deep neural networks (DNNs) belong to this category. PGMs, such as hidden Markov chains and Bayesian networks, often use latent variables along with posterior distributions to express the data representation, but they could be intractable when the dimensions of dependency increase (Bengio et al., 2013). In terms of DNNs, most of the prevalent ones (e.g., convolutional neural network) use the supervised training mode to approximate conditional or joint distribution of input data and label, and therefore interpreted as approximator rather than feature learner. Besides. in our case, the supervision mode would require true OD flows as labels, which may not even exist. Meanwhile, auto-encoders (AE), a special class of DNN that work in self-supervision mode, have been suggested as an extraordinary feature learner and exhibited promising performance in learning robust features by setting up parametric maps between data and latent variables (Vincent et al., 2008).

Therefore, based on the existing architecture of the AEs, we proposed a latency-constrained autoencoder (LCAE) in this study to denoise the prior OD flows. The general architecture of the LCAE is illustrated in Fig. 5. As described previously, the LCAE is a self-supervised as the model input also serves as the target label. It comprises two major parts—the encoder ($\mathscr{M}_{\boldsymbol{\theta}}$, parametrized by $\boldsymbol{\theta}$) and the decoder ($\mathscr{R}_{\boldsymbol{\omega}}$, parametrized by $\boldsymbol{\omega}$). Then, the denoising mechanism of LCAE could be attributed to the following three aspects:

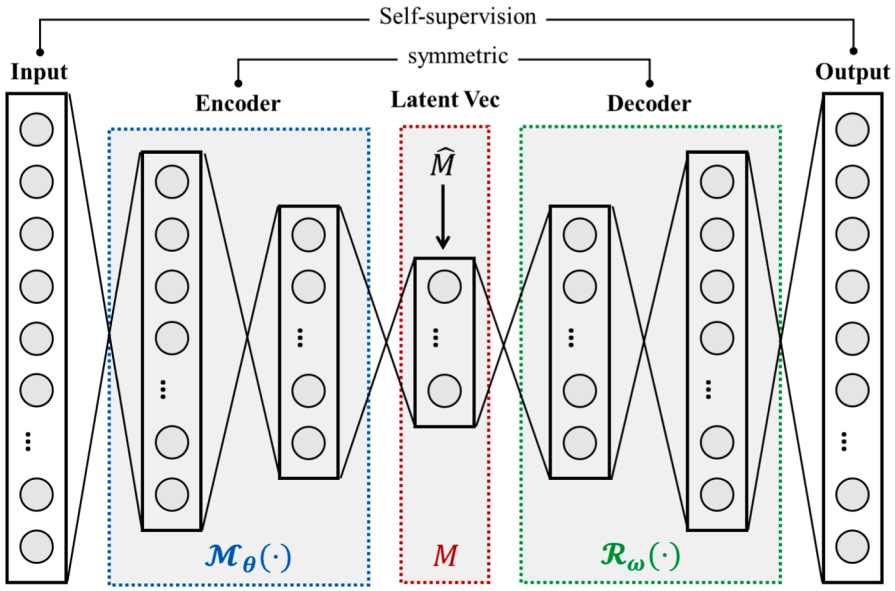


Fig. 5. Illustration of LCAE architecture.

robust latent vector, limited feature dimension, and the regularization term. Firstly, the encoder encodes prior OD flows into robust latent variables that are hardly affected by noise, whereas the decoder recombines them to approximate the true OD flows. Secondly, within the encoder, the number of neurons in each layer is decreasing as the network go deep and the trend is the opposite in the decoder. In other words, only a limited number of features could be recognized as robust for recovering the OD flow. Last but not least, the L2 regularization terms penalize the larger parameters in both encoder and decoder, which means the encoder and decoder is supposed to approximate the OD flow and link flow by looking evenly at effective features rather than relying on a few features and abandon those neutral features as much as possible. Here, the flow observations from AVI detectors are used as the latent variables, including link flows (Q^o_{ia}) and matched flows (Q^o_{iab}). The reason for this manipulation is simple and straightforward: among the observations of AVI and CVs, the flows are the most representative and reliable ones. Both the encoder and decoder comprise multiple generic neural network layers, and the simplest form, i.e., a fully connected layer with batch normalization (Ioffe & Szegedy, 2015) is used for each layer.

To train the LCAE model, the optimization objective is defined as the sum of three terms: the first term is the deviation between the encoded and observed latent vectors, which supervises the robust latent variables; the second term is the self-supervised error, which contributes to the recovery of OD flow; and the third term is the L2 norm penalty on model parameters, which is in essence penalize the model complexity to prevent overfitting on the prior OD flows. Then, the objective can be formulated as Eq. (11)–(13). Note that two types of flow observations in the latent vector are externally and separately normalized to ensure a smoother loss surface and facilitate the training process (Ioffe & Szegedy, 2015).

$$\mathscr{L} = \|\boldsymbol{M} - \boldsymbol{Q}\|_2^2 + \|\boldsymbol{R} - \boldsymbol{Z}\|_2^2 + \lambda^{`}\left(\|\boldsymbol{\theta}\|_2^2 + \|\boldsymbol{\omega}\|_2^2\right) \tag{11}$$

$$\boldsymbol{M} = \mathscr{M}_{\boldsymbol{\theta}}(\boldsymbol{Z}) \tag{12}$$

$$\boldsymbol{R} = \mathscr{R}_{\boldsymbol{\omega}}(\boldsymbol{M}) \tag{13}$$

where $\boldsymbol{Q} = \left[\frac{\boldsymbol{Q}_a - \mu_Q}{\sigma_Q}, \frac{\boldsymbol{Q}_p - \mu_Q}{\sigma_Q}\right]^T, \boldsymbol{Q}_a = \left[Q^o_{ia}, \cdots, Q^o_{(i+\Delta)a}\right], \boldsymbol{Q}_p = \left[Q^o_{ip}, \cdots, Q^o_{(i+\Delta)p}\right], \boldsymbol{Z} = \left[\frac{X_{i1} - \mu_X}{\sigma_X}, \cdots, \frac{X_{(i+\Delta)1} - \mu_X}{\sigma_X}, \cdots, \frac{X_{i|RS|} - \mu_X}{\sigma_X}, \cdots, \frac{X_{(i+\Delta)|RS|} - \mu_X}{\sigma_X}\right]^T$; μ_Q, σ_Q is the mean and standard deviation of the whole dataset of AVI flow, respectively; μ_X, σ_X are the mean and standard deviation of the prior OD flow, respectively; Δ is the window size for generate each sample $\boldsymbol{Z}$; and $\lambda^{`}$ is the weight of L2 regularization.

Using the optimization process with any prevalent stochastic gradient descent algorithm, we can obtain parameters of the LCAE and can form a reliable estimator to output the final OD flows, as shown in Eqs. (14)–(15):

$$\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\omega}} = \operatorname{argmin} \mathscr{L} \tag{14}$$

$$\widehat{\boldsymbol{Z}} = \mathscr{R}_{\widehat{\boldsymbol{\omega}}}\left(\mathscr{M}_{\widehat{\boldsymbol{\theta}}}(\boldsymbol{Z})\right) \tag{15}$$

Notably, the way for searching optimal OD flows in LCAE is generally consistent with that of conventional mathematical optimization programs (e.g., GLS-based models). As illustrated in Eqs. (16)–(17), where a general formulation of conventional

mathematical optimization programs is presented, the optimal OD flows are determined by minimizing certain kind of deviation (e.g., squared error and entropy) and satisfying the DTA constraints. Usually, this program could be solved through iterative, gradient-based method. Given the estimated prior and specific DTA model, then the solution space is fixed. Once the used prior is biased or the DTA process deviates from the realistic condition, the resulting solution would probably be a local optimum.

$$\min_{X,Q} \sum_{i=1}^{I} \sum_{rs \in RS} f_1\left(X_{irs}, \widetilde{X_{irs}}\right) + \sum_{i=1}^{I} \sum_{a \in A^o} f_2\left(Q_{ia}, Q_{ia}^o\right) \tag{16}$$

$$\text{Subjectto.} \quad Q_{ia} = DTA\left(X_{1,1}, \cdots, X_{irs}\right), \forall i \in I, a \in A, rs \in RS \tag{17}$$

Replacing the flow variable Q_{ia} in the objective by constraint (17), the model could be re-written into Eq. (18). Then, it is evident that the program in Eq. (18) is very similar to the objective (or loss function) presented in Eq. (11). The major difference lies in the relationship expression between OD flows and link flows. In Eq. (18), DTA models, whether analytical or simulation-based (Peeta & Ziliaskopoulos, 2001), are incorporated to constrain the program. The constraints provided by DTA model are in essence conservation constraints based on user choice assumptions and certain model of traffic flow propagation. Instead, in our model, the conservation constraint (or the entire DTA process) is replaced by sparse parametric mapping, where the sparseness is induced by the regularization.

$$\min_{X} \sum_{i=1}^{I} \sum_{rs \in K} f_1\left(X_{irs}, \widetilde{X_{irs}}\right) + \sum_{i=1}^{I} \sum_{a \in A^o} f_2\left[DTA\left(X_{1,1}, \cdots, X_{irs}\right), Q_{ia}^o\right] \tag{18}$$

This is a similar idea to the metamodeling that has been justified in calibrating OD demand (Osorio, 2019) as well as direct approximation of dynamic network loading (DNL) model (Song et al., 2018). Osorio (2019) used a metamodel to analytically and iteratively linearize the DTA simulator while Song et al. (2018) applied the statistical learning method of Kriging to establish a surrogate DNL model that could be analytically expressed. Differ from these metamodels that approximated from perfect simulated data, our proposed LCAE approach focuses on robustly approximate the multi-dimensional correlations between imperfect OD flows and link flows, and lacks of an elegant analytical from (which is supposed to be explored in our future research). This LCAE approach also resembles the idea of using a computational graph for modelling the hierarchy of traffic demand (Ma et al., 2020; Wu et al., 2018). The forward pass on the computational graph is in essence the demand assignment process while the backward pass propagates the gradients by chain rule to update certain parameters to be estimated. The training of LCAE also adopts the forward–backward algorithm as it is currently a fundamental building block for training DNNs, and the physical meaning also resembles the hierarchical flow network (HFN) proposed by (Wu et al., 2018). What differs the LCAE from the HFN is that the LCAE does not provide concrete physical interpretation for each node of the computation graph while the HFN does. Again, this is induced by the idea that the LCAE target at setting up robust mappings to denoise the prior OD flow. And the data fusion exploration on this highly flexible framework of HFN is also within our research interest in the future.

2.4.2. ASC algorithm

As stated in Section 2.4.1, the optimization framework of LCAE is considerably similar to that of conventional models. In other words, the local optimum problem would possibly pose an issue even when LCAE is well regularized. In other words, the LCAE would possibly converges to the prior OD flow that carried a considerable amount of error as itself serves as the training target. Furthermore, the training of LCAE follows the method of training DNNs and splitting the dynamic OD flow into samples. Then, each sample covers only a few time intervals, and the within-day smoothness and consistency of traffic flow characteristics would not be guaranteed. Therefore, the obtained estimates may involve significant and unrealistic variance even for very similar samples. To deal with these two problems, the periodicity and general features of traffic demand, i.e., day-to-day recurrences and variations, could be utilized to adaptively update the priors in the optimization objective of the LCAE. In this way, the solution space becomes dynamic (differ in each iteration) to avoid the local minimum and the estimation variance among similar samples could also be reduced.

In this study, we propose an ASC algorithm and integrate it into the training of LCAE. The detailed algorithm integrating LCAE and ASC algorithms is presented in Algorithm 1. The ASC algorithm selects the sub-samples that cause the largest variance and interpolates these sub-samples with the neighboring ones. This process is similar to first-order exponential smoothing, where the average of neighboring sub-samples is regarded as the smoothing benchmark. Here, the sub-sample refers to a slice of an OD flow sample, i.e., a time sequence of OD flows for an OD pair. For example, given an OD flow sample covers several time intervals and K OD pairs, then K sub-samples could be split from the sample, with each sub-sample representing the flow series of a specific OD pair. The reason of using sub-samples instead of samples is that the similarity between samples may be dominated by large OD flows, whereas sub-samples are more flexible.

Algorithm 1. Integrated LCAE and ASC

Input: prior OD flow sample set Z and latent vector sample set Q; sub-sample set Y; initialized encoder and decoder $\mathscr{T}_{\theta}, \mathscr{G}_{\varphi}$; batch size M; outer and inner loop iteration T_1, T_2; regularization weight λ; initial learning rate δ_0; Nearest neighbors set J; LID noisy sample portion p (noisy sample amount $N = p \cdot |Z|$); vector distance measure$f(\cdot)$

1. **For** $t_1 = 1$ to T_1 **do**:
2. **For** $t_2 = 1$ to T_2 **do**:
3. **For** $\tau = 1$ to $\mathrm{mod}(|Z|, BS)$ **do**:

(*continued on next page*)

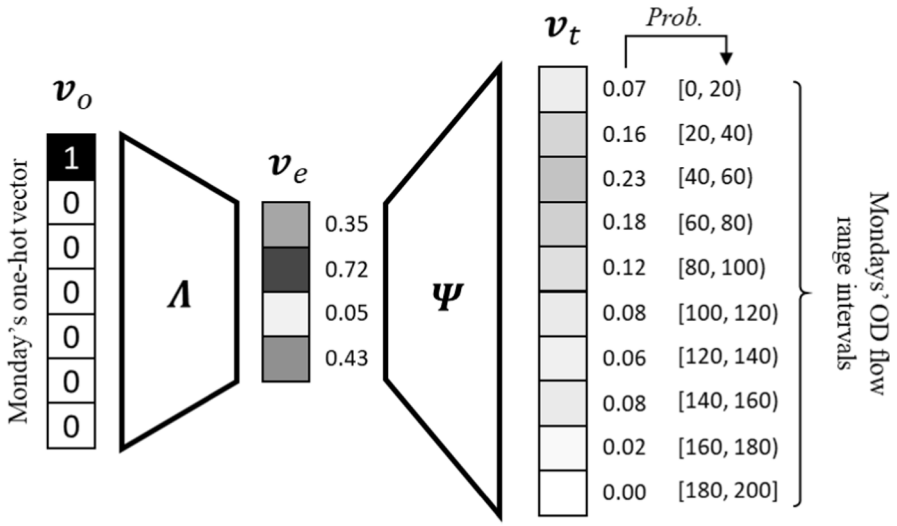


Fig. 6. Illustration of vector embedding.

(*continued*)

Algorithm 1. Integrated LCAE and ASC

Sample a batch $(Z, Q)^{(\tau)}$ from dataset$Z$
$\boldsymbol{M} = \mathscr{M}_{\boldsymbol{\theta}}(\boldsymbol{Z}), \boldsymbol{R} = \mathscr{R}_{\boldsymbol{\omega}}(\boldsymbol{M})$ // forward pass
$\mathscr{L} = \|\boldsymbol{M} - \boldsymbol{Q}\|_2^2 + \|\boldsymbol{R} - \boldsymbol{Z}\|_2^2 + \lambda^{'}\left(\|\boldsymbol{\theta}\|_2^2 + \|\boldsymbol{\varphi}\|_2^2\right)$ // calculate loss
$\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \mathrm{Adam}(\delta, \nabla_{\boldsymbol{\omega}}\mathscr{L}), \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mathrm{Adam}(\delta, \nabla_{\boldsymbol{\theta}}\mathscr{L})$ // gradient descent
End
End
$Y \leftarrow \mathrm{split}(Z)$ // split each sample into sub-samples
For $y = 1$ to $|Y|$ **do**:
$$\widehat{LID}_y = -\left[\frac{1}{|J_y|}\sum_{j=1}^{|J_y|}\log\frac{f_j(\boldsymbol{Y})}{\max\left(f_j(\boldsymbol{Y})\right)}\right]^{-1}$$ //calculate LID score for each sub-sample
End
$Z_1^{'}, \cdots, Z_N^{'} = \mathrm{argmax}\left(\widehat{LID}_i, \cdots, \widehat{LID}_{|Y|}\right)$ // recognize most noisy sub-samples
For $n = 1$ to N **do**:
$\gamma_n \sim \mathrm{Beta}(\alpha, \beta)$, $Z_n^{'} \leftarrow \gamma_n \cdot Z_n^{'} + \frac{1-\gamma_j}{|J_n|}\sum_{j=1}^{|J_n|} Z_j^{'}$ // interpolate inconsistent sub-samples
End
$Z \leftarrow \mathrm{recombine}(Y, Z_1^{'}, \cdots, Z_N^{'})$
End
Output: $\mathscr{M}_{\widehat{\boldsymbol{\theta}}}, \mathscr{R}_{\widehat{\boldsymbol{\omega}}}, Z$

Before applying the algorithm, two questions need to be answered: one, how to measure the sub-sample similarity when considering discrete contextual information (e.g., time-of-day and day-of-week)? Two, how to recognize sub-samples that cause the largest variance?

The OD flow patterns across different OD pairs, time-of-days, or day-of-weeks considerably differ. However, these discrete variables could not be directly measured to reflect the differences in the OD flow pattern. Thus, for the first question, the vector embedding technique is applied (Mikolov et al., 2013). Vector embedding is a parametric transformation to map discrete variables to continuous vectors. As illustrated by Eq. (19), two parametric matrices ($\boldsymbol{\Lambda}, \boldsymbol{\Psi}$) are randomly initialized and optimized by minimizing the squared error between the estimated target vector and label vector. Then, the embedding vector $\boldsymbol{v}_e = \boldsymbol{\Lambda} \cdot \boldsymbol{v}_o^T$ is regarded as a joint representation of vector $\boldsymbol{v}_o$ and $\boldsymbol{v}_t$. In our case, the input vector $\boldsymbol{v}_o$ is a one-hot vector of contextual variables including time-of-day, day-of-week, and OD pair index, and the target vector is a correspondingly discretized distribution of prior OD flow. Taking day-of-week embedding as an example, which is illustrated in Fig. 6, the input vector of Monday is a one-hot vector with the first entry being one, while the target vector is a probability vector with each entry equaling the probability of the OD flow range. As presented in Eq. (20), the $\mathrm{softmax}(\hat{\mathrm{A}}\cdot)$ function forms a valid probability term through natural constant exponential normalization. By specifying the dimensions of $\boldsymbol{v}_e$ and solving the program, each sub-sample could be represented by a dense vector and any generic distance function could be applied for measuring the similarity.

$$\min\left\|\mathrm{softmax}\left(\boldsymbol{\Psi}\cdot\left(\boldsymbol{\Lambda}\cdot\boldsymbol{v}_o^T\right)^T\right)-\boldsymbol{v}_t\right\|_2^2 \tag{19}$$

$$\mathrm{softmax}(v_0,\cdots,v_n)=\left[\frac{\exp v_0}{\sum_{i=0}^{n}\exp v_i},\cdots,\frac{\exp v_n}{\sum_{i=0}^{n}\exp v_i}\right]^T \tag{20}$$

For the second question, a distributional measure of datasets called the locally intrinsic dimensionality (LID) is applied (Houle, 2017). Given a data sample x, let $r>0$ be a random variable denoting the distance from x to other data samples; then, a maximum likelihood estimator of LID of x at distance r could be given by Eq. (21). As presented in the formula, LID actually measures the inconsistency of a sample to its neighboring samples, which indicates the local manifold in the geometric space. For the detailed derivation, proof, and effectiveness indication of LID, interested readers can refer to (Houle, 2017) and (X. Ma et al., 2018). In our case, x is an embedded vector of each sum-sample and the Euclidean distance is used as the distance measure $f(\cdot)$.

$$\widehat{LID}(x)=-\left[\frac{1}{|J|}\sum_{j=1}^{|J|}\log\frac{f_j(x)}{\max\left(f_j(x)\right)}\right]^{-1} \tag{21}$$

where $\widehat{LID}(\cdot)$ is the calculator of the LID score, J is the set of nearest neighbors of x, $|J|$is the number of nearest neighbors, and $f_j(x)$ is used to calculate the distance of x to its j-th nearest neighbor.

In each training epoch of LCAE, a portion of sub-samples with the highest LID scores is recognized as the portion of noisy samples. Then, these noisy sub-samples are interpolated by its k-nearest neighbors with a random coefficient γ following beta distribution $B(\alpha, \beta)$. By adaptively correcting noisy sub-samples and re-combining them into full samples, the LCAE can gradually denoise the prior samples and can finally output clean OD flows.

3. Validation

3.1. Evaluation metrics

In this study, the estimation performance is indicated by four error metrics—mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), and mean square percentage error (MSPE). MAE and MAPE are prevalent indicators of the estimation performances for various tasks. RMSE and MSPE are also included in this study because they can better reveal the estimation performance on larger values (larger OD flows tend to be more relevant). Corresponding formulas are presented in Eq. (22)–(25).

$$MAE=\frac{1}{|Z|}\sum_{z=1}^{|Z|}\left|X_{irs}-\widehat{X}_z\right| \tag{22}$$

$$RMSE=\sqrt{\frac{1}{|Z|}\sum_{z=1}^{|Z|}\left(X_z-\widehat{X}_z\right)^2} \tag{23}$$

$$MAPE=\frac{1}{|Z|}\sum_{z=1}^{|Z|}\frac{\left|X_z-\widehat{X}_z\right|}{\widehat{X}_z+\varepsilon} \tag{24}$$

$$MSPE=\frac{1}{|Z|}\sum_{z=1}^{|Z|}\frac{\left(X_z-\widehat{X}_z\right)^2}{\widehat{X}_z^{\,2}+\varepsilon} \tag{25}$$

where ε is a small positive number for avoiding dividing by zero, and it is set as 0.01 here.

In addition, to measure the computation efficiency of the proposed method, the experiments were all conducted on a server with a 3.2 GHz 8-core CPU, a GTX-1060Ti GPU, and a 32 GB RAM. All codes are implemented in Python, the deep neural networks are built upon the Tensorflow framework, and the computation time was then recorded and compared.

3.2. Empirical case

3.2.1. Dataset description

First, an empirical dataset is used to validate the effectiveness of the proposed method. The data were collected from an urban arterial in Changzhou, China. This is a toy network with trivial route choices. In this toy network, LPR cameras are installed at all approaches of each intersection; thus, the true OD flow can be obtained by matching vehicle IDs. Missing detection is not considered in this case because LPR cameras are newly installed and traffic police enforcement has claimed a very low missing detection rate. The layout of the arterial network is presented in Fig. 7. It can be seen that there are six signalized intersections, 14 origin/destination

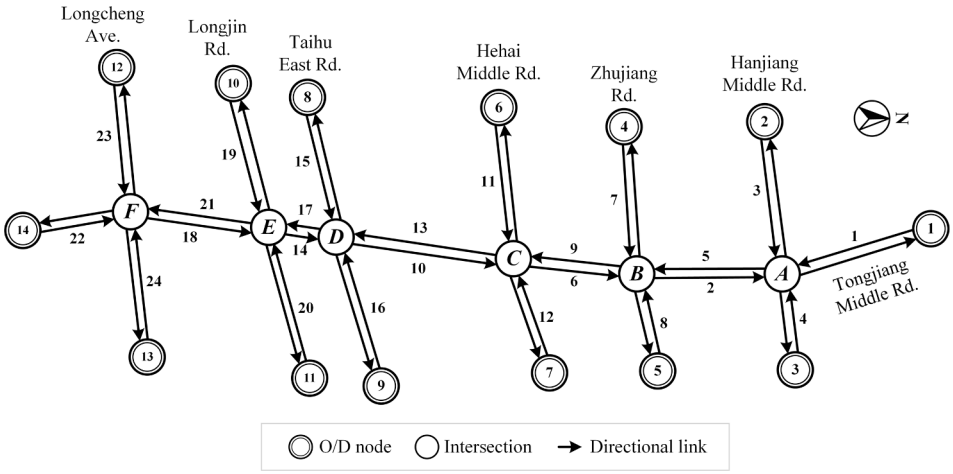


Fig. 7. Validation site of empirical case.

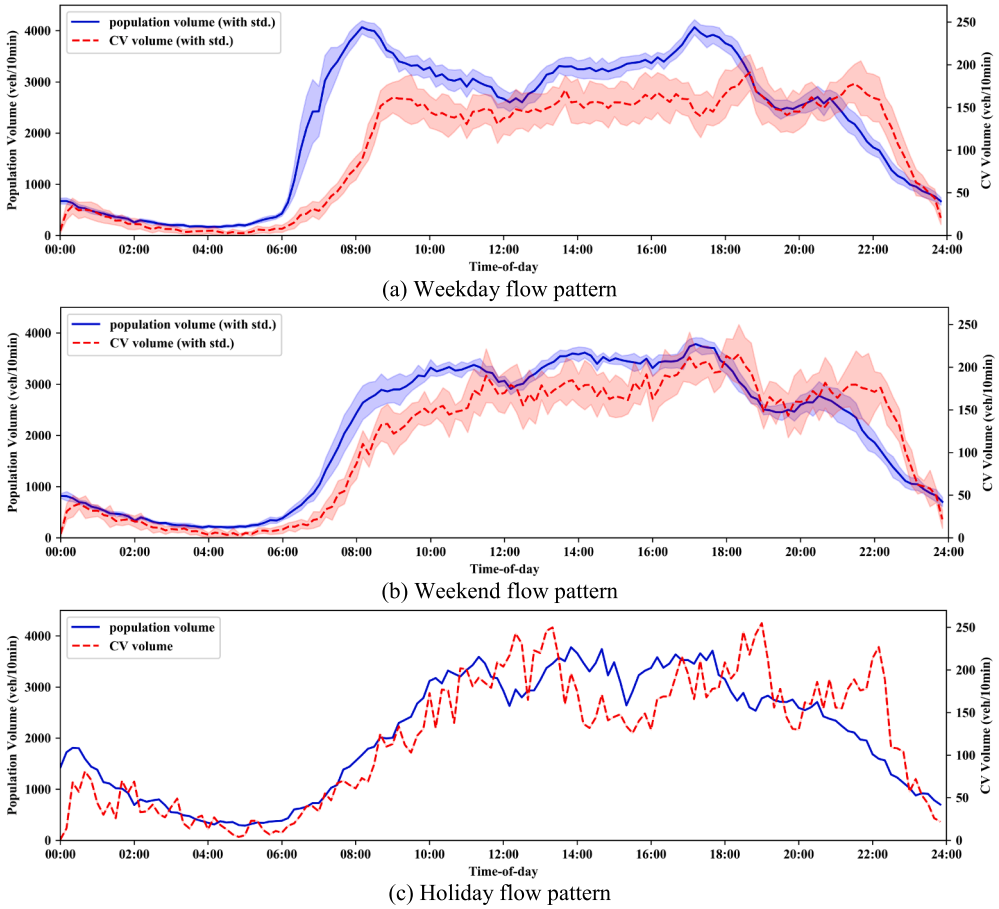
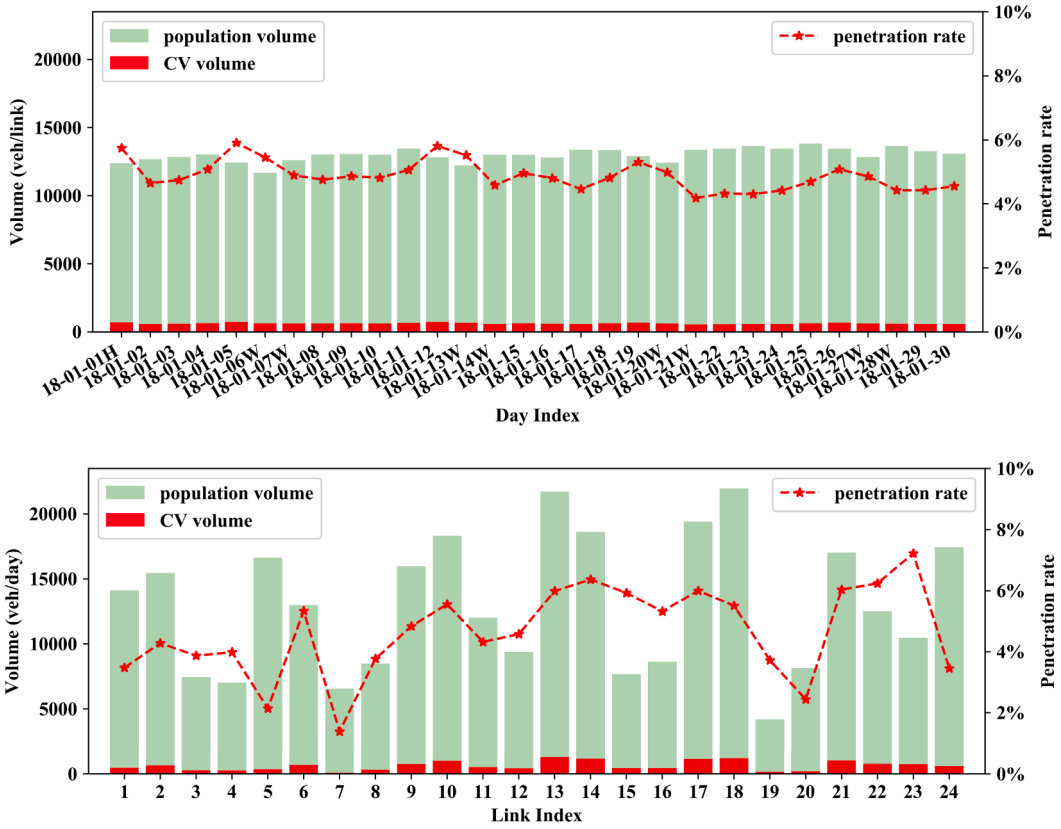


Fig. 8. Flow patterns of the target area.

nodes, 38 links, and $14 \times 14 = 196$ OD pairs (in theory). Here, the OD pairs whose maximum OD flow within an interval is less than 5 vehicles are removed; thus, 31 OD pairs remain to be estimated.

Meanwhile, the trajectories of CVs traveled within this area are collected from E-hailing service vehicles of DiDi company, and the CV penetration rates on this arterial network are calibrated based on the fraction between the total CV counts and AVI counts. The average penetration rate of this arterial network is 5.35% with a standard deviation of 1.68%, which is low for OD estimation but could



(b) Spatial variation of penetration rates

Fig. 9. Variation of penetration rates.

be often seen in real-world cases. The timespan of both data sources is from 2018/01/01 to 2018/01/30, i.e., 30 days. Each day in this dataset is split into identical time intervals of 10 min, and the total time-of-day volume as well as CV volume on this arterial network with respect to weekdays (21 days), weekends (8 days), and holidays (2018/01/01) are separately presented in Fig. 8(a)–(c). The population and CV volumes are respectively aggregated from all AVI detectors and CV trajectories within this network. As can be seen in Fig. 8, the CV penetration rates vary across different time-of-days, which is the within-day temporal variation discussed previously in Section 2.3. In general, the penetration rates are lower during the morning peak and higher during the evening peak, and the peaks of CV are delayed compared with the population vehicles, especially for weekdays.

In addition to within-day variation of penetration rate, the day-to-day temporal variation as well as variation across different links could can be observed. In Fig. 9(a), the dates of weekends and holidays are respectively marked as W and H. Thus, as can be seen, there is a gradual increase in weekdays and a consecutive drop in weekends in each week. In terms of spatial variation, as presented in Fig. 9 (b), the highest link average penetration rate is approximately 8%, while the lowest is less than 2%.

3.2.2. Experiment results

In our experiment, four out of 24 links (16.7% coverage) were randomly selected and regarded as observed links, while others were not. Based on three different combinations of observed links, the proposed method was validated. First, the hyperparameter settings within our methodology were specified. The NTD process requires a pre-defined rank of core tensor ($\mathscr{C}$), which can be interpreted as the number of unique patterns within each dimension. Here, the rank is set as (25, 120, 25), which indicates that 25, 120, and 25 unique patterns are recognized in the days, time-of-day intervals, and OD pairs, respectively. In the constructed tensor, all the values below 10 veh/10 min is regarded as values to be imputed, and the values are replaced after NTD. For the optimization, the regularization weight (λ) is set to 0.01 and the maximum iteration is 50. Preliminary experiments indicated that NTD always reaches convergence within 50 iterations and is insensitive to the aforementioned hyperparameters.

Different from most DNN applications where exact labels are available, the self-supervision character of LCAE-ASC determined the hyperparameters could hardly be optimized for field implementation. In this case, the architecture of LCAE is determined according to minimization of the reconstruction loss along with the design rule of decreasing neuron and increasing neuron in encoder and decoder, respectively. The dimension of input is 31 OD pairs times 6 intervals (one-hour long) and equals to186, and that of latent vector is 8 flow observations times 6 intervals and equals to 48. Preliminary results indicate that encoder (and decoder) with 3 to 4 layers could effectively reconstruct prior OD flows. Then, we used 3 layers in both encoder and decoder, and the number of neurons in encoder is set

Table 2
General evaluation results of the empirical case.

	MAE (veh/10 min)	RMSE (veh/10 min)	MAPE	MSPE	Time Cost (s)
ELP	12.91	21.99	71.62%	51.27%	0.02
NTD	10.97	18.25	60.86%	35.31%	0.58
LCAE-ASC	8.27	12.28	45.90%	15.98%	166.99

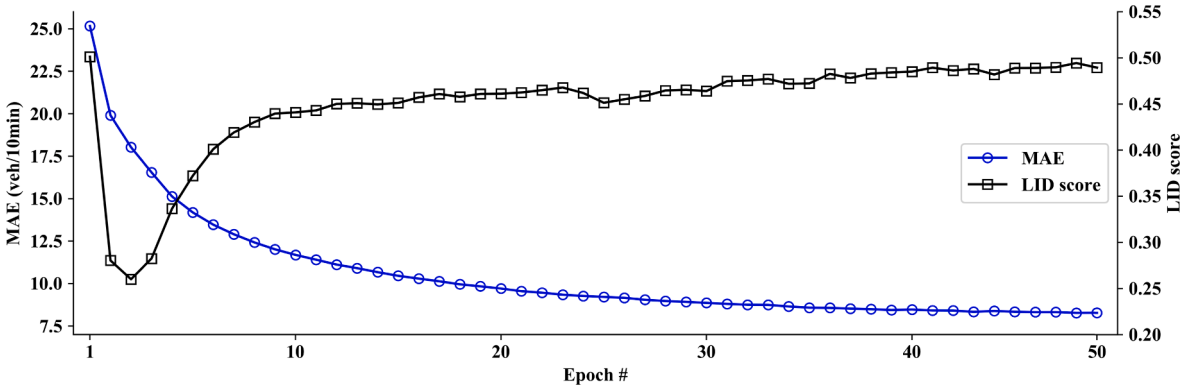


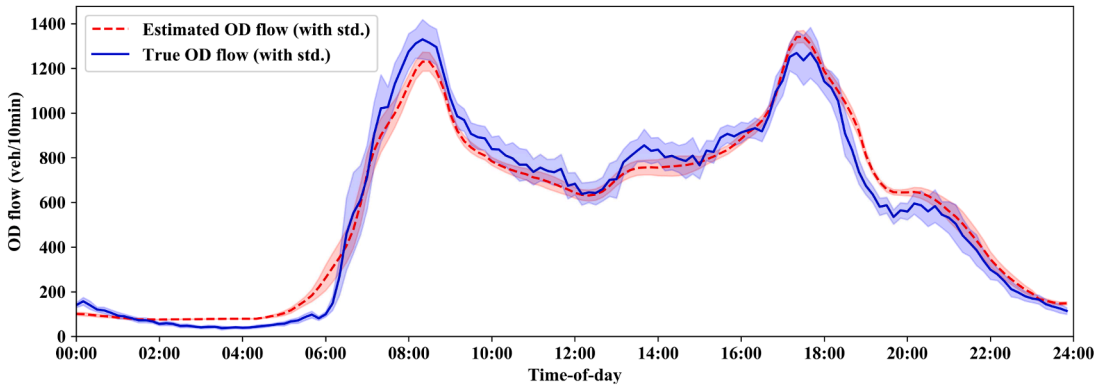
Fig. 10. LCAE-ASC iteration process.

to 160, 128, 64 whereas that of the architecture of decoder is symmetrical to the encoder, and a 9-layer DNN is finally formed. Afterward, the prevalent training hyperparameter choices of the deep neural network are used: the regularization weight (λ) is set as 0.01, the model is trained by the Adam optimizer for 50 epochs (T_1) and 10 (T_2) with a batch size (M) of 1024 and Dropout rate of 10%, and the initial learning rate (δ_0) is set as 0.0003. In each ASC iteration, 20% of noisy samples within the dataset are recognized according to the LID score and are split into sub-samples. Each of these sub-samples is interpolated by the average of the 30 nearest neighbors using the interpolating coefficient sampled from the Beta distribution $B(9,1)$. The number of nearest neighbors is set to 30 in that this is a 30-day dataset, and the distribution $B(9,1)$ leads to coefficients with a mean coefficient of 0.9. The numerical results are presented in Table 2.

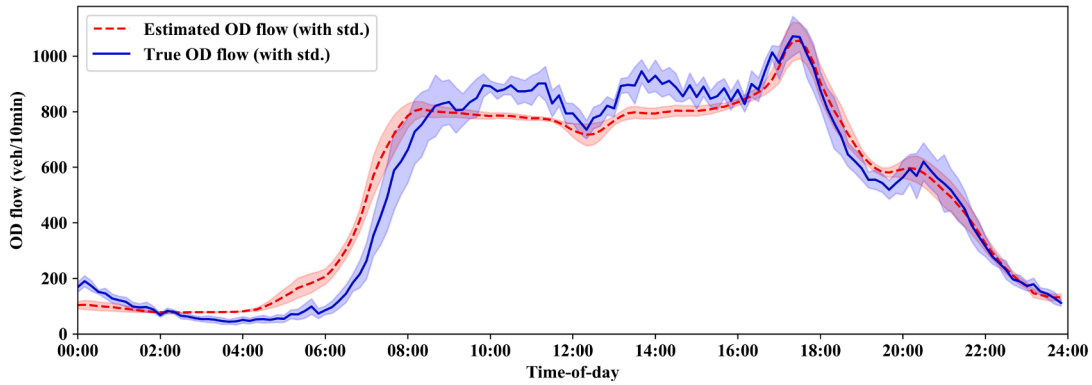
As can be seen in Table 2, the ELP produced a rough prior estimate with an RMSE of 22 veh/10 min and an MSPE of 51.27%. One of the significant contributors to the error is sparsity. The experimental results indicated that the sparse rate (fraction of sparse entries) of the projected prior OD flow is 52.8%, which reduced to 16.2% after NTD, while the true sparse rate was 16.1%. Therefore, the NTD also resulted in an obvious decrease in all error metrics. Additionally, to further validate the effectiveness of NTD, the simple average method is also applied to as a comparing benchmark. For the defined sparse entries, the simple average method directly calculates the average value of the OD flow across the whole timespan (with sparse ones excluded) and replace the sparse values. The results show that the simple average method reduce the sparse rate to 23.8%, whereas the MAE and MAPE respectively increased to 13.93 veh/10 min and 77.28%, which validate that the consideration on multi-dimensional correlation of NTD could deal with the sparsity problem more properly.

As shown in Fig. 10, The LCAE-ASC further corrected the samples via an iterative process. As can be seen, MAE gradually decreased and reached convergence. The LID score, which indicate the local variance, has significantly dropped at the first few epochs and gradually increased to a stable value around 0.5. This phenomenon is induced by the common training character of deep neural networks: in the earliest stage, the parameters of neural network is initialized randomly, and thus the difference between samples are not significant (less varied, lower LID score); then, the robust latent variables and ASC algorithm gradually affects the optimization direction and finally reach a balance state where the final OD flows simultaneously resembles neighboring ones and is consistent with corresponding AVI flow observations. Notably, although the LID curve in Fig. 10 is generally in an upward trend, the initial LID score of the prior OD flow is actually 0.69, which validate that the ASC algorithm substantially reduced variance between similar sub-samples. The performance improvement by LCAE-ASC is also indicated by two other error metrics—RMSE and MSPE—which are more affected by larger values and which decreased by 32.7% and 54.7%, respectively. Even though the final MAPE is greater than 45%, the final MSPE is only approximately 16%, which indicated that the large MAPE is mainly caused by smaller values, especially considering that OD flow distribution with respect to OD pairs is often long-tailed. In conclusion, the proposed method achieved accurate estimation in this case, especially when considering the CV penetration rate of 5% and AVI coverage of 16%.

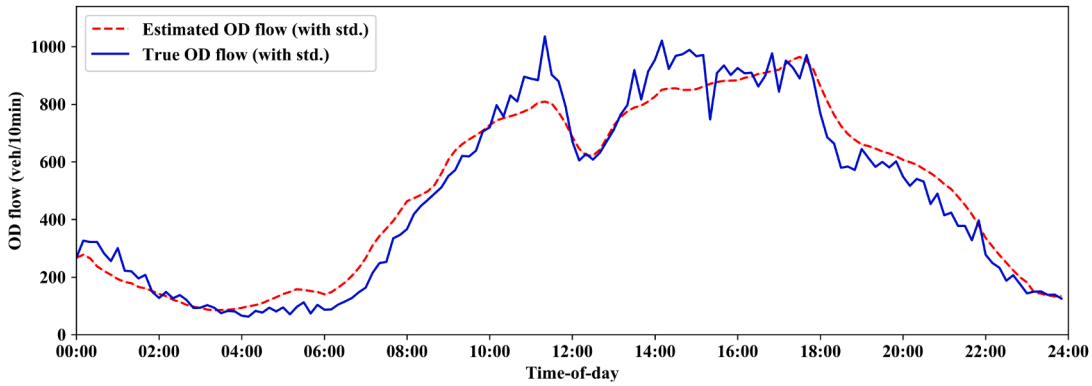
In terms of recovering the time-of-day OD flow pattern, Fig. 11 presents the estimated and true aggregated OD flow pattern for weekdays, weekends, and the holiday and Fig. 12 presents the true and estimated OD flow with respect to each OD pair. It is evitable that the patterns are generally recovered and the proposed method generated smoother estimates (from both figures) as the LCAE mainly looked at robust components and neglected much of the noises. However, the variance produced (presented by the shadow area) is in general less than the actual one. This is induced by the ASC algorithm, which interpolated mostly varied sub-samples with neighboring ones and therefore reduced variance. A special note should be given to the pattern on the holiday, as one may naturally



(a) Estimated OD flow pattern on weekdays



(b) Estimated OD flow pattern on weekends



(c) Estimated OD flow pattern on the holiday

Fig. 11. Estimated OD flow pattern.

questioning if the proposed method (especially the neural network) could handle abnormal conditions when physical process and human behavior assumptions are not included. In this dataset, the holiday pattern is significantly different from those of weekdays and weekends, but the LCAE-ASC could also reproduce the pattern thanks to the robust latent variables, which have provided concrete evidence to enable the algorithm to differ the holiday from other days.

3.2.3. Comparison with existing models

To better illustrate the effectiveness and efficiency of the proposed method, an existing benchmark method (Yang et al., 2017) was also tested. This benchmark method extended the classical GLS framework for integrating probe vehicle trajectories and link count observations. Two single-level GLS models are proposed in this work, and the formulations are presented in Appendix A. The first one, called the scaled probe as prior (SPP), scales probe OD flows as a prior and forms a convex quadratic program. The second model, called the penetration rate assignment (PRA), further extends the SPP model by adding constraints concerning penetration rates and forms a non-convex quadratic program. The benchmark study is originally tested on a simulation network within a 3-h period. Here, we adopted the same period and tested both models in the period of 7:00–10:00 a.m. on 2018/01/08. The corresponding numerical

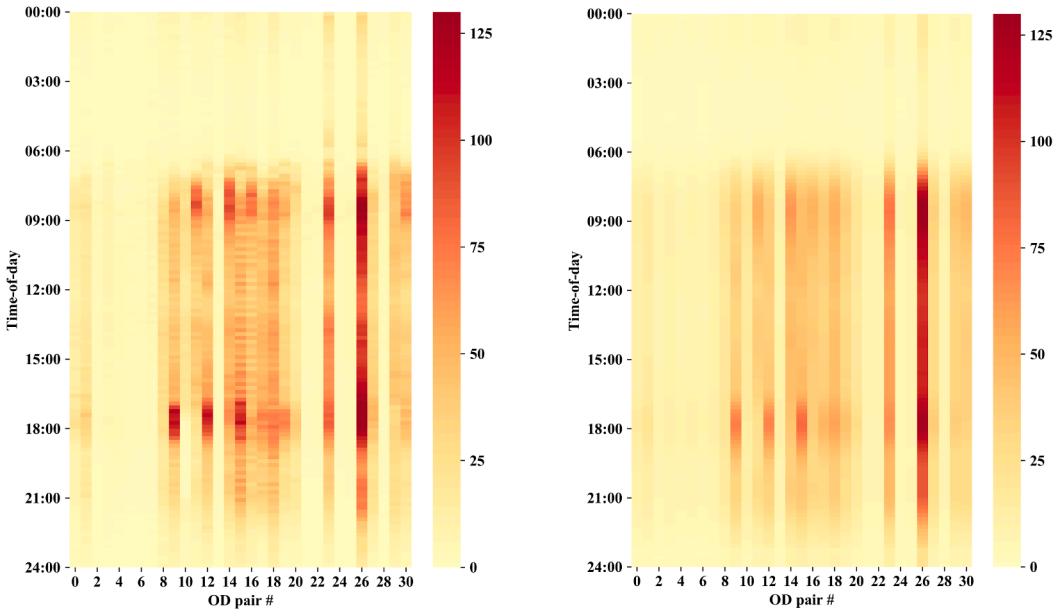


Fig. 12. Estimated OD flow for each OD pair.

Table 3
Evaluation results of model comparison.

	MAE (veh/10 min)	RMSE (veh/10 min)	MAPE	MSPE	Time Cost (s)
Prior	25.53	37.21	69.82%	52.89%	–
SPP	20.36	30.20	55.69%	34.82%	0.20
PRA	21.45	31.21	58.67%	37.21%	276.57
Our method	11.44	16.32	31.28%	10.17%	0.34

results are listed in Table 3, and the regression plots are presented in Fig. 13(a)–(d).

As can be seen in Table 3, the proposed method showed significant improvement compared with both benchmarking models. Both the SPP and PRA models showed obvious improvement based on the prior. The RMSE and MPSE of these models were 30 veh/10 min and approximately 35%, respectively. Furthermore, this of our method were 11 veh/10 min and approximately 10%, respectively, indicating that our method was improved over both benchmark models.

As can be observed in Fig. 13, the estimates produced by SPP and PRA are almost similar to the prior estimates. As justified by the existing studies (Cascetta et al., 2013; Marzano et al., 2009) and discussed in Section 1, the reason that led to this phenomenon is that the GLS-based models are susceptible to the quality of prior estimates and the quantity of physical constraints. When an unreliable prior estimate is provided, the GLS-based models can be trapped into the local minima, resulting in unsatisfactory estimation. In this case, the CV penetration rate is too low to produce an effective prior as well as reliable assignment ratio, and the number of observed links is lower than the number of OD flows to be estimated. Thus, these two models are largely weakened under such circumstances, which is also consistent with the discussion on penetration rates in the original paper (70% of MAPE were reported under 5% penetration rate). Note that this comparison is not completely fair as the SPP and PRA models focused on estimation of a much lower period, while our method utilized day-to-day variation information.

In terms of computation cost, the SPP model performed the best because of the convexity, whereas the cost of the proposed method was the second, and that of the PRA model was the highest because it is non-convex program with bilinear constraints and requires a well-designed solution algorithm. In conclusion, the proposed method provided more accurate estimates than these two models and presented the most cost-effective solution in this case.

3.2.4. Determine the influence of the dataset timespan

In this section, the influence of the dataset timespan is determined by a set of comparing experiments. Originally, the timespan of the dataset is 30 days (about four weeks), three comparing groups are then created by dumping one-week data each time. The size of core tensor in NTD as well as the number of nearest neighbors are accordingly reduced with the dataset size, while other model parameters remain unchanged. Then, for each dataset, the method is applied from scratch, and the corresponding results are presented in Table 4.

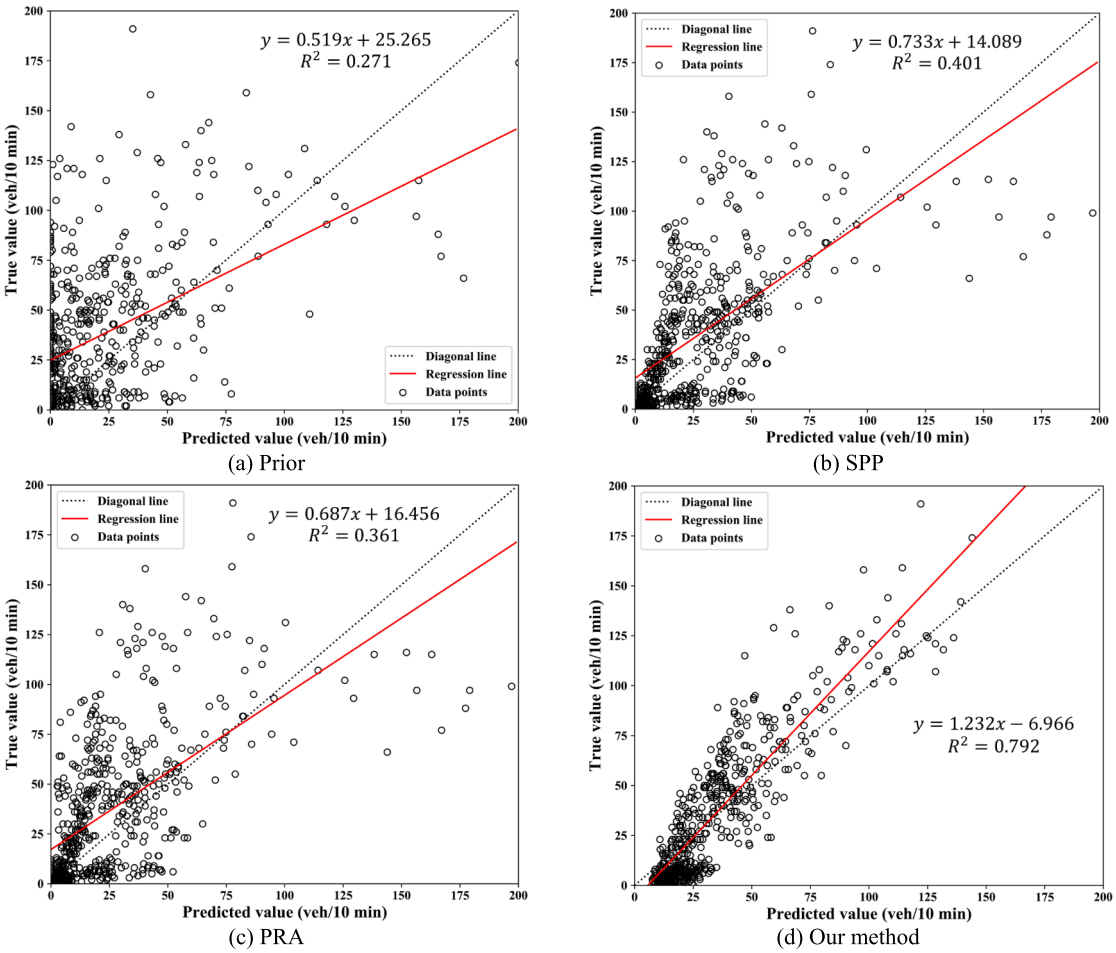
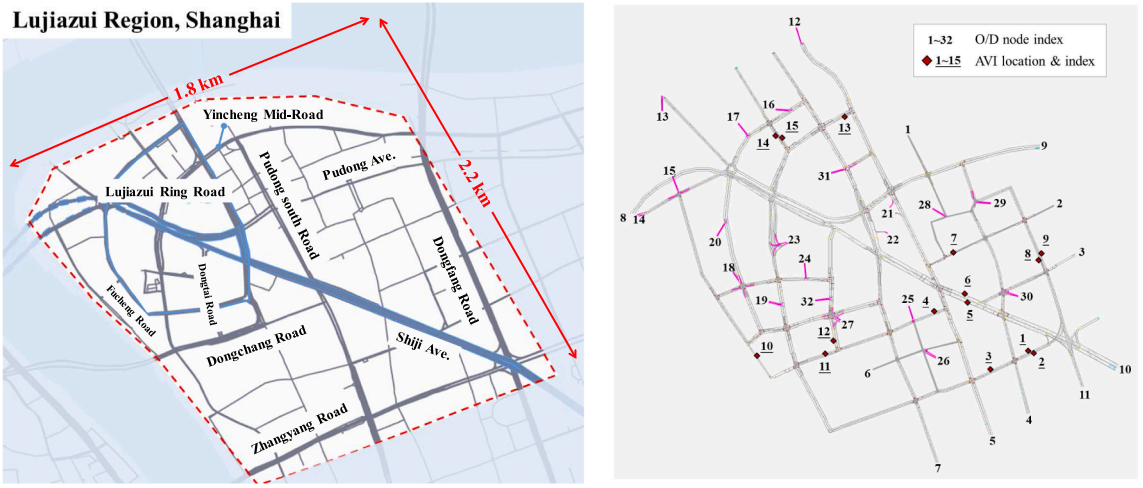


Fig. 13. Regression plots of comparison models.

Table 4
Evaluation results under different dataset timespan.

		Four weeks' dataset	Three weeks' dataset	Two weeks' dataset	One week's dataset
ELP	MAE (veh/10 min)	12.91	12.66	12.33	12.11
	RMSE (veh/10 min)	21.99	21.31	20.7	20.42
	MAPE	71.62%	70.82%	69.85%	69.66%%
	MSPE	51.27%	48.62%	46.38%	45.97%
NTD	MAE (veh/10 min)	10.97	10.59	10.35	10.2
	RMSE (veh/10 min)	18.25	18.73	18.32	18.1
	MAPE	60.86%	59.21%	58.61%	58.69%
	MSPE	35.31%	37.56%	36.32%	36.11%
LCAE-ASC	MAE (veh/10 min)	8.27	9.47	9.91	10.27
	RMSE (veh/10 min)	12.28	15.9	17.78	18.26
	MAPE	45.90%	52.94%	56.11%	59.08%
	MSPE	15.98%	27.08%	34.22%	36.78%

From the table, it can be seen that the first module (including ELP and NTD) is generally not affected by the reduced timespan. For ELP, the scaling process does not require day-to-day information. As for NTD that emphasizes multi-dimensional pattern mining, the reduced timespan only weakened the day-to-day correlation mining, while the correlations of other dimensions (time-of-day and OD pair) could still be utilized. For the LCAE-ASC module, there is an obvious trend that the error increases with the decrease of dataset timespan, as the dataset size for training LCAE is decreased and the day-to-day recurrence is less utilized as references are reduced. Despite the increasing error, the LCAE-ASC would at worst produce the estimates close to that of NTD estimates because of the self-supervision character, which could be indicated by the errors of one week's dataset.



(a) validation site (b) simulation model

Fig. 14. Simulation Network.

Table 5
General evaluation results of the baseline case.

Sampling	Models	MAE (veh/10 min)	RMSE (veh/10 min)	MAPE	MSPE	Time Cost (s)
Homogeneous Sampling	ELP	5.19	8.35	68.87%	36.35%	0.03
	NTD	2.68	4.72	36.43%	11.98%	3.73
	LCAE-ASC	1.60	2.51	21.78%	3.39%	236.51
Heterogeneous Sampling	ELP	5.13	8.27	68.13%	35.66%	0.03
	NTD	2.81	4.98	37.30%	12.95%	4.49
	LCAE-ASC	1.73	2.81	23.50%	4.24%	228.64

3.3. Simulation case

3.3.1. Dataset description

For further validation and investigation of the influencing factors of the proposed method, traffic simulation experiments were conducted using a realistic urban road network in Lujiazui, Shanghai. This regional traffic network consisted of 188 links, 35 intersections (including signalized and unsignalized intersections), and 32 origin/destination nodes, as presented in Fig. 14(a). Based on this network, a simulation network was established using the VISSIM simulator, as presented in Fig. 14(b). In this network, 297 major OD pairs were calibrated using data from the Sydney Coordinated Adaptive Traffic System and data from LPR cameras (Tang et al., 2014), and trivial OD flows were redistributed to larger flows during the calibration. To establish a larger dataset to represent day-to-day conditions, Gaussian noise was added for each OD pair on both the route choice probabilities and the input volumes based on the calibrated dataset. The detailed descriptions are provided in (Tang et al., 2020). Finally, a 30-day dataset for model evaluation was established. Following the previous experimental settings, the length of each time interval was set to 10 min.

The following sections are arranged as follows: Section 3.3.2 forms a baseline case with empirical AVI locations and 10% CV by homogeneous sampling; Section 3.3.3 discusses the impact of heterogeneous CV sampling based on the baseline; Section 3.3.4 details the investigation of the model sensitivity to CV penetration rates and AVI missing detection rates; and Section 3.3.5 shows the investigation of the sensitivity to different AVI locations and coverage.

3.3.2. Baseline case

In this section, a baseline case is formed for basic model validation and further discussion. In the network specified in Section 3.3.1, 15 empirical AVI locations are selected (i.e., 8% coverage) without missing detection, as shown in Fig. 14(b). Meanwhile, 10% of vehicles are homogeneously sampled as CV. Here, the homogeneous sampling refers to that the OD pair penetration rates are sampled from uniform distribution with narrow bandwidth, while heterogeneous sampling refers to that the OD pair penetration rates are sampled from a nonuniform distribution (e.g., Gaussian distribution). Generally, the difference lies in the penetration rate variation among different OD pairs: the resulted penetration rates of homogeneous sampling would become nearly the same, whereas that of heterogeneous sampling would much larger variation. The rank of the core tensor in the Tucker decomposition is set (25, 120, 200), while other model parameters follow the same settings in the empirical case. The results are presented in the homogeneous sampling case in Table 5. As can be seen, the proposed method is still effective in this case because the estimation errors decreased step by step and our method finally achieved an MAE of only 1.6 veh/10 min and an MSPE of less than 4%.

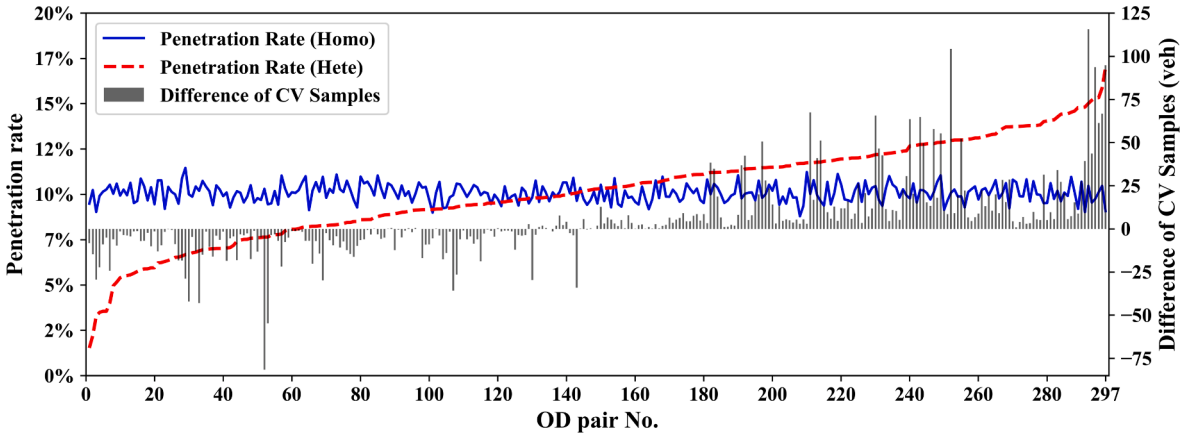


Fig. 15. Homogeneous and heterogeneous sampling.

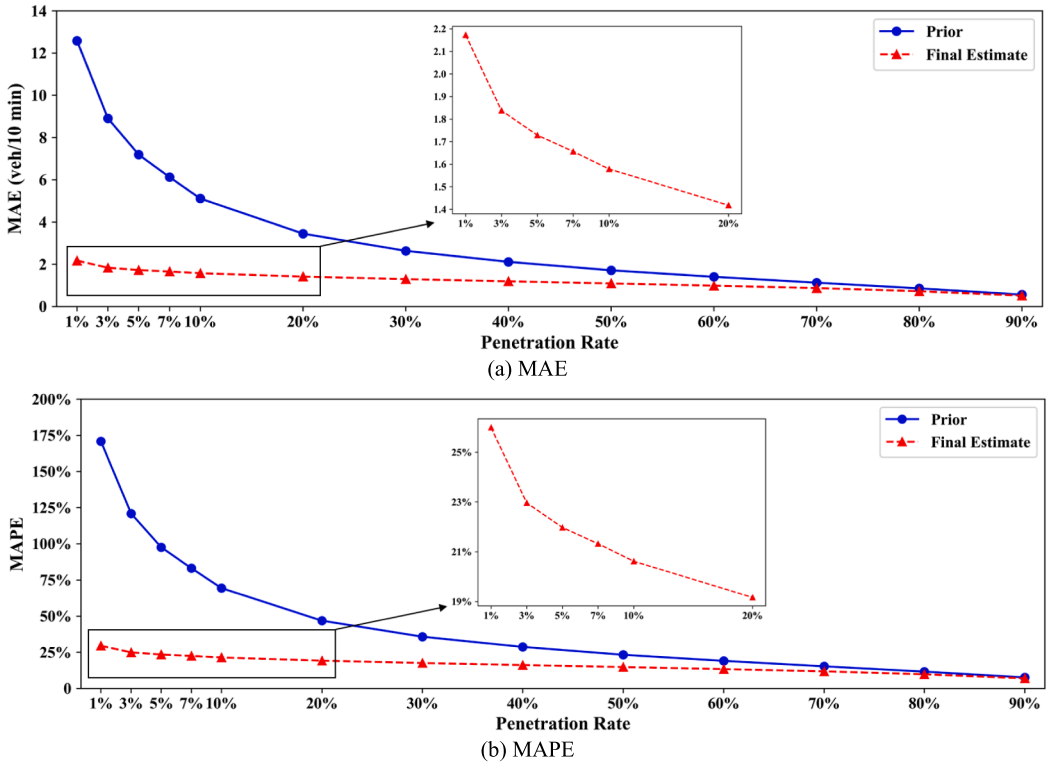


Fig. 16. Sensitivity to penetration rates.

3.3.3. *Impact of heterogeneous CV sampling*

In this section, the sampling strategy for CV is changed to heterogeneous sampling. As presented by the two curves in Fig. 15, the average penetration rates for different OD pairs are sampled from the uniform distribution $U(9, 11)$ for homogeneous sampling, whereas those of heterogeneous sampling are sampled from Gaussian distribution $N(0.1, 0.02^2)$. As presented by the bars in Fig. 15, the difference in CV samples is significant under different sampling strategies, and the estimation results in this case are compared with the baseline case.

As presented in Table 5, the MAE, RMSE, MAPE, and MSPE of the heterogeneous sampling case are 1.73 veh/10 min, 2.81 veh/10 min, 23.50%, and 4.24%, respectively. As can be seen, the estimation errors slightly increased compared with the baseline case because this heterogeneity brought the systematic bias to prior OD flow in the ELP process. However, in general, the estimation performance is satisfactory because the MAE is still less than 2 veh/10 min and the MSPE less than 5%, especially when considering the severe observing condition (8% AVI coverage) in this case.

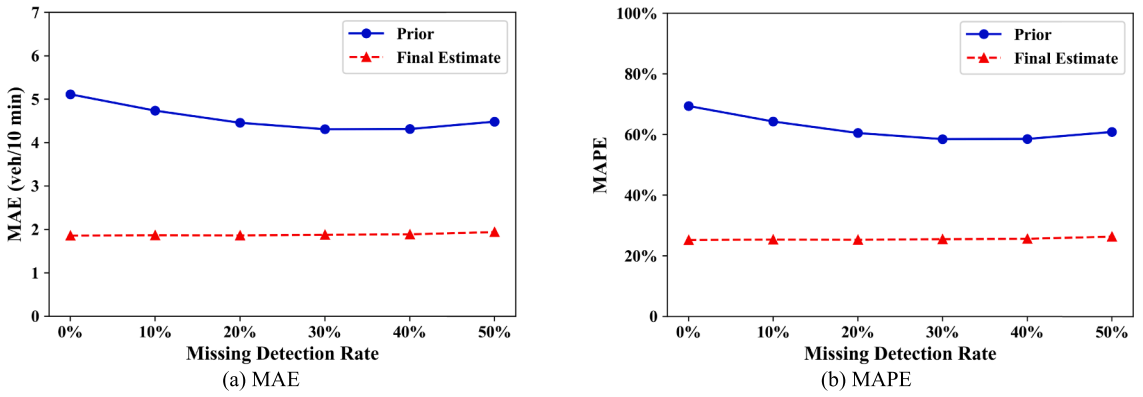


Fig. 17. Sensitivity to missing detection rates.

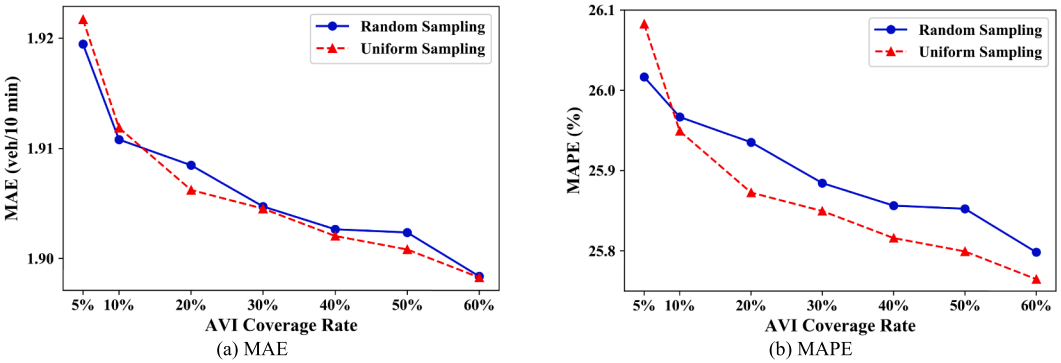


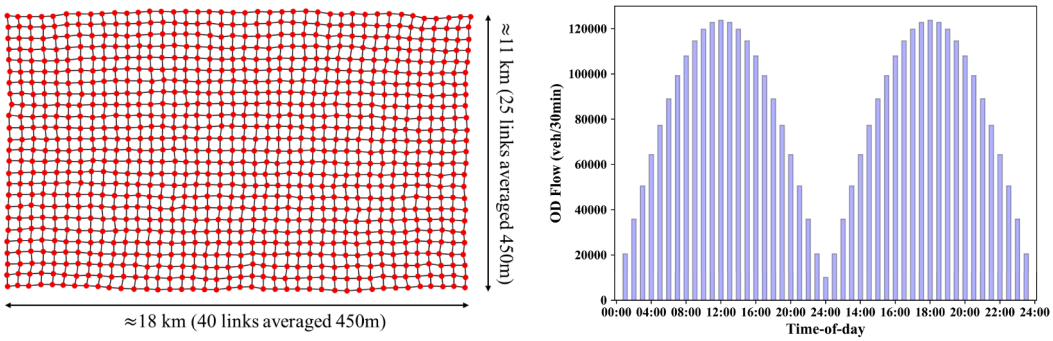
Fig. 18. Sensitivity on location and coverage rate of AVI detectors.

3.3.4. Sensitivity to penetration rates and missing detection rates

In this section, we conducted a sensitivity analysis with respect to CV penetration rates and missing detection rates of AVI. We selected thirteen groups of penetration rates (1–7% with a step of 2% and 10–90% with a step of 10%) and six groups of missing detection rates (0–50% with a step of 10%) to reveal the sensitivity.

As is shown in Fig. 16(a)–(b), with the increasing penetration rates, the prior estimation performance largely reduced. Meanwhile, as presented in both zoom-in plots, the final estimates also showed gradual improvement and became constantly better than the prior. Generally, the increase in CV penetration rate could bring obvious improvement. In addition, the proposed method showed acceptable performance even when only 1% of the vehicles were sampled as CV. Under such circumstances, the prior estimate showed large errors (more than 12 veh/10 min of MAE and around 175% of MAPE), whereas the final estimates achieved a MAE of approximately 2 veh/10 min and a MAPE of approximately 25%. This is largely enabled by the ASC algorithm as it could utilize the neighboring sub-samples to adaptively reduce the noise and estimation variance. The regression plot under different penetration rates is presented in Appendix B to provide more straightforward view of the model performance as well as improvement.

The sensitivity to missing detection rates of AVI detectors is explored under an empirical location of AVI detectors and 10% penetration of CV; the results are presented in Fig. 17(a)–(b). With the increase of missing detection rates, both MAE and MAPE of the prior estimates showed slight improvement in the range of 0–30%, which is counter-intuitive. In this case, when AVI detectors have no missing detection, the estimated penetration rates are generally lower than the true value. Then, when the missing detection rate increased, the denominator of Eqs. (2) and (3) would slightly decrease and lead to penetration rate estimates closer to the true values. When the missing detection rate is larger than 30%, the estimated penetration rates gradually exceed the true values, thus leading to errors. In terms of final estimates, there was a slight increase in errors with increasing missing detection rate, but it remained mostly stable in general. The final MAE was stable at approximately 2 veh/10 min, and the MAPE was stable at approximately 25%. This robustness is largely due to the assumption that AVI detectors could recognize connected and unconnected vehicles. Under such assumption, the average missing detection rate could be estimated and used to correct the flow observations. Though the AVI and CV system are currently not sharing the ID encryption code, we believe that it will not be a problem when vehicle-to-infrastructure (V2I) communication is available in the near future. In conclusion, the impact of missing detection rates is rather trivial, and the regression plot under different missing detection rates is presented in Appendix B.



(a) Network topology (25×40) (b) Time-of-day demand pattern

Fig. 19. Sensitivity to location and coverage rate of AVI detectors.

3.3.5. *Sensitivity to location and coverage of AVI detectors*

The sensitivity to location and coverage rates of AVI detectors is explored in this section. In total, 7 groups of AVI coverage rates (5% and values range from 10% to 60% with interval of 10%) and two types of sampling (random sampling and uniform sampling) were selected for discussion. Here, random sampling means a set of links are randomly sampled from the link set A^o, while uniform sampling means that links are selected according to the index with a constant interval. The corresponding results are presented in Fig. 18.

As the coverage rate increases, the errors showed a decreasing trend, but the contribution of additional AVI detectors was rather trivial as the MAE only decreased by 0.02 veh/10 min and the MAPE decreased by 0.3%. This phenomenon also indicated that the proposed method does not rely much on the quantity of AVI detectors. The reason concerns the usage of AVI observations, and it is two-fold. In the ELP process, the AVI detectors are paired to better approximate the penetration rates of OD pairs. With less AVI detectors available, the unobserved OD pairs would increase but the penetration rate estimates would not deviate severely as it is replaced by average values. In the LCAE, the flow observations are recognized as latent variables. However, because neural networks could effectively capture correlations even when given observations are limited, this usage is also insensitive to the coverage rate of AVI detectors. In terms of sampling type, the results have shown insignificant difference as the proposed method did not include any spatial information regarding AVI detectors.

3.4. *Numerical case*

The proposed method was comprehensively evaluated and analyzed in a realistic arterial network and calibrated simulation regional network. Although good performance was indicated, the test networks were small and medium-scale especially when considering DTA-related applications. Therefore, in this section, a numerical examination was conducted on a synthetic large-scale grid network to investigate the scalability of the proposed method.

The synthetic network was a 25-by-40 (1000 intersections) grid network with link length following uniform distribution $U(400, 500)$, as presented in Fig. 19(a). In this network, 100 nodes are randomly selected as the origin/destination nodes and produce 10,000 OD pairs. The synthetic daily OD demands with respect to OD pairs follow the Poisson distribution with a mean of 10 veh/h. The time-of-day traffic demand pattern is generated by an absolute sine curve, as presented in Fig. 19(b). Between each OD pair, 10 shortest paths based on uncongested path travel time are generated, and the path choice probability is generated by multinomial logit model (Sheffi, 1985). Then, path flows as well as link flows could be aggregated. Considering the size of this network, the aggregation duration, or the length of the time interval, was set be 30 min to increase the trip completion rate within an interval. Following the experiment settings in previous sections, 10% links were randomly selected to be installed with AVI detectors and 5% vehicles were regarded as CV. The missing detection of the AVI detector was omitted in this case, and the average penetration rate of each OD pair was generated from Gaussian distribution, i.e., $\pi_{irs}\ N(0.05, 0.02^2)$.

Within our methodology framework, the scalability issue may occur due to two aspects. The first concerns the sparsity issue. Normally, in a large-scale network, the OD flows tend to be sparser as the travel time lengthens. Then, the sparsity of linear projected prior OD flow matrix would also increase and affect the performance. The second issue concerns the computational efficiency. In the proposed method, the computation bottleneck lies in the *k*-nearest neighbors (*k*-NN) algorithm for obtaining sub-sample similarities. Nevertheless, the *k*-NN algorithm applied in our case could be conducted in parallel (Garcia et al., 2010); after implementing the *k*-NN algorithm on graphical processing unit (GPU), the process is largely accelerated and the computational cost scaled mildly; the corresponding results are listed in Table 6.

As can be seen, the sparsity of the linearly projected OD flow matrix is higher than 65%, which has reduced by nearly 60% after applying the NTD method. Meanwhile, the LCAE-ASC algorithm led to additional reduction and a sparsity of 2.71%, while that of the true OD flow was 2.24%. Thus, we conclude that the sparsity would not tamper the effectiveness of the proposed method in large road networks. Regarding computational efficiency, the total computation time was approximately 390 s, which is much less than the duration of a time interval, 30 min, in this case. This is deemed as an efficient solution for a network of such a large scale, especially

Table 6
General evaluation results of the numerical case.

	Sparsity	MAE (veh/30 min)	MSPE	Time Cost (s)
ELP	68.55%	6.54	66.07%	0.15
NTD	8.01%	1.94	6.45%	26.92
LCAE-ASC	2.71%	1.64	4.77%	361.62

when considering the estimates covered the whole network with a timespan of a month.

4. Conclusion and future work

In this paper, we developed a novel methodology for estimating the dynamic OD flows under day-to-day context based on the fusion of CV trajectories and AVI observations. This method requires neither any external or historical prior information nor assumptions on route choice behavior and dynamic network loading process, and thus, it could be recognized as a generalizable method. In this methodology, two remaining research problems are solved: obtaining reliable prior OD flows given limited observations and effective determination of optimal OD flow estimates based on the priors. Two modules were developed in this methodology to separately deal with the problems. Targeted at the first problem, when there are a few AVI detectors and CVs available, conventional linear projection was extended to lower the temporal and spatial bias, and then, the NTD method was adopted to impute the sparse entries in the projected OD flow. Regarding the second problem, a self-supervised neural network, called the LCAE, is introduced to reconstruct the prior OD flow given robust latent variables. To achieve more accurate estimation and avoid undesirable local optima, day-to-day OD flow recurrences and variations were quantified by vector embedding, and then, the ASC algorithm was developed to make corrections on prior OD flows adaptively. Generally, this methodology is computationally efficient because the estimation for multiple days can be achieved within one effort. In addition, the two proposed modules are both easy to extend and adjust because the OD flow is the only variable passing between them. Moreover, more information that could reflect the dynamics of traffic states (e.g., traffic accident occurred in some day) could be incorporated through extra vector embedding.

The proposed method was examined on a real-world urban arterial network, a regional simulation network, and a synthetic large-scale grid network. The obtained estimates showed competitive accuracies on all networks, and the method was found to outperform two benchmarking GLS models that are largely enabled by CV trajectories. The MSPE in the empirical case was approximately 10%, which indicated substantial improvement compared to benchmarking GLS models, and the MSPE was no greater than 5% in the other two cases. Besides, the proposed method has reacted satisfactorily to different OD flow patterns, e.g., weekdays, weekends, and even holiday. In addition, sensitivity analyses on the regional simulation network are also conducted. The proposed method could produce acceptable estimates when penetration rate is low, and perform slightly better in homogeneous CV sampling case. Generally, the method is insensitive to location of AVI detectors, and the estimation error decreases mildly with increasing AVI coverage. In conclusion, the proposed method could provide effective estimation even when the available data are very limited (e.g., 5% CV and 8% coverage rate of AVI detectors). Additionally, the proposed method has also proved to be efficient in terms of computation cost: similar time cost with convex quadratic programming in a small-scale network and scaled mildly to large networks through proper parallel computing (less than a time interval).

Despite the promising results, the method has certain shortcomings. The method has largely benefited from the optimization of LCAE as it emphasizes on correlations between OD flow and robust network observations. In other words, instead of posing physical constraints (e.g., flow conservation), the time-series characteristics, similarities of day-to-day traffic and numerical mappings (or meta-modeling) between OD flow and network observations are more emphasized in this methodology. As a result, only the OD flow could be effectively obtained while the dynamic path flow, path travel times, and network-wide link flows remained unknown. Although the external route choice models could be applied for inferring meso- and microscopic variables, the consistency and realism may not be guaranteed. Therefore, in the future, we will attempt to achieve a simultaneous solution for OD flow, path flow, and link flow under this framework. Another issue concerns the scalability on larger networks, especially computational burden. Although the method has proven to be time-efficient in the numerical case, the peak memory usage reached more than 8 GB for the synthetic large-scale network. The existing methods for improving the computation performance without harming estimation accuracies will be considered. Besides, for real-world networks, there are often heterogeneous regions with distinct travel pattern, which also requires further investigation. Furthermore, it is also worth mention that there is a collection of studies focused on using mobile phone data (call detail records or GPS data) to estimate OD flow (Bachir et al., 2019; Ge & Fukuda, 2016). Generally, the mobile phone data has larger sample size and coverage rate in the network comparing with the used CV and AVI data, but the data is often aggregated with rougher granularity concerning the privacy issue. The data fusion between these difference data sources (see an example in Wu et al., 2015) is also considered promising in the future.

CRediT authorship contribution statement

Yumin Cao: Conceptualization, Methodology, Validation, Visualization, Writing - original draft. **Keshuang Tang:** Writing - review & editing, Supervision, Funding acquisition. **Jian Sun:** Writing - review & editing, Supervision, Funding acquisition. **Yangbeibei Ji:** Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research is jointly sponsored by the National Key Research and Development Program of China (2018YFB16005), the National Natural Science Foundation of China (61673302, U1764261), and the Shanghai Science and Technology Commission Fund Project (19DZ1208800). The authors would also like to thank the constructive comments of the three anonymous reviewers. Any opinions, findings and conclusions are the responsibility of the authors alone.

Appendix A

The model formulation of Scaled Probe as Prior (SPP) is presented in Eqs. (A.1)–(A.4):

$$\min_{X,Q} \sum_{i=1}^{I} \sum_{rs \in K} \frac{\left(X_{irs} - X_{irs}\right)^2}{\sigma_X^2} + \sum_{i=1}^{I} \sum_{a \in A^o} \frac{\left(Q_{ia} - Q_{ia}^o\right)^2}{\sigma_Q^2} \tag{A.1}$$

$$s.t. Q_{ia} = \sum_{\tau=1}^{|\tau|} \sum_{rs \in K} \vartheta_{rs,a}^{\tau} \cdot X_{(i-\tau)rs}, a \in A^o \tag{A.2}$$

$$-\Delta \le \frac{X_{(i+1)rs} - X_{irs}}{X_{irs}} \le \Delta, i \in I, rs \in K \tag{A.3}$$

$$X_{irs} \ge 0, i \in I, rs \in K \tag{A.4}$$

The model formulation of Penetration Rate Assignment (PRA) is presented in Eqs. (A.5)–(A.9):

$$\min_{X,Q,\pi} \sum_{i=1}^{I} \sum_{rs \in K} \frac{\left(X_{irs} - X_{irs}\right)^2}{\sigma_X^2} + \sum_{i=1}^{I} \sum_{a \in A^o} \frac{\left(Q_{ia} - Q_{ia}^o\right)^2}{\sigma_Q^2} + \sum_{i=1}^{I} \sum_{a \in A^o} \frac{\left(\pi_{ia} - \pi_{ia}^o\right)^2}{\sigma_\pi^2} \tag{A.5}$$

$$s.t. Q_{ia} = \sum_{\tau=1}^{|\tau|} \sum_{rs \in K} \vartheta_{rs,a}^{\tau} \cdot X_{(i-\tau)rs}, a \in A^o \tag{A.6}$$

$$\pi_{ia} = \sum_{\tau=1}^{|\tau|} \sum_{rs \in K} h_{rs,a}^{\tau} \cdot \frac{N_{(i-\tau)rs}}{X_{(i-\tau)rs}}, a \in A^o \tag{A.7}$$

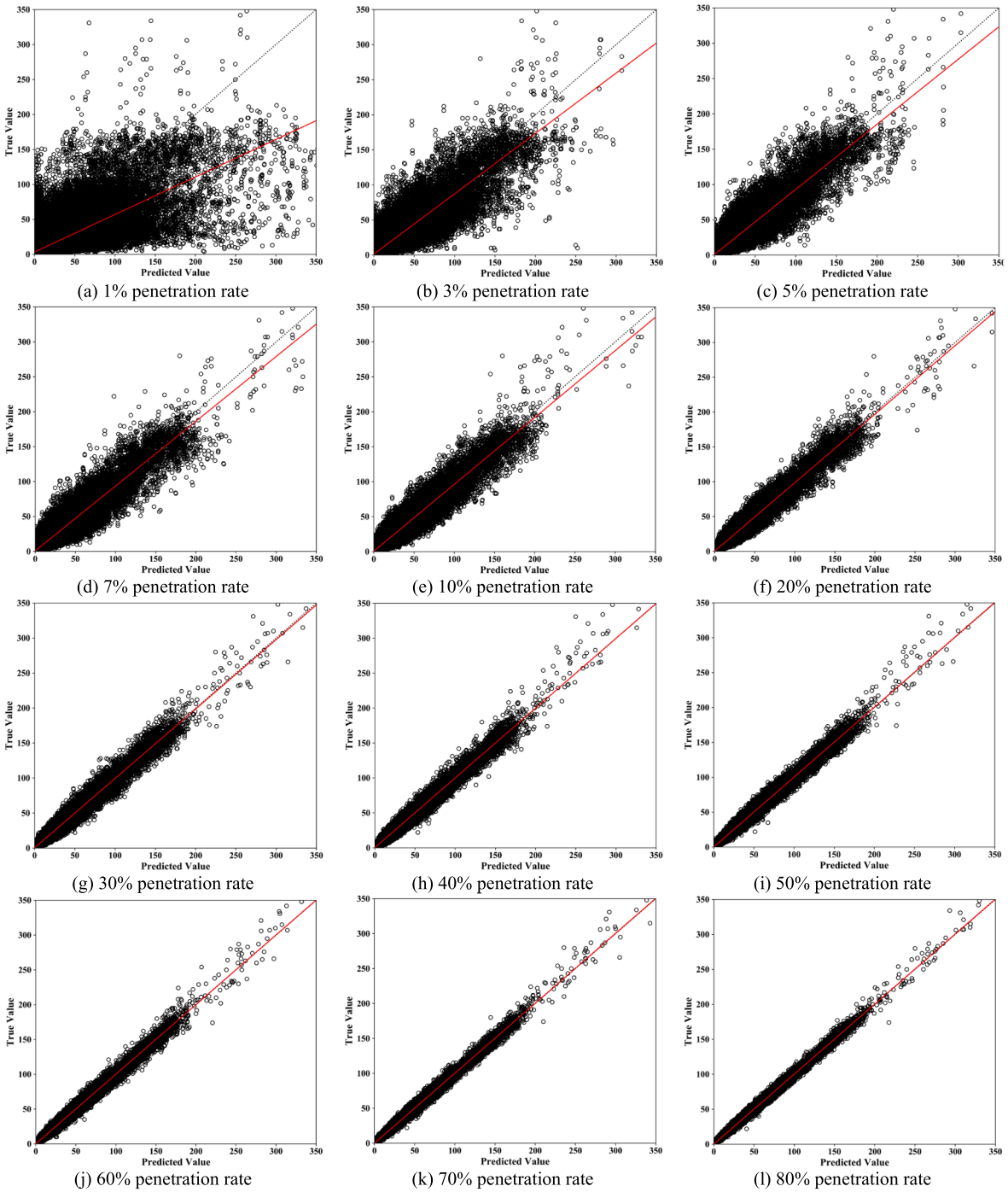
$$-\Delta \le \frac{X_{(i+1)rs} - X_{irs}}{X_{irs}} \le \Delta, i \in I, rs \in K \tag{A.8}$$

$$X_{irs} \ge 0, i \in I, rs \in K \tag{A.9}$$

where, $\sigma_X^2, \sigma_X^2, \sigma_\pi^2$ are respectively the variance of prior matrices $(X_{irs}, Q_{ia}^o, \pi_{ia}^o)$, $\vartheta_{rs,a}^{\tau}, h_{rs,a}^{\tau}$ is the volume and penetration rate assignment fraction from OD pair rs to link a after τ intervals, Δ is a predefined threshold to constrain the change rate of OD flow between two consecutive time intervals.

Appendix B

See Figs. B1 and B2.



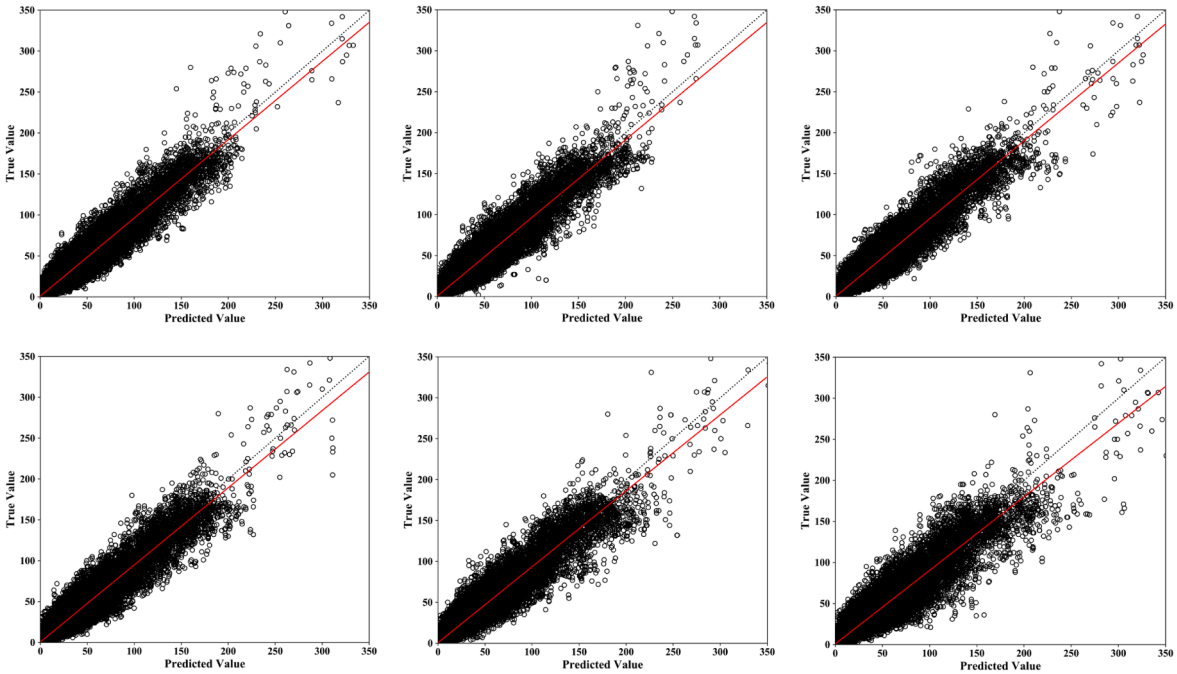
(a) 1% penetration rate (b) 3% penetration rate (c) 5% penetration rate

(d) 7% penetration rate (e) 10% penetration rate (f) 20% penetration rate

(g) 30% penetration rate (h) 40% penetration rate (i) 50% penetration rate

(j) 60% penetration rate (k) 70% penetration rate (l) 80% penetration rate

Fig. B1. Sensitivity on penetration rate of CVs.



(d) 30% missing detection (e) 40% missing detection (f) 50% missing detection

Fig. B2. Sensitivity on missing detection rate of AVI detectors.

References

Arsava, T., Xie, Y., Gartner, N., 2018. OD-NETBAND: an approach for origin-destination based network progression band optimization. Transport. Res. Rec.: J. Transport. Res. Board, 036119811879300.

Ásmundsdóttir, R., 2008. Dynamic OD Matrix Estimation using Floating Car Data (Msc). Delft University of Technology.

Ásmundsdóttir, R., Chen, Y., van Zuylen, H.J., 2010. Dynamic origin-destination matrix estimation using probe vehicle data as a priori information. In: Barceló, J., Kuwahara, M. (Eds.), Traffic Data Collection and its Standardization. Springer New York, New York, NY, pp. 89–108.

Bachir, D., Khodabandelou, G., Gauthier, V., El Yacoubi, M., Puchinger, J., 2019. Inferring dynamic origin-destination flows by transport mode using mobile phone data. Transport. Res. Part C: Emerg. Technol. 101, 254–275. https://doi.org/10.1016/j.trc.2019.02.013.

Bell, M.G.H., 1991. The estimation of origin-destination matrices by constrained generalised least squares. Transport. Res. Part B: Methodol. 25 (1), 13–22. https://doi.org/10.1016/0191-2615(91)90010-G.

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 35 (8), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50.

Cao, P., Miwa, T., Yamamoto, T., Morikawa, T., 2013. Bilevel generalized least squares estimation of dynamic origin-destination matrix for urban network with probe vehicle data. Transp. Res. Rec. 2333 (1), 66–73. https://doi.org/10.3141/2333-08.

Carrese, S., Cipriani, E., Mannini, L., Nigro, M., 2017. Dynamic demand estimation and prediction for traffic urban networks adopting new data sources. Transport. Res. Part C: Emerg. Technol. 81, 83–98. https://doi.org/10.1016/j.trc.2017.05.013.

Cascetta, E., 1984. Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator. Transport. Res. Part B: Methodol. 18 (4), 289–299. https://doi.org/10.1016/0191-2615(84)90012-2.

Cascetta, E., Papola, A., Marzano, V., Simonelli, F., Vitiello, I., 2013. Quasi-dynamic estimation of o–d flows from traffic counts: formulation, statistical validation and performance analysis on real data. Transport. Res. Part B: Methodol. 55, 171–187. https://doi.org/10.1016/j.trb.2013.06.007.

Castillo, E., Conejo, A.J., Menéndez, J.M., Jiménez, P., 2008a. The observability problem in traffic network models. Comput.-Aided Civ. Infrastruct. Eng. 23 (3), 208–222. https://doi.org/10.1111/j.1467-8667.2008.00531.x.

Castillo, E., Menéndez, J.M., Jiménez, P., 2008b. Trip matrix and path flow reconstruction and estimation based on plate scanning and link observations. Transport. Res. Part B: Methodol. 42 (5), 455–481.

Castillo, E., Menéndez, J.M., Sánchez-Cambronero, S., 2008c. Traffic estimation and optimal counting location without path enumeration using Bayesian networks. Comput.-Aided Civ. Infrastruct. Eng. 23 (3), 189–207. https://doi.org/10.1111/j.1467-8667.2008.00526.x.

Chen, X., He, Z., Wang, J., 2018. Spatial-temporal traffic speed patterns discovery and incomplete data recovery via SVD-combined tensor decomposition. Transport. Res. Part C: Emerg. Technol. 86, 59–77. https://doi.org/10.1016/j.trc.2017.10.023.

De Lathauwer, L., De Moor, B., Vandewalle, J., 2000. On the best rank-1 and rank-(R1, R2,., RN) approximation of higher-order tensors. SIAM J. Matrix Anal. Appl. 21 (4), 1324–1342. https://doi.org/10.1137/S0895479898346995.

Dixon, M.P., Rilett, L.R., 2002. Real-time OD estimation using automatic vehicle identification and traffic count data. Comput.-Aided Civ. Infrastruct. Eng. 17 (1), 7–21.

Eisenman, S.M., List, G.F., 2004. Using probe data to estimate OD matrices. In: Paper presented at the Proceedings of The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749).

Feng, Y., Sun, J., Chen, P., 2015. Vehicle trajectory reconstruction using automatic vehicle identification and traffic count data. J. Adv. Transport. 49 (2), 174–194. https://doi.org/10.1002/atr.1260.

Garcia, V., Debreuve, E., Nielsen, F., Barlaud, M., 2010, 26–29 Sept. 2010. K-nearest neighbor search: Fast GPU-based implementations and application to high-dimensional feature matching. In: Paper presented at the 2010 IEEE International Conference on Image Processing.

Ge, Q., Fukuda, D., 2016. Updating origin–destination matrices with aggregated data of GPS traces. Transport. Res. Part C: Emerg. Technol. 69, 291–312. https://doi.org/10.1016/j.trc.2016.06.002.

Houle, M.E., 2017. Local Intrinsic Dimensionality I: An Extreme-Value-Theoretic Foundation for Similarity Applications. In: Paper presented at the Similarity Search and Applications, Cham.
Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv e-prints, arXiv:1502.03167. Retrieved from https://ui.adsabs.harvard.edu/abs/2015arXiv150203167I.
Kim, Y., Choi, S., 2007. Nonnegative tucker decomposition. Paper Presented at the 2007 IEEE Conference on Computer Vision and Pattern Recognition.
Kolda, T.G., Bader, B.W., 2009. Tensor decompositions and applications. SIAM Rev. 51 (3), 455–500. http://www.jstor.org/stable/25662308.
Krishnakumari, P., van Lint, H., Djukic, T., Cats, O., 2020. A data driven method for OD matrix estimation. Transport. Res. Part C: Emerg. Technol. 113, 38–56. https://doi.org/10.1016/j.trc.2019.05.014.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444. https://doi.org/10.1038/nature14539.
Lu, C.-C., Zhou, X., Zhang, K., 2013. Dynamic origin–destination demand flow estimation under congested traffic conditions. Transport. Res. Part C: Emerg. Technol. 34, 16–37. https://doi.org/10.1016/j.trc.2013.05.006.
Lu, L., Xu, Y., Antoniou, C., Ben-Akiva, M., 2015. An enhanced SPSA algorithm for the calibration of Dynamic Traffic Assignment models. Transport. Res. Part C: Emerg. Technol. 51, 149–166. https://doi.org/10.1016/j.trc.2014.11.006.
Ma, W., Pi, X., Qian, S., 2020. Estimating multi-class dynamic origin-destination demand through a forward-backward algorithm on computational graphs. Transport. Res. Part C: Emerg. Technol. 119, 102747. https://doi.org/10.1016/j.trc.2020.102747.
Ma, W., Qian, Z., 2018a. Estimating multi-year 24/7 origin-destination demand using high-granular multi-source traffic data. Transport. Res. Part C: Emerg. Technol. 96, 96–121. https://doi.org/10.1016/j.trc.2018.09.002.
Ma, W., Qian, Z., 2018b. Statistical inference of probabilistic origin-destination demand using day-to-day traffic data. Transport. Res. Part C: Emerg. Technol. 88, 227–256. https://doi.org/10.1016/j.trc.2017.12.015.
Ma, X., Wang, Y., Houle, M. E., Zhou, S., Erfani, S. M., Xia, S.-T., et al., 2018. Dimensionality-Driven Learning with Noisy Labels. arXiv e-prints, arXiv:1806.02612. Retrieved from https://ui.adsabs.harvard.edu/abs/2018arXiv180602612M.
Maher, M.J., 1983. Inferences on trip matrices from observations on link volumes: a Bayesian statistical approach. Transport. Res. Part B: Methodol. 17 (6), 435–447. https://doi.org/10.1016/0191-2615(83)90030-9.
Marzano, V., Papola, A., Simonelli, F., 2009. Limits and perspectives of effective O-D matrix correction using traffic counts. Transport. Res. Part C: Emerg. Technol. 17 (2), 120–132.
Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space. arXiv e-prints, arXiv:1301.3781. Retrieved from https://ui.adsabs.harvard.edu/abs/2013arXiv1301.3781M.
Mo, B., Li, R., Dai, J., 2020. Estimating dynamic origin–destination demand: a hybrid framework using license plate recognition data. Comput.-Aided Civ. Infrastruct. Eng. https://doi.org/10.1111/mice.12526.
Naveh, K.S., Kim, J., 2019. Urban trajectory analytics: day-of-week movement pattern mining using tensor factorization. IEEE Trans. Intell. Transp. Syst. 20 (7), 2540–2549. https://doi.org/10.1109/TITS.2018.2868122.
Nie, Y.M., Zhang, H.M., 2010. A relaxation approach for estimating origin-destination trip tables. Netw. Spatial Econ. 10 (1), 147–172. https://doi.org/10.1007/s11067-007-9059-y.
Osorio, C., 2019. High-dimensional offline origin-destination (OD) demand calibration for stochastic traffic simulators of large-scale road networks. Transport. Res. Part B: Methodol. 124, 18–43. https://doi.org/10.1016/j.trb.2019.01.005.
Peeta, S., Ziliaskopoulos, A.K., 2001. Foundations of dynamic traffic assignment: the past, the present and the future. Netw. Spatial Econ. 1 (3), 233–265. https://doi.org/10.1023/A:1012827724856.
Rao, W., Wu, Y.-J., Xia, J., Ou, J., Kluger, R., 2018. Origin-destination pattern estimation based on trajectory reconstruction using automatic license plate recognition data. Transport. Res. Part C: Emerg. Technol. 95, 29–46. https://doi.org/10.1016/j.trc.2018.07.002.
Shao, H., Lam, W.H.K., Sumalee, A., Chen, A., Hazelton, M.L., 2014. Estimation of mean and covariance of peak hour origin–destination demands from day-to-day traffic counts. Transport. Res. Part B: Methodol. 68, 52–75. https://doi.org/10.1016/j.trb.2014.06.002.
Sheffi, Y., 1985. Urban Transportation Networks, vol. 6. Prentice-Hall, Englewood Cliffs, NJ.
Song, W., Han, K., Wang, Y., Friesz, T.L., del Castillo, E., 2018. Statistical metamodeling of dynamic network loading. Transport. Res. Part B: Methodol. 117, 740–756. https://doi.org/10.1016/j.trb.2017.08.018.
Spiess, H., 1987. A maximum likelihood model for estimating origin-destination matrices. Transport. Res. Part B: Methodol. 21 (5), 395–412. https://doi.org/10.1016/0191-2615(87)90037-3.
Tan, C., Yao, J., Tang, K., Sun, J., 2019. Cycle-based queue length estimation for signalized intersections using sparse vehicle trajectory data. IEEE Trans. Intell. Transp. Syst. 1–16 https://doi.org/10.1109/TITS.2019.2954937.
Tang, K., Cao, Y., Chen, C., Yao, J., Tan, C., Sun, J., 2020. Dynamic origin-destination flow estimation using automatic vehicle identification data: a 3D convolutional neural network approach. Comput.-Aided Civil Infrastruct. Eng. 1–17 https://doi.org/10.1111/mice.12599.
Tang, K., Mei, Y., Li, K., 2014. A simulation-based evaluation of traffic state estimation accuracy by using floating car data in complex road networks. J. Tongji Univ.: Natl. Sci. 9, 1347–1351.
Van Aerde, M., Hellinga, B., Yu, L., Rakha, H., 1993. Vehicle probes as real-time ATMS sources of dynamic OD and travel time data. In: Paper presented at the Large Urban Systems-Proc. of the Advanced Traffic Management Conference.
Van Zuylen, H.J., Willumsen, L.G., 1980. The most likely trip matrix estimated from traffic counts. Transport. Res. Part B: Methodol. 14 (3), 281–293. https://doi.org/10.1016/0191-2615(80)90008-9.
Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A., 2008. Extracting and composing robust features with denoising autoencoders. In: Paper Presented at the Proceedings of the 25th International Conference on Machine Learning. https://doi.org/10.1145/1390156.1390294.
Wong, W., Wong, S.C., 2019. Unbiased estimation methods of nonlinear transport models based on linearly projected data. Transportation Science 53 (3), 665–682. https://doi.org/10.1287/trsc.2018.0856.
Wu, C., Thai, J., Yadlowsky, S., Pozdnoukhov, A., Bayen, A., 2015. Cellpath: Fusion of cellular and traffic sensor data for route flow estimation via convex optimization. Transport. Res. Part C: Emerg. Technol. 59, 111–128. https://doi.org/10.1016/j.trc.2015.05.004.
Wu, X., Guo, J., Xian, K., Zhou, X., 2018. Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph. Transport. Res. Part C: Emerg. Technol. 96, 321–346. https://doi.org/10.1016/j.trc.2018.09.021.
Yang, H., Iida, Y., Sasaki, T., 1991. An analysis of the reliability of an origin-destination trip matrix estimated from traffic counts. Transport. Res. Part B: Methodol. 25 (5), 351–363. https://doi.org/10.1016/0191-2615(91)90028-H.
Yang, H., Sasaki, T., Iida, Y., Asakura, Y., 1992. Estimation of origin-destination matrices from link traffic counts on congested networks. Transport. Res. Part B: Methodol. 26 (6), 417–434. https://doi.org/10.1016/0191-2615(92)90008-K.
Yang, J., Sun, J., 2015. Vehicle path reconstruction using automatic vehicle identification data: An integrated particle filter and path flow estimator. Transport. Res. Part C: Emerg. Technol. 58, 107–126. https://doi.org/10.1016/j.trc.2015.07.003.
Yang, X., Lu, Y., Hao, W., 2017. Origin-Destination Estimation Using Probe Vehicle Trajectory and Link Counts. Journal of Advanced Transportation 2017, 1–18. https://doi.org/10.1155/2017/4341532.
Yang, Y., Fan, Y., Wets, R.J.B., 2018. Stochastic travel demand estimation: Improving network identifiability using multi-day observation sets. Transport. Res. Part B: Methodol. 107, 192–211. https://doi.org/10.1016/j.trb.2017.10.007.
Yao, J., Li, F., Tang, K., Jian, S., 2019. Sampled Trajectory Data-Driven Method of Cycle-Based Volume Estimation for Signalized Intersections by Hybridizing Shockwave Theory and Probability Distribution. IEEE Trans. Intell. Transp. Syst. 1–13 https://doi.org/10.1109/TITS.2019.2921478.
Yildirimoglu, M., Kahraman, O., 2017, 16-19 Oct. 2017. How far is traffic from user equilibrium? In: Paper presented at the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC).

Zhang, H., Chen, P., Zheng, J., Zhu, J., Yu, G., Wang, Y., Liu, H.X., 2019. Missing data detection and imputation for urban ANPR system using an iterative tensor decomposition approach. Transport. Res. Part C: Emerg. Technol. 107, 337–355. https://doi.org/10.1016/j.trc.2019.08.013.

Zheng, J., Liu, H.X., 2017. Estimating traffic volumes for signalized intersections using connected vehicle data. Transport. Res. Part C: Emerg. Technol. 79, 347–362. https://doi.org/10.1016/j.trc.2017.03.007.

Zhou, X., Mahmassani, H.S., 2006. Dynamic origin-destination demand estimation using automatic vehicle identification data. IEEE Trans. Intell. Transp. Syst. 7 (1), 105–114.

Zhu, S., Levinson, D., 2015. Do people use the shortest path? An empirical test of Wardrop's first principle. e0134322-e0134322 PLoS ONE 10 (8). https://doi.org/10.1371/journal.pone.0134322.