Network-wide link travel time and station waiting time estimation using automatic fare collection data: A computational graph approach

Jinlei Zhang, Feng Chen, Lixing Yang, Wei Ma, Guangyin Jin, and Ziyou Gao

***Abstract*—Urban rail transit (URT) system plays a dominating role in many megacities like Beijing and Hong Kong. Due to its important role and complex nature, it is always in great need for public agencies to better understand the performance of the URT system. This paper focuses on an essential and hard problem to estimate the network-wide link travel time and station waiting time using the automatic fare collection (AFC) data in the URT system, which is beneficial to better understand the system-wide real-time operation state. The emerging data-driven techniques, such as the computational graph (CG) method in the machine learning field, provide a new solution for solving this problem. In this study, we first formulate a data-driven estimation optimization framework to estimate the link travel time and station waiting time. Then, we cast the estimation optimization model into a CG-based framework to solve the optimization problem and obtain the estimation results. The methodology is verified on a synthetic URT network and applied to a real-world URT network using the synthetic and real-world AFC data, respectively. Results show the robustness and effectiveness of the CG-based framework. To the best of our knowledge, this is the first time that the CG is applied to the URT. This study can provide critical insights to better understand the operational state in URT.**

***Index Terms*—Urban rail transit, link travel time estimation, station waiting time estimation, computational graph model**

I. Introduction

With the increasing urbanization, urban rail transit (URT) plays an important role in urban transportation systems and becomes increasingly congested. To better operate and manage the complex URT system, understanding passengers' spatiotemporal congestion distributions in transit systems is in great need [1, 2]. ***To this end, this paper aims to estimate the link travel time, which refers to the in-vehicle time from one station to the adjacent station, as well as station waiting time, which refers to the waiting time at the platform of all stations.*** The accurate estimation of both link travel time and station waiting time could help us better understand the operation status of the URT system.

- First, different travel times could help the public agencies track the spatiotemporal congestion patterns and identify the bottlenecks of the URT system. For example, we can identify the congestion level at different subway stations and in different time intervals. Long station waiting time might suggests that the station is likely to be very congested, and then operators can take measures to alleviate congestions for the identified stations.
- Second, the estimated travel times are the essential input to the transit assignment model CITE. To be specific, we can compute the travel time of a specific path, which is the summation of several link travel times and station waiting times. With the path travel time as the disutility, transit assignment can be implemented to obtain the sectional flow. There are four subway operating companies cooperatively operating different lines in Beijing, China. They share revenues according to the sectional flow volume. Therefore, the estimation of link travel time and station waiting time can help these companies calculate revenue splits.
- Third, with accurate path travel time, passengers can arrange their schedules, thus helping them save time and costs [3]. Advertisers can post advertisements in subway stations according to the station waiting time. If there exists a long waiting time in a station, advertisers can post more advertisements to propagandize their products [4].
- Finally, link travel time and station waiting time are fundamental inputs for many other applications, such as inferring passengers' trajectories, estimating path travel time distributions, and analyzing activities [5-7].

Some studies have dived into estimating the link travel time and station waiting time in the URT systems by using the automatic fare collection (AFC) data.

- ***In terms of the link travel time,*** the most straightforward way to estimate the link travel time is to directly extract from the train timetable [8, 9]. However, the **scheduled timetable** data may not be officially provided in

Manuscript received August 4, 2021; date of current version November 20, 2021. This work was supported by the Fundamental Research Funds for the Central Universities (No. 2021RC270), the National Natural Science Foundation of China (Nos. 71621001, 71825004, 52102385), the State Key Laboratory of Rail Traffic Control and Safety（Contract No. RCS2022ZQ001）, Beijing Jiaotong University), and a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU/25209221).

Jinlei Zhang, Lixing Yang, and Ziyou Gao are with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, No.3 Shangyuancun, Haidian District, Beijing 100044, China. (e-mail: zhangjinlei@bjtu.edu.cn; lxyang@bjtu.edu.cn; zygao@bjtu.edu.cn). *(Corresponding author: Feng Chen and Wei Ma.)*

Feng Chen is with the School of Civil Engineering, Beijing Jiaotong University, No.3 Shangyuancun, Haidian District, Beijing 100044, China. (e-mail: fengchen@bjtu.edu.cn).

Wei Ma is with the Department of Civil and Environmental Engineering, the Hong Kong Polytechnic University, Kowloon, Hong Kong SAR, China. (e-mail: wei.w.ma@polyu.edu.hk).

Guangyin Jin is with the College of System Engineering, National University of Defense Technology, Changsha 410005, China (e-mail: jinguangyin18@nudt.edu.cn).

megacities like Beijing, China. Even the **scheduled timetable** data is available, it might not reflect the **actual timetable** as the the trains could not be punctual sometimes. For example, if passengers' clothes were **clamped** by a train gate, the train can be delayed for several seconds [10]. There are indicators in the subway stations showing the real-time **actual timetable** information such as how long the next train will arrive, it is still impractical to conduct a large-scale onsite survey to investigate all the train timetables throughout a day in hundreds of stations [10].

• ***In terms of the station waiting time,*** existing studies generally use a half of the headway as the average station waiting time [8], which cannot reflect the actual traffic conditions. In many real-world URT systems, the dynamic and true station waiting times are not available. ***Besides,*** many studies only estimate the station waiting time at a small number of subway stations or on a few lines [4, 8, 11]. The involved estimation methods include clustering, regression, and maximum likelihood method. In recent years, URT networks in many cities expand rapidly, and hence efficient end-to-end network-wide estimation method is thus in urgent need.

The emerging data-driven computational graph (CG) method in the machine learning area provides a new solution scheme for the estimation of link travel time and station waiting time on a network scale. The CG is a type of computational framework, including nodes and edges. The nodes represent variables or operations on variables. The variables can be scalars, vectors, matrices, or tensors, etc., and operations on variables can be a recurrent neural network, conventional neural network, or fully connected network, etc. The edges represent the data dependency between variables, that is, a flow relationship between variables. Therefore, the CG can also be viewed as a data flow graph. Fig. 1 shows a CG with two fully connected layers in PyTorch. It transforms the input tensor of (64, 1000) dimension into a tensor with a dimension of (64, 100) through the first fully connected layer, and then into a tensor of (64, 10) dimension through the second fully connected layer, which is the final output. The backward propagation algorithm in machine learning is based on the chain rule. In strict accordance with the data flow relationship in the CG, the error is backpropagated from the output node to the input node, so as to optimize parameters on the computational graph. In recent years, CG-based framework have been widely used in various transportation applications, such as the four-step traffic demand model, the OD estimation problem [12, 13], to estimate traffic parameters leveraging multi-source traffic data. ***As an emerging approach to estimate traffic parameters, the data-driven CG-based framework provides a powerful and promising solution for the scenario of traffic problems. However, existing related models focus on road networks, while there is no study using the CG-based framework in URT systems.***

To summarize, there are several challenges in the estimation of link travel time and station waiting time in URT.

• First, the timetable data and real-time operation data may not be available. Therefore, how to estimate these travel times using only the AFC data is challenging.

• Second, with the increasing scale of the URT systems, it is impractical to conduct an onsite investigation (or field survey) to obtain some supplementary information to assist the estimation process.

• Third, existing models generally consist of multiple sub-modules, which makes the entire estimation process complicated. How to incorporate the complicated estimation process in an efficient end-to-end framework also deserves to be explored.

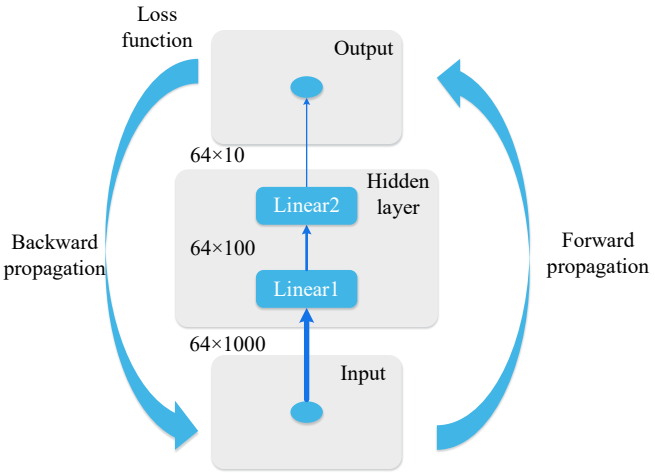


Fig. 1 Diagram of the computational graph

To this end, we apply the CG-based framework to the link travel time and station waiting time estimation problem in the network-wide URT system. Firstly, we formulate the data-driven estimation problem for link travel time and station waiting time. Then, all the involved variables are vectorized in the estimation problem and cast into the CG-based framework. Finally, the methodology is verified using a synthetic URT network as well as the real-world URT network in Beijing, China. ***The contributions of this study are summarized as follows.***

(1) Making use of the AFC data, we formulate a data-driven framework to estimate the link travel time and station waiting time for all the stations and links on a large URT network. No additional information and data are needed in the framework.

(2) It is the first time that the computational graph model is applied to URT systems. The proposed framework is end-to-end in the sense that the AFC data is input into the CG-based framework and the link travel time and station waiting time can be directly estimated.

(3) The CG-based framework is strictly verified on a synthetic URT network and is applied to the real-world URT network in Beijing, China. It is also proved to be more efficient in solving the problem than the three existing benchmark methods, especially for the large-scale transit netwrok.

The remainder of this paper is organized as follows. Existing related works are reviewed in Section II. We define the

problem and notations in Section III. In Section IV, we present the estimation problem and CG-based framework formulation. The case study is conducted in Section IV including the synthetic case and the real-world case. Finally, we draw findings and conclusions in Section VI.

II. LITERATURE REVIEW

A. Link travel time and station waiting time estimation

With the increasing popularity of the automatic fare collection (AFC) system in the URT system, AFC data becomes accessible, and it contains the origin, destination, entry time, and exit time of each trip record. There is a great potential to extract useful travel time information from the AFC data. Therefore, it is of great interest and importance to develop models to estimate the accurate link travel time and station waiting time using AFC data.

Many studies are using the AFC data for different purposes. AFC data can be utilized to analyze the route choice model in URT systems [14-18], and the travel cost can also be estimated from AFC data using conventional statistical methods. The traffic assignment model was thoroughly explored in URT systems with AFC data [19, 20]. Some studies dived into the commuting pattern or travel pattern analysis [21-23], and short-term passenger flow predictions [24-27] using the AFC data. Additionally, passengers' trajectories were also estimated via AFC data [28, 29]. ***According to solution methods, literature in the estimation for the link travel time and station waiting time can be classified into two categories. One aim is to directly estimate these travel times, and the other one is to conduct transit assignment or route choice models, from which the travel times can be estimated.***

In the first category, some researchers classified all trips into transfer trips and non-transfer trips. Zhang et al [10] classified trips into one-transfer and non-transfer trips. They segmented the travel time into walking time, waiting time, in-vehicle time, and transfer time. Clustering and stochastic methods were applied in their study. They also conducted a field survey to obtain the train dwell time, which is supplementary means for the estimation process. However, in some megacities like Beijing, it is infeasible to conduct such a large-scale field survey. Qu et al and Yang et al classified trips into one-transfer, two-transfer, and non-transfer trips to estimate these information [28, 30]. Qu et al [30] only take several origin stations as examples to only estimate the waiting time. Yang et al [28] aimed to reconstruct passengers' trajectories. Both of them used the timetable data as supplementary data to obtain the in-vehicle time. However, in some megacities like Beijing, the timetable data for the entire URT network is hardly available by the public. ***Some researchers only take several stations or lines as examples rather than make network-wide estimations. Only some components of the travel time are estimated rather than all components of the travel time.*** Lee et al. [4] decomposed the travel time into walking time, riding time (in-vehicle time), and waiting time in subway networks, which is similar to our study. Besides, a statistic- and regression-based method is developed to conduct the estimation on a few stations. In our study, we further divide the riding time into link travel times between stations. This kind of partition method is more meaningful because we can get more fine-grained link travel time information. Wu et al. [9] divided the total travel time into walking time, in-vehicle time, and transfer time. The timetable data is available in their study. The similarity of these two studies was that they firstly estimated the walking time using non-transfer AFC records, and then estimated the transfer time using transfer AFC data. Oh et al. [11] incorporated the AFC data and train operation data to estimate the dwell time via the support vector regression-based model. However, the model is specifically for Seoul, South Korea because of the data specificity. Wahaballa et al [31, 32] only studied the platform waiting time and transfer time using multi-source data. They only took several OD pairs as examples and is hard to scale to the network-wide estimation. Zhang et al [33] proposed a Bayesian inference method to infer the walking time, waiting time, in-vehicle time, and transfer time. However, prior knowledge of the in-vehicle time was obtained from the train timetable data. Moreover, they only utilized OD pairs with only one effective route and the Markov Chain Monte Carlo method was applied to solve the problem, which is inefficient for a large-scale network.

In the second category, link travel time and station waiting time are obtained when conducting the assignment model or route choice model. In these studies, the time information usually constitutes the cost function of traffic assignment models [34, 35]. Zhu et al proposed a "passenger to train" assignment framework [5, 36]. During the process of assigning an individual passenger to train, some travel time components could be estimated using AFC data and automated vehicle location data. Under the same framework, Singh et al [37] estimated the entry, on-train, and exit time, and Li et al. [8] estimated the waiting and walking time in subway stations. In their study, only a small portion of the stations instead of the entire subway network was studied. Sun et al [19] proposed ***Bayesian inference method to conduct flow assignment model.*** During the assignment process, some travel time information can be estimated. However, in their study [19], the network is undirected and the travel costs of the same route for different directions are the same. Tian et al [38] proposed a "TripDecoder" framework using ***maximum likelihood estimator and stochastic gradient descent method*** to simultaneously infer the link travel times and route choice model. Sun and Xu [39] obtained the walking time and transfer time during the inference of route choice proportions. The two studies were designed only for single OD pairs and was challengeable when the network is critically large.

All in all, existing link travel time and station waiting time estimation-related studies only consider limited OD pairs, stations, or subway lines. Train timetable or other real-time operation data are generally supplementary data required in the estimation process, and existing models are challenging to scale up for the network-wise estimation. To highlight the uniqueness of this paper, differences between some important existing studies and this study are summarized in Table 1.

Table 1 Comparison between similar studies

	Studies	Information estimated*	Cases	Data	Supplementary data	Supplementary means	Methods
Direct estimation	Zhang et al [10]	①②③④	Subway network	AFC data	None	Onsite survey	Clustering, statistic method
	Qu et al [30]	②	One subway line (25 stations)	AFC data	Train timetable	None	Statistic method
	Yang et al [28]	①③	One OD pair	AFC data	Train timetable	None	Clustering, statistic method
	Lee et al. [4]	①②④	Two stations	AFC data	None	None	Clustering, statistic method, regression
	Wu et al. [9]	①③④	Ten OD pairs	AFC data	Train timetable	None	Clustering, statistic method, normal distribution method
	Oh et al. [11]	⑤	One subway line (twelve stations)	Smart card-based passenger information data	Real-time metro operation data	None	Support vector regression
	Wahaballa et al [32]	②	Three origin and 27 destination stations	AFC data	Train timetable and performance report data	None	Likelihood estimation method, stochastic frontier modeling
	Zhang et al [33]	①②③④	OD pairs with one route	AFC data	Train timetable	Travel behavior survey	Bayesian inference method
Indirect estimation (leverage assignment or route choice model)	Zhu et al [36]	①③④	One subway line (Five stations)	AFC data	Automated vehicle location data	None	"Passenger to train" assignment framework
	Singh et al [37]	①④	Three line sections	AFC data	Train location data	None	Regression, "passenger to train" assignment framework
	Li et al [8]	①②	Four stations	AFC data	Train timetable	None	Clustering, maximum likelihood method, "passenger to train" assignment framework
	Sun et al [19]	①②③④	Subway network	AFC data	None	None	Bayesian inference method, flow assignment model
	Tian et al [38]	⑥	OD pairs with one route	AFC data	None	None	Maximum likelihood estimation, stochastic gradient descent, route choice model
	Sun et al [39]	②③	Two OD pairs	AFC data	Train timetable	Manual surveys	Statistic method, route choice model
	This study	***⑥***	***Subway network***	***AFC data***	***None***	***None***	***Computational graph model***

*Information estimated: ①: walking time; ②: waiting time; ③: transfer time; ④: in-vehicle time; ⑤: dwell time (The duration of time of a train stopped at a station); ⑥: link travel time and station waiting time.

*Note that the link travel time includes the in-vehicle time, entry walking time, exit walking time, and transfer time. The station waiting time includes the waiting time at origin stations and transfer stations.

B. Computational graph

In recent years, CG-based framework have been widely used in various transportation applications. Some studies began to cast the conventional four-step traffic demand model and OD estimation problem into the CG-based framework. Leveraging the backward propagation algorithm in TensorFlow or PyTorch, the CG can solve the specific parameters in the model. For example, Wu et al. [13] cast household travel surveys, mobile phone sample data, sensor data, floating car data, and vehicle location/identification data into a hierarchical CG-based framework. Through the forward propagation of multiple data sources and the backward propagation of the loss errors of travel time, many demand variables in the three of the four-step traffic demand model (namely trip generation, spatial distribution estimation, and path flow-based traffic assignment) were estimated. This framework has favorable interpretation and was beneficial to better understand the traffic behaviors. Similar to this study, Sun et al. [40] further analyzed the marginal effects of all kinds of traffic policies, e.g., adding the toll on links, adding flows to OD pairs, removing flows from OD pairs. The marginal effects and potential traffic migration caused by various traffic strategies were quantified by casting the various traffic policies individually or cooperatively into the above hierarchical CG-based framework. Ma et al [12] solved the dynamic OD estimation problem under a large-scale road network using the CG-based framework. Through the forward propagation of initialized OD demand and the backward propagation of the loss errors of link flows or link speeds, many parameters used in the dynamic traffic assignment models and route choice models, such as OD demand, route choice probability, dynamic assignment ratio, and link/path travel time, were estimated. Focusing on discrete choice modeling applications, Kim et al [41] integrated the strengths of econometric models and machine learning algorithms using the CG-based framework. They proposed a CG-oriented cost functional representation and results showed that the CG-based choice modeling can produce consistent estimates of parameters with substantial computational efficiency. ***In summary, the data-driven CG-based framework provides a powerful and promising solution for the scenario of traffic problems. However, existing related models are all about the road network. There is no study using the CG-based framework in URT.***

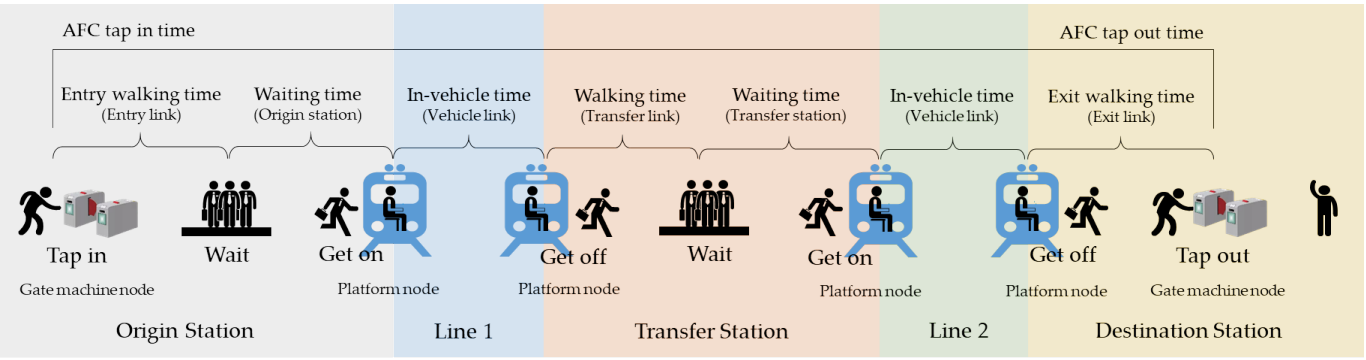


Fig. 2 Diagram of passengers' trip behavior

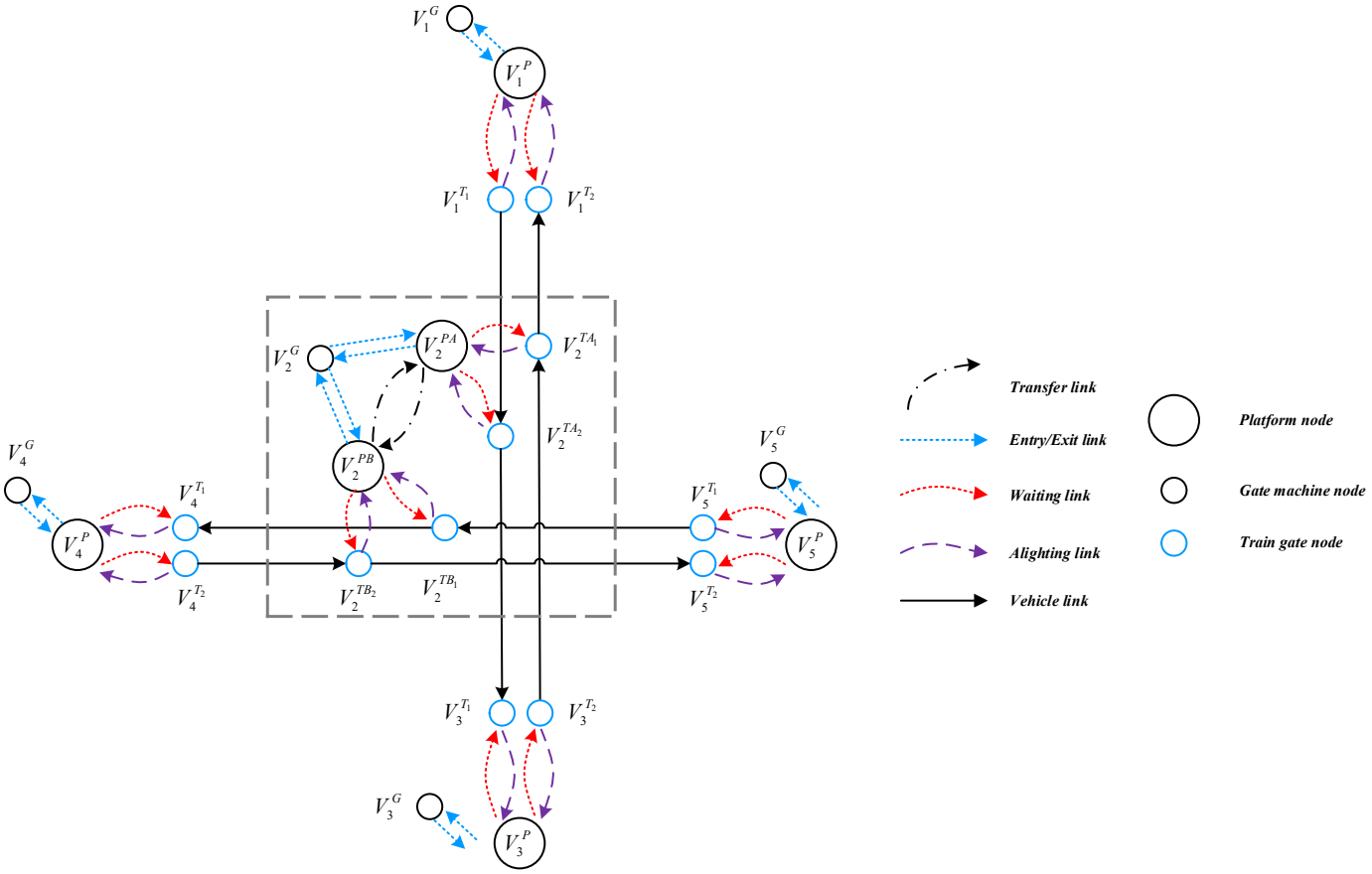


Fig. 3 Diagram of network representation

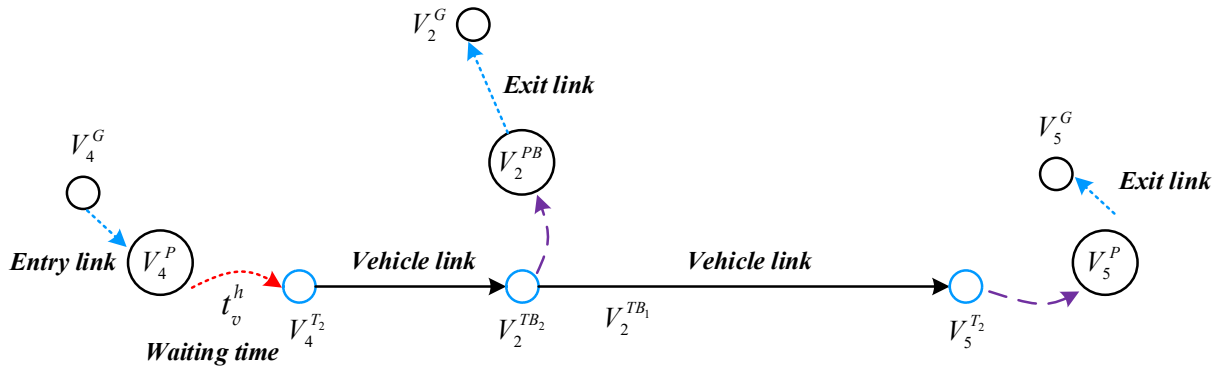


Fig. 4 An example of the problem statement

III. PROBLEM DEFINITION

A. Travel time decomposition

Each trip of a passenger from tap-in to tap-out in URT systems can be represented into several phases as shown in Fig. 2 [9]. In the origin station, the passenger will tap in and walk to the platform to wait for the next train. If there are queues, he/she cannot board the closest train and have to wait for another train. Therefore, the waiting time will depend on the queue length. After departing from the origin station, the passenger will take line 1 to go to the transfer station, if any. In the transfer station, he/she will walk to another platform and wait for the next train. Similar to the origin station, there is a waiting time. After departing from the transfer station, the passenger will take line 2 to go to the destination station and walk to tap out.

The complete trip can be encapsulated into the subway network graph $G = (V, E)$, where V is the set of all nodes, including the gate machine node, platform node, and train gate node. E is the set of all links, including the entry link, exit link, vehicle link, transfer link, and the virtual waiting link and alighting link. ***Different links can be either a uplink or a downlink, and the overall set is E.*** If the direction of link a→b is the uplink, the direction of link b→a is the downlink. The links from different directions have different attributes are modeled separately in ***E***. In summary, a complete trip includes six types of links and three types of nodes. The meanings of the notations are listed as follows.

For links:

(1) Entry link: The links from the entry gate machine to the platform. It can be also called the "boarding link".
(2) Exit link: The links from the platform to the exit gate machine.
(3) Vehicle link: The links that passengers are in the vehicle.
(4) Transfer link: The links that passengers transfer from one subway line to another subway line at transfer stations.
(5) Alighting link: Virtual links denote the link from train gate node to platform node. In this study, we assume that there in no time spent on the alignting link.
(6) Waiting link: Virtual links denote the link from platform node to train gate node. In this study, the time spent on the waiting link is to be estimated.

For nodes:

(7) Gate machine node: The node at the gate machine.
(8) Platform node: The node at the platform.
(9) Train gate node: The node at the train gate.

The AFC data can record the card number, entry station, entry time, exit station, exit time. Therefore, the true travel time of a single trip is available.

B. Network representation

In order to further illustrate the problem, a simplified URT network with 5 subway stations and two subway lines is constructed as shown in Fig. 3. Note that this simplified network is just for illustrative purpose, and in the case study section, we will construct a complete synthetic network to demonstrate the performance of the proposed method.

The station can be divided into the general station and transfer station. All stations are numbered according to the line number and station adjacency relationship. During the network modeling process, we divide the general station into four nodes, namely one gate machine node, one platform node, and two train gate nodes, as shown in Fig. 3. The number of the gate machine is the original station number with the letter G, such as V_1^G, V_2^G, V_3^G. The number of the platform node is the original station number with the letter P, such as V_1^P, V_2^P, V_3^P. The number of the train gate nodes are the original station number with the letter T_1 and T_2, indicating different directions, such as $V_1^{T_1}, V_1^{T_2}$. We divide the transfer station into seven nodes, denoting the gate machine node V_2^G, the original platform node V_2^{PA}, the virtual platform node V_2^{PB}, and four train gate nodes $V_2^{TA_1}, V_2^{TA_2}, V_2^{TB_1}, V_2^{TB_2}$, as shown in Fig. 3.

The links in the synthetic are divided into entry link, exit link, vehicle link, and transfer link, as shown in Fig. 3. Every link includes the attributes of the origin, destination, direction, line number, travel time, and link name.

C. Problem statement

In this study, we will estimate the link travel time and the station waiting time, as shown in Fig. 3, only using the AFC data. The link travel time includes walking time on the entry link, transfer link, and exit link, as well as the in-vehicle time on the vehicle link. The station waiting time includes waiting time at the platform node of the origin station or transfer station. The problem is cast into the CG-based framework, with inputs being the AFC data and outputs being the journey time components.

D. An intuitive example

Taking path 1 (station 4 to station 2) and path 2 (station 4 to station 5) as shown in Fig. 4 as an example, we will estimate the waiting time t_v^h at the node V_4^P of the station 4, as well as the link travel time of the entry link, vehicle link, and exit link only using the AFC data. For path 1 and path 2, passengers only wait at station 4. Therefore, the station waiting time only exists at station 4. Passengers tap in at node V_4^G and wait at node V_4^P to board. Some passengers alight at node $V_2^{TB_2}$ and tap out at node V_2^G, and others alight at node $V_5^{T_2}$ and tap out at node V_5^G. Hence, the travel time of five links needs to be estimated. In brief, we will estimate the link travel time (five links) and station waiting time (one station) for path 1 and path 2.

E. Notations

All notations are summarized in Table 2.

Table 2 Notations in the framework

Notations	Definitions
G	The URT network graph
V	The set of all nodes in the URT network
E	The set of all links in the URT network
$a \in E$	The link in the URT network. All links are directed.
$v \in V$	The node in the URT network

r	The origin station
s	The destination station
K_q	The set of all OD pairs
k	The k_{th} shortest path
K_{rs}	The set of k shortest paths from station r to s
$h \in H=(h_1, h_2, h_3 \ldots)$	The time interval that passengers enter the URT network
t_a	The link travel time that passengers pass the link a
t_v^h	The passengers' waiting time at station v during the time interval h
α_{rs}^{ka}	The coefficient of t_a
β_{rs}^{kvh} and γ_{rs}^{kvh}	The coefficient of t_v^h
c_{rs}^{kh}	The cost function
$\boldsymbol{\Psi}_{rs}^{kh}$	The generalized route choice model
θ	The parameter of the logit model
p	The probability that passengers choose the k path
$\tilde{\boldsymbol{c}}$	The vector of the true travel time
$\boldsymbol{B}$	The vector of the link
$\boldsymbol{M}^h$	The vector of the node
$\boldsymbol{A}$	The combined vector of the link and node
$\boldsymbol{t}_a$	The vector of the link travel time
$\boldsymbol{t}_v^h$	The vector of the link travel time and station waiting time
$\boldsymbol{t}$	The combined vector of the link travel time and the station waiting time
$\boldsymbol{P}$	The vector of the route choice probability

IV. METHODOLOGY

In this section, we present the estimation framework in large-scale URT networks. The framework comprises two parts: (1) Define the estimation framework on the basis of the passengers' route choice model; (2) Variable vectorization and development of a CG-based framework to solve this problem.

A. Passengers' route choice model

In Section 2.1, we have described the passengers' behavior during a single trip. Before choosing this route, passengers' route choice behaviors are affected by many factors including the walking time on the entry link and exit link, in-vehicle time on the vehicle link, transfer walking time on the transfer link, waiting time at the platform nodes of the origin station or transfer station, transfer times, and ticket fare, etc. In this study, we choose the link travel time, station waiting time, and ticket fare as the generalized cost function, as shown in Eq. (1). In Beijing, China, the ticket fare is determined according to the distance. Generally, it is 0.5 Yuan per kilometer.

$$c_{rs}^{kh} = \sum_a \alpha_{rs}^{ka} t_a + \sum_v \beta_{rs}^{kvh} t_v^h + \sum_v \gamma_{rs}^{kvh} t_v^h + \sum_a 0.5\delta_a l_a \quad (1)$$

where c_{rs}^{kh} is the generalized cost, $rs \in K_q$, K_q is the set of OD pairs, $k \in K_{rs}$, K_{rs} is the path set from the origin station r to destination station s, $h \in H=(h_1, h_2, h_3 \ldots)$ is the time interval that passengers enter the URT network, $a \in E$ is the link edge set in the URT network graph G, l_a is the length of link a, $v \in V$ is the station node set in the URT network graph G, t_a is the travel time that passengers pass the link a, t_v^h is the passengers' waiting time at station v during the time interval h, α、β and γ are the coefficient of t_a and t_v^h. The δ is the coefficient of link a. They are defined as follows.

The α_{rs}^{ka} equals 1 if the path k of OD pair rs passes the link a, otherwise equals 0.

The β_{rs}^{kvh} equals 1 if the path k of OD pair rs boarding at the origin or transfer station v in the uplink during time interval h, otherwise equals 0.

The γ_{rs}^{kvh} equals 1 if the path k of OD pair rs boarding at the origin or transfer station v in the downlink during time interval h, otherwise equals 0.

The δ_a equals 1 if the link a is vehicle link otherwise equals 0.

In this study, we assume that all passengers have the same waiting time at a certain station and in a specific time interval. The dwell time is included in the waiting time as passengers are still waiting in the train. The overall waiting time and dwell time will be amortized to each passenger in order to obtain an averaged waiting time t_v^h.

Several meanings should be noted in the generalized cost function.

(1) The h is the time interval that passengers enter the URT network. Hence, the c_{rs}^{kh} is associated with the time interval h. For the same route, there is different c_{rs}^{kh} in different time intervals.

(2) The t_v^h is also associated with the time interval h. For the same route, there is a different t_v^h in different time intervals.

(3) The t_v^h is associated with the direction of the train. For the same station in the same time interval, there is a different t_v^h. Therefore, the coefficients of t_v^h for uplink and downlink are β_{rs}^{kvh} and γ_{rs}^{kvh}, respectively.

(4) We assume that the walking time and in-vehicle time is independent of the time interval. Hence, α_{rs}^{ka} and t_a is not associated with the time interval h.

(5) We assume that the stochastic cost is subject to the independent identical distribution and Gumbel distribution. Therefore, the route choice model can be defined as Eq. (2).

$$p_{rs}^{kh} = \boldsymbol{\Psi}_{rs}^{kh}(\boldsymbol{c}, \boldsymbol{t}) \quad (2)$$

$$\mathbf{c} = \left\{ c_{rs}^{kh} \middle| h \in H, rs \in K_q, k \in K_{rs} \right\}$$

$$\mathbf{t} = \{ t_a, t_v^h \,|\, h \in H, a \in E, v \in V \}$$

where p_{rs}^{kh} is the probability that passengers choose the k path between OD pair rs within the time interval h. $\boldsymbol{\Psi}_{rs}^{kh}$ is a generalized route choice model that is used to compute the probability. c_{rs}^{kh} is the cost function as shown in Eq. (1), including the link travel time t_a and station waiting time t_v^h.

B. Estimation framework: A data-driven optimization model

In this section, we will formulate the estimation framework leveraging the AFC data. Through minimizing the difference between the estimated OD travel time and the true OD travel time, we will estimate the link travel time (including the walking time when entering or exiting the station and in-vehicle time) and station waiting time (including the waiting time at the origin station and the transfer station). Therefore, we will first formulate the true OD travel time and the estimated OD travel time.

C. The true OD travel time

For one piece of the AFC data, let the entry station, entry time, exit station, exit time be the r, $\tilde{t}_{ir}$, s, $\tilde{t}_{is}$, respectively. Therefore, the true travel time can be formulated as Eq. (3).

$$\tilde{c}_{irs}^{h} = \tilde{t}_{is} - \tilde{t}_{ir} \tag{3}$$

where $\tilde{c}_{irs}^{h}$ is the true travel time of passenger i for the OD pair rs. Because the AFC data cannot record that which path of the OD pair rs the passengers chose, the $\tilde{c}_{irs}^{h}$ is not associated with the path k of the OD pair rs.

There are many records for the same OD pair in the same time interval. Among these records, the travel time is generally different because of the different entry time and different routes. To reduce the computational cost, we choose the true average OD travel time as the true OD travel time, as Eq. (4).

$$\tilde{c}_{rs}^{h} = \frac{1}{N_{rs}^{h}} \sum_{i=1}^{N_{rs}^{h}} \tilde{c}_{irs}^{h} = \frac{1}{N_{rs}^{h}} \sum_{i=1}^{N_{rs}^{h}} (\tilde{t}_{is} - \tilde{t}_{ir}) \tag{4}$$

where N_{rs}^{h} is the record number of the OD pair rs in the time interval h, $\tilde{c}_{irs}^{h}$ is the true travel time for the specific OD pair rs in the time interval h.

D. The estimated OD travel time

Given the route choice probability p_{rs}^{kh} and travel cost c_{rs}^{kh}, the estimated average OD travel time of the OD pair rs in the time interval h is shown as Eq.(5). When estimating the OD travel time, we should consider the probability of different paths of the OD pair rs. Therefore, the c_{rs}^{kh} is associated with the path k of the OD pair rs.

$$c_{rs}^{h} = \sum_{k} p_{rs}^{kh} \, c_{rs}^{kh} \tag{5}$$

E. The optimization model

The objective is to minimize the difference between the estimated OD travel time and the true OD travel time，as shown in Eq. (6)

$$\min \quad \sum_{rsh} |\tilde{c}_{rs}^{h} - c_{rs}^{h}|_{p}^{2} \tag{6}$$

Considering the Eqs. (1)-(6), the optimization model used in this study can be formulated as Eq. (7). The objective function is to minimize the difference between the estimated OD travel time and the true OD travel time. The decision variables are the link travel time t_a and the waiting time t_v^h.

$$\begin{aligned} \min_{\{t_a\}_a, \{t_v^h\}_{v,h}} \quad & \sum_{rsh} \left| \tilde{c}_{rs}^{h} - \sum_{k} p_{rs}^{kh} \, c_{rs}^{kh} \right|_{p}^{2} + |\tilde{t}_a - t_a|_{p}^{2} + |\tilde{t}_v^h - t_v^h|_{p}^{2} \\ s.t. \quad & t_a \geq 0, a \in E \\ & t_v^h \geq 0, v \in V \\ & \text{Eqs. (1)-(2)} \end{aligned} \tag{7}$$

F. Variable vectorization

Before casting the optimization model into a CG-based framework to obtain the optimal solution, we will vectorize the variables used in Eq. (7), including the true travel time $\tilde{c}_{rs}^{h}$, the links a, the nodes v, the link travel time t_a, the station waiting time t_v^h, and the route choice probability p. After vectorization, the supervised tasks under the CG-based framework are constructed.

The summary of all variable vectorizations is shown in Table 3. The detailed vectorizing process is shown in the supplementary file.

Table 3 Variable vectorization

Variable	Scalar	Vector	Dimension	Type
True travel time	$\tilde{c}_{r_i s_i}^{h}$	$\tilde{\boldsymbol{c}}$	$R^{(N_{rs}\times H)\times 1}$	Dense
Links	α_{rs}^{ka}	$\boldsymbol{B}$	$R^{(N_{rs}\times k)\times N_a}$	Sparse
Nodes	$\beta_{rs}^{kvh}, \gamma_{rs}^{kvh}$	$\boldsymbol{M}^h$	$R^{(N_{rs}\times k)\times N_v}$	Sparse
Links and nodes	$\alpha_{rs}^{ka_1}, \beta_{rs}^{kvh}, \gamma_{rs}^{kvh}$	$\boldsymbol{A}$	$R^{(N_{rs}\times k\times H)\times(N_a+N_v\times H)}$	Sparse
Link travel time	t_a	$\boldsymbol{t}_a$	$R^{N_a\times 1}$	Dense
Station waiting time	t_v^h	$\boldsymbol{t}_v^h$	$R^{N_v\times 1}$	Dense
Variable to be estimated	t_a、t_v^h	$\boldsymbol{t}$	$R^{(N_a+N_v\times H)\times 1}$	Dense
Route choice probability	p_{rs}^{kh}	$\boldsymbol{P}$	$R^{(N_{rs}\times H)\times(N_{rs}\times k\times H)}$	Sparse

Based on Table 3, the optimization model in Eq. (7) can be reformulated as Eq. (8).

$$\begin{aligned} \min_{\{\boldsymbol{t}\},\theta} \quad & \|\tilde{\boldsymbol{c}} - \boldsymbol{P}\boldsymbol{A}\boldsymbol{t}\|_2^2 \\ s.t. \quad & \boldsymbol{t} \geq 0 \\ & \boldsymbol{P} = logit(\boldsymbol{A}; \boldsymbol{t}; \theta) \end{aligned} \tag{8}$$

The main decision variable in the above optimization model is $\boldsymbol{t}$, including the link travel time (walking time when entering or exiting the station and in-vehicle time) and station waiting time (waiting time at origin station or transfer station). When the matrix $\boldsymbol{A}$ is with full rank, there is the only solution for the optimization problem.

G. A computational graph for the optimization model

In order to solve the optimization model, we cast it into the CG framework.

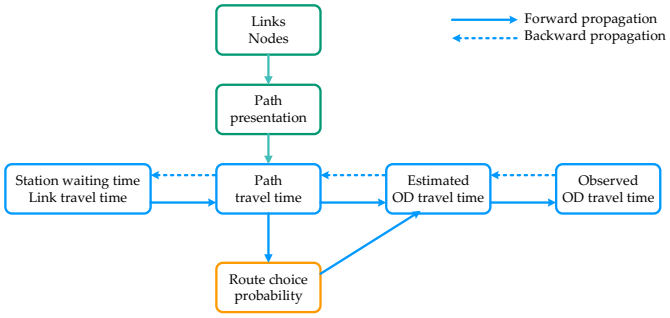


Fig. 5 shows the CG framework of this problem. The framework consists of two parts, the forward propagation part and the backward propagation part.

The forward propagation part is shown as the solid line in

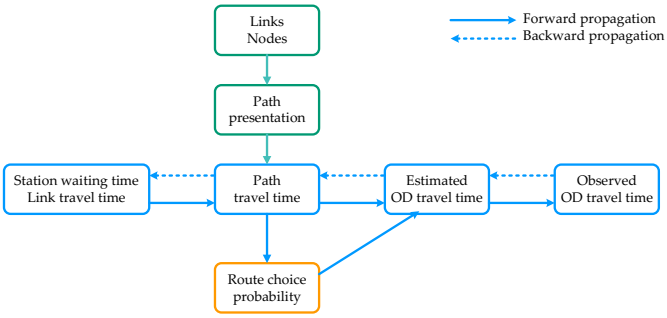


Fig. 5. During forward propagation, we assume that the station waiting time and link travel time are fixed. By calculating the path travel time and route choice probability, the estimated OD travel time is obtained, and the estimated OD travel time is compared with the true one to obtain the estimated error. In the backward propagation part, which is shown as the dashed line in

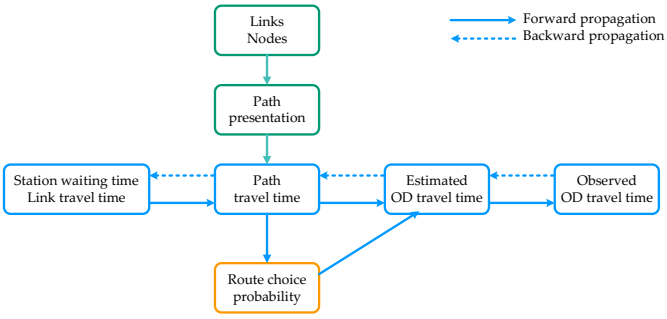


Fig. 5, the error is backward propagated according to the original path to adjust the station waiting time and link travel time. In this way, repeated iteration calculation is carried out until the iteration stops, and the final estimated station waiting time and link travel time are obtained.

Fig. 6 shows the corresponding variable propagation process. The solid line represents the forward propagation, and the dashed line represents the backward propagation. Through the forward propagation of variables and the backward propagation of errors, the final $\boldsymbol{t}$ composed of station waiting time and link travel time is obtained.

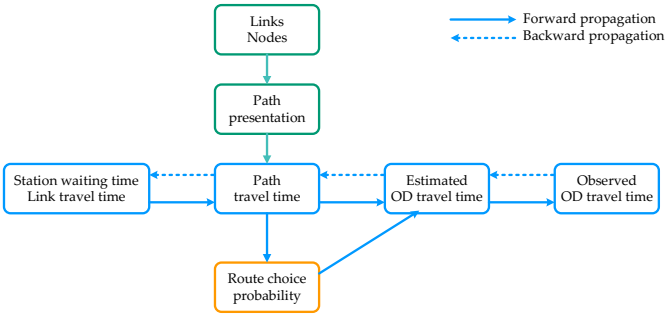


Fig. 5 The computational graph framework: a diagram of the forward-backward algorithm

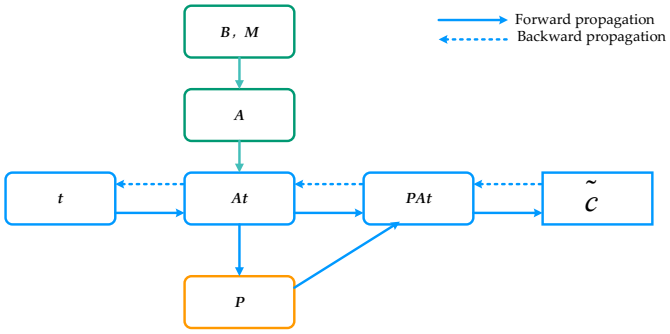


Fig. 6 The diagram of the forward-backward algorithm among all variables

Forward propagation: The input is the randomly initialized $\boldsymbol{t}$ composed of the station waiting time $\boldsymbol{t}_v^h$ and link travel time $\boldsymbol{t}_a$. Together with the path representation $\boldsymbol{A}$, the path travel time $\boldsymbol{c}_{path}$ is obtained. According to the path travel time, the route choice probability $\boldsymbol{P}$ is obtained. The path travel time $\boldsymbol{c}_{path}$ multiply the route choice probability $\boldsymbol{P}$ to get the estimated OD travel time.

The objective of the forward propagation is to obtain the estimated OD travel time. Let the objective function of Eq. (8) be the loss function in the machine learning field, the Eq. (8) can be decomposed into Eq.(9).

$$
\begin{aligned}
&L = \|\tilde{\boldsymbol{c}} - \boldsymbol{c}\|_p^2 \\
&\boldsymbol{c} = \boldsymbol{P}\boldsymbol{c}_{path} \\
&\boldsymbol{P} = \mathrm{softmax}(-\theta \boldsymbol{c}_{path}) \\
&\boldsymbol{c}_{path} = \boldsymbol{A}\boldsymbol{t}
\end{aligned}
\tag{9}
$$

where $\tilde{\boldsymbol{c}}$ is the true average OD travel time extracted from the AFC data,$\boldsymbol{c}$ is the estimated average OD travel time, $\boldsymbol{P}$ is the route choice model obtained using the logit model, $\boldsymbol{c}_{path}$ is the path travel time.

Backward propagation: During this process, we take the route choice probability $\boldsymbol{P}$ as a known variable. Then, Eq. (8) can be simplified into Eq. (10).

$$
\begin{aligned}
&\min_{\{\boldsymbol{t}\},\theta} \quad \|\tilde{\boldsymbol{c}} - \boldsymbol{P}\boldsymbol{A}\boldsymbol{t}\|_2^2 \\
&s.t. \quad \boldsymbol{t} \geq 0
\end{aligned}
\tag{10}
$$

Further, the gradient of the objective function can be obtained as shown in Eq. (11).

$$
\begin{aligned}
&\frac{\partial L}{\partial \boldsymbol{c}} = 2(\tilde{\boldsymbol{c}} - \boldsymbol{P}\boldsymbol{A}\boldsymbol{t}) \\
&\frac{\partial L}{\partial \boldsymbol{c}_{path}} = \boldsymbol{P}^T \frac{\partial L}{\partial \boldsymbol{c}} \\
&\frac{\partial L}{\partial \boldsymbol{t}} = \boldsymbol{A}^T \frac{\partial L}{\partial \boldsymbol{c}_{path}}
\end{aligned}
\tag{11}
$$

By consolidating Eq. (11), the gradient of the objective function L against $\boldsymbol{t}$ can be obtained, as shown in Eq.(12). Finally, the problem of the Eq. (8) can be solved leveraging the backpropagation algorithm.

$$
\frac{\partial L}{\partial \boldsymbol{t}} = \boldsymbol{A}^T \boldsymbol{P}^T 2(\tilde{\boldsymbol{c}} - \boldsymbol{P}\boldsymbol{A}\boldsymbol{t}) \tag{12}
$$

V. CASE STUDY

In this section, we first present the synthetic and real-world data. The model configuration is then described. The result analyses are finally shown.

A. Data Description

In this section, the synthetic and real-world data are described in detail. Due to the page limit, the detailed data examples (Table I to IV) are presented in the supplementary material.

1) Synthetic data

In order to verify the proposed framework, a synthetic URT network is constructed. The network modeling method is the same as that in Section III. Fig. 7 shows the synthetic subway network with 3 subway lines, including one ring line and 2 general lines, and 14 subway stations, including 8 general stations and 6 transfer stations. The green line is line 1, including 7 stations (stations 1, 2, 3, 4, 5, 6, 7). The yellow line is the ring line 2, including 5 stations (stations 8, 9, 104, 10, 102). Station 104 and station 102 are the same as station 4 and station 2, respectively. The blue line is the line 3, including 8 stations (station 11, 109, 103, 110, 12, 13, 106, 14). Station 109, 103, 110, and 106 are the same as stations 9, 3, 10, 6. In the diagram, the blue nodes are the general stations and the green nodes are the transfer stations.

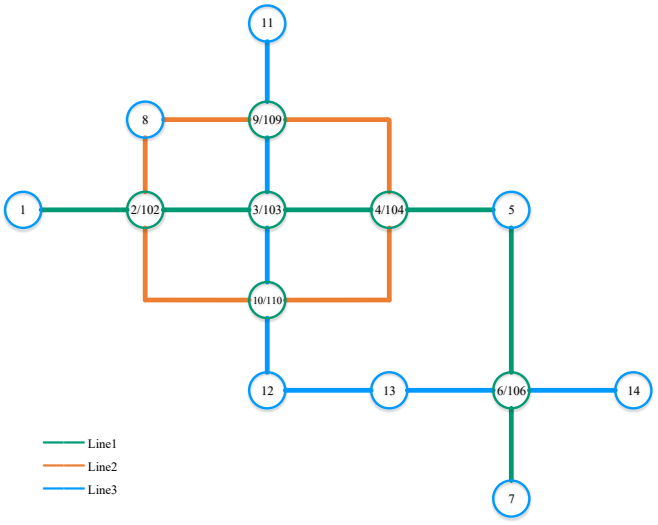


Fig. 7 Diagram of the synthetic URT network

In the synthetic network, the link travel time and station waiting time are artificially generated. According to this information, the synthetic AFC data are artificially constructed. Therefore, all the true traffic state information in this synthetic network is available.

The synthetic AFC data is shown in Table I in the supplementary material. The travel time includes theoretical true travel time and travel time with errors. The theoretically true travel time does not include any noises. The travel time with white noises is obtained by applying a random noise subject to normal distribution to the theoretically true travel time in order to simulate the real travel environment.

When constructing AFC data, note that: (1) To verify the proposed methodology, we construct enough AFC data for all OD pairs. However, when conducting sensitivity analysis, we will delete some data records of some OD pairs to simulate the actual AFC data. (2) The number of data records can be different for different OD pairs. (3) For the same OD pair, the data records for different k paths are generated according to the logit model. (4) For the same OD pair, the number of data records varies in different time intervals in order to simulate the demand changes in peak hours and off-peak hours. (5) To verify the proposed methodology, the travel time is the theoretically true travel time without white noises. However, when conducting sensitivity analysis, we will use the travel time with white noises to simulate the travel time obtained from the actual AFC data. The example of synthetic AFC data for the same OD pair and different k paths is shown in Table II in the supplementary material.

2) Real-world data

The dataset from Beijing Subway in China on March 7, 2016, is used in the experiment, as shown in Table III in the supplementary material. The travel time is obtained by subtracting the entry time from the exit time, and the station number is unique for each station in Beijing. The entry time and exit time are converted to minutes (There are 1440 minutes throughout the day). The topology structure of the URT network in Beijing is also available, which includes link distance information between adjacent stations and the transfer walking time. We set the average train speed of the Beijing subway as 30 km/h [42]. The walking time is further transformed into the driving distance of the train to search for the k shortest path. An example of the link distance information is shown in Table IV in the supplementary material.

3) Comparison between the synthetic and real-world network

To present a straightforward overview of the synthetic and real-world networks, we list their detailed information as shown in Table 4.

Table 4 The detailed information of the synthetic and real-world network

Attributes		Synthetic network	Real-world network
Node number	Total	74	1254
	Gate machine node	14	276
	Platform node	20	326
	Train gate node	40	652
Link number	Total	168	2689
	Entry link	20	326
	Waiting link	40	652
	Vehicle link	36	633
	Alighting link	40	652
	Exit link	20	326
	Transfer link	12	100
OD pair number		182	75, 900
Routes number		426	204, 627
Line number		3	16
Station number	Total	14	276
	General station	8	226
	Transfer station	6	50
AFC data number		18.1 million	20.9 million
Capacity of trains		/	1920
Capacity of platforms		/	5 p/m^2*
Time period		7:00 am to 12:00 am,	
Time interval		30 min	
Optimizer		AdaGrad	
Learning rate		0.1	
Loss function		MSE	
Batch size		8	1024*5

Epochs		20	10
Time consumed		8.81 s	1, 135 s
Desktop Specs	Intel Core i7-8700K, 24GB RAM, NVIDIA GeForce GTX 1070 Ti		

* This data is from "*Code for Design of Urban Rail Transit*" in the Chinese standard.

B. Model configuration, efficiency, and accuracy

1) Configuration

For the synthetic data and real-world data, the configurations are all listed in Table 4. We randomly choose 100 epochs to show the loss variation as in Fig. 8 and Fig. 9. After 20 and 10 epochs for synthetic and real-world data, respectively, the model remains stable with a good convergence effect.

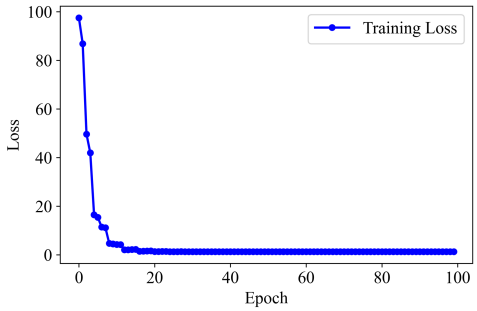


Fig. 8 The convergence curve of training loss for synthetic data

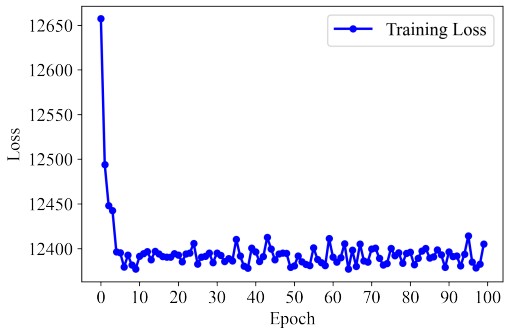


Fig. 9 The convergence curve of training loss for real-world data

2) Efficiency

To analyze the efficiency of the proposed CG-based framework, we adopt the delayed Stochastic Gradient Descent (SGD) framework [12, 43]. Suppose that there are n workers process and one master process, the communication time between the worker and master is τ_c, and τ_I represents the time cost for completing the forward and backward iteration once. To run the CG for N times, the total time cost is $N\tau_c + \left\lceil \frac{N}{n} \right\rceil \tau_I$. Note that each process can be viewed as a computer, a central processing unit (CPU), or a graphics processing unit (GPU). For large-scale networks, τ_I is larger than τ_c, hence the CG-based framework with multiple process could speed up the solution process significantly by the factor of n.

Table 5 Mean and standard deviation of the running time for different methods (unit: seconds)

Methods	Number of Time Intervals							
	10	20	30	40	50	60	70	80
OSQP	14.19±0.02	42.23±0.29	115.54±0.34	276.61±5.19	504.70±6.28	—	—	—
SCS	27.65±0.54	83.89±1.99	197.95±5.34	405.7±2.73	695.30±19.53	—	—	—
NNLS	28.85±0.36	161.71±3.43	535.17±24.01	1300.09±63.76	2509.36±32.28	—	—	—
CG	**8.79±0.44**	**25.89±1.53**	**75.18±8.43**	**179.24±6.64**	**329.17±13.34**	**531±16.54**	**791±19.61**	**1112±22.97**

Table 6 The residual error for different methods (unit: No. of passengers / 30mins)

Methods	Number of Time Intervals							
	10	20	30	40	50	60	70	80
OSQP	7.76±0.00	11.12±0.00	11.31±0.00	**11.82±0.00**	12.12±0.00	—	—	—
SCS	7.76±0.00	11.12±0.00	11.31±0.00	**11.82±0.00**	12.12±0.00	—	—	—
NNLS	7.98±0.01	11.38±0.02	11.61±0.02	12.09±0.01	12.46±0.01	—	—	—
CG	**7.75±0.00**	**11.11±0.01**	**11.29±0.00**	11.83±0.01	**12.10±0.02**	**12.51±0.01**	**12.98±0.01**	**13.27±0.00**

In terms of the time complexity, we first collect the overall time consumed both for the synthetic and the real-world network. For the synthetic network, the algorithm is nearly real-time, only consuming 8.81 seconds under 10 time intervals, indicating extremely high efficiency. For the real-world network, the time consumed for the 5-hour morning peak period in Beijing is 1135 seconds (about 20 minutes), which is also acceptable.

Then, we compare the proposed CG-based framework with existing solvers such as Non-Negative Least Square (NNLS) solver [44], Operator Splitting Solver for Quadratic Programs (OSQP) [45], and Splitting Conic Solver (SCS) [46]. On the synthetic network, we run the experiments 5 times with time intervals being [10, 20, 30, 40, 50, 60, 70, 80], and the mean and standard deviation of the running time is presented in Table 5. As is shown, the CG-based framework is significantly faster than the existing methods, showing its high advantages in solving this problem. With the increasing of time interval number, the efficiency advantage of CG becomes more obvious. When the time intervals are larger than 50, all other methods fail to give the result on the synthetic network.

On the real-world network, however, no matter how many the time interval is, all other methods fail to give the result because of the memory overflow or exceeding the maximum time budget (24 hours).

3) Accuracy

Because all other methods fail to give the result for the real-world netwrk, we only compare the estimation accuracy on the synthetic network. On the synthetic network, we run the experiments 5 times with time intervals being [10, 20, 30, 40, 50], and the residual error for different methods is presented in Table 6.

C. Result analysis for synthetic subway networks

In this section, we will verify the methodology using the synthetic subway network and the synthetic AFC data. For the synthetic network, all the true traffic states are available, including the true link travel time, the true station waiting time, and the true path travel time. The estimated $\boldsymbol{t}$ includes the estimated link travel time and station waiting time in different time intervals. Leveraging these estimated link travel times and station waiting times, we can reconstruct the estimated path travel time. Therefore, the estimated and the true link travel time, station waiting time, and path travel time are all available. The summary of this information is shown in Table 7.

Table 7 Information summary for the synthetic and real-world subway network

		Estimated values	True values
Synthetic	Link travel time	√	√
	Station waiting time	√	√
	Path travel time	√	√
Real-world	Link travel time	√	×
	Station waiting time	√	×
	Path travel time	√	√

1) Link travel time and station waiting time

The comparison between the true and the estimated link travel time and station waiting time (delay time) is shown in Fig. 10. The estimated values and the true values are also generally equal to each other. The R^2 reaches up to 0.999, which proves the effectiveness of the proposed methodology.

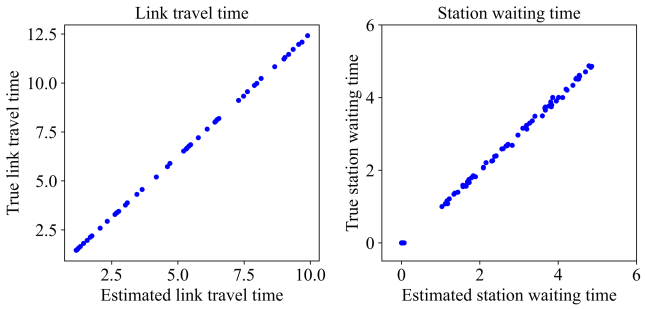


Fig. 10 The comparison between the estimated and the true time (unit: mins)

2) Path travel time

The comparison between the true and the estimated path travel time is shown in Fig. 11. The estimated values and the true values are generally equal to each other. The R^2 reaches up to 0.999, which proves the effectiveness of the proposed methodology.

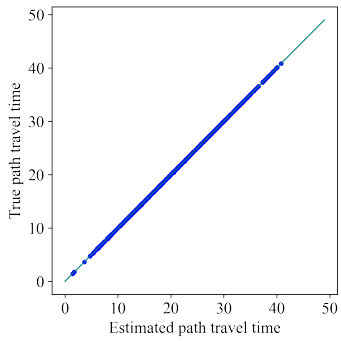


Fig. 11 The comparison between the estimated and the true OD path travel time (unit: mins)

3) Sensitivity analysis

For the real-world AFC data, the path travel time always contains noises. Moreover, there is no AFC record for many OD pairs. To simulate the real-world AFC data, we conduct the sensitivity analysis in this section. At first, we add 10% and 20% white noise to the path travel time of all AFC records. On the basis of the 20% white noise, we randomly delete the records of 20% and 50% OD pairs. For the deleted OD pairs, the corresponding true average OD path travel time in the vector $\boldsymbol{t}$ in Eqs. (4) is zero. Therefore, we use the "softimpute" method to conduct matrix completion. The R^2 between the true and the estimated values are computed for the four cases as shown in Table 8. Given the page limit, we put the detailed results into the supplementary material.

From the first case to the fourth case, the errors continue to increase, which is reasonable because more interference factors are imposed onto the original AFC data. However, the estimated path travel times are always highly aligned with their corresponding true values, showing that the proposed methodology is effective.

Table 8 The goodness-of-fitting (R^2) in different cases

	Link travel time	Station waiting time	Path travel time
Synthetic-Original	0.999	0.999	0.999
Adding 10% white noise	0.999	0.999	0.999
Adding 20% white noise	0.998	0.987	0.999
Deleting 20% OD pairs	0.996	0.972	0.999
Deleting 50% OD pairs	0.981	0.827	0.998
Real-world	/	/	0.700

D. Result analysis for real-world subway networks

In this section, we will apply the methodology to the Beijing Subway network using real-world AFC data. For the real-world network, the true link travel time and the true station waiting time are unavailable. However, the true path travel time is available. The estimated $\boldsymbol{t}$ includes the estimated link travel time and station waiting time in different time intervals. Leveraging these estimated link travel time and station waiting time, we can reconstruct the estimated path travel time. Therefore, the estimated path travel time is also available. The summary of this information is shown in Table 7.

1) Convergence and goodness-of-fitting

In the real-world subway network, the comparison between the true and the estimated path travel time is shown in Fig. 12. As we can see, there exist some errors in the real-world network.

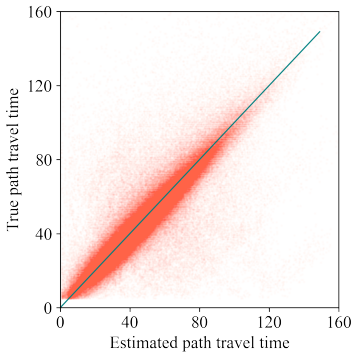


Fig. 12 The comparison between the estimated and the true OD path travel time (unit: mins)

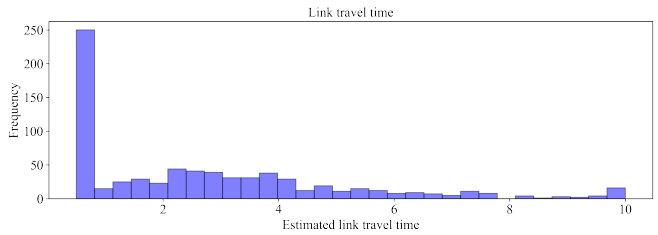


Fig. 13 The summary of the estimation results for all link travel times (unit: mins)

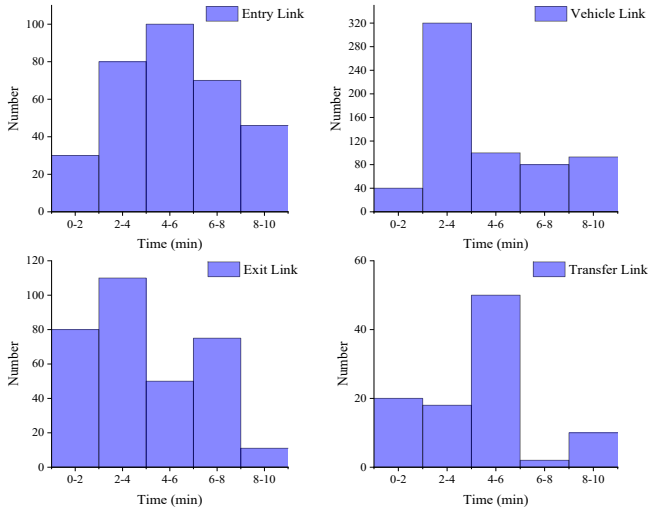


Fig. 14 The summary of the estimation results for different link types (unit: mins)

However, the estimated values and the true values are generally aligned with each other. The R^2 of real-world results can reach up to 0.700 as shown in Table 8, which is generally acceptable because so many external factors affect the travel time. Therefore, results show the effectiveness of the proposed methodology for the application in the real world.

2) Link travel time

The summary of the estimation results for all link travel times and different link types is shown in Fig. 13 and Fig. 14. There are nearly 250 links that cost small travel time (less than 1 minute), such as the transfer links in the same platforms. Most link travel time ranges from 1 minute to 5 minutes, which conforms to real-world operation environments. The estimation results are also consistent with common sense. Besides, we compute the average speed on the vehicle links using the estimated time and the distance, which is known. The vehicle speed ranges from 29 km/h to 38 km/h, which is compatible with the official average speed, namely 30 km/h [42].

3) Station waiting time in different time intervals

A summary of the estimation results for the station waiting time is shown in Fig. 15. From the first time interval (07:00-07:30) to the tenth time interval (11:30-12:00), the number of the larger waiting times increases, while the number of the smaller waiting times decreases, which is in line with common sense. In the sixth time interval (09:00-09:30), the station waiting time seems to be the largest. We select several specific stations to present their waiting time variations in different time intervals as shown in Fig. 16. The estimation results are also reasonable, which shows the effectiveness of the proposed methodology.

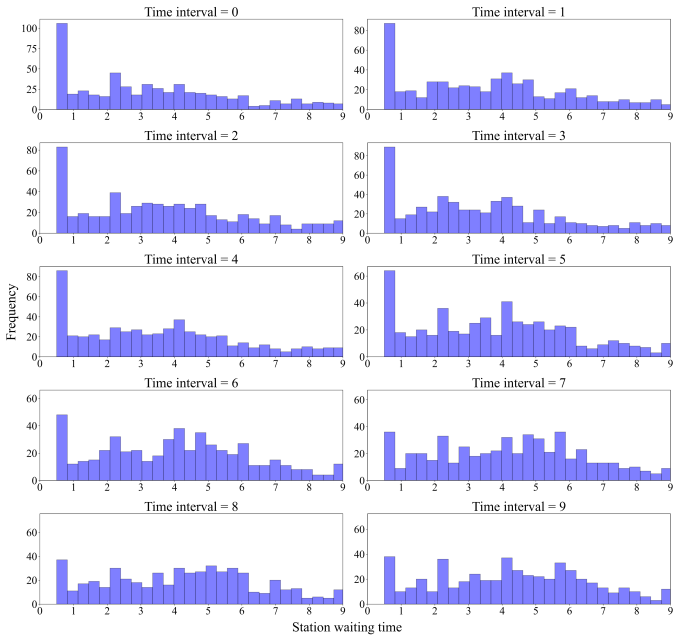


Fig. 15 The summary of the estimation results for the station waiting time in different time intervals (unit: mins)

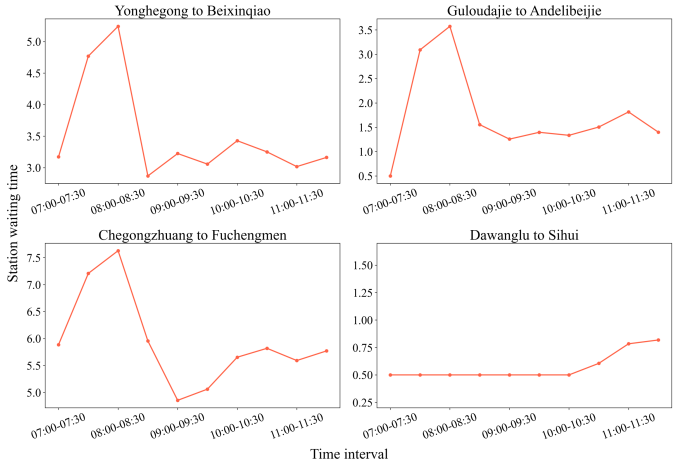


Fig. 16 The specific station waiting time variation in different time intervals

VI. Conclusion

In this study, we propose a CG-based framework to estimate the link travel time and station waiting time using the AFC data for large-scale URT networks. We first formulate a data-driven optimization model to estimate the link travel time and station waiting time. The optimization model is then cast into a CG-based framework to obtain the estimation results efficiently. The proposed framework is strictly verified on a synthetic URT network and applied to a real-world URT network. To the best of our knowledge, this is the first time that the CG-based method is applied to the URT system. Several critical findings are summarized as follows.

(1) The CG-based framework is proved to be effective and efficient in the estimation of the link travel time and station waiting time under a URT network level.

(2) Sensitivity analysis on the synthetic network indicates that the proposed CG-based framework is robust to the data noise and data missing.

(3) The CG methodology can be verified using the AFC data in real-world, and the estimated OD path travel time in the real-world network is proved to be reasonable.

Overall, these findings can provide critical insights for future research. For example, the CG-based framework has potentials to be applied to the multi-modal traffic systems [47]. If the AFC data for both subway and bus systems are available, the developed CG-based framework can be extended to estimate the transfer time between the adjacent subway station and bus station. Travel time reliability can be measured and passengers' trajectories can be estimated using the estimation results [48, 49].

In practice, several considerations should be taken. For example, when we apply this research in the real world, the entry/exit/transfer walking time needs to be carefully calibrated using field survey data or timetable data. The large-scale timetable data can also be integrated into the estimation process to improve the estimation accuracy. Computer vision techniques can be used to detect and count passengers. Therefore, the monitoring video data from cameras in the subway station, together with the congestion identification results according to the station waiting time, can be used to simultaneously monitor the congestion. We assume that all passengers board the train simultaneously and once passengers board the train, the train will drive to the next station, which should be further refined in future study. Some accidents should also be taken into consideration when applying this research. The error distribution in the link travel time and station waiting time for the real-world URT systems should be analyzed in detail, and the estimated travel time should also be incorporated into the proposed CG-based framework. We assume that the link travel time is not time-dependent, which may not be true in some of the URT systems and should be calibrated in practice. Moreover, whether the CG-based framework can be applied on weekends is to be further explored.

VII. References

[1] J. Zhang, H. Che, F. Chen, W. Ma, and Z. He, "Short-term origin-destination demand prediction in urban rail transit systems: A channel-wise attentive split-convolutional neural network method," Transportation Research Part C: Emerging Technologies, vol. 124, pp. 102928, 2021.

[2] J. Zhang, F. Chen and Q. Shen, "Cluster-based LSTM network for short-term passenger flow forecasting in urban rail transit," IEEE Access, vol. 7, pp. 147653-147671, 2019.

[3] J. Zhang, F. Chen, Y. Guo, and X. Li, "Multi-graph convolutional network for short-term passenger flow forecasting in urban rail transit," IET Intelligent Transport Systems, vol. 14, pp. 1210-1217, 2020.

[4] H. Lee, D. Zhang, T. He, and S. H. Son, "Metrotime: Travel time decomposition under stochastic time table for metro networks," in 2017 IEEE International Conference on Smart Computing (SMARTCOMP), 2017, pp. 1-8.

[5] Y. Zhu, H. N. Koutsopoulos and N. H. Wilson, "Passenger itinerary inference model for congested urban rail networks," Transportation Research Part C: Emerging Technologies, vol. 123, pp. 102896, 2021.

[6] Z. Dai, X. Ma and X. Chen, "Bus travel time modelling using GPS probe and smart card data: A probabilistic approach considering link travel time and station dwell time," Journal of Intelligent Transportation Systems, vol. 23, pp. 175-190, 2019.

[7] L. Yang, Y. Zhang, S. Li, and Y. Gao, "A two-stage stochastic optimization model for the transfer activity choice in metro networks," Transportation Research Part B: Methodological, vol. 83, pp. 271-297, 2016.

[8] W. Li, X. Yan, X. Li, and J. Yang, "Estimate Passengers' Walking and Waiting Time in Metro Station Using Smart Card Data (SCD)," IEEE Access, vol. 8, pp. 11074-11083, 2020.

[9] J. Wu, Y. Qu, H. Sun, H. Yin, X. Yan, and J. Zhao, "Data-driven model for passenger route choice in urban metro network," Physica A: Statistical Mechanics and its Applications, vol. 524, pp. 787-798, 2019.

[10] F. Zhang, J. Zhao, C. Tian, C. Xu, X. Liu, and L. Rao, "Spatiotemporal segmentation of metro trips using smart card data," IEEE Transactions on Vehicular Technology, vol. 65, pp. 1137-1149, 2015.

[11] Y. Oh, Y. Byon, J. Y. Song, H. Kwak, and S. Kang, "Dwell time estimation using real-time train operation and smart card-based passenger data: A case study in Seoul, South Korea," Applied Sciences, vol. 10, pp. 476, 2020.

[12] W. Ma, X. Pi and S. Qian, "Estimating multi-class dynamic origin-destination demand through a forward-backward algorithm on computational graphs," Transportation Research Part C: Emerging Technologies, vol. 119, pp. 102747, 2020.

[13] X. Wu, J. Guo, K. Xian, and X. Zhou, "Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph," Transportation Research Part C: Emerging Technologies, vol. 96, pp. 321-346, 2018.

[14] B. Mo, Z. Ma, H. N. Koutsopoulosc, and J. Zhao, "Assignment-based path choice estimation for metro systems using smart card data," arXiv preprint arXiv:2001.03196, 2020.

[15] C. Yu, H. Li, X. Xu, and J. Liu, "Data-driven approach for solving the route choice problem with traveling backward behavior in congested metro systems," Transportation Research Part E: Logistics and Transportation Review, vol. 142, pp. 102037, 2020.

[16] Y. Zhang, E. Yao, K. Zheng, and H. Xu, "Metro passenger' s path choice model estimation with travel time correlations derived from smart card data," Transportation Planning and Technology, vol. 43, pp. 141-157, 2020.

[17] Y. Zhang, E. Yao, J. Zhang, and K. Zheng, "Estimating metro passengers' path choices by combining self-reported revealed preference and smart card data," Transportation Research Part C: Emerging Technologies, vol. 92, pp. 76-89, 2018.

[18] J. Zhao, F. Zhang, L. Tu, C. Xu, D. Shen, C. Tian, X. Li, and Z. Li, "Estimation of passenger route choice pattern using smart card data for complex metro systems," IEEE Transactions on Intelligent Transportation Systems, vol. 18, pp. 790-801, 2016.

[19] L. Sun, Y. Lu, J. G. Jin, D. Lee, and K. W. Axhausen, "An integrated Bayesian approach for passenger flow assignment in metro networks," Transportation Research Part C: Emerging Technologies, vol. 52, pp. 116-131, 2015.

[20] F. Zhou and R. Xu, "Model of passenger flow assignment for urban rail transit based on entry and exit time constraints," Transportation Research Record, vol. 2284, pp. 57-61, 2012.

[21] X. Ma, C. Liu, H. Wen, Y. Wang, and Y. Wu, "Understanding commuting patterns using transit smart card data," Journal of Transport Geography, vol. 58, pp. 135-145, 2017.

[22] X. Ma, Y. Wu, Y. Wang, F. Chen, and J. Liu, "Mining smart card

data for transit riders' travel patterns," Transportation Research Part C: Emerging Technologies, vol. 36, pp. 1-12, 2013.
[23] J. Zhao, Q. Qu, F. Zhang, C. Xu, and S. Liu, "Spatio-temporal analysis of passenger travel patterns in massive smart card data," IEEE Transactions on Intelligent Transportation Systems, vol. 18, pp. 3135-3146, 2017.
[24] J. Zhang, F. Chen, Z. Wang, and H. Liu, "Short-term origin-destination forecasting in urban rail transit based on attraction degree," IEEE Access, vol. 7, pp. 133452-133462, 2019.
[25] D. Yang, K. Chen, M. Yang, and X. Zhao, "Urban rail transit passenger flow forecast based on LSTM with enhanced long-term features," IET Intelligent Transport Systems, vol. 13, pp. 1475-1482, 2019.
[26] Z. Guo, X. Zhao, Y. Chen, W. Wu, and J. Yang, "Short-term passenger flow forecast of urban rail transit based on GPR and KRR," IET Intelligent Transport Systems, vol. 13, pp. 1374-1382, 2019.
[27] J. Zhang, F. Chen, Z. Cui, Y. Guo, and Y. Zhu, "Deep learning architecture for short-term passenger flow forecasting in urban rail transit," IEEE Transactions on Intelligent Transportation Systems, vol. 22, pp. 7004-7014, 2021.
[28] T. Yang, P. Zhao and X. Yao, "A method to estimate URT passenger spatial-temporal trajectory with smart card data and train schedules," Sustainability, vol. 12, pp. 2574, 2020.
[29] L. Sun, D. Lee, A. Erath, and X. Huang, "Using smart card data to extract passenger's spatio-temporal density and train's trajectory of MRT system," in Proceedings of the ACM SIGKDD international workshop on urban computing, 2012, pp. 142-148.
[30] H. Qu, X. Xu and S. Chien, "Estimating wait time and passenger load in a saturated metro network: A data-driven approach," Journal of Advanced Transportation, vol. 2020, pp. 4271871, 2020.
[31] A. M. Wahaballa, F. Kurauchi, J. Schmöcker, and T. Iwamoto, "Estimation of transfer time distribution parameters with automatic fare collection data: Stochastic frontier model," Journal of Transportation Engineering, Part A: Systems, vol. 147, pp. 04021031, 2021.
[32] A. M. Wahaballa, F. Kurauchi, T. Yamamoto, and J. Schmöcker, "Estimation of platform waiting time distribution considering service reliability based on smart card data and performance reports," Transportation Research Record, vol. 2652, pp. 30-38, 2017.
[33] Y. Zhang and E. Yao, "Splitting travel time based on AFC data: estimating walking, waiting, transfer, and in-vehicle travel times in metro system," Discrete Dynamics in Nature and Society, vol. 2015, pp. 539756, 2015.
[34] Z. Xu, J. Xie, X. Liu, and Y. M. Nie, "Hyperpath-based algorithms for the transit equilibrium assignment problem," Transportation Research Part E: Logistics and Transportation Review, vol. 143, pp. 102102, 2020.
[35] P. Shang, R. Li, J. Guo, K. Xian, and X. Zhou, "Integrating Lagrangian and Eulerian observations for passenger flow state estimation in an urban rail transit network: A space-time-state hyper network-based assignment approach," Transportation Research Part B: Methodological, vol. 121, pp. 135-167, 2019.
[36] Y. Zhu, H. N. Koutsopoulos and N. H. Wilson, "A probabilistic passenger-to-train assignment model based on automated data," Transportation Research Part B: Methodological, vol. 104, pp. 522-542, 2017.
[37] R. Singh, D. Hörcher, D. J. Graham, and R. J. Anderson, "Decomposing journey times on urban metro systems via semiparametric mixed methods," Transportation Research Part C: Emerging Technologies, vol. 114, pp. 140-163, 2020.
[38] X. Tian, B. Zheng, Y. Wang, H. Huang, and C. Hung, "TRIPDECODER: Study Travel Time Attributes and Route Preferences of Metro Systems from Smart Card Data," ACM/IMS Transactions on Data Science, vol. 2, pp. 1-21, 2021.
[39] Y. Sun and R. Xu, "Rail transit travel time reliability and estimation of passenger route choice behavior: Analysis using automatic fare collection data," Transportation Research Record, vol. 2275, pp. 58-67, 2012.
[40] J. Sun, J. Guo, X. Wu, Q. Zhu, D. Wu, K. Xian, and X. Zhou, "Analyzing the impact of traffic congestion mitigation: From an explainable neural network learning framework to marginal effect analyses," Sensors, vol. 19, pp. 2254, 2019.
[41] T. Kim, X. Zhou and R. M. Pendyala, "Computational graph-based framework for integrating econometric models and machine learning algorithms in emerging data-driven analytical environments," Transportmetrica A: Transport Science, vol 17, pp. 1-35, 2021
[42] W. Li, L. Weili, D. Li, L. Dong, X. Zhang, Z. Xiaochen, J. Cao, and C. Junci, "Status and research progress of the linear rail transit system in China," Transportation Systems and Technology, vol. 2, pp. 16-41, 2016.
[43] J. Langford, A. J. Smola and M. Zinkevich, "Slow learners are fast," in Proceedings of the 22nd International Conference on Neural Information Processing Systems, 2009, pp. 2331-2339.
[44] C. L. Lawson and R. J. Hanson, Solving least squares problems: SIAM, 1995.
[45] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," Mathematical Programming Computation, vol. 12, pp. 637-672, 2020.
[46] B. O Donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," Journal of Optimization Theory and Applications, vol. 169, pp. 1042-1068, 2016.
[47] S. Krygsman, M. Dijst and T. Arentze, "Multimodal public transport: an analysis of travel time elements and the interconnectivity ratio," Transport Policy, vol. 11, pp. 265-275, 2004.
[48] M. Bagherian, O. Cats, N. van Oort, and M. Hickman, "Measuring passenger travel time reliability using smart card data," in Tristan ix: Triennial symposium on transportation analysis, oranjestad, aruba, 2016.
[49] P. Kumar, A. Khani and Q. He, "A robust method for estimating transit passenger trajectories using automated data," Transportation Research Part C: Emerging Technologies, vol. 95, pp. 731-747, 2018.

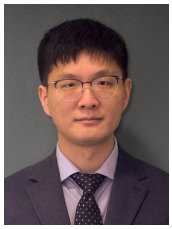
Jinlei Zhang was born in Hebei Province, China. He received a Ph.D. degree from Beijing Jiaotong University, China. He is currently an assistant professor at Beijing Jiaotong University. His research interests include machine learning, deep learning, traffic data mining and applications, and dynamic traffic modeling and management.

Feng Chen received his B.S. degree, M.A. degree, and a Ph.D. degree in railway engineering from Beijing Jiaotong University, Beijing, China, in 1983, 1990, and 2007, respectively. He is currently working as a professor at Beijing Jiaotong University and is the president of the China University of Petroleum. His research interests include passenger flow management, traffic data mining, and application for urban rail transit. Professor Chen won the first prize of national science

and technology progress as the second accomplisher in 2017.

Lixing Yang received the B.S. and M.S. degrees from the Department of Mathematics, Hebei University, Baoding, China, in 1999 and 2002, respectively, and the Ph.D. degree from the Department of Mathematical Sciences, Tsinghua University, Beijing, China, in 2005.

Since 2005, he has been with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, where he is currently a Professor. He is the author or co-author of more than 80 papers published in national conferences, international conferences, and premier journals. His current research interests include stochastic programming, fuzzy programming, intelligent systems, and applications in transportation problems and rail traffic control systems.



Wei Ma received bachelor's degrees in Civil Engineering and Mathematics from Tsinghua University, China, master's degrees in Machine Learning and Civil and Environmental Engineering, and a Ph.D. degree in Civil and Environmental Engineering from Carnegie Mellon University, USA. He is currently an assistant professor with the Department of Civil and Environmental Engineering at the Hong Kong Polytechnic University (PolyU). His research focuses on the intersection of machine learning, data mining, and transportation network modeling, with applications for smart and sustainable mobility systems.

Guangyin Jin is a Ph.D. candidate at the College of Systems Engineering of the National University of Defense Technology.

His research interest falls in the area of spatial-temporal data mining, urban computing, and intelligent transportation.

Ziyou Gao received the Ph.D. degree from the Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China, in 1994.

He is currently a Professor with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University. He also serves as the President of the Society of Management Science and Engineering of China, the Vice-President of the Systems Engineering Society of China, and the Vice-President of the Chinese Society of Optimization, Overall Planning and Economic Mathematics. In 2003, he was elected as a foreign member of the Russian Academy of Natural Sciences. His research interests include urban traffic management, management and optimization of urban rail transit, complexity of transportation network, and transport modeling and planning.