Multi-task Weakly Supervised Learning for Origin–Destination Travel Time Estimation

Hongjun Wang, Zhiwen Zhang, Zipei Fan, Jiyuan Chen,
Lingyu Zhang, Ryosuke Shibasaki and Xuan Song

Abstract—Travel time estimation from GPS trips is of great importance to order duration, ridesharing, taxi dispatching, etc. However, the dense trajectory is not always available due to the limitation of data privacy and acquisition, while the origin-destination (OD) type of data, such as NYC taxi data, NYC bike data, and Capital Bikeshare data, is more accessible. To address this issue, this paper starts to estimate the OD trips travel time combined with the road network. Subsequently, a Multi-task Weakly Supervised Learning Framework for Travel Time Estimation (MWSL-TTE) has been proposed to infer transition probability between roads segments, and the travel time on road segments and intersection simultaneously. Technically, given an OD pair, the transition probability intends to recover the most possible route. And then, the output of travel time is equal to the summation of all segments' and intersections' travel time in this route. A novel route recovery function has been proposed to iteratively maximize the current routes' co-occurrence probability, and minimize the discrepancy between routes' probability distribution and the inverse distribution of routes' estimation loss. Moreover, the expected log-likelihood function based on a weakly-supervised framework has been deployed in optimizing the travel time from road segments and intersections concurrently. We conduct experiments on a wide range of real-world taxi datasets in Xi'an and Chengdu and demonstrate our method's effectiveness on route recovery and travel time estimation.

Index Terms—Travel Time Estimation, Urban Computing, Weakly Supervised Learning

✦

1 INTRODUCTION

With the emergence of newly-developed applications, estimating travel time has become one of the hottest topics, which is of great importance to route planning, taxi dispatching, and ride-sharing in recent years. In the early phase, the data of real traffic state is mainly collected from loop sensors, which can only provide the individual travel time in a certain road segment and usually face the sparse issue. Recently, an alternative solution is to use floating car data. The floating cars equipped with GPS receivers, including taxis, buses, private cars, and online ride-hailing, record time stamps, longitude, latitude, speed, and other information at regular intervals, which can reflect the vehicle's operation status.

As a result, a good deal of travel time estimation techniques based on floating car data have been proposed in different scenes, such as dense trajectory [1], [2], [3], low-

- Hongjun Wang, Jiyuan Chen and Xuan Song are with (1) SUSTech-UTokyo Joint Research Center on Super Smart City, Department of Computer Science and Engineering (2) Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology (SUSTech), Shenzhen, China. E-mail: wanghj2020,11811810@mail.sustech.edu.cn and songx@sustech.edu.cn.

- Zipei Fan, Ryosuke Shibasaki and Zhiwen Zhang are The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa-shi, Chiba, 277-8561, Japan; emails: zhangzhiwen@csis.u-tokyo.ac.jp, fanzipei@iis.u-tokyo.ac.jp, and shiba@skl.iis.u-tokyo.ac.jp

- Lingyu Zhang is Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology (SUSTech), Shenzhen, China. emails: zhanglingyu@didiglobal.com

- Corresponding to Zipei Fan, Xuan Song;
- Hongjun Wang, Zhiwen Zhang equal contribution;

sampling-rate trajectory [4], [5], [6]. However, due to the privacy concern and data acquisition problems, extensive works focus on inferring the travel time from OD data, which only gives the origin-destination location, such as finding nearby neighbors [7], such as distance-based [8] or representation-based [9] neighbors. In general, the OD type of data is more available than the dense type, and multiple sources of OD data have been released, for example, the NYC taxi data[1], NYC bike data[2] and Capital Bikeshare Data[3]. However, as far as we know, previous literature omits the factor of the road network, which often leads to a high estimating error. Since the total travel time of the trajectory is equal to the sum of the travel time of all road segments and intersections (e.g., waiting traffic signal). Each traffic condition in road segment change will affect the total travel time. With the road network introduced, here we face three intractable problems:

1) *How to recover the route when only OD pairs are given.*
2) *How to effectively estimate the travel time when the route has been obtained.*
3) *How to learn features from complex road network.*

At first glance, given a pair of origin-destination, the shortest path algorithm (e.g., Dijkstra's algorithm) is a natural choice for problem 1) because people usually choose paths that are similar to the shortest path with less number of turns. However, the shortest path in the geometry aspect may not always match the definition of the 'shortest path' in the driver's route choice. For example, some resident or

---

1. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
2. https://data.cityofnewyork.us/Transportation/Bike-Data/374u-5ie7
3. https://www.capitalbikeshare.com/system-data

tertiary types of road are shorter than primary and trunk types of road, but they are more vulnerable to congestion, since the complex traffic state (many pedestrians), or narrowness of road width. How to encode the road features into the road search procedure? One way can be done by learning the transition probability between road segments and inferring the route via the search for the maximum route probability, where the superiority of this approach is that the character of the road will be considered in every search process. Inspired by [10], this article employs the graph neural network (GCN) to learn the features, such as road type, road length, road sign, and road lanes, of each road segment. Consequently, the problem of ignoring natural road networks in the shortest-path algorithm can be alleviated. Fig. 1 shows an example of searching candidate path through transition probability. Based on the Markov assumption, the routing probability can be obtained by multiplying the probabilities and equal to the sum of the log probabilities. The candidate paths r_1, r_2, r_3 are acquired by Depth First Searching (DFS) algorithm with pruning operation.

As we mentioned, this paper infers the overall travel time of a given route by summing up the travel time of all the road segments and intersections on that route. This raises the question of how to estimate the travel time of road segments and intersections reasonably. One concern is how to model the differences in individual driving behaviors, since given a specific OD pair, the travel time in the same time interval is varied. To address this problem, we here model the travel time with uncertainty, which means that each road/intersection travel time follows certain distribution (e.g., Lognormal). Those roads/intersections tend to be provided a large variance σ, for example, with large crowd flow. In conclusion, the uncertain travel time of route generated by route searching has been obtained. To effectively optimize the distribution, this paper formulates it as the inexact supervision learning problem [11]. One of the most well-known examples of inexact supervision is the drug activity prediction problem [12], which predicts if a molecule induces a given effect. Inexact supervision deals with training data arranged in sets called *bags*, and the labels are only annotated on *bags*. For modeling the uncertainty, given a pair of origin and destination, we consider the travel time label annotated on the unobserved routes (*bags*) and use the normal distribution [13], [14], [15] to model the uncertainty for each road segment and the interaction travel time in *bags*. We drive the objective function (Eq. (2)) based on the assumption of aggregated observation and Markov chain, and solve it with a general inexact learning probabilistic framework [16] and an iterative route recovery algorithm.

After solution 1) and 2) has been discussed, we finally introduce how to learn the meaningful features from a complicated road network, since multiple factors will affect the traffic condition, such as road types, road lanes, speed limitations, and traffic signals. To learn the intricate relation of road networks, many existing works modeling the problem of estimating travel time from either road segments [1], [3] or intersections [17], but do not assemble those features simultaneously. However, we argue that this will cause error accumulation with the road segments increasing. To fill this
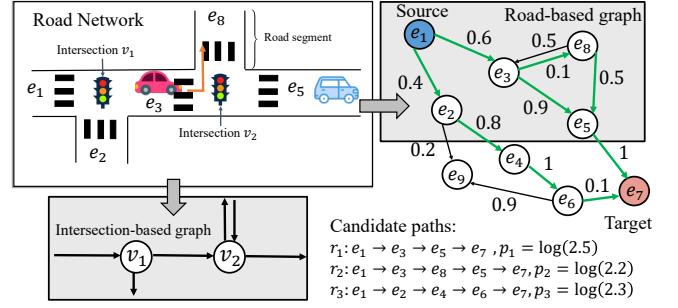

Fig. 1. The motivation in this paper is illustrated above. Given any OD pair, we want to recover the path by searching the maximum route probability learned by the model, and construct a dual graph from road-based and intersection-based aspects to estimate the travel time of road segment and intersection respectively.

gap, in this work, we construct a dual graph comprised of the road-based graph and the intersection-based graph, and estimate the travel time by summing up all road segments and intersections on one route. In the meanwhile, we also take the connected relation[4] into consideration from one road segment to one road segment, such as *primary* → *secondary*, or *trunk* → *residential*, and one intersection to one intersection, such as *tertiary* and *residential*. We introduce the Relational Graph Convolutional Networks (R-GCNs) [18] to learn the complex connected relation of the road network. Moreover, to solve the problem of losing local patterns when expanding the receptive neighborhood in GCN, we combine the Relational Graph Convolutional Networks, and gated recurrent unit (GRU) [19] together as the stacked architecture to capture both global and local features [20]. Fig. 1 represents the procedure to construct the dual graph (gray box) of the road network and to recover the route from the candidate set that is searched from the transition probability.

The main contributions of this paper can be summarized as follows.

- We propose a multi-task framework to estimate the travel time of road segments and intersection, and transition probability simultaneously. To the best of our knowledge, it is the first attempt to recover the route and jointly model the factor of intersection and road segments in the general OD travel time estimation problem.
- For the first time, we consider the estimation of the OD travel time as the weakly supervised learning problem, since the observation of the OD travel time is annotated with a bag of unobserved routes. This paper aims to infer each road segment and the intersection travel time distribution from the aggregation observation.
- We validate the effectiveness of travel time estimation and route recovery using large-scale datasets from the real world in Chengdu and Xi'an, respectively, which significantly outperform current methods.

Here, we list the organization in this paper: Sec. 2 gives the related works, including weakly supervised learning, travel time estimation, as well as route estimation. Sec. 3 introduces the preliminary knowledge, such as the road net-

---

4. https://wiki.openstreetmap.org/wiki/Key:highway

work, the origin-destination. Sec. 4 provides the definition of our formulation, assumptions, and objective function. Sec. 5 gives the methodology of our MWSL-TTE. Sec. 6 and Sec. 7 conduct the qualitative and quantitative experiments respectively to demonstrate the superiority of MWSL-TTE. Sec. 8 gives a summary of this paper and future work.

2 RELATED WORK

In this section, we will discuss several relevant topics about weakly supervised learning, travel time estimation, as well as route estimation.

2.1 Weakly Supervised Learning

Since the general supervised learning method requires each data in the training set to be labeled, this expensive labeling consumes a lot of manpower and time. Therefore, learning under the condition of weakly supervised information has become a hot research topic in the field of machine learning in recent years [21], [22], [23]. The weakly supervised learning methods focus on addressing the low-quality labels scenarios 1) incomplete supervision [24] : only part of data can be labeled. 2) inexact supervision [12]: only have coarse-grained labels . 3) inaccurate supervision [25] : only part of the data owns true labels. The task of estimating travel time from OD can be considered as the inexact supervision belonging to the category of Weakly Supervised Learning, where the only observations are the total travel time and OD locations, but the accurate routes are unknown. Different from traditional approaches [7], [26] searching similar historical trajectories from data but ignoring the city road network structure, in this paper, we aim to infer the potential route from OD by using transition probability between road segments, where a set of potential routes can be seen as a *bag*, and the OD travel time can be obtained by summing up the estimated times of the road segments in *bag*. To the best of our knowledge, we are the first to introduce the problem of travel time estimation into the inexact supervision framework.

2.2 Travel Time Estimation

Various TTE implementations were classified into three groups, traditional approaches, deep learning-based approaches and graph neural network-based approaches. Traditional approaches for TTE include the road-segment-based and path-based methods. The road segment-based methods [27], [28] coarsely forecast the route travel time by summing up all estimated times of roads by using the data collected from sensors like magnetometer detectors or highway cameras, which omitted the necessary factor of intersection and relationship among road segments. And the path-based methods address the above challenges mainly by nearest neighbors search [7], [26] and trajectory regression [28], [29], [30]. Nearest neighbors search (NNS) finds nearby historical trajectories according to the assumption that the routes with similar origins and destinations own close travel time. Trajectory regression methods predict the whole route travel time based on the given historical trajectories.

Recently, deep learning-based approaches have become especially important in the task of TTE. These approaches can be divided into two groups, classical deep learning-based methods and graph deep learning-based methods. Some classical methods such as deep neural networks (DNNs) [1] and convolutional neural networks (CNNs) [2], [31] have been successfully applied in TTE. For example, Deep-TTE [1] proposed a CNN-based framework to integrate various types of attribute information (such as weather, time ID and driver ID) for TTE. However, most of these methods model the road network as a grid-based map, but they ignore the graphical structure of real-world road network.

To fully utilize spatial information, GNN is an emerging tool to analyze the topological relations of graph-structured traffic data. Especially, Spatial-Temporal Graph Neural Network (STGNN) [32] is a framework that integrates GNN and temporal processing modules, which can handle spatial relations and temporal trends simultaneously. Due to the spatio-temporal characteristics of the real-world road network, STGNN are widely adopted in TTE. For example, diffusion convolutional recurrent neural network (DCRNN) [33] modeled the graph-structured traffic data as a diffusion process on a directed graph and transformed spatio-temporal features into a seq2seq framework. ASTGNN [34] proposes a trend-aware multi-head attention mechanism to capture multiple potential correlations in traffic forecasting. However, these works only consider the spatio-temporal attributes of road segments but ignore the interactive correlations between intersections and road segments. Meanwhile, both the real route of OD pairs and road condition also have an important influence on TTE.

2.3 Route Estimation

Another bunch of research in travel time estimation is to solve the issue of sparse trajectory due to the privacy, business competition [4], [5], [6], and limitation of GPS devices, or in the scene of ETC [35], [36] and surveillance cameras [37]. A common strategy for solving the sparse trajectory is to infer the potential route based on the information of the road network. Reference [4] applies the inverse reinforcement learning to learn the latent cost (reward) of a road through historical data, and proposed Exact Route Search approach to find the maximum probabilistic route based on dynamic programming. However, route search-based algorithms are only adapted in low-sampling rate trajectories, but not OD problem, due to the heavy computational cost. Because of the large distance between a pair of toll stations or surveillance cameras, a frequently used path inferring algorithm is based on the Depth First Searching algorithm to find all possible *simple paths* that the one road segment can only appear at most once. However, those methods omitting the real traffic condition tend to generate the unreal route in the path inferring procedure. To this end, in this paper, we combine the transition probability and route search approach together to find the optimal route based on the real travel time and road network structure.

3 PRELIMINARIES

We start with giving the definition about the road network, Origin-Destination, simple path as well as route.
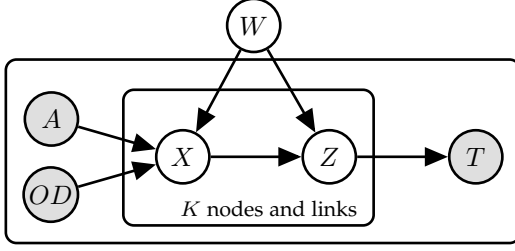
Fig. 2. The graphical model of the data generating process.

Definition 3.1. *Road Network. A road network is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \pi)$, where $\mathcal{V}$ denotes the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of directed edges, and π is a node's feature set. This paper uses $\mathcal{G}_v$ and $\mathcal{G}_e$ to denote the node-wise and link-wise graphs respectively. For the π in $\mathcal{G}_v$, the features of $\mathcal{V}$ can be such as, junction types, and traffic signals. For the π in $\mathcal{G}_e$, the features of v can be such as, road types, road lanes, and speed limits.*

Definition 3.2. *Origin-Destination (OD). In this paper, a OD pair represents a tuple with $\{e_i, e_j, t^{th}\}$, where e_i and e_j denote the start and end road segment, respectively, and t^{th} is the start time interval of a day (e.g., 8:00am-9:00am). Note that we assume the traffic conditions for all road segments and intersections are invariable within the same time slot.*

Definition 3.3. *Simple Path. A simple path t_r can be presented as a series of time-ordered links. We have $t_r : e_1 \to e_2 \to \cdots \to e_{|t_r|}$, where each link satisfied $e_i \neq e_j$.*

Definition 3.4. *Route. A route r in this paper is a sequence alternating with links and intersections. We have $r : e_1 \to v_2 \to e_3 \to \cdots e_{K-1} \to v_K$, where v_i is the intersection of edges pair (e_{i-1}, e_i), and K is the length of r.*

4 PROBLEM FORMULATION

To overcome the previous issue with ignoring the road network in the OD-TTE problem, we here intend to give a formulation under the weakly supervised learning. This solution is motivated by the advance in weakly supervised learning and GCN path inference.

Given a pair of origin and destination, estimating the travel time is to infer the total time cost. Traditionally, the formulation of TTE can be divided into two parts: 1) inferring the future traffic through historical state [3], [38], 2) online infer traffic through real-time trajectories [39]. The previous one is mainly related to the robust traffic state, which is calculated on either dense trajectories [3] or loop detectors [32]. Another one intends to resolve the sparse issue by estimating (imputation) citywide-level traffic state from fewer real-time trajectories. Since this paper is under the OD scenario and hard to give a valid historical traffic state, we here follow the online TTE formulation. Formally, this paper follows the assumption that the traffic state at one road segment in a specific time interval $\Delta t = 60mins$, e.g., 10:00 am-11:00 am, under the same distribution, such as, Gaussian distribution $\mathcal{N}(\mu, \sigma)$ with $\mu = 60s$ and $\sigma = 1$. Subsequently, suppose the current time is $\mathbf{t}$, we train the real-time OD pairs in time slot $[\mathbf{t}, \mathbf{t} + \Delta \mathbf{t}_1)$, the online traffic state will be completed and evaluated with the OD pairs in $[\mathbf{t} + \Delta \mathbf{t}_1, \mathbf{t} + \Delta \mathbf{t})$.

TABLE 1
Partial Symbols Description.

Notation	Description
A	Links inner transition probability matrix
Q	Aggregate function
Ω_{OD}	Set of candidate paths from O to D
f	Multi-task function
X	Features of links and nodes
Z	Unobservable travel time of links and nodes
$\mathcal{G}_v$ and $\mathcal{G}_e$	Node-wise and link-wise graph respectively
T and $\tilde{T}$	Ground truth and estimated of travel time
$\mathcal{T}$	Time embedding vector
W	Learnable parameters

About the training procedure of MWSL-TTE, Fig. 2 illustrates the data generating process with graphical representation. OD are the features of origin-destination locations. $X_{1:K}$[5] stands for the features vectors of unobserved K nodes and links in route $r : e_1 \to v_2 \to e_3 \to \cdots e_{K-1} \to v_K$, and $Z_{1:K}$ is the unobservable *travel time* of nodes and links. We assume that r is conditioned on OD and transition matrix A, where $A_{i,j}$ indicates the transition probability from e_i to e_j. Therefore, we have $p(r|A; OD) = p(X_{1:K}|A; OD)$, where A is generated by multi-task function f with $A = f(W_A; \mathcal{G}_e)$, $\mathcal{G}_e$ is the link graph, and W is the learnable parameters. Meanwhile, Z also under a parametric distribution $p(Z_{1:K}|\theta_z = f(X_{1:K}; W_Z;)) = p(Z_{1:K}|(\mu, \sigma) = f(X_{1:K}; W_Z;))$ on the factors of $X_{1:K}$ and W_Z, where μ and σ are the mean and variance of Gaussian distribution. For the relation between $Z_{1:K}$ and aggregate observation of travel time T, we have the following definition:

Definition 4.1. *Given a route $r_i : e_1 \to v_1 \to e_2 \to \cdots v_{|r_i|-1} \to e_{|r_i|}$ under a pair of OD, and it's unobserved travel time $Z_{1:K} = \{\tilde{t}_{e_1}, \tilde{t}_{v_2}, \tilde{t}_{e_3}, \cdots \tilde{t}_{v_{K-1}}, \tilde{t}_{e_K}\}$. The aggregate functions Q can be defined as*

$$Q(Z) = \tilde{t}_{e_1} + \tilde{t}_{v_1} + \tilde{t}_{e_2} \cdots + \tilde{t}_{v_{K-1}} + \tilde{t}_K, \quad (1)$$

where an aggregate function $Q : Z \to T$ is a mapping function from unobserved variable Z to observation T. Since we assume Z follows a Gaussian distribution, we can written $T = Q(Z) = \sum_i \tilde{t}_i = \sum_i \mu_i$ according to the nature of additivity: $X + Y \sim N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$, where $X \sim N(\mu_1, \sigma_1^2) Y \sim N(\mu_2, \sigma_2^2)$.

Subsequently, we summarize our assumptions below.
Assumption 1(Aggregate observation assumption) $p(T \mid X_{1:K}, Z_{1:K}) = p(T \mid Z_{1:K})$

We here assume that the observation T is conditionally independent $X_{1:K}$ when given $Z_{1:K}$ (Def. 4.1). Intuitively, given Z, in fact, the travel time T can be obtained by summing up all components in Z.
Assumption 2 (Markov chain assumption) $p(Z_{1:K} \mid X_{1:K}) = p(Z_1 \mid X_1) \prod_{i=2}^{K} p(Z_{i+1} \mid X_i; Z_i)$

We assume that the road travel times Z_{i+1} are mutually independent except for Z_i, which is consistently under the

5. The subscript for example $X_{1:K}$ denotes an abbreviation for the set $\{X_1, X_2, \cdots, X_K\}$

(a) Overview of Our Model

(b) Detailed Structure of Stacked R-GCNs

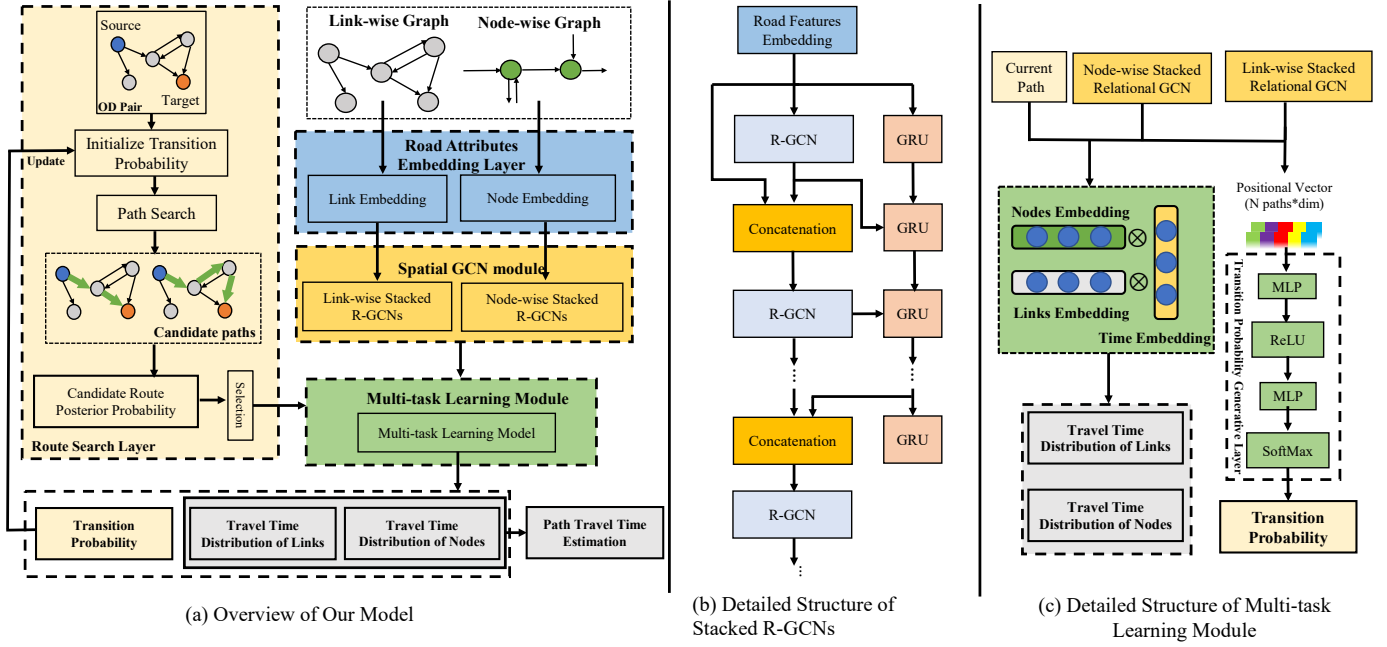(c) Detailed Structure of Multi-task Learning Module

Fig. 3. Framework design of MWSL-TTE. The overview of our proposed model is depicted in (a), which includes the route search layer, road attributes embedding layer, spatial GCN module, and multi-task learning module. (b) is the architecture of the stacked R-GCNs layer, which attempt to capture both global and local relation by modeling the GRU and R-GCNs together. (c) is our multi-task learning layer, which will estimate the travel time of links, intersections, and transition probability simultaneously. The transition probability will be updated when the multi-task learning layer is output, and then, the path search algorithm will work to find the candidate paths.

assumption of Markov chain and extensive applications in trajectory data mining [40], [41]. Furthermore, since T can be determined by function Q, the conditional probability for $p(T \mid Z_{1:K}) = \delta_{Q(Z_{1:K})}(T)$, where $\delta(\cdot)$ represents the *Dirac delta function*.

To sum up, the objective function in this paper can be defined as

$$
\begin{aligned}
\max_{X_{1:K}} \ & p\left(T; X_{1:K} \mid A; OD\right) \\
=& p\left(T \mid X_{1:K}\right) p(X_{1:K} \mid A; OD) \\
=& \int_{Z^K} p\left(T; Z_{1:K} \mid X_{1:K}\right) d_{Z_{1:K}} \, p(X_{1:K} \mid A; OD) \\
=& \int_{Z^K} \delta_{Q(z_{1:K})}(T) p(Z_{1:K} \mid X_{1:K}) d_{Z_{1:K}} \, p(X_{1:K} \mid A; OD) \\
=& \mathop{\mathbb{E}}_{Z_i \sim p(Z_i \mid X_{i-1}; Z_{i-1})} \left[\delta_{Q(Z_{1:K})}(T)\right] p(X_{1:K} \mid A; OD)
\end{aligned}
\tag{2}
$$

Therefore, according to Eq. (2), our training procedure could be split into two stages: 1) maximizes the posterior probability $p(X_{1:K} \mid A; OD)$. 2) maximizes the conditional probability $p(Z_{1:K} \mid X_{1:K})$ to estimate each road segments and intersection travel time, which can be optimized by *expected log-likelihood* [16]:

$$
\ell(W) = \mathbb{E}\left[\log p\left(T \mid X_{1:K}; W\right)\right]
\tag{3}
$$

For ease of reference, some important notations are summarized in Table 1.

We here give a brief summarization of our problem formulation. In this paper, we aim to solve the two challenges in OD travel time estimation, which are uncertain routes and uncertain travel time. We intend to infer the potential route r between source and destination by transition probability
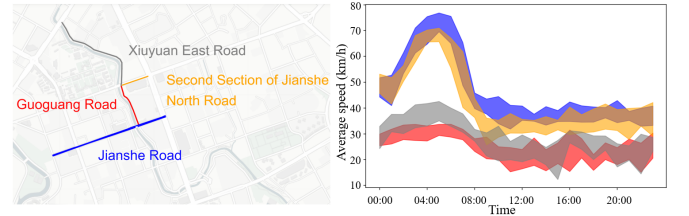

Fig. 4. In the speed distributions of neighbor road segments during the 7 days of National Day, we observe that the speed distribution is highly consistent with the connected type. The color indicate the road correspondingly.

A. Subsequently, we optimize the travel time distribution $Z_{1:K} = \{\tilde{t}_{e_1}, \tilde{t}_{v_2}, \tilde{t}_{e_3}, \cdots \tilde{t}_{v_{K-1}}, \tilde{t}_{e_K}\}$ at r via weakly supervised learning (Eq. (3)). We assume $\tilde{t}$ under Gaussian distribution $\mathcal{N}(\mu, \sigma)$. Therefore, transition probability A and parameter at Gaussian distribution can be generated by $(A, \mu, \sigma) = f(W; \mathcal{G}_v; \mathcal{G}_e)$, where μ, σ are the mean and variance in Gaussian distribution, respectively.

5 METHODOLOGY

In this section, the MWSL-TTE will be detailedly introduced. Specifically, the overview of MWSL-TTE has been depicted in Fig. 3 (a) including with four components included road attributes embedding layer, spatial GCN module, route search layer, and multi-task learning module. Fig. 3 (b) shows the inner structure of the stacked R-GCNs layer, and Fig. 3 (c) represents the multi-task learning module.

5.1 Road Attributes Embedding layer

Let each latent variable $\tilde{t}_i \in Z$ belong to Gaussian distribution $\mathcal{N}(\mu_i, \sigma_i)$, which is a common assumption and widely be used in modeling the travel time distribution [13], [14], [15]. Given a pair of OD, we have the conditional probability $p(Z_{1:K}|\theta_z = f(X_{1:K}; W_Z)) = p(Z_{1:K}|(\mu, \sigma) = f(X_{1:K}; W_Z))$. Therefore, one of the tasks for neural network f is to estimate the distribution parameters μ, σ for each road segment and intersection. Since the road features $X_{1:K}$ could affect the travel time estimation Z, we consider the follow statistical spatial factors as the important matters for road segments:

- *Road types*: e.g., primary, primary link, secondary, secondary link, tertiary, residential, service road, etc.;
- *Number of lanes*: how many traffic lines in the road;
- *Otherwise features*: e.g., road length, whether it is a one way or not, limiting velocity, unique ID;

and intersections, such as

- *Node tags*: e.g., speed camera, traffic signal, crossing sign, turn circle, stop sign;
- *Node street count*: e.g., T-juction X-junction, and 5-way junction:
- *Otherwise features*: e.g., unique ID, GPS coordinate.

To obtain the feature representations of both links and nodes, we use the embedding method [42] to transform each categorical attribute into a low-dimensional feature vector by multiplying the spatial feature embedding matrices $E \in R^{n^{(s)} \times d^{(s)}}$. Here, $n^{(s)}$ represents the number of possible values of the categorical features, and $d^{(s)}$ represents the embedding dimension. This allows us to share efficient information among different road segments or intersections, so that rarely traveled segments could be learned from those frequently traveled with similar semantic meaning. Besides the categorical road attributes, we concatenate the obtained embedded feature vectors together with other road attributes (e.g., road length and GPS coordinate). Based on the above feature representations of the dual graph (link-wise and node-wise), we can obtain the corresponding input for the subsequent spatial GCN module.

5.2 Spatial GCN Module

After the embedding characteristics of the road attributes have been obtained, we next introduce the spatial GCN module serving as modeling the complex spatial relations from the dual graph. The motivation for introducing R-GCNs in the travel-time estimation problem has been represented in Fig. 4. We can observe that the travel speed is highly similar with the connected types. For specifically, even though the Second Section of JianShe North road is the neighbor of Xiuyuan East and Guoguang road, its speed distributions are more related to JianShe road, where their road types are the same. Therefore, we are concerned that the features of *road types*, the connected types, for example, *resident → secondary* (link level), and *secondary* (node level), are also important. To this end, here we introduce the Re-

lational Graph Convolutional Networks [18] in our model, which can be defined as

$$\boldsymbol{h}_i^{(l+1)} = \sigma \left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(l)} \boldsymbol{h}_j^{(l)} + W_0^{(l)} \boldsymbol{h}_i^{(l)} \right), \quad (4)$$

where $\boldsymbol{h}_i^{(l)} \in \mathbb{R}^{d^{(l)}}$ is the hidden state of road r_i in the lth layer of model with dimensionality $d^{(l)}$. $\mathcal{N}_i^r$ denotes the set of neighboring road segment/intersection indices under the relation $\boldsymbol{r}$. $W_r^{(l)}, W_0^{(l)}$ are the learnable parameters, $c_{i,r}$ is the normalization constant, and $\sigma(\cdot)$ is the activation function. In this paper, we set $c_{i,r} = |\mathcal{N}_i^r|$.

Next, we will introduce the stacking operation based on R-GCNs. The stacking operation has recently been demonstrated to prevent local information loss [20], [43]. Thus, we model the temporal trends in the stacked GCN architecture combined with the spatial feature representations of both nodes and links. In this paper, we use a gated recurrent unit (GRU) [19] as a temporal processing module to incrementally concatenate multi-scale features, which can be written as

$$
\begin{aligned}
\boldsymbol{c}^{(0)} &= \text{GRU}\left(\boldsymbol{h}^{(0)}, \boldsymbol{c}^{(-1)}\right), \\
\boldsymbol{h}^{(1)} &= \sigma \left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(0)} \boldsymbol{h}_j^{(l)} + W_0^{(0)} \boldsymbol{h}^{(0)} \right), \\
\boldsymbol{h}^{(2)} &= \sigma \left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(1)} \left[\boldsymbol{h}_j^{(0)}, \boldsymbol{h}_j^{(1)} \right] + W_0^{(1)} \left[\boldsymbol{h}^{(0)}, \boldsymbol{h}^{(1)} \right] \right), \\
\boldsymbol{h}^{(l+1)} &= \sigma \left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(l)} \left[\boldsymbol{c}_j^{(l-1)}, \boldsymbol{h}_j^{(l)} \right] + W_0^{(l)} \left[\boldsymbol{c}^{(l-1)}, \boldsymbol{h}^{(l)} \right] \right), \\
l &= 2, 3, \dots, n-1,
\end{aligned}
\tag{5}
$$

where $\boldsymbol{c}^{(l)}$ is the hidden state of the output of GRU, and $\boldsymbol{c}^{(-1)}$ is initialized with 0. $\boldsymbol{h}^{(l)}$ is the latent state of the link and node at l^{th} hop. The detailed architecture of the stacked R-GCNs has been shown in Fig. 3 (b), and the formula of GRU can be expressed as

$$
\begin{aligned}
o &= \sigma \left(\boldsymbol{W}_{o_1} \boldsymbol{h}(t) + \boldsymbol{O}_{o_1} \boldsymbol{c}(t-1) + \boldsymbol{b}_{o_1} \right), \\
q &= \sigma \left(\boldsymbol{W}_{q_1} \boldsymbol{h}(t) + \boldsymbol{O}_{q_1} \boldsymbol{c}(t-1) + \boldsymbol{b}_{q_1} \right), \\
\boldsymbol{c}'(t) &= \tanh \left(\boldsymbol{W}_{h_1} \boldsymbol{h}(t) + \boldsymbol{O}_{h_1}(q \odot \boldsymbol{c}(t-1)) + \boldsymbol{b}_{h_1} \right), \\
\boldsymbol{c}(t) &= o \odot \boldsymbol{h}(t-1) + (1-o) \odot \boldsymbol{c}'(t)
\end{aligned}
\tag{6}
$$

where $\boldsymbol{W}_{o_1}, \boldsymbol{O}_{o_1}, \boldsymbol{W}_{q_1}, \boldsymbol{O}_{q_1}, \boldsymbol{W}_{h_1}, \boldsymbol{O}_{h_1}$ are the learnable parameters, and $\boldsymbol{b}_{o_1}, \boldsymbol{b}_{q_1}, \boldsymbol{b}_{h_1}$ are biases.

5.3 Multi-task Learning Module

In this section, we will introduce the productions of MWSL-TTE and the route recovery algorithm together.

5.3.1 Generating nodes and links travel time

As we mentioned in Sec. 4, the task of TTE can be formulated as given the real-time OD pairs and the corresponding observation T in $[\mathbf{t}, \mathbf{t} + \Delta \mathbf{t}_1)$, we aim to complete the travel time for all links and nodes, and evaluate them using the OD pairs in $[\mathbf{t} + \Delta \mathbf{t}_1, \mathbf{t} + \Delta \mathbf{t})$. To address the data sparse issue, this paper formulates it as the problem of tensor completion [44] by tensor decomposition technique, a popular method

for traffic missing value imputation method. Since urban travel time has typical temporal and spatial distribution characteristics, it can frequently be divided into two levels: one is the modeling of road segments or intersections in space, and the other is temporal embedding in time (such as Weather ID and Holiday ID). Based on the above spatio-temporal embedding, we finally employ the 1st order CP decomposition to reconstruct the travel time distribution as

$$\mu_e = (\boldsymbol{h}_e^{(l+1)} W_{\mu_e} + b_{\mu_e}) \otimes \mathcal{T}_i, \ \sigma_e = (\boldsymbol{h}_e^{(l+1)} W_{\sigma_e} + b_{\sigma_e}) \otimes \mathcal{T}_i,$$
$$\mu_v = (\boldsymbol{h}_v^{(l+1)} W_{\mu_v} + b_{\mu_v}) \otimes \mathcal{T}_i, \ \sigma_v = (\boldsymbol{h}_v^{(l+1)} W_{\sigma_v} + b_{\sigma_v}) \otimes \mathcal{T}_i,$$

where $\mu_e, \sigma_e \in \mathbb{R}^{|\mathcal{E}| \times 1}$, and $\mu_v, \sigma_v \in \mathbb{R}^{|\mathcal{V}| \times 1}$ are the mean and variance in Gaussian distribution for link and node respectively. $W_{\mu_e}, W_{\sigma_e}, W_{\mu_v}, W_{\sigma_v} \in \mathbb{R}^{d^{(l)} \times d^{(l+1)}}$, $b_{\mu_e}, b_{\sigma_e} \in \mathbb{R}^{|\mathcal{E}|}$, and $b_{\mu_v}, b_{\sigma_v} \in \mathbb{R}^{|\mathcal{V}|}$ are the parameters in the fully connected layer (FC). $\mathcal{T} \in \mathbb{R}^{d^{(l+1)} \times I}$ is the embedding tensor of time, and $\mathcal{T}_i \in \mathbb{R}^{d^{(l+1)} \times 1}$ denotes the embedding vectors in real-time interval. We discretize the day of time into I time slots (e.g., Δt=15 minutes). According to *expected log-likelihood* in Eq. (3), since the normal distribution is closed with addition, the loss function can be derived as

$$L_{\mu,\sigma} = -\frac{(Q(Z_{1:K}) - Q(\mu_{1:K}))^2}{2 \sum_i (\sigma^2)^i} - \frac{1}{2} \log \left(2\pi \sum_i (\sigma^2)^i \right)$$
$$= -\frac{(T - Q(\mu_{1:K}))^2}{2\sigma^2} - \frac{1}{2} \log \left(2\pi\sigma^2 \right), \quad (7)$$

where Q is the aggregation function defined in Def. 4.1, and K denotes the number of samples in *bag*. In this paper, *bag* is equal to *route* (Def. 3.4) between origin-destination.

5.3.2 Transition Probability Generative Layer

We thereafter introduce the detailed structure of the transition probability generative layer to generate the link transition probability by using edges features $\boldsymbol{h}^{(l+1)}$ (Eq. (5)). Technically, for the last two layers, we use the multi-layer perceptron (MLP) to produce the weights of links

$$a_{i \to j} = \text{MLP} \left(\boldsymbol{h}_i^{(l+1)} \| \boldsymbol{h}_j^{(l+1)} \right),$$

where $\|$ is the operator of features-wise concatenation, and then apply the softmax layer over outgoing links

$$p(e_j \mid e_i) = \frac{\exp(a_{i \to j})}{\sum_{v_n \in \mathcal{V}_i} \exp(a_{i \to n})} \quad (8)$$

After that, the transition probability in Eq. (8) will be employed in the Route Search Layer. For simplicity, we use the transition matrix $A \in \mathcal{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ to represent all links' possibilities; for example, we have $A[i,j] = P(e_j \mid e_i)$.

5.3.3 Route Search Layer

As aforementioned, the shortest route algorithms omit the condition of the road in practice. To solve this problem, we intend to construct the transition probability between road segments and combine the transition probability to infer the route. However, considering the complex highway graph, there are a tremendous number of routes for any OD pair. It would be reasonable to prune the routes through some thresholds. Therefore, in this paper, we prune the route from two aspects: 1) The lengths of the simple route r from origin

o to destination d should satisfy $r.length < r_{short} + \delta_{lens}$, where r_{short} is the shortest simple route and δ_{lens} is the distance threshold. 2) the co-occurrence probability of a simple route $P(r|OD; A)$ should also meet the criteria $P(r|OD; A) > \delta_{probs}$, where δ_{probs} is the probability threshold. Next, we will introduce the definition of probability $P(r|OD; A)$. According to Assumption 2 and Eq. (8), the co-occurrence route probability

$$p(r) = P(e_1, e_2, \cdots e_K) = p(e_0) \prod_{i=1}^{K} p(e_i \mid e_{i-1}), \quad (9)$$

where e_0 is the origin location, and we have $p(e_0) = 1$.

After the strategy of pruning has been introduced, the candidate routes can be obtained by the Depth First Search (DFS) algorithm. Specifically, we generate the candidate route set Ω_{OD} via posterior probability $p(r|OD; A)$ and choose the route with maximum probability as the optimal solution. Formally, given an OD pair and observation T, the optimal route r^* can be written as

$$r^* = \arg \max_{r_i \in \Omega_{OD}} p(r_i \mid OD; A; T; Z)$$
$$= \arg \min_r |T - \sum_{e_i \in r_j} \mu_e^{(i)} - \sum_{v_i \in r_j} \mu_v^{(i)}|, \ \forall r_j \in \Omega_{OD}. \quad (10)$$

Eq. 10 selects the most suitable route r^* regarding current travel time μ.

5.3.4 Model Training

Next, we will introduce the optimizing procedure of MWSL-TTE. As we discussed in Sec. 5.3.3, the top m maximum routes have been obtained, and we chose the most satisfied one by Eq. (10). However, such a choice may fall into a local solution, and other candidate routes might never be picked. To address this problem, we here introduce the ϵ-greedy algorithm, which means that the route satisfied Eq. (10) will be chosen in $1 - \epsilon$ probability. Otherwise, randomly select the routes with top m maximum probability in ϵ probability. Moreover, we wish that the transition probability could help us infer the most possible route based on the ground truth (observation travel time). So, we here adopt the Kullback-Leibler (KL) divergence to measure the coherence, which can be written as

$$D_{\text{KL}}(\mathcal{P} \| \mathcal{Q}) = -\sum_i \mathcal{P}(i) \ln \frac{\mathcal{Q}(i)}{\mathcal{P}(i)}, \quad (11)$$

where $\mathcal{P}$ is the probability distribution of each route in the candidate set, and $\mathcal{Q}$ is the inverse estimation loss distribution between $Q(\mu)$ and ground truth. In other words, the optimization direction is towards both higher route probability and more accurate travel time estimation. For the route picked up through Eq. (10), the Negative Log Likelihood (NLL) loss has been employed to minimize the negative log likelihood function, which can be defined as

$$L_{tp} = -\sum_{i=2}^{|t_r|} \log(p(e_i \mid e_{i-1}, \theta)), \quad (12)$$

where θ is the model's trainable parameters to represent the posterior probability. By fusing all objective functions

together, our model is trained to minimize the weighted combination of three loss terms

$$\mathcal{L} = \alpha L_{\mu,\sigma} + \beta L_{tp} + (1 - \alpha - \beta)D_{\mathrm{KL}}(\mathcal{P}\|\mathcal{Q}) \qquad (13)$$

where α and β are the const parameter to balance three loss terms $L_{\mu,\sigma}, L_{tp}$ and $D_{\mathrm{KL}}(\mathcal{P}\|\mathcal{Q})$. The training pseudocode of MWSL-TTE has been depicted in Algorithm 1.

Algorithm 1: Training Procedure of MWSL-TTE

Input: OD datasets $\mathcal{D}$, node-wise graph $\mathcal{G}_v$, link-wise graph $\mathcal{G}_e$, and number of candidates routes N
Output: OD TTE estimation function f
1 **while** *not convergence* **do**
2 $\quad (\mu, \sigma, A) = f(W; \mathcal{G}_v; \mathcal{G}_e)$
3 $\quad$ **for** $OD \in \mathcal{D}$ **do**
$\quad\quad$ 1. Generating Top N candidates route set Ω_{OD}
$\quad\quad$ 2. Select the route r_i from Ω_{OD} through ϵ-greedy
$\quad\quad$ 3. Calculate the loss by Eq. (13) and update the parameters through back-propagation.
4 $\quad$ **end**
5 **end**
6 **return** f

---

6 EXPERIMENTS

In this section, various experiments will be conducted based on a wide range of public real-world taxi dataset in Xi'an, and Chengdu to evaluate the superiority of MWSL-TTE in TTE and route recovery aspects.

6.1 Datasets

Road Networks. We use two road networks: Chengdu Road Network and Xi'an Road Network. Both of them are extracted from OpenStreetMap [45], and include nine road types (trunk, trunk link, freeway link, primary, primary link, secondary, secondary link, tertiary, tertiary link). Here, Chengdu road network contains 8221 edges and 5182 nodes, which ranges from 30.63° to 30.69° in latitude and 104° to 104.07° in longitude. And Xi'an road network contains 4780 edges and 3782 nodes, ranging from 34.20° to 34.29° in latitude and 108.90° to 108.99° in longitude.

Taxi OD Orders. We use two public taxi trajectory datasets come from the Didi Express platform to generate the OD orders. Each generated order corresponds to a trip record that consists of the time-stamps and locations of an OD. Here, we implement Xi'an dataset that is from 10/10/2016 - 10/22/2016, and Chengdu Dataset is from 08/18/2014 - 08/24/2014 (a whole week from Monday to Sunday). The GPS points of both two datasets have been tied to the road and the interval of sample trajectory points is 2-4s, ensuring that the vehicle trajectory can correspond to the actual road information. Especially, we generate the ground truth route of the original vehicle trajectories for the route recovery task via a map-matching tool FMM [46].

6.2 Baseline Methods and Metrics

We first compare our models with six baseline methods for the task of OD travel time estimation:

- **TEMP**: Temporally weighted neighbors [7] is a nearest-neighbor-based approach that estimates the OD travel time by averaging the travel time of all historical trajectories falling in the same time slot with a similar origin and destination.
- **GBDT**: Gradient boosting decision tree [47] is used for the regression of OD travel time estimation.
- **STNN**: Spatio-temporal deep neural network [8] is a deep neural network-based approach that first predicts the travel distance given an OD pair, and then combines this prediction with the departure time to estimate the travel time.
- **MURAT**: Multi-task representation learning [9] is a deep neural network-based approach that jointly predicts the travel distance and the travel time for taxi orders by learning representations of road segments and the origin-destination information.
- **DCRNN**: It exploits GCN to capture spatial dependency, and then uses recurrent neural networks to model temporal dependency [33]. We implemented this model based on OD estimation prediction of road network. The hidden vector size of GCN and GRU are set as 20 and 128, respectively.
- **ConSTGAT**: This model adopts a graph attention mechanism to explore the joint relations of spatio-temporal information [48]. The parameter setting is basically same with the original model. In the integration module, we also use two-layer MLP.
- **ASTGNN**: This model consider multiple factors in traffic forecasting, such as, periodicity, spatial heterogeneity by leveraging a trend-aware multi-head attention mechanism [34]. The number of layers for both encoder and decoder is set to 3. And the kernel size of convolution is set to 5.

Moreover, for the task of OD route recovery, we compare with two representative baselines of route recovery from sparse trajectories. Both of them are based on inverse reinforcement learning to capture the spatial transition probabilities, and the difference between these two models is that the temporal components:

- **STRS**: Spatio-temporal-based route recovery system [4] seeks to recover the route from sparse trajectories. The temporal components of STRS comprise a matrix factorization-based method.
- **DeepGTT-STRS**: Li et al. [3] proposes a deep generative travel time estimation model named DeepGTT that replaces the temporal component of STRS.

Evaluation Metrics. For the OD travel time estimation of our MWSL, we evaluate the performance with RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error). Then we adopt the accuracy of route recovery as the main performance metric for the route recovery task. It is defined as the ratio of the length of a correctly inferred route to the length of the ground truth route R_G or the inferred route R_I whichever is longer, i.e., $accuracy = \frac{(R_G \cap R_I).len}{max\{R_G.len, R_I.len\}}$.

6.3 Experimental Settings

The experiments are implemented with PyTorch 1.6.0 and Python 3.6, and trained with a RTX2080 GPU. The platform ran on Ubuntu 16.04 OS. We trained the models using Adam

TABLE 2
Performance of MWSL and its variants for OD travel time estimation, compared with other baseline methods.

Models	Xi'an			Chengdu		
	RMSE (sec)	MAE (sec)	MAPE	RMSE (sec)	MAE (sec)	MAPE
TEMP	398.95	277.56	34.24%	446.98	327.99	32.00%
GBDT	365.72	250.63	31.27%	435.88	303.83	30.35%
STNN	353.06	241.19	30.43%	425.53	293.10	28.25%
MURAT	538.23	512.65	127.87%	519.20	503.36	118.62%
DCRNN	282.54	191.41	24.94%	392.45	263.91	25.95%
ConSTGAT	283.89	195.31	25.32%	403.31	280.90	28.16%
ASTGNN	259.46	179.08	23.86%	362.48	244.03	23.52%
N-Node	253.62	173.71	23.04%	367.04	236.18	23.69%
N-GRU	259.52	181.37	24.25%	371.34	240.45	24.03%
N-R-GCN	263.46	184.73	24.60%	364.58	234.37	23.46%
N-PathUpdate	257.62	178.25	23.59%	358.09	229.56	23.15%
MWSL-TTE	**238.86**	**162.37**	**21.33%**	**341.02**	**215.03**	**22.27%**

TABLE 3
Inference time cost comparison for GCN-based travel time estimation models. The time unit is second.

Models	Xi'an	Chengdu
	Time (sec)	Time (sec)
DCRNN	0.15	0.16
ConSTGAT	0.41	0.45
ASTGNN	0.34	0.37
MWSL-TTE	0.23	0.25

TABLE 4
Performance of MWSL and STRS-based baseline methods for route recovery. The column T.time refers to the training time of the model and its unit is hour.

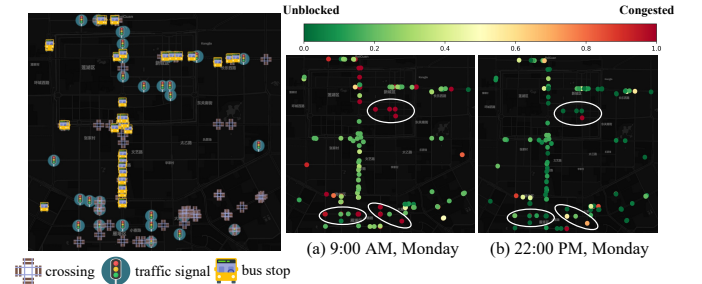| Models	Xi'an		Chengdu	
	Acc	T. time	Acc	T. time
STRS	82.71%	3.18	71.64%	4.74
DeepGTT-STRS	79.39%	3.37	68.72%	5.02
MWSL-TTE	**86.25%**	**0.52**	**77.03%**	**0.69**

Fig. 5. The time-consuming ratio for some nodes in the Xi'an road network. Here, we select three types of nodes including "crossing", "traffic signal" and "bus stop".

optimizer with an initial learning rate of 0.001 on both Chengdu and Xi'an datasets, and early stopping is used on the validation dataset. Especially, we run each experiment for three times.

The main hyper-parameter settings of our proposed method are described as follows:

- In the generation of a candidate route set $\Omega_{a,b}$, m candidate routes are selected between the OD pairs. Here, $m = 6$ is used for Xi'an, and $m = 8$ for Chengdu, respectively. Both two hyper-parameters can ensure that over 90 percents of ground truth route can be acquired from Ω_{OD}.
- The number of stacked R-GCNs is set to 3.
- In the road attributes embedding layer, the embedding sizes of link feature representation (road ID, road types, number of lanes and one way or not) are set to 128, 8, 4 and 2, respectively. And the embedding size of node feature representation (node ID, node type and node street count) are set to 96, 2 and 2, respectively.
- In the temporal embedding component, we embed Weather ID and Holiday ID in R^8 and R^4, respectively.

6.4 Experimental Results

We compare our MWSL-TTE with other baseline methods under two datasets.

Performance on Travel Time Estimation. Table 2 shows the overall performance of estimating the travel times of Taxi OD orders. From the performance comparison, we find that our MWSL-TTE achieves the best performance than

other methods in terms of all three metrics. The better prediction results can be explained in two aspects. First, our model implements an effective graphical structure to capture the prior information of road network. Although all DCRNN, ConSTGAT and ASTGNN model the link-wise adjacency of road segments, the node-wise features especially traffic signal junctions are ignored. Thus, these three models without considering the node-wise adjacency cannot achieve higher accuracy. Second, given an OD pair, our weak supervision-based method can study the travel time distributions of the links and nodes. Compared with Taxi OD estimation baselines such as STNN and MURAT, in-process information of OD pairs improves our estimation results.

Ablation Study. In Table 2, except for the comparison experiment with the six baseline methods, we also con-

TABLE 5
Performance of MWSL under different combinations of α and β based on Xi'an and Chengdu dataset for both OD travel time estimation and route recovery.

| Datasets	Xi'an/Chengdu			
Parameters	TTE			Route recovery
	RMSE (sec)	MAE (sec)	MAPE (%)	Acc (%)
(α=1, β=0)	240.62/354.36	161.12/233.27	21.62/23.17	\
(α=0.8, β=0.2)	243.63/347.48	164.15/221.19	20.86/22.64	85.94/73.36
(α=0.8, β=0.1)	238.86/341.02	162.37/215.03	21.33/22.27	86.25/77.03
(α=0.6, β=0.2)	248.41/352.29	167.56/232.34	22.06/22.83	86.92/78.14
(α=0.6, β=0.3)	247.46/358.75	166.55/239.49	22.12/23.39	84.17/73.22
(α=0.6, β=0.1)	249.00/365.27	166.59/237.92	21.84/23.78	80.83/67.50
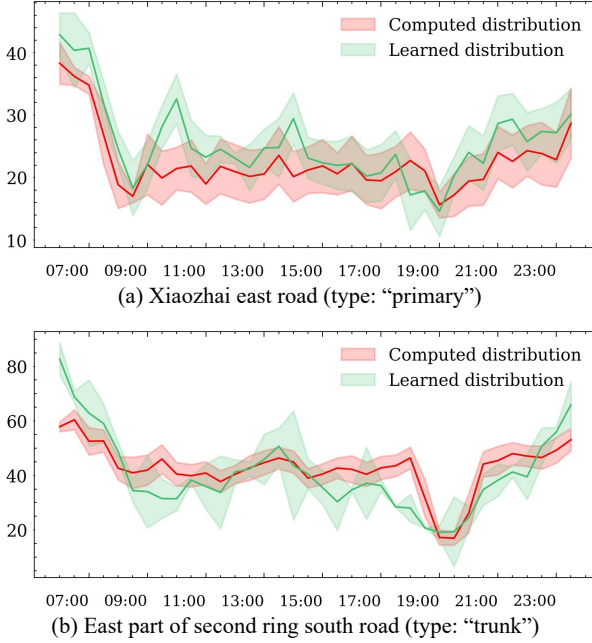
Fig. 6. Learned speed distributions of two sample links by MWSL, compared with the speed distributions computed by the original taxi trajectories.

duct the ablation study by replacing our MWSL-TTE with four variations, namely N-Node, N-GRU, N-R-GCN and N-PathUpdate, to evaluate the effectiveness of different modules in MWSL-TTE (see Fig. 3). In N-Node, we remove the node-wise estimation. In N-GRU, we remove the stacked GRU and only use the same number of layers of R-GCN. In N-R-GCN, we remove the R-GCN and replace it with the same number of layers of normal GCN [49]. In N-PathUpdate, we only implement the initial path as the in-process trajectories of OD orders and do not update the path based on the learned travel time and transition probability. The result comparison, it shows that R-GCN and node-wise estimation are the most critical parts. Regardless of the node-wise aspect, the performance becomes worse, and it proves the importance of modeling the complex adjacency for both the links and nodes. Furthermore, stacked R-GCNs with GRU integrate the multiscale information to capture both global and local features, and path update is also important in improving the travel time estimation. To sum up, the key designs of MWSL-TTE are effective.

Time Cost Analysis. Due to the online travel time estimation of our proposed MWSL-TTE, we further compare the inference time with main GNN-based baselines. As Table 2 shows, GNN-based travel time estimation models have significantly better performance than common deep learning-based methods, with a batch size 32 using a RTX2080 GPU card. Table 3 provides the inference time cost comparison for main GNN-based models. Although our proposed model is slower than DCRNN, the inference time of our model can acquire a faster inference time comapred with other relatively complex models. And this inference time result also proves that our model can process an online travel time estimation.

Performance on Route Recovery. Table 4 shows the performance of the route recovery task by given OD pairs. We can find that our path recovery of MWSL has better recovery accuracy and shorter model training time. STRS-based methods are very time-consuming due to the long iterations of inverse reinforcement learning to acquire the transition probability among road segments. Observe that the accuracy of Deep-STRS is always worse than that of STRS. The reason is that a larger sampling time interval between OD pairs leads to a more inaccurate grid-based traffic tensor which is the input of DeepGTT to model the traffic representation. In particular, a more complex Chengdu road network leads that the recovery accuracy of three methods dropping as expected.

Hyper-parameter Analysis. To further show the effectiveness of multi-task components of our model, we conduct experiments under different combinations of parameters α and β based on both of the two datasets. As observed in Table 5, on one hand, we find that in terms of the TTE task, the overall TTE performance improves as α changes from 0.6 to 0.8 under the datasets of two cities. However, $\alpha = 1$ doesn't achieve the best TTE performance. This is because that the β that controls the loss terms of route recovery also has some impacts on TTE prediction. More accurate path updates can bring the improvement of TTE prediction. On the other hand, the route recovery performance achieves the best accuracy when $\alpha = 0.8$ and $\beta = 0.1$. This indicates that the $D_{\text{KL}}(\mathcal{P}\|\mathcal{Q})$ also plays its part in obtaining a higher accuracy for route recovery. Furthermore, we compare the route recovery performance when $\alpha = 0.6$. The optimal hyper-parameter is the combination of L_{tp} and $D_{\text{KL}}(\mathcal{P}\|\mathcal{Q})$ as well, but excessive loss weight of $D_{\text{KL}}(\mathcal{P}\|\mathcal{Q})$ would
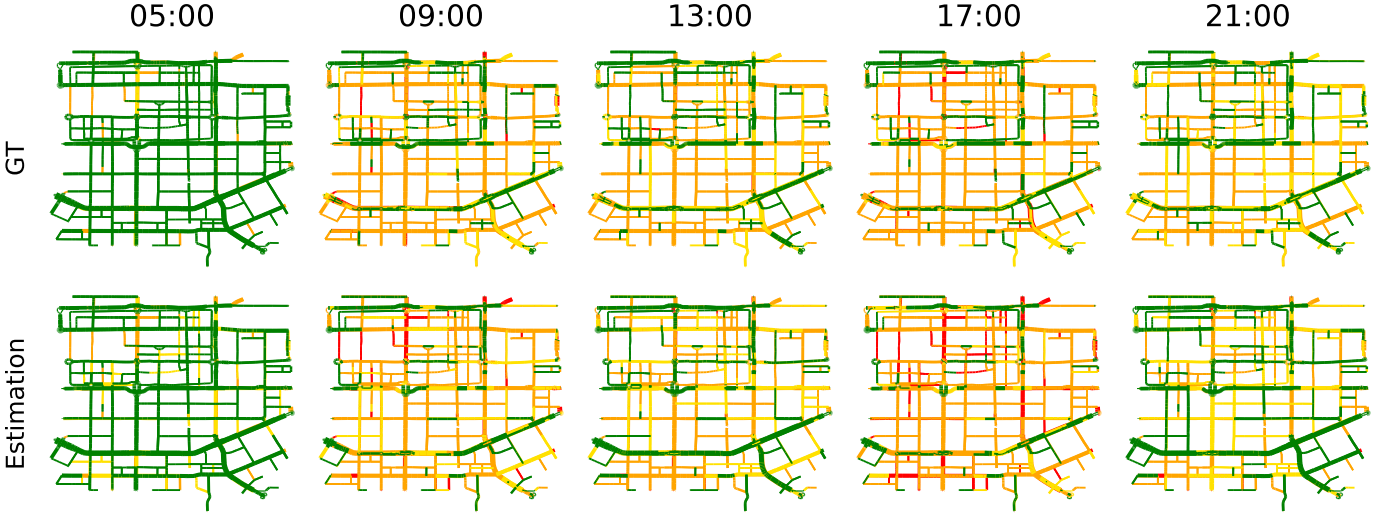
Fig. 7. Comparison between ground truth traffic state and estimated traffic state from MWSL-TTE. Here we use four kinds of colors to represent the different road states, which can be defined as 1) red - very congested, 2) orange - congested, 3) yellow - slow, and 4) green - unblocked.

cause the poor prediction performance. To sum up, we conclude that the experimental results demonstrated the superiority and generality of the multi-task components of our proposed MWSL.

7 CASE STUDY

Our MWSl-TTE not only can conduct path travel time estimation and route recovery for OD GPS trips, but also learn the travel time distributions of the links and nodes based on weakly supervised learning. Thus, in addition to the quantified evaluations described in Section 6.4, we also conduct a real-world case study in Xi'an, which visualizes the learned distributions of the links and nodes from the road network. Especially, we conducted the comparison with road condition computed by original taxi trajectories.

7.1 Learned Distributions for Nodes and Links

On one hand, we first provide the estimation results for different types of nodes, which are depicted in Fig. 5. We select three types of nodes in a road network and use the

time-consuming ratio to represent congestion status. The time-consuming ratio is calculated by dividing the corresponding mean value of learned travel time distributions by the maximum mean value among all nodes (Noted that we filter out top 1% largest node travel time). From Fig. 5, we can find that nodes with the "traffic signal" type are more time-consuming than the other two types of nodes, and most nodes in the morning peak are easy to become congested. These indicate that our node travel time estimation is reasonable in both spatial and temporal aspects.

On the other hand, we transform the travel time distribution into speed distributions in terms of links, and this transformation process is based on [41]. The reason for this transformation is that most users drive cars with a normal speed range (e.g., 10 kmph to 120 kmph), and thus we can easily analyze the rationality of learned speed distribution compared with link travel time. Two links' speed distributions were generated by the proposed MWSL, as is shown in Fig. 6, and we compared them with the speed distribution that is computed by the original taxi trajectories. To test the generalization of our model, we select two types of links, where the Xiaozhai east road is a busy link (type: "primary"), and we can find that the morning and evening peaks are obvious in both learned speed distribution and computed distribution. Another link is a highway link, and the commuting pattern only appears in the evening for both distributions. Based on the above analysis, it is concluded that the learned distributions of links can effectively represent different functional types of links. Furthermore, the mean values μ of learned speed distributions are closer to the ground truth.

7.2 A Demo of the Generated Road Conditions

We generate the road conditions by our MWSL based on taxi OD trips, and we compare with the ground truth computed by the original taxi trajectories. Especially, we mark it with the unblocked state for the road segments without taxi trajectories. Since the speed limit for each road is varied, which is primarily defined by road type or road length,
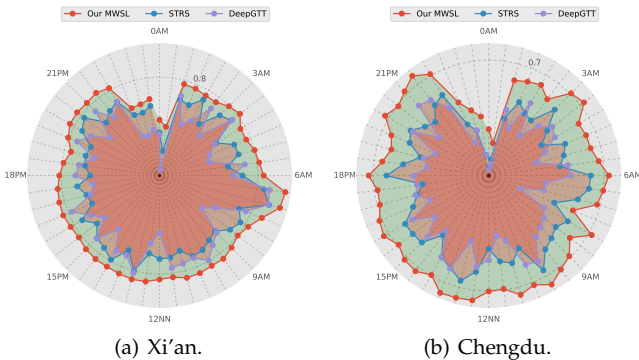

(a) Xi'an.

(b) Chengdu.

Fig. 8. The 24-h divergence between generated road conditions and ground truth under datasets of both Xi'an and Chengdu, compared with the temporal components of two baseline methods.

we use four kinds of colors to represent the different road states (very congested, congested, slow and unblocked). We divide the limiting-velocity for each road type equally, for example, the rate-limiting of primary road is $60kph$, so the interval between very congested is $[0, 15)$, congested is $[15, 30)$,slow is $[30, 45)$,unblocked is $[45, 60)$. The compared result is shown in Fig. 7. From the comparison of the generated road condition and ground truth at several time slots, we can acquire the following insights: 1) similar traffic state. The road condition generated by our model is similar to the ground truth. Most road segments have the same road states, and those road segments with different road states frequently have a consistent tendency; 2) rational adjacency correlation. We can find that the road segments with neighbor segments often have the similar road state for the generated road condition map. This indicates that our model can learn adjacency correlation of road network.

To better illustrate the model performance for generated road conditions, we also conduct quantitative analysis under Xi'an and Chengdu datasets. As is shown in Fig. 8, we computes the 24-h divergence between generated road conditions and ground truth. The prediction accuracy is relatively worse among these three methods from 22:00 PM to 2AM. This is because that a very small number of OD pairs in these time slots can not provide efficient model training for better prediction performance. However, the plots show that the generated conditions achieve the accuracy of around 80% and 70% at ordinary time slots under the datasets of both Xi'an and Chengdu, respectively. Compared with ground truth, our weak supervision learning method can provide believable road conditions only relying on the OD pairs.

8 CONCLUSION

For the first time, we consider the OD travel time estimation as an inexact supervision problem and propose a multi-task framework to infer the optimal route based on transition probability and learn the travel time distribution for each road segment and intersection through *expect MLE framework* [16]. The stacked R-GCN architecture has been employed to learn the complex relations of the road network, and we generate the travel time distribution for both road segments and intersections by 1st-order CP decomposition. Finally, we produce the transition probability between road segments by multi-layer perception. Moreover, an iterative update strategy has been proposed to update the transition probability and candidate paths during the training process. We evaluate our model on two real-world public datasets and verify the effectiveness of our proposed algorithm.

Future work can be concluded in four parts. Firstly, more potential superior distribution can be developed under the assumptions of weakly supervised learning, since in this paper, only log-normal distribution has been employed. Secondly, more advanced route search algorithms could be designed based on, for example, transition probability, travel time, or route probability. Thirdly, more urban scenarios, such as buses, subways, and people, can be tried to extend the applications of weakly supervised learning or travel time distribution. Lastly, a federated learning-based

method can be designed, since the OD types of data save abundant storage costs on the client's mobile phone.

9 ACKNOWLEDGMENT

REFERENCES

[1] Z. Wang, K. Fu, and J. Ye, "Learning to estimate the travel time," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 858–866.

[2] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, "When will you arrive? estimating travel time based on deep neural networks," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[3] X. Li, G. Cong, A. Sun, and Y. Cheng, "Learning travel time distributions with deep generative model," in *The World Wide Web Conference*, 2019, pp. 1017–1027.

[4] H. Wu, J. Mao, W. Sun, B. Zheng, H. Zhang, Z. Chen, and W. Wang, "Probabilistic robust route recovery with spatio-temporal dynamics," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1915–1924.

[5] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 25–34.

[6] I. Sanaullah, M. Quddus, and M. Enoch, "Developing travel time estimation methods using sparse gps data," *Journal of Intelligent Transportation Systems*, vol. 20, no. 6, pp. 532–544, 2016.

[7] H. Wang, X. Tang, Y.-H. Kuo, D. Kifer, and Z. Li, "A simple baseline for travel time estimation using large-scale trip data," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–22, 2019.

[8] I. Jindal, X. Chen, M. Nokleby, J. Ye *et al.*, "A unified neural network approach for estimating travel time and distance for a taxi trip," *arXiv preprint arXiv:1710.04350*, 2017.

[9] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1695–1704.

[10] J.-B. Cordonnier and A. Loukas, "Extrapolating paths with graph neural networks," *arXiv preprint arXiv:1903.07518*, 2019.

[11] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National science review*, vol. 5, no. 1, pp. 44–53, 2018.

[12] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.

[13] A. Richardson and M. Taylor, "Travel time variability on commuter journeys," *High Speed Ground Transportation Journal*, vol. 12, no. 1, 1978.

[14] H. A. Rakha, I. El-Shawarby, M. Arafeh, and F. Dion, "Estimating path travel-time reliability," in *2006 IEEE Intelligent Transportation Systems Conference*. IEEE, 2006, pp. 236–241.

[15] M. AREZOUMANDI, "Estimation of travel time reliability for freeways using mean and standard deviation of travel time," *Journal of Transportation Systems Engineering and Information Technology*, vol. 11, no. 6, pp. 74–84, 2011.

[16] Y. Zhang, N. Charoenphakdee, Z. Wu, and M. Sugiyama, "Learning from aggregate observations," *arXiv preprint arXiv:2004.06316*, 2020.

[17] M.-x. Wang, W.-C. Lee, T.-y. Fu, and G. Yu, "Learning embeddings of intersections on road networks," in *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2019, pp. 309–318.

[18] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European semantic web conference*. Springer, 2018, pp. 593–607.

[19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[20] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, "Traffic flow prediction via spatial temporal graph neural network," in *Proceedings of The Web Conference 2020*, 2020, pp. 1082–1092.

[21] D. Zhang, J. Han, G. Cheng, and M.-H. Yang, "Weakly supervised object localization and detection: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 5866–5885, 2021.

[22] Y.-F. Li, L.-Z. Guo, and Z.-H. Zhou, "Towards safe weakly supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, pp. 334–346, 2019.

[23] P. Nodet, V. Lemaire, A. Bondu, A. Cornuéjols, and A. Ouorou, "From weakly supervised learning to biquality learning: an introduction," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–10.

[24] B. Settles, "Active learning literature survey," 2009.

[25] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2013.

[26] D. Tiesyte and C. S. Jensen, "Similarity-based prediction of travel times for vehicles traveling on known routes," in *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, 2008, pp. 1–10.

[27] C.-H. Wu, J.-M. Ho, and D.-T. Lee, "Travel-time prediction with support vector regression," *IEEE transactions on intelligent transportation systems*, vol. 5, no. 4, pp. 276–281, 2004.

[28] R. Sevlian and R. Rajagopal, "Travel time estimation using floating car data," *arXiv preprint arXiv:1012.4249*, 2010.

[29] W. Luo, H. Tan, L. Chen, and L. M. Ni, "Finding time period-based most frequent path in big trajectory data," in *Proceedings of the 2013 ACM SIGMOD international conference on management of data*, 2013, pp. 713–724.

[30] B. Yang, J. Dai, C. Guo, C. S. Jensen, and J. Hu, "Pace: a pa th-ce ntric paradigm for stochastic path finding," *The VLDB Journal*, vol. 27, no. 2, pp. 153–178, 2018.

[31] T.-y. Fu and W.-C. Lee, "Deepist: Deep image-based spatio-temporal network for travel time estimation," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 69–78.

[32] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[33] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[34] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[35] Y. Yang, F. Zhang, and D. Zhang, "Sharededge: Gps-free fine-grained travel time estimation in state-level highway systems," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, pp. 1–26, 2018.

[36] H. Chen, B. An, G. Sharon, J. Hanna, P. Stone, C. Miao, and Y. Soh, "Dyetc: Dynamic electronic toll collection for traffic congestion alleviation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[37] K. Shao, K. Wang, L. Chen, and Z. Zhou, "Estimation of urban travel time with sparse traffic surveillance data," in *Proceedings of the 2020 4th High Performance Computing and Cluster Technologies Conference & 2020 3rd International Conference on Big Data and Artificial Intelligence*, 2020, pp. 218–223.

[38] G. Jin, H. Yan, F. Li, J. Huang, and Y. Li, "Spatial-temporal dual graph neural networks for travel time estimation," *arXiv preprint arXiv:2105.13591*, 2021.

[39] J. James, "Citywide estimation of travel time distributions with bayesian deep graph learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[40] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, "On map-matching vehicle tracking data," in *Proceedings of the 31st international conference on Very large data bases*, 2005, pp. 853–864.

[41] M. Li, A. Ahmed, and A. J. Smola, "Inferring movement trajectories from gps snippets," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, 2015, pp. 325–334.

[42] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[43] S. Luan, M. Zhao, X.-W. Chang, and D. Precup, "Break the ceiling: Stronger multi-scale deep graph convolutional networks," *arXiv preprint arXiv:1906.02174*, 2019.

[44] Y. Li, Z. Li, and L. Li, "Missing traffic data: comparison of imputation methods," *IET Intelligent Transport Systems*, vol. 8, no. 1, pp. 51–57, 2014.

[45] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive computing*, vol. 7, no. 4, pp. 12–18, 2008.

[46] C. Yang and G. Gidofalvi, "Fast map matching, an algorithm integrating hidden markov model with precomputation," *International Journal of Geographical Information Science*, vol. 32, no. 3, pp. 547–570, 2018.

[47] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[48] X. Fang, J. Huang, F. Wang, L. Zeng, H. Liang, and H. Wang, "Constgat: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2697–2705.

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

Hongjun Wang is working toward the M.S. degree in computer science and technology from Southern University of Science and Technology, China. He received the B.E. degree from the Nanjing University of Posts and Telecommunications, China, in 2019. His research interests are broadly in machine learning, with urban computing, explainable AI, data mining, data visualization.

Zhiwen Zhang received the B.E. and M.S. degree in Artificial Intelligence from Nankai University, China, in 2016 and 2019 respectively. From 2019, he is currently pursuing a Ph.D. degree at the Department of Socio-Cultural Environmental Studies, The University of Tokyo. His current research interests include urban computing and data visualization.

Zipei Fan received his B.S. degree in Computer Science from Beihang University, China, in 2012, both M.S. and a Ph.D. degree in Civil Engineering from The University of Tokyo, Japan, in 2014 and 2017 respectively. He became Project Researcher and Project Assistant Professor in 2017 and 2019, and he has promoted to Project Lecturer at the Center for Spatial Information Science, the University of Tokyo in 2020. His research interests include ubiquitous computing, machine learning, Spatio-temporal data mining, and heterogeneous data fusion.

Jiyuan Chen is working towards his B.S. degree in Computer Science and Technology from Southern University of Science and Technology, China. His major research fields include artificial intelligence, deep learning, urban computing and data mining.

Prof. Xuan Song received the Ph.D. degree in signal and information processing from Peking University in 2010. In 2017, he was selected as an Excellent Young Researcher of Japan MEXT. In the past ten years, he led and participated in many important projects as a principal investigator or primary actor in Japan, such as the DIAS/GRENE Grant of MEXT, Japan; Japan/US Big Data and Disaster Project of JST, Japan; Young Scientists Grant and Scientific Research Grant of MEXT, Japan; Research Grant of MLIT, Japan; CORE Project of Microsoft; Grant of JR EAST Company and Hitachi Company, Japan. He served as Associate Editor, Guest Editor, Area Chair, Program Committee Member or reviewer for many famous journals and top-tier conferences, such as IMWUT, IEEE Transactions on Multimedia, WWW Journal, Big Data Journal, ISTC, MIPR, ACM TIST, IEEE TKDE, UbiComp, ICCV, CVPR, ICRA and etc.

Lingyu Zhang joined Baidu in 2012 as a search strategy algorithm research and development engineer. He joined Didi in 2013 and served as senior algorithm engineer, technical director of taxi strategy algorithm direction, and technical expert of strategy model department. Currently a researcher at Didi AI Labs, he used machine learning and big data technology to design and lead the implementation of multiple company-level intelligent system engines during his work at Didi, such as the order distribution system based on combination optimization, and the capacity based on density clustering and global optimization. Scheduling engine, traffic guidance and personalized recommendation engine, "Guess where you are going" personalized destination recommendation system, etc. Participated in the company's dozens of international and domestic core technology innovation patent research and development, application, good at using mathematical modeling, business model abstraction, machine learning, etc. to solve practical business problems. He has won honorary titles such as Beijing Invention and Innovation Patent Gold Award and QCon Star Lecturer, and his research results have been included in top international conferences related to artificial intelligence and data mining such as KDD, SIGIR, AAAI, and CIKM.

Ryosuke Shibasaki was born in Fukuoka, Japan. He received his B.S., M.S., and Doctoral degrees in Civil Engineering from The University of Tokyo, Japan, in 1980, 1982, and 1987, respectively. From 1982 to 1988, he was with the Public Works Research Institute, Ministry of Construction. From 1988 to 1991, he was an Associate Professor in the Civil Engineering Department, The University of Tokyo. In 1991, he joined the Institute of Industrial Science, The University of Tokyo. In 1998, he was promoted to Professor in the Center for Spatial Information Science, The University of Tokyo. His research interests cover three-dimensional data acquisition for GIS, conceptual modeling for spatial objects, and agent-based microsimulation in a GIS environment.