Journal Pre-proof

Multi-attention graph neural networks for city-wide bus travel time estimation using limited data

Jiaman Ma, Jeffrey Chan, Sutharshan Rajasegarar, Christopher Leckie

PII: S0957-4174(22)00471-7
DOI: https://doi.org/10.1016/j.eswa.2022.117057
Reference: ESWA 117057

To appear in: *Expert Systems With Applications*

Received date : 31 October 2021
Revised date : 16 February 2022
Accepted date : 28 March 2022

Please cite this article as: J. Ma, J. Chan, S. Rajasegarar et al., Multi-attention graph neural networks for city-wide bus travel time estimation using limited data. *Expert Systems With Applications* (2022), doi: https://doi.org/10.1016/j.eswa.2022.117057.

Multi-attention Graph Neural Networks for City-wide Bus Travel Time Estimation using Limited Data

Jiaman Ma[a], Jeffrey Chan[b], Sutharshan Rajasegarar[c], Christopher Leckie[d]

[a] *Hangzhou Innovation Institute of Beihang University, China, jiaman.ma@outlook.com,*
[b] *RMIT University, Australia, jeffrey.chan@rmit.edu.au,*
[c] *Deakin University, Geelong, Australia, sutharshan.rajasegarar@deakin.edu.au,*
[d] *The University of Melbourne, Australia, caleckie@unimelb.edu.au,*

Abstract

An important factor that discourages patronage from using bus systems is the long and uncertain waiting times. Therefore, accurate bus travel time prediction is important to improve the serviceability of the bus transport systems. Many researchers have proposed machine learning and deep learning-based models for bus travel time predictions. However, most of the existing models focus on predicting the travel times using complete data. Moreover, with the dramatically increasing population, bus systems also expand and upgrade their routes to provide improved coverage. Consequently, predicting the routes with sparse or no historical records become vital in this situation, and have not been well addressed in the literature. In particular, the challenges involved in this prediction includes discovering routes with sparse records, discovering newly deployed routes, and finding the roads that need new routes. In order to address these, we propose a **M**ulti-**A**ttention **G**raph neural network for city-wide bus travel time estimation (TTE), especially for the routes with limited data, called **MAGTTE**. In particular, we first represent the bus network using a novel multi-view graph, which can automatically extract the stations and paths as nodes and weighted edges of bus graphs, respectively. Using inductive learning on dynamic graphs, we propose a multi-attention graph neural network with novel masks to capture the global and local spatial dependencies using limited data, and formulate a framework with LSTM and transformer layers to learn short and long-term temporal dependencies. Evaluation of our model on a real-world bus dataset from Xi'an, China demonstrates that the proposed model is superior compared to nine baselines, and robust to highly sparse data.

Keywords: Bus travel time prediction, graph neural network, multi-attention, sparse data.

1. Introduction

The growing number of vehicles and the fast growth of urbanization in modern cities have caused many traffic, health, and economic challenges. Public transit systems, especially bus systems, are vital in this situation for providing efficient service, and have attracted much attention as an efficient and environment-friendly alternative to motorized urban travel (Liu et al., 2020). The long waiting times and uncertainty in journey times can discourage patronage from public transportation. Hence, accurately predicting the travel times for existing bus routes is important. However, with the population increasing dramatically in urban areas, bus systems are required to evolve to meet the travel demands. According to the report in (of Transport of the People's Republic of China, 2018), the number of routes and the total length of the bus routes have doubled in China during the last decade. This brings many challenges. First, many bus routes in the suburbs with infrequent service have uncertain travel times. Second, the newly deployed routes in developing areas have no travel records, hence are unable to provide accurate arrival times at various stops. Third, when planning for new routes, travel time predictions between stops become important. Therefore, rather than merely forecasting the travel times with complete and dense travel records/data, the new challenge of predicting the bus routes with sparse or no historical records is crucial for realising intelligent public transport systems.

In this paper, we investigate the modeling of travel times of bus systems from a new perspective, i.e., city-wide travel time prediction with limited data. The aim is to predict the travel times of each route in the bus network under different conditions (with dense, sparse, and no records) in a given time slot.

Unfortunately, in spite of many existing studies focusing on travel time modeling, the results achieved are hardly satisfactory for city-wide bus travel time prediction. In the past decades, using historical data obtained by the Global Positioning System, various forecasting models have been proposed to estimate the bus travel time, including hybrid statistical models (Chiabaut and Faitout, 2021), machine learning, and deep learning methods, such as support vector regression (SVR) (Chen et al., 2020), recurrent neural network-based models(RNN) (Zhang et al., 2018), and graph convolutional networks (Jin et al., 2022). However, these methods are data-driven approaches and rely on large volume of high-quality labeled travel records to achieve high prediction accuracy. Therefore, bus routes with sparse and no historical records are unable to provide learnable travel patterns for prediction. To address this issue, some methods have been proposed in the literature for traffic prediction with limited data using taxi trajectories, such as (Tang et al., 2018; Tang et al., 2018). However, these methods have limitations for using with the bus systems, as they have more complex network structures (fixed routes and stop locations). Several significant challenges exist for city-wide, public transport travel time prediction, especially for the routes with sparse and no historical data. They include:

1) *Difficulty to learn temporal dependencies of bus routes with sparse data*: Because of the absence of historical travel information, the travel patterns of target routes learned from temporal features have much noise and lead to low prediction accuracies. How to infer the missing history values of the bus routes with sparse data and under design is the primary issue to be solved.

2) *Challenges to represent correlations among routes*: It is desirable to learn travel patterns of the sparsely recorded routes from the spatial neighbors with rich information. The existing methods usually divide a bus route by stop or links, which are unable represent the network structures and the correlations/similarity among bus routes. To the best of our knowledge, all existing works pay very limited attention to finding a proper representation method for the relationships among bus routes, especially for travel time prediction. One of the important reasons is that the bus routes have independent travel structures, and are influenced by many traffic factors. For example, two taxis may have similar travel times when running on the same road simultaneously. However, as shown in Figure 1, due to different numbers of stops and boarding/alighting passengers, bus routes can have different travel patterns on roads of same length. It is difficult to identify the neighboring routes with similar patterns. Therefore, a bus network representation method for effectively discovering spatial-temporal dependencies is the key to city-wide prediction.

3) *Without considering spatial and context features*: Since the city-wide bus travel time prediction is relatively a new problem, the spatial and context similarities among bus routes have not been studied deeply in the literature. Because of the complexity of bus network structures and the sparse travel records of the bus routes under different conditions, learning the global de-

pendencies become challenging using existing models, such as CNN-based methods. A deep learning model, which can learn spatial and context features with limited data, esspecifically for bus systems, is important to achieve highly precise prediction.

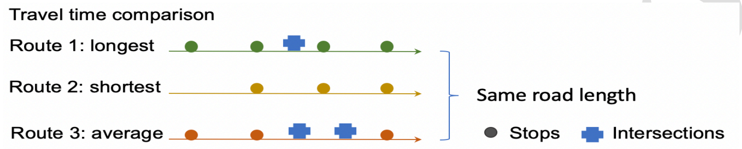


Figure 1: An illustration of the complex and independent travel patterns of public transportation routes.

To tackle the aforementioned problems of city-wide bus travel time prediction with limited data, we develop a novel neural network architecture named multi-attention graph neural network, called **MAGTTE**, which takes into account the heterogeneous information and characteristics of bus systems to estimate future travel times of bus routes with different conditions. The primary aim to learn city-wide travel patterns of bus routes under different conditions (with sparse or no historical records) is to discover spatio-temporal dependencies from the routes with fully-recorded data. Therefore, representing the bus network in a proper form is vital. As bus networks are difficult to be defined as regular grids, CNN-based methods are difficult to be utilized. We represent geo-structures and correlations of city-wide bus routes using a novel graph view. It can automatically locate important travel locations (e.g., stops, intersections) and represent the spatial-temporal dependencies among them by a proposed density-based representation method. Note the nodes represent travel segments extracted between every two adjacent stops. The spatial feature (location) of the nodes is determined based on their geo-structures. The edges represent the dependencies and correlations between nodes with weights acquired from different views(e.g., functionality similarity, spatial distances). It can help identify each route's travel patterns and represent correlations among them effectively. To jointly capture the time-varying spatial correlations among local and global neighbors with rich and stable information simultaneously, we propose a spatial attention module constructed by a multi-attention graph neural network with designed masks and a temporal attention module that includes LSTM and transformer layers. In the spatial attention module, we build an attention block based on graph attention network (GAT) with well-designed multiply attention heads and masks. Unlike previous works using graph convolutional network (GCN) that neglected travel time predictions with incomplete records, we exploit the advantages of GAT networks on inductive learning and handling dynamic graphs. We propose three types of designed attention masks from various views and combine the learned hidden features into multiple heads, effectively ensemble the dependencies of impact factors from global bus routes with rich information. It explicitly complements the incomplete travel time records of the bus routes with sparse and no historical data. In the temporal attention module, we first LSTM units explore the short-term historical travel patterns. Meanwhile, to enhance the individual information for each part of city-wide bus routes, we involve the Transformer layer to capture global temporal dependencies. Compared with the traditional RNN-based models, adding a self-attention model has the advantages of low cumulative error and efficient training in capturing long-term dependencies. Our main contributions can be summarized as follows:

- We present an end-to-end multi-attention graph neural network-based model to estimate the city-wide bus travel times with limited data, called **MAGTTE**. To the best of our knowledge, it is the first time exploration to achieve bus travel time prediction under different conditions, particularly for those with sparse or no historical records. This approach is not limited to buses with labeled data, but significantly improves the accuracy of the routes under design.

- We propose a bus network construction method from a novel multi-graph perspective. A designed weighted density-based representation method represents the road segments of bus routes as nodes and jointly defines the weighted relationships among them as edges. It can automatically discover and extract important geo-locations (stops and intersections) in the networks and can be effectively applied to neural networks to capture dynamic and complex travel patterns for all buses.

- We propose a spatial-temporal attention network to learn travel patterns of bus routes. A multi-head graph attention block with designed masks inductively learns the spatial dependencies from nearby and global neighboring spatial areas, which can help to infer the missing records for the buses with sparse and no records. A temporal module consisting of LSTM units and a transformer layer can discover current traffic conditions and long-term stable impacts. To the best of our knowledge, it is the first time the city-wide spatial correlations among different roads are considered in public transport systems.

- We evaluated our model using large-scale and real-world datasets, consisting of public transportation datasets of Xi'an. The experiments results show that the proposed model achieves significant improvement over several baselines on city-wide bus travel time estimation accuracy. It is more robust to the buses with high sparse travel records and applicable to travel times estimation for designing bus routes.

The remainder of the paper is organized as follows. In the next section, related work and relevant literature are reviewed. Section 3 introduces definitions and preliminaries. Section 4 presents the proposed **MAGTTE** model in more detail. Section 5 introduces the evaluation datasets and preparation details, and compares our proposed model with several baselines, where the results are presented and discussed. Finally, we conclude on the work in Section 6.

2. Related Work

In this section, we discuss the related work for city-wide bus travel time predictions with limited data.

In the past decade, various techniques and methods have been proposed to predict the travel time of public transportation systems based on GPS data, such as GPS trajectories (As and Mine, 2018) and smart card information (Zhou et al., 2017). Recently, data-driven approaches have been receiving increased attention and gained interest within the field of intelligent public transport systems. These methods can be classified into the following categories: (1) statistical models, such as historical average methods (He et al., 2019); (2) machine learning models, including support vector machine methods (Chen et al., 2020), Kalman filtering models (Achar et al., 2020); (3) deep learning models, such as recurrent network(RNN)-based models (Wang et al., 2018),convolutional neural networks (Petersen et al., 2019) and hybrid neural network models (Pang et al., 2019).

However, the existing studies mainly focus on single bus journey time forecasting with complete labeled data, which are rather limited. As mentioned in the introduction, at least three major situations have not been studied in previous bus travel time predictions but are significant to bus system optimization. First, the travel time prediction for the lines with infrequent service or in the suburbs. Besides, the newly deployed lines in developing areas do not have sufficient data to acquire travel times under different situations. The third is planning new routes for unserved areas. Current planners usually adopt real-road trails to estimate the travel times, which are time and manual costing. We conclude these new problems as city-wide bus travel time prediction with sparse data. We divide the related work of this problem into two parts, which are *city-wide travel time prediction of public transport systems* and *traffic prediction using limited data.*

City-wide travel time prediction of public transport systems

The above methods use complete bus data to predict the travel times by capturing their temporal dependencies. It leads to the low accuracy of city-wide prediction. The reason is that these methods only focus on the time aspect. As proved in many traffic prediction studies, neighbors with similar social functions and spatial closeness have high travel pattern correlations with each other (Liu et al., 2019). Therefore, the ignorance of spatial and contextual dependencies among bus routes or different parts of one bus line limits their performance on travel time prediction. Although some convolutional and recurrent neural networks (CNN and RNN) based models have been proposed in recent years (Wu et al., 2020), they still are used for temporal feature discovery in public transport systems.

Existing bus representation methods for travel time predictions are stop-based (Ma et al., 2019) and link-based (Kumar et al., 2019) segmentation, which divides the bus routes by stop and important intersection locations, have challenges to be utilized to learn spatial correlations in neural networks. They regard spatial travel patterns of each part of route as distinct and independent and without considering the correlation between the roads (Wang et al., 2018).

There are some models that consider spatial-temporal features in taxis' travel time prediction, such as the models of (Lv et al., 2018; Zhang et al., 2018; Abdollahi et al., 2020). For example, Wang et al. (2018) combines a geo-convolution layer and LSTM layers capture the local spatial correlations. The work of Fang et al. (2020) represents city network as a graph to learn different features based on graph neural networks. However, these models cannot be directly applied to public transport systems. Bus networks are not only affected by traffic conditions (vehicle speed, traffic flow, social functions etc.), but also by factors such as stop locations, dynamic bus load, bus schedules and fixed paths. Compared to travel time prediction for cars, travel time prediction for buses is more complex and have significant challenges. For example, difficulty in the discovery of spatial dependencies due to bus routes usually designed to be cover more areas in cities and independent to each other. Thus, city-wide bus travel time prediction model that consider spatial, temporal and contextual dependencies is needed.

Traffic Prediction With Sparse Trajectories.

An open challenge of city-wide traffic prediction is data sparsity. The existing bus travel time prediction methods, such as deep-learning-based models, are all based on the fully-recorded historical travel data, as shown in Table 1. With no or sparse trajectories, spatial and temporal dependency of travel patterns are difficult to capture, and the prediction accuracy of travel times of these models will significantly decrease. There are some methods proposed to estimate the traffic speed or travel time of freeways and paths using missing or sparse data(Tang et al., 2018) (Cui et al., 2020). The basic theory of these methods is to discover the similar travel patterns between the fully-recorded trips and the target paths at specific time slots to estimate their missing travel times (Rahmani et al., 2017). Therefore, identifying similar traffic patterns is the core of the methods. Several models have been proposed in recent years, such as a knn-based model (Habtemichael and Cetin, 2016), tenser decomposing methods (Liu et al., 2019), deep learning models (e.g., auto-encoder structures (Zhao et al., 2020)). However, due to the complex structures of bus networks, as mentioned in the last sections, these methods have difficulties in being utilized in city-wide or sparse bus travel time prediction. Therefore, these challenges cause the sparse and city-wide bus travel time problem rarely being proposed and not well solved.

Table 1: A comparison of deep-learning based travel time prediction approaches.

Work	Model	Data source	Target	Representation	Limited data	Impact factors				
						Road Network	Bus structures	Spatial	Contextual	Route correlation
Wang et al. (2018)	Geo-ConvLSTM	Taxi trajectories	path	-	✓	-	-	✓	-	-
Fu et al. (2020)	GAT+LSTM	Taxi trajectories	path	-	-	-	-	✓	✓	-
Jin et al. (2022)	ST-GCN	Taxi trajectories	path	-	-	✓	-	✓	✓	-
Achar et al. (2020)	Spatial-KF	One busline traj	one line	trip-based	-	-	-	✓(local)	-	-
Kumar et al. (2019)	knn+KF	One busline traj	individual buslines	stop-based	-	-	-	-	-	-
Petersen et al. (2019)	Convolutional LSTM	multi-busline traj	multi-lines(same path)	link-based	-	✓	-	-	-	Independent
He et al. (2019)	LSTM+interval HA	individual busline traj	individual buslines	stop-based	-	✓	-	-	-	Independent
Pang et al. (2019)	RNN	individual busline traj	multi-lines	link-based	-	✓	-	✓(local)	-	Independent
Barnes et al. (2020)	DeepTTE+Realtime	multi-busline traj	multi-lines	link-based	-	✓	✓	✓(local)	✓	Independent
Chen et al. (2020)	Deep believe network	individual busline traj	one line	link-based	-	-	✓	-	✓	Independent
Wu et al. (2020)	Attention-ConvLSTM	individual busline traj	one line	link-based	-	-	✓	-	-	Independent
MAGTTE	GAT(mask)+LSTMTransformer	City-wide limited bus traj	City-wide buslines	graph-based	✓	✓	✓	✓(global)	✓	**Neighbor learning**

In summary, the existing travel time models have different drawbacks for achieving city-wide prediction, such as without considering bus network structures and spatial correlations among bus stops and routes. For addressing the challenges, we propose a multi-attention graph neural network, called

MAGTTE in this paper. It attempts to develop a model to predict the city-wide bus travel times that considering spatial, temporal and contextual information from a global graph view. Second, it tries to fill up the blank field of bus travel time prediction with sparse data. To the best of our knowledge, this is the first work of city-wide bus travel time prediction using limited data and the first work that targeted on the buses in suburbs and under design.

3. Preliminaries

In this section, we present the definitions used in this paper and the problem definition of city-wide bus travel time prediction using limited data.

Definition 1 (Bus trajectories). Bus trajectories of a route can be represented as $\tau = \{p_1, ..., p_n\}$, where $p = \{long, lat, t\}$ contains a longitude $long$, a latitude lat at the create t-th interval.

Definition 2 (Public transportation Network Graph). We represent a public transportation network as a weighted graph $G = (S, E, \mathbf{W})$, whose nodes S are the road segments between two adjacent stops of one bus route and the edges represent the correlations among road segments, $\mathbf{W}$ denotes the weights of edges and represents the relationship strength in the graph, e.g., the road length similarity, social function similarity. A larger weight means that the two corresponding road segments have more similar travel patterns. The details of how to construct the public transport network graph will be elaborated in the next section.

Definition 3 (Travel time). A bus travel time record on a segment s (nodes in Graph G) of the i-th time interval is defined as t_i^s. The set of travel time records of all segments S at time slot t is represented as T_t.

Problem Definition. Given a transport network graph G generated from Definition 1, the historical travel time records of nodes (road segments) in the graph G, which is represented as $T = \{T_1, T_2 ..., T_t\}$, the objective of our problem is to predict the future travel time T_{t+1} in the next time slot for all bus trips.

4. MAGTTE

In this section, we introduce proposed multi-attention graph neural network approach for bus city-wide travel time estimation **MAGTTE** using limited data.

4.1. Framework Overview

Figure 2 gives an overview of our model **MAGTTE**. It consists of two main parts, which are *multi-view transport network graph construction* and *MAGTTE Network*.

multi-view ransport network graph construction: We construct three transport networks G from a novel graph view by considering both the spatial, temporal and contextual characteristics of bus systems. We first propose a geo-structure discovery algorithm and a node extraction algorithm to automatically extract the bus network graph's nodes (road segments between stops). Then, we construct edges graphs by proposing multiple adjacency matrices to represent the different relationships (e.g., the distance between segments, the similarity of path length, traffic conditions).

MAGTTE network: Specifically, we present a multi-attention-based graph neural network with designed masks and attention heads to learn the global and local spatial travel patterns of bus routes. It can learn the travel patterns of city-wide bus routes with complete and stable records to infer the missing values of target ones (e.g., with sparse and no records). We construct a temporal attention module consisting of LSTM and transformer layers to obtain the far-distance and recent temporal patterns. It combines spatio-temporal dependencies under different conditions (e.g., normal and abnormal conditions in traffic congestion and weather conditions) to predict bus travel times in the future time slots.

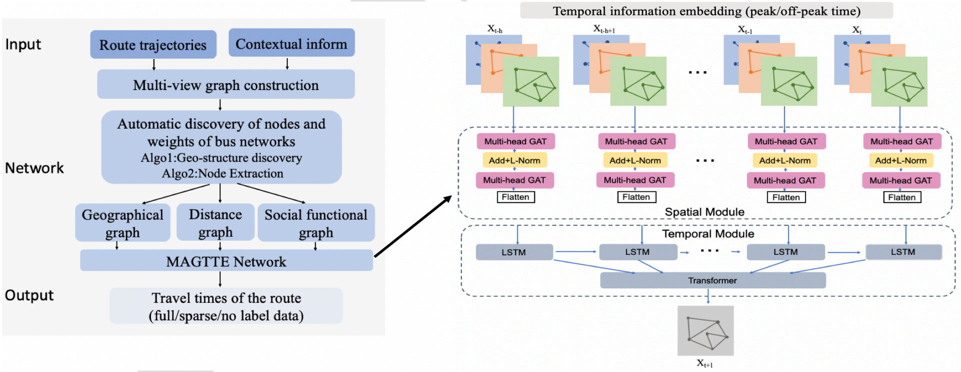


Figure 2: An overview of the model - **MAGTTE**, where consists of two components: multi-view transport graph construction(Section 4.2), multi-attention graph neural network (Section 4.3).

4.2. Multi-view transport graph construction

In this part, we describe the details of how to construct the public transport network graph. For predicting city-wide bus travel times, especially for the buses with sparse data, constructing a proper bus network for learning travel patterns from the bus routes with complete records is important.

As mentioned, since the travel structures of buses are usually divided by stops or important intersections, the grid-based methods, such as CNN, have difficulty in representing the bus network. Therefore, we propose to build a transport network from a novel graph view, which can effectively learn correlations between city-wide neighbors. Accurately generating transport graphs is the key to discovering spatial and temporal travel patterns of bus trips. We have two components in this method: nodes with features and edges with weights (adjacency matrix). In the graph, the nodes are the subsegments between every two nearby stops in one route. The edges represent relationships between travel segments. Based on the idea, we first propose a density-based representation method to extract the geo-structures of bus routes and then elect appropriate locations to represent the travel segments between stops by embedding the spatial and temporal information, such as peak/off-peak time period, weekday/weekends, by the widely used Word2Vec in our model. Secondly, we construct multiple weighted adjacency matrices to represent the correlations between segments based on the impact factors extracted.

Nodes and node features. The popular representation methods for travel time prediction are stop-based and link-based segmentation, which divides the bus routes by stop and important intersection locations. Based on these experiences, we construct the transportation network graph using these locations to represent the geo-structures of routes. First, we utilize a density-based clustering algorithm (DBSCAN) to extract the accurate locations and density of the intersections and stops. Instead of directly using the labeled stop information, we extract the locations from the algorithm for two reasons. First, as the travel segments we proposed are stop-based, some situations, such as two routes having the same stops but with different paths, e.g., see example in Figure 3, may occur. Using only the stop locations to extract the geo-structures of routes is therefore not accurate. Secondly, the waiting times at each stop and intersection are usually influenced by passenger numbers and traffic lights. It can result in the travel times of different routes having a significant difference. For instance, the waiting times of intersections can be different due to traffic light optimization, and popular stops usually have longer dwelling times than less popular ones. DBSCAN is a density-based algorithm used to find dense regions without requiring a cluster number or fixed shape setting. It requires two parameters: *eps* and the minimum number of points $minPts$. Our model is set according to the number of GPS points that have a speed value around zero in the database. Further, it should be small enough to find all the stopping areas along the route, including stops and intersections. Then, based on the numbers of points in each stop and intersection, we incorporated the importance of stops and intersections via weights.

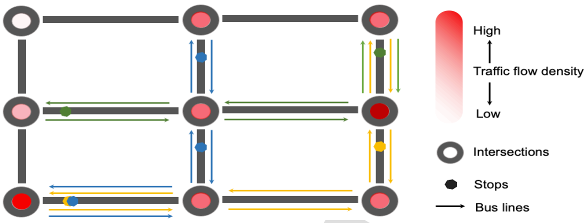


Figure 3: An example of bus trajectories distribution of a transportation network.

The density-based representation method consists of two steps, which are geo-structure discovery (Algorithm 1) and node extraction (Algorithm 2). The first step is to discover stop and intersections by Algorithm 1. After discovering important locations along the route, we select locations to represent the travel segments as node locations by the process of node extraction. The details as follow:

Algorithm 1 Geo-structure discovery.

```
Input: $\tau = \{p_1, ..., p_n\}$ in $\boldsymbol{G}$, where speed of $p = 0$.
Output: The geo-structure of bus routes, which represented by a sequence of locations, including stops and intersections $C = \{c_1, ..., c_n\}$. /* $c$ is a cluster generated by DBSCAN.
while $\tau$ of bus routes do $DBSCAN(P, minPts, eps)$
        return $\{c_1, ..., c_n\}$ /*$c$ is a cluster generated by DBSCAN.*/
    if unique number of (c) < number of stops then
        $minPts = minPts$ -1
    for $c \in C$ do
        $c(long, lat, num, st) = (centerof(c.long), (c.lat), numberofc) \cap stops(long, lat)$
```

Algorithm 1 outroutes the geo-structure discovery process. For discovering the bus dwelling areas, we first input bus trajectories that have zero speed, including intersections and stops. In steps 3 - 6, we process DBSCAN to the trajectories for each independent bus route until the number of clusters c is larger than the number of stops st. The extracted clusters c includes three features. They are longitude $long$, latitude lat, and the number of GPS num points in the cluster. The locations of clusters can help to identify the locations of travel segments S in the graph and the road length of S, num of clusters can help to identify dwelling times of intersections and stops. Then, for labeling the stop locations and in case the stop information still missing from the algorithm, we combine the stop locations with the clusters as the fourth feature (stop label), where $c_i(st)$ is a boolean index of a stop (1 means g_i contains stops). If the stop location is inside the cluster, the st of c is 1; otherwise, it is 0. If the stop location is not in any cluster, the stop location will be as a new cluster added into the set of C. The output of Algorithm 1 is the input of the Algorithm of node extraction.

Algorithm 2 Nodes Extraction.

```
1: Input: $C = \{c_1, ..., c_n\}$ in $\boldsymbol{G}$.
2: Output: The node set $S$ of transport network graph $\mathbf{G}$, which represented
   by a sequence of selected locations $\{s_1, ..., s_n\}$.
3: for all $c_i$ in $C$, if $c_i(st)$ == 1, $c_{i+k}(st)$ ==1 do
4:     extract $\{c_i, c_{i+1} ..., c_{i+k}\}$
5:     for do$\{c_{i+1}$): $c_{i+k-1}\}$
6:         $c_{i+1}(dist)$ = perpendicular distance $(c_i, c_{i+k}, c_{i+1})$
7: $s(long, lat)$ = locations of $\{c_i, c_{i+1} ..., c_{i+k}\}$
```

Algorithm 2 shows how to select locations to represent the travel segments between each two adjacent stops. First, we extract all locations between two nearby stops c_i and c_{i+k} in steps 3 - 4. Then, in steps 5 - 7, for each window (between every two stops c_i and c_{i+k}), we calculate the farthest perpendicular distance between each intersection to the line of stops. Finally, we choose the middle location between intersections and stops are the node locations to represent the travel segments, as shown in Figure 4. It can distinguish different paths with the same stop locations. Furthermore, we maintain the features of each cluster extracted from Algorithm 1 as an impact factor for building the spatial correlation between segments. For each node s, we record the travel time at each time slot as the node feature in one graph slice. The temporal information is also considered in our model. We utilized the widely-used Word2Vec method to embed the temporal information (peak/off-peak time slot, weekday/weekend) as another node feature. It can effectively learn the temporal dependencies by the multi-attention graph network, as shown in Figure 5.

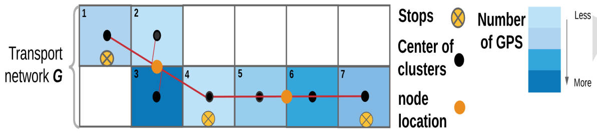
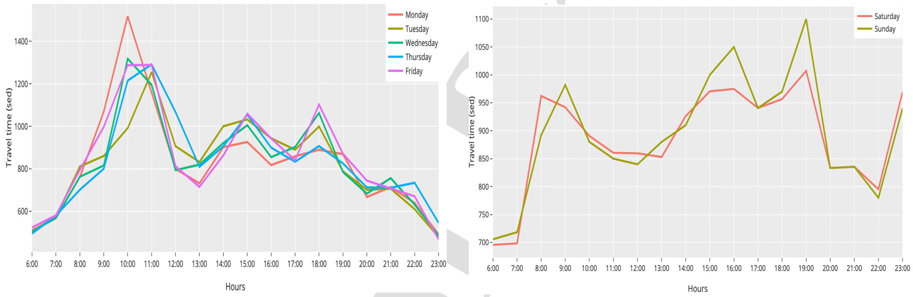


Figure 4: An illustration of the process of discovering important locations between each public transportation stations in a transport network.



a. Weekday travel times.

b. Weekend travel times.

Figure 5: Examples of travel times of a station in weekdays and weekends.

Weighted adjacency matrix generation. In this part, we introduce the processes of selecting and generating a weighted adjacency matrix for representing multiple correlations between each two travel segments. For predicting travel times of the city-wide buses, especially for those with sparse and no historical records, learning travel patterns from the fully recorded routes from a complete network view is important. Therefore, accurately representing the relationships between each two travel segments in the graph is necessary. We analyze various impact factors and build multiple weighted adjacency matrix in this part to represent the spatial dependencies of each segment. As mentioned in the Introduction, the travel patterns of public transportation routes are more complex than other types of vehicles, such as taxis, since they also have unique traffic information. Besides influenced by road length, direction, and interaction numbers (Wang et al., 2018), public transits are also impacted by departure time schedules, numbers, and locations of stops (dwelling times). Furthermore, with different social functions/land use around stops, the routes in transport networks have relatively different and independent travel patterns, even share part of the same paths. Selecting impact factors to build an adjacency matrix is an important task. Hence, for accurately discovering the travel patterns of routes, three aspects - geographical information, traffic environment and social functions are taken into consideration. The form of adjacency matrix $\boldsymbol{A}$ of the graph $G = (S, E, \mathbf{W})$ is shown as follow, where *corr* represent weights w of edges E. If the correlation is the similarity between nodes, we utilize the function of cosine similarity to define W.

$$\boldsymbol{A} = \in R^{N \times N} \begin{bmatrix} corr_{(1,1)} & \dots & corr_{(1,n)} \\ corr_{(2,1)} & \dots & corr_{(2,n)} \\ \vdots & \ddots & \vdots \\ corr_{(n,1)} & \dots & corr_{(n,n)} \end{bmatrix}$$

Distance graph: In one transport network, travel times/ speed of road segments usually relate to each other. Near neighbors (nearby travel segments) are more related than distant neighbors since they share similar real-time traffic conditions. For example, when traffic congestion occurs on a road segment, the drivers may choose the nearby roads to their destinations. The speeds of a range of areas around the central road with traffic congestion can all be decreased. Following this idea, we first use distance to identify the weight between two travel segments to distinguish near and distant neighbors. Here, we use reciprocal distance to decide the weights between nodes. The closer to the travel segments, the higher weights they have. We define the distance graph as $G_d = (S, E, \mathbf{W})$, weights $W = distance^{-1}$. Then, we normalized the weights of the distance graph into 0 to 1.

Geographical graph: Generally, the road segments with similar geo-structures have a high possibility of sharing similar travel patterns. In addition, the length of roads usually influences the travel times of vehicles. After extracted important geo-structures using the proposed algorithms, we consider two impact factors in geographical influence to travel time pattern discovery: the length and the number of intersections between each public transportation station. First, the length of road segments. Vehicles running on two roads with the same distance usually have similar travel times under similar traffic conditions. $G_{gl} = (S, E, \mathbf{W})$, weights $W = corr(road\ length)$. Second, the dwelling times during the path. The number of intersections, the waiting time of stops, and intersections highly influence the travel times of each part of transit trips Ma et al. (2019). Based on the number of clusters C for one travel segment S and number of points in one cluster $c(num)$, we define the dwelling time graph as $G_{gd} = (S, E, \mathbf{W})$, weights $W = corr(C_i, C_j)$, where C_i denotes the set of clusters of travel segment s_i.

Social functional graph: As demonstrated by existing models of traffic prediction, the regions/stations share similar travel patterns. In the problem of travel time prediction, it can impact the traffic flow and the patterns of passenger demands of stops. Based on the density and categories of Point of Interest data (POI), we define the social functional graph as $G_{gd} = (S, E, \mathbf{W})$, weights $W = corr(f_i, f_j)$, where f_i represents the social functions f of travel segment s_i.

4.3. Multi-attention graph neural network

In this section, we introduce the proposed multi-attention graph neural networks for predicting city-wide bus travel times using limited data. This model can effectively learn and predict the travel time of each bus route segment at a city-wide level, especially for the routes in the suburbs and the paths have no historical records. It can help to update the existing public transport systems by adjusting the outdated timetables of the routes in developing areas and help to choose new routes' paths and design new routes by providing travel times under normal and abnormal traffic conditions of each path that without public transits running. This model contains two modules, which are the spatial and temporal attention modules. It focuses on discovering spatial dependencies and temporal influence and combining them based on global and local attention mechanisms. We regard the travel time sequence as a time series $T = \{t_1, ..., t_n\} = \{x_1, ..., x_n\}$ in this section.

Spatial Attention Module. We apply a graph attention network in the spatial module to learn the spatial dependency among different travel segments. Due to the data sparsity problem of many bus routes, e.g., buses in the suburbs and newly deployed, the neural networks that rely on reliable labeled data have challenges to be utilized in city-wide prediction. Compared with the Graph Convolution Network (GCN) (Jin et al., 2022), the Graph attention network's ability to handle dynamic graphs and inductive learning is more suitable for city-wide travel time prediction with limited data. Based on the attention mechanism, the travel patterns of a bus route that with unstable records can be learned from others with rich information based on the weight aggregation process. The graph attention layer is the base component of GAT (Fang et al., 2020), which can learn the correlation between each pair of nodes and update the hidden feature of each of them. In general, we denote the node feature in layer l in time interval t as $h_i^t \in R^{d(l)}$, where $d(l)$ is the length of the feature of travel segment s_i at layer l. In the first layer, h_i^t is the input travel time record of segment s_i and the embedding temporal information. The attention coefficient of s_i and s_j can be represented as:

$$e_{ij}^t = a(Wh_i^t, Wh_j^t) \tag{1}$$

where W is the learnable parameters of layer l, a(.) is the function that calculates the correlations. We utilize *LeakyReLU* active function to train the feedforward neural network. For each layer, we normalize the output by a softmax function into [0,1].

$$a_{ij}^t = softmax(LeakyReLU(a(Wh_i^t, Wh_j^t))) \tag{2}$$

In order to obtain more abundant travel pattern combination, we extend the spatial attention to the multi-head attention. K independent and parallel attention heads with different learnable parameters are concatenated to achieve the final results:

$$h_i' = ||_{k=1}^{K} \sigma(\sum_{j=1}^{N} a_{ij}^t W h_i^t) \tag{3}$$

where || represents concatenation here.

Multi-masks:

Within each head, we propose to add an attention mask matrix to exploit the mutual importance information of the nodes in the graph. The reasons that we introduce mask matrix in our model come from two aspects. First reason is to effectively increase the prediction accuracy of the bus trips in the suburbs and under design (with sparse and no historical records). When the target segment has fully recorded data, its own travel records could be a reliable support to travel time prediction. However, when the target segments have significant sparse records or without records, most of their historical data are zero. With the dense noise, prediction models lead to the accuracy decreased. Therefore, adding masks to focus on the neighbors with fully-recorded data helps to predict the travel time of the particular types of bus trips. Second, considering the whole graph of each head in **MAGTTE** can cause computing waste. We use a mask matrix to provide a guide with label information to decide in what direction to update (i.e, shares the similar label) in certain layers can gain improved feature characteristics. Therefore, along with regular attention matrix (weighted adjacent matrix) A, we also propose multiple mask matrices to decide which nodes (neighbors) can be aggregated to focus on different impact factors and neighbors.

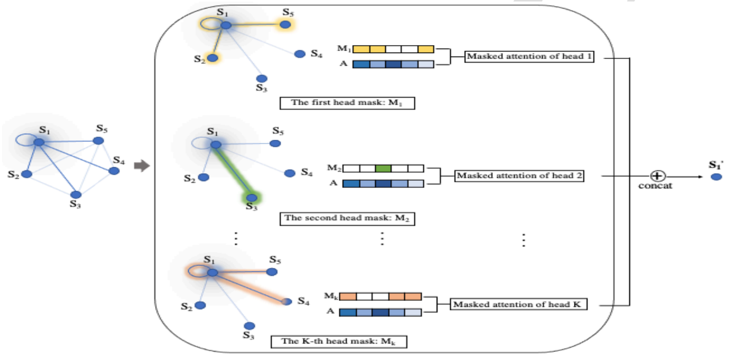


Figure 6: An illustration of masks in MAGTTE.

Typically, the function of mask matrix M of node s_i is to determining whether a node s_j can be aggregated as a feature support to s_i. For example, if we focus on the nearby neighbors of s_i, the mask M should prevent the nodes that the geo-distance is farther than a parameter γ. If we focus on the travel patterns of the neighboring nodes instead of it owns, the mask matrix M should eliminate the edges to itself in attention matrix $\mathbf{A}$. By considering different behaviors of travel patterns, the mask is defined as:

$$m_t^{(l)} = 1, a(Wh_i^t, Wh_j^t) > \gamma \tag{4}$$

$$m_t^{(l)} = -1, a(Wh_i^t, Wh_j^t) < \gamma \tag{5}$$

where m_t is a $N \times N$ matrix, which is as same as attention matrix $\mathbf{A}$. For each layer, we can get a normalized $N \times N$ attention matrix $\mathbf{A}$ with mask $\mathbf{A}$.

The mask matrix functions as an attention mask, when element wisely multiplied to the attention matrix, it ensures that the attention values between nodes labeled is positive, and the attention values between nodes without in the certain consideration, such as the previous historical data of the trips with many missing values, are negative. The output of X of layer l has the following formulation:

$$X^{l'} = (\mathbf{A} \circ \mathbf{M})X' \tag{6}$$

where $\mathbf{A} \circ \mathbf{M}$ is the element-wise product of $\mathbf{A}$ and $\mathbf{M}$.

Temporal Attention Module. After discovering the spatial dependency of time slot t, we connect a temporal module with it. The connection structure of spatial and temporal module is shown in Figure 2. The travel times of each bus trip is highly influenced by traffic conditions. For instance, when the traffic condition is normal, the previous far-distance historical records at the same time slot of the target route may have highly similar travel times at the current time slot. However, when traffic congestion happens, the travel patterns could be unstable, but still may have similar patterns with the very recent time slots. Therefore, the global and local(recent) temporal travel pattern all need to be considered for different traffic conditions.

Local temporal attention. The recurrent neural network (RNN) is an artificial neural network that especially suitable to capture the temporal dependency in sequence learning. However, previous studies show that RNN is usually hard to train long sequence due to the problem of vanishing and exploding gradients. To overcome these drawbacks, Long Short-Term Memory (LSTM) (Petersen et al., 2019) was developed by introducing an input gate and a forget gate to determine the optimal time lags automatically. Therefore, we build a model based on LSTM to focus on local temporal information.

So the input gate i_t of LSTM can be represented as:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \tag{7}$$

where h_t is the output vector of the LSTM unit, W and b are learnable parameter matrices and bias vectors in recurrent layer, σ is the standard sigmoid function. In our model, as the historical travel time records are learnt from other routes, over fitting can be a problem. Previous studies

(Srivastava et al., 2014) and following experiments have shown that dropout can efficiently reduce the problem of over fitting.

Global temporal attention. To discover temporal information from a global view, we introduce transformer layer in temporal module.

For single head self-attention layer, there are commonly three types of vectors, query, key and value for each node in the transportation graph, which denote as Q, K, V. The latent subspace learning process can be formulated as:

$$Q = X_i W^Q, K = X_i W^K, V = X_i W^V. \tag{8}$$

The global temporal attention output is calculated based on the scaled dot-product attention:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}}) \tag{9}$$

where dk is a normalization factor and its value is consistent with the feature dimension of Q and K.

Loss function. When we have obtained the high-dimensional spatiotemporal features, we make a prediction through a linear layer. Our **MAGTTE** can be trained to predict X_{t+1} from the previous h inputs of by minimizing mean squared error between our desired output $\hat{X}_{t+1}$ and use Mean Square Error (MSE) loss to train the model which can be formulated as:

$$L(\theta) = ||X_{t+1} - \hat{X}_{t+1}||_2^2 \tag{10}$$

where θ are learnable parameters in the model.

5. Experiments

In this section, we present extensive experiments to evaluate the performance of proposed **MAGTTE** on the problem of city-wide travel time prediction of public transport systems with different level of limited data. We used real-world public transportation dataset collected in Xi'an, China to evaluate the effectiveness of our framework.

5.1. Experiment Setup

Datasets. Three types of data are used in our experiments, which are bus trajectories, POI information, and weather reports, all acquired from the transport department, Xi'an, in June 2017. The bus trajectories consist of location, timestamp, speed, and bus ID information. The average sampling frequency is 30 seconds per points. The POI datasets consist of building location and category (social functions). We further collected the weather conditions for each day, which are divided into eight types. Using cross-validation, three routes are chosen as the target routes to evaluate the performance and robustness of our model. The three testing bus routes we selected are located in different areas of the city, which includes developed central areas, remote areas, and the paths linking central and suburbs. Then results are averaged. We randomly remove 40%, 60%, and 80% GPS recorded points in each trajectory of the three testing public transport routes to test the prediction accuracy for different degrees of sparsely recorded routes. We also removed all of their historical records and regarded them as three routes under design (without any historical travel time records) to test the travel time estimation performance of new routes (stop locations and the path are designed), which help evaluate whether the proposed **MAGTTE** is efficient for developing and optimizing routes.

Table 2: Dataset details

Datasets	Xi'an Bus GPS
Latitude	[34.1115, 34.3344]
Longitude	[108.786, 109.171]
No. of traj	300,000 per day
No. of Routes	278
Peak Times	7am-9am,5pm-7pm

Table 3: Training details

Variable	Value
Learning rate	0.001
Epochs	200
batch size	8
Decay ratio	0.1
Optimizer	Adam

Parameters and Evaluation metric. In our experiments, the time interval is 15 minutes, which is set larger than the departure time gap between each two buses. The first 25 days are set as the training set, the last five days are test set. We implement our **MAGTTE** with the popular deep learning framework PyTorch. The input data and the ground-truth of output are normalized with Z-score Normalization before used in the model. The filter weights of all layers in GAT are initialized using Xavier. The training details are presented in Table 3. On each benchmark, we train the model with the training set and a validation set to test the performance. In total, there are two hyperparameters used in our model and the number of attention heads K and the dimensionality d of each attention head. We tune these parameters on the validation set and observe the best performance on the setting $K = 3$, and $d = 8$. Finally, the trained model is evaluated on the testing set. Same as some previous works (Jin et al., 2022), we choose two common metrics to evaluate our proposed models and all baseline models, including Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). In detail, RMSE can reflect the absolute errors between the predicted results and the ground truths, where RMSE is more sensitive for some large errors. MAPE is a percentage error to measure the estimation accuracy. The lower RMSE and MAPE, the better the model perform. The definition of the metrics as follows:

$$MAPE = \frac{\sum_{i=1}^{n} |T_i - \hat{T}_i|}{T_i} \times 100\% \tag{11}$$

$$RMSE = \sqrt{\frac{1}{n} \Sigma_{i=1}^{n} (T_i - \hat{T}_i)^2} \tag{12}$$

Approaches for comparison. We compare with ten of the state-of-art methods:

1. Historical Average model (HA) (He et al., 2019): It predicts travel time by calculating the average historical travel time of bus routes in each time slot (15 mins);
2. Spatio-temporal Artificial neural network (ST-ANN) (As and Mine, 2018): It is a model built with three-layer and five hidden neurons with peak/off-peak times.
3. Kalman Filtering (KF) (Achar et al., 2020);
4. Support Vector Regression (SVR) (Sharmila et al., 2019): We apply radial basis function (RBF) kernel that was suggested by authors;
5. E-knn (Habtemichael and Cetin, 2016): This is a model proposed based on a weighted enhanced k-NN method, In here, we set the travel pattern similarity over 90% as k neighbors.
6. RnnTTE (Pang et al., 2019): This model based on LSTM neural network that contains a single fully connected LSTM layer with 128 hidden units.

7. DeepTTE (Wang et al., 2018): This model combines a geo-covlutional layer and a lstm layer to predict the travel times. In the geo-convolutional layer, the number of filters c = 32, the kernel size is set as 3. In the recurrent layer, the size of the hidden vector h_i as 128.
8. ConvLSTM (Petersen et al., 2019): the bus segment sequence into a 64-dimensional feature vector. Then, it is put forward into a two-layer fully-connected network to estimate bus time.
9. Attention-ConvLSTM (Wu et al., 2020): This model contains a ConvLSTM2D , a flatten layer, and a self-attention layer.
10. ST-GCN (Jin et al., 2022): This model architecture has spatial–temporal learning module. The embedding size of GCN model and the dimension set as 32.

Since the previous models represent the public transportation routes as sequences of road segments and none of them consider spatial dependencies, we first evaluate the performance of the spatial module and then show the performance of different models. This arrangement is to show the usages of spatial modules, and then to add it to the baselines for completely comparing the performances of different baselines with ours.

5.2. Performance of Spatial Module

In this part, we first evaluate the performance of the spatial module. The spatial module is based on graph attention blocks, it is used to learn travel patterns from the other fully-recorded data. It can be utilized in missing historical travel time inference independently. To test the performance of the spatial module, we use the MAPE and RMSE to compare the missing values we inferred with the other popular missing value estimation methods. The following baselines, which are popular methods used in the previous research with sparse data and the simple version of our method, are considered in the comparisons:

1. Historical Average value(HA) (He et al., 2019): the missing values are inferred based on the existing records of target routes;
2. Tensor construction (TC)(Tang et al., 2018): the model based similarity of segments from spatial and temporal aspects as a tensor $t_i = \frac{\sum_{j=1}^{n} sim(s_i,s_j)*t_j}{\sum_{j=1}^{n} sim(s_i,s_j)}$;
3. Weighted Tensor construction(WCT): This model uses the weighted adjacent matrix we built to add the importance of each fully-recorded

neighbors on the three views (distance, social functions and geo-length) into TC model.

Our method could be represented as multi-attention graph block (MAG). The comparison results are summarized in Table 4 and Figure 7.

Table 4: Travel time construction performance comparison.

	80% missing		60% missing		40% missing	
No.36	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
HA	23.47%	166.12s	18.44%	143.56s	12.35%	91.51s
TC	7.23%	54.43s	6.43%	50.27s	4.86%	37.05s
WTC	6.17%	48.13s	6.00%	46.85s	4.22%	31.64s
MAG	**6.08%**	**46.27s**	**5.88%**	**44.41s**	**3.74%**	**29.35s**
No.43	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
HA	17.67%	134.79s	14.08%	106.44s	11.26%	85.94s
TC	13.22%	100.04s	8.88%	67.93s	7.05%	53.63s
WTC	10.56%	80.57s	8.45%	64.45s	6.33%	47.77s
MAG	**9.89%**	**76.56s**	**8.19%**	**62.24s**	**6.28%**	**47.55s**
No.215	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
HA	28.43%	170.11s	21.33%	127.75s	14.22%	85.18s
TC	18.93%	112.39s	14.11%	79.36s	9.14%	56.54s
WTC	15.14%	90.70s	10.35%	68.02s	7.56%	41.28s
MAG	**13.60%**	**80.51s**	**9.89%**	**61.13s**	**6.81%**	**37.73s**

Table 4 shows the filled missing values of different methods for three bus routes with sparse records. As shown in the table, the performance of WTC and MAG outperforms the other two methods for all of the three bus routes, which means that traditional methods to predict travel times of sparsely recorded and new bus routes can lead to great error due to the limitations on handling data sparsity and without considering the different correlation among bus routes. For the bus line with a popular path like No.36 sharing part of the path with many bus routes, the segments with high similarity of fully recorded bus routes are easy to find. Under this situation, TC, WCT, and MAG all show good performance to generate the missing values. However, when the travel patterns become more complex, and the similar bus routes are less, like bus line No.215, the performance of the TC method is decreased, but the MAG method still maintains a stable accuracy.

Figure 7 shows the inferred travel time records of a bus line during the designing process. In this experiment, we give stop locations of the bus lines but do not provide any historical data of them. For example, we used No.43 as the assumed new bus route by deleting all its historical records and predicting its travel times, then using the real travel records to compare the models' performances. The table shows that the missing travel time records filled by TC method are all smaller than the real travel time records because the travel times of similar segments are mostly shorter than the target one. It shows the disadvantage of the TC method when the selected segment records are limited. MAG method shows a better result for the new bus line, which can conclude that the combination of different segments (sub-paths) and the weights of each of the segments are all important for the travel time construction.

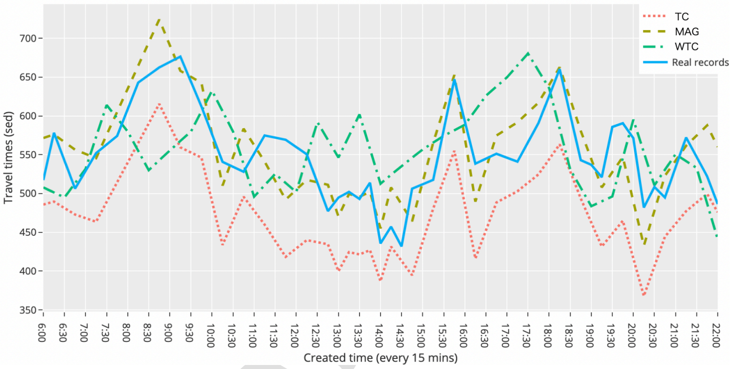


Figure 7: Travel time construction for a bus trip assumed in the designing process (No.43).

5.3. Performance of prediction models

We implement experiments to evaluate the performance of our model with the other state-of-art method on three dimensions:1) the different degrees of missing values, including no historical records; 2) the performance comparison of several public transportation travel time prediction models with **MAGTTE**; 3) The performance comparison of the baselines added spatial models (WCT) with ours. The results are shown in Table 5.

In Table 5, 40% missing means that 40% travel time records of target bus routes are randomly eliminated from the full records dataset. The first set of experiments in Table 5 (above the line) shows that compared with the existing travel time prediction methods with other or no missing value

Table 5: Travel time model performance comparison.
(NA means the baseline method have difficult to get the predicted result.)

MAPE	Fully Recorded	20% missing	40% missing	60% missing	80% missing	Designing
RnnTTE (Pang et al., 2019)	4.95%	12.56%	24.95%	60.51%	85.98%	NA
HA (He et al., 2019)	12.66%	12.89%	13.57%	16.87%	24.59%	NA
HA+SVR(Sharmila et al., 2019)	7.42%	11.60%	17.95%	23.33%	27.95%	NA
HA+RnnTTE	4.95%	13.61%	15.88%	16.93%	20.21%	NA
HA+DeepTTE (Wang et al., 2018)	4.50%	10.00%	19.97%	22.21%	24.88%	NA
HA+ConvLSTM (Petersen et al., 2019)	5.16%	12.78%	19.97%	22.76%	25.99%	NA
HA+Attention-ConvLSTM (Wu et al., 2020)	6.27%	15.67%	23.12%	25.21%	30.48%	NA
HA+ST-GCN (Jin et al., 2022)	5.77%	14.97%	20.11%	23.56%	25.46%	NA
E-knn(Habtemichael and Cetin, 2016)	12.66%	12.78%	14.93%	15.14%	19.47%	19.75%
WTC+ST-ANN (As and Mine, 2018)	9.98%	10.45%	10.86%	11.07%	11.78%	12.54%
WTC+KF(Kumar et al., 2019)	8.51%	9.19%	9.88%	10.32%	10.52%	11.21%
WTC+SVR	7.42%	8.26%	8.66%	8.69%	8.91%	9.03%
WTC+RnnTTE	4.95%	6.01%	6.61%	7.23%	7.83%	8.19%
WTC+ConvLSTM (Petersen et al., 2019)	5.16%	7.61%	7.94%	8.25%	8.86%	9.24%
WTC+Attention-ConvLSTM (Wu et al., 2020)	6.27%	6.58%	7.24%	7.54%	7.79%	8.21 %
WTC+ST-GCN(Jin et al., 2022)	5.77%	7.31%	7.75%	8.34%	8.56%	8.61%
MAGTTE	3.98%	4.32%	5.68%	6.55%	6.84%	7.14%

inference, the prediction errors are lower than the methods with our WTC method (under the line). Specifically, the missing values bring large bias to the prediction results. When the missing records increase, the first method RnnTT have issues in predicting the travel times. The existing popular travel time prediction methods such as SVR, ConvLSTM, DeepTTE, and ST-GCN have infinite MAPE values when training for designing bus routes (100% missing). The self-attention mechanism has low accuracy when the missing value inference ability is not great. For example, the accuracy of same model (ConvLSTM) increases 17.13%. In other words, these models cannot be used when the historical data are unknown. With the records becoming more sparse, simply using the average historical travel time records leads to a great decrease in accuracy. We can see that the traditional data missing value inference methods for travel time prediction (HA and E-knn) cannot provide accurate results for public transportation systems. However, the E-knn method has better performance than the other methods above the line, meaning that a good missing value inference method is important to the travel time prediction with sparse records. It demonstrates that the global spatial attention module and the self-attention mechanism are effective to city-wide bus travel time prediction with limited data.

5.4. Computation Time

We present the training times and prediction time of ST-GCN, Attention-ConvLSTM, ST-GAT (without masks), and our proposed model **MAGTTE** on Xi'an dataset. The results are shown in Table 6. STGCN is the most ef-

Table 6: The computation Time on the Xi'an dataset.

Model	Training(s/epoch)	Inference(s)
ST-GCN	56.33	101.88
Attention-ConvLSTM	176.64	32.98
ST-GAT	188.25	21.54
MAGTTE	229.13	18.76

ficient but with poor prediction since it does not have the ability to infer missing values of the target buses with sparse data. MAGTTE is less efficient as it includes the self-attention mechanism and the missing value inference module. For buses with dense data, the ST-GAT demonstrates that MAGTTE is an efficient method with high performance.

6. Conclusion

In this paper, we bring a new problem ignored by previous studies, which is modeling of city-wide bus travel times of bus systems with limited (dense, sparse, and no records) data. We propose a **M**ulti-**A**ttention **G**raph neural network for city-wide bus travel time estimation (TTE), especially for the routes with limited data, called **MAGTTE**. Specifically, we first automatically extract the stations and paths as nodes and weighted edges of bus graphs by a novel multi-view graph construction method. We then propose a multi-attention graph neural network with designed masks to capture global and local spatial dependencies using limited data, and present well-designed LSTM and transformer layers to learn short and long-term temporal dependencies. To the best of our knowledge, it is the first work that can introduce global spatial dependencies and infer the travel patterns of buses with sparse/no records from the city-wide buses with rich information. Experiments on real-world datasets show that MAGTTE achieves state-of-art results, and the advantages are more evident as the predictions are made into the buses with high data sparsity. In the future, we will further explore the city-wide travel prediction on comprehensive transportation systems.

References

Abdollahi, M., Khaleghi, T., Yang, K., 2020. An integrated feature learning approach using deep learning for travel time prediction. Expert Systems With Applications 139, 112864.

Achar, A., Bharathi, D., Kumar, B.A., Vanajakshi, L., 2020. Bus arrival time prediction: A spatial kalman filter approach. IEEE Transactions on Intelligent Transportation Systems 21, 1298–1307.

As, M., Mine, T., 2018. Dynamic bus travel time prediction using an ann-based model, in: Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication, Association for Computing Machinery, New York, NY, USA. p. 8.

Barnes, R., Buthpitiya, S., Cook, J., Fabrikant, A., Tomkins, A., Xu, F., 2020. Bustr: Predicting bus travel times from real-time traffic, in: Proceedings of the 26th ACM SIGKDD, pp. 3243–3251.

Chen, C., Wang, H., Yuan, F., Jia, H., Yao, B., 2020. Bus travel time prediction based on deep belief network with back-propagation. Neural Computing and Applications 32.

Chiabaut, N., Faitout, R., 2021. Traffic congestion and travel time prediction based on historical congestion maps and identification of consensual days. Transportation Research Part C: Emerging Technologies 124, 102920.

of Transport of the People's Republic of China, M., 2018. Report of transportation development of china cities .

Cui, Z., Lin, L., Pu, Z., Wang, Y., 2020. Graph markov network for traffic forecasting with missing data. Transportation Research Part C: Emerging Technologies 117, 102671.

Fang, X., Huang, J., Wang, F., Zeng, L., Liang, H., Wang, H., 2020. Constgat: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps, in: Proceedings of the 26th ACM SIGKDD, pp. 2697–2705.

Fu, K., Meng, F., Ye, J., Wang, Z., 2020. Compacteta: A fast inference system for travel time prediction, in: Proceedings of the 26th ACM SIGKDD, pp. 3337–3345.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks, in: AISTATS.

Habtemichael, F.G., Cetin, M., 2016. Short-term traffic flow rate forecasting based on identifying similar traffic patterns. Transportation research Part C: emerging technologies 66, 61–78.

He, P., Jiang, G., Lam, S.K., Tang, D., 2019. Travel-time prediction of bus journey with multiple bus trips. IEEE Transactions on Intelligent Transportation Systems 20, 4192–4205.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9, 1735–1780.

Jin, G., Wang, M., Zhang, J., Sha, H., Huang, J., 2022. STGNN-TTE: Travel time estimation via spatial–temporal graph neural network. Future Generation Computer Systems 126, 70–81.

Kumar, B.A., Jairam, R., Arkatkar, S.S., Vanajakshi, L., 2019. Real time bus travel time prediction using k-nn classifier. Transportation Letters 11, 362–372.

Liu, L., Qiu, Z., Li, G., Wang, Q., Ouyang, W., Lin, L., 2019. Contextualized spatial–temporal network for taxi origin-destination demand prediction. IEEE Transactions on Intelligent Transportation Systems 20, 3875–3887.

Liu, S., Yamamoto, T., Yao, E., Nakamura, T., 2020. Exploring travel pattern variability of public transport users through smart card data: role of gender and age. IEEE Transactions on Intelligent Transportation Systems .

Lv, Z., Xu, J., Zheng, K., Yin, H., Zhao, P., Zhou, X., 2018. Lc-rnn: A deep learning model for traffic speed prediction., in: Proceedings of IJCAI, pp. 3470–3476.

Ma, J., Chan, J., Ristanoski, G., Rajasegarar, S., Leckie, C., 2019. Bus travel time prediction with real-time traffic information. Transportation Research Part C: Emerging Technologies 105, 536–549.

Pang, J., Huang, J., Du, Y., Yu, H., Huang, Q., Yin, B., 2019. Learning to predict bus arrival time from heterogeneous measurements via recurrent neural network. IEEE Transactions on Intelligent Transportation Systems 20, 3283–3293.

Petersen, N.C., Rodrigues, F., Pereira, F.C., 2019. Multi-output bus travel time prediction with convolutional lstm neural network. Expert Systems With Applications 120, 426–435.

Rahmani, M., Koutsopoulos, H.N., Jenelius, E., 2017. Travel time estimation from sparse floating car data with consistent path inference: A fixed point approach. Transportation Research Part C: Emerging Technologies 85, 628–643.

Sharmila, R., Velaga, N.R., Kumar, A., 2019. Svm-based hybrid approach for corridor-level travel-time estimation. IET Intelligent Transport Systems 13, 1429–1439.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15, 1929–1958.

Tang, K., Chen, S., Khattak, A.J., 2018. Personalized travel time estimation for urban road networks: A tensor-based context-aware approach. Expert Systems With Applications 103, 118–132.

Tang, K., Chen, S., Liu, Z., 2018. Citywide spatial-temporal travel time estimation using big and sparse trajectories. IEEE Transactions on Intelligent Transportation Systems 19, 4023–4034.

Wang, D., Zhang, J., Cao, W., Li, J., Zheng, Y., 2018. When will you arrive? estimating travel time based on deep neural networks, in: Proceedings of AAAI, pp. 1–8.

Wu, J., Wu, Q., Shen, J., Cai, C., 2020. Towards attention-based convolutional long short-term memory for travel time prediction of bus journeys. Sensors 20, 3354.

Zhang, H., Wu, H., Sun, W., Zheng, B., 2018. Deeptravel: a neural network based travel time estimation model with auxiliary supervision. arXiv preprint arXiv:1802.02147 .

Zhao, J., Nie, Y., Ni, S., Sun, X., 2020. Traffic data imputation and prediction: An efficient realization of deep learning. IEEE Access 8, 46713–46722.

Zhou, Y., Yao, L., Chen, Y., Gong, Y., Lai, J., 2017. Bus arrival time calculation model based on smart card data. Transportation Research Part C: Emerging Technologies 74, 81–96.

Highlights

- First time to achieve city-wide bus travel time prediction with limited data.
- First time to build bus networks based on graph for travel time predic- tion.
- A spatial-temporal graph attention network to learn travel patterns from each other.
- Test results show the model can accurately predict bus travel time with limited data.

ORCID Information
Jiaman Ma: https://orcid.org/0000-0002-5685-6145

Credit Author Statement

Jiaman Ma: Paper idea, data collection and cleaning, methodology, experiments and paper writing; **Jeffrey Chan**: Paper idea, methodology and paper editing; **Sutharshan Rajasegarar**: Paper idea, methodology and paper editing; **Christopher Leckie**: Paper idea, methodology and paper editing;

Declaration of interest statement

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

Dr. Jiaman Ma