This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3417432

GT-TTE: Modeling Trajectories as Graphs for Travel Time Estimation

Yunjie Huang, Xiaozhuang Song, Shiyao Zhang[†], *Member, IEEE*, Lei Li, and James Jianqiao Yu[†], *Senior Member, IEEE*

***Abstract*—Travel time estimation (TTE) aims to predict travel duration and provide reliable planning for residential travel schedules. Trajectories naturally contain sequential features in form of GPS points with temporal precedence, which can be leveraged to improve prediction performance. Besides, the spatial information, i.e. the graph structure of the road network, can well represent the road highly and is commonly used to capture spatial information in traffic networks. However, extracting regional spatial information from trajectory data, in addition to its latitude and longitude information, poses a significant challenge due to the inherent format in which the trajectory data is recorded. In light of this, we propose a Graph-Transformer for Travel Time Estimation (GT-TTE) to utilize a Graph Transformer to adapt effectively to trajectories' sequential and spatial characteristics for improved TTE performance. By traversing the trajectory nodes with GT-TTE, we construct a graph structure for all trajectory points, thereby obtaining the relative spatial information of each point. Further, we obtain a region adjacency empirically more feature-rich over the sequential data. We evaluate GT-TTE on three real-world representative datasets and observe improvement by approximately 17% compared to the state-of-the-art baselines.**

***Index Terms*—Travel Time Estimation, Trajectory, Graph Learning, Attention Mechanism.**

I. Introduction

Travel Time Estimation (TTE) is pivotal in the realm of intelligent transportation systems and represents a domain intricately entwined with vehicle connectivity, which entails establishing connections between vehicles and other devices. [1]. It supports a large number of downstream applications, including but not limited to route planning [2], navigation [3], and traffic scheduling [4]. Precise travel time estimations can significantly reduce traffic congestion and improve the user experience while facilitating trajectory similarity calculation [5] and travel speed prediction [6]. The potential applications of TTE make it a valuable tool for optimizing transportation systems and enhancing urban mobility.

Yunjie Huang and Lei Li are with the Thrust of Data Science at the Hong Kong University of Science and Technology, Guangzhou, China (e-mail: yhuang863@connect.hkust-gz.edu.cn, thorli@ust.hk).

Xiaozhuang Song is with the School of Data Science at the Chinese University of Hong Kong, Shenzhen, China (e-mail: xiaozhuang-song1@link.cuhk.edu.cn).

Shiyao Zhang is with the Research Institute for Trustworthy Autonomous Systems, Southern University of Science and Technology, Shenzhen, China (e-mail: zhangsy@sustech.edu.cn).

James Jianqiao Yu is with the Department of Computer Science, University of York, York YO10 5GH, United Kingdom (e-mail: jqyu@ieee.org).

[†]James Jianqiao Yu and Shiyao Zhang are corresponding authors.

This work was supported in part by the Stable Support Plan Program of Shenzhen Natural Science Fund under Grant 20220815111111002.

TTE has been extensively studied with classical time-series and statistical learning models [7], [8], followed by data-driven models such as pooled regression tree models [9], [10]. However, these models are limited in capturing complex spatio-temporal relationships. In recent years, deep learning-based models, such as Convolutional Neural Networks (CNNs) [11] and Recurrent Neural Networks (RNNs) [12] have been used to extract both spatial and temporal features, respectively. Composite models of CNNs and RNNs [13]–[15] have been employed to discover spatio-temporal relationships in transportation data. Besides, Graph Neural Networks have emerged as a promising approach for learning spatial topology representations by modeling a city's road network as a graph, which better aligns with the geographical structure of the city.

Graph-based learning has shown impressive results in transportation, but incorporating graphs into trajectories remains a topic of debate. Some studies use map-matching to connect trajectory data to roads, requiring highly accurate road network data [16], [17], while others, like [5], create their own graph structures from trajectory data by exploring the local linkage between trajectory points. The latter provides a more flexible and scalability way to construct graphs including the spatial and temporal information instead of relying on other data sources. We follow this approach to construct the trajectory graph in spatial, and improve it by enhancing trajectory features.

In the meantime, the sequential information also plays a critical role in transportation tasks. RNNs are commonly used in traffic tasks due to their ability to process sequential data. Despite the ability of RNN models to accommodate sequences of varying lengths using padding and masking techniques [13], [18], such operations can introduce computational overhead and undermine computational efficiency [19]. This is particularly evident when applied to trajectory data characterized by significant disparities between the minimum and maximum numbers of recorded points.

Differently, this paper employs the Graph Trajectory Enhancement (GTE) module to capture local spatial features and the attention mechanism to extract global information, allowing each trajectory to observe all data points without truncation or interference and free from the introduced external operation and other issues.

By the discussion, we find the current TTE faces several challenges.

- Trajectory data, which captures spatial-temporal information for each GPS point, is subject to variability due to the movement of sampling devices. This type of data,

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3417432

while traffic-related, differs from the average speed or read occupancy data collected by fixed sensors installed on roads. This variability in trajectory data can pose unique challenges for TTE to exploit the graph spatial information.
- RNN-like models can effectively capture sequential dependencies in data but require input data to be in a sequential format and may require some preprocessing to standardize the input format. In the case of trajectory data, the number of recorded data points can vary significantly between trajectories, ranging from ten to thousand or more. Traditional methods of processing such data involve truncating trajectories into smaller segments of equal length for estimation. However, such truncation may change the starting and ending points and may introduce extraneous information that is irrelevant to the analysis, which can spoil the accuracy.

To address the above challenges, this paper proposed a Graph-Transformer for Travel Time Estimation (GT-TTE) with a Graph Trajectory Enhanced (GTE) method. GT-TTE combines the spatial information embedded in graph structures and the contextual learning abilities of Transformers, a powerful synergy capable of capturing and utilizing spatial-temporal dependencies within datasets spanning geographical and temporal dimensions. The proposed method begins by constructing a graph representation of the GPS points and extracting spatial relationships. Prior research, such as the work presented in [5], has endeavored to develop graph structures for trajectories. Limitations in terms of enhanced trajectory points exist in applying the proposed method to TTE due to its original focus on trajectory similarity. To address this, we restructured a region-passing graph A_{reg} to improve its suitability for TTE. We further incorporated Graph Convolutional Network (GCN) into Transformer, which gathers information from both A_{reg} and an order graph A_{odr} to obtain spatial information from both a local and global perspectives. This approach provides the added benefit of accommodating data with different lengths while avoiding interference caused by truncated data.

Our principle contributions can be summarized as follows:

- We propose a Graph-Transformer for Travel Time Estimation (GT-TTE) with a enhanced graph trajectory. GT-TTE constructs a hierarchical graph by extracting point-area relationships between GPS points, as distinguished from the more common models that use the adjacency matrix of road segments as the graph structure. This approach aims to improve the efficiency of message passing in long-distance trajectories and increase the spatial region information of the trajectory itself without adding extra data. To the best of our knowledge, this is the first time a graph creation technique like this has been performed for a TTE problem, and the accuracy of the suggested model is noticeably superior to the existing baseline.
- We propose a trajectory representation that enables efficient batch processing without the need for truncation.
- We optimize the method of constructing trajectory graphs to better extract spatial features for TTE-Tasks.
- We conduct extensive experiments on three real-life datasets to evaluate GT-TTE and previous state-of-the-art methods. Experimental results demonstrate the superiority of GT-TTE in comparison of state-of-the-art baselines.

The rest of this paper is organized as follows. We first review the background of Graph Convolution Network, Transformer, and works for TTE-Task in Sec. II. Then, we present the preliminaries in Sec. III. Sec. IV elaborates on the proposed framework. We conduct case studies and provide analytical discussions in Sec. V. Finally, we conclude this paper in Sec. VI.

II. Related work

This section provides an overview of the relevant research on GCNs and Transformers, and a summary of the approaches specifically developed for TTE.

A. Graph Convolution Networks

In the field of transportation, Graph Neural Networks (GNNs) offer a distinct advantage in effectively representing the non-Euclidean structure of road networks. This characteristic has been demonstrated by T-GCN [20], a GNN model specifically designed for transportation applications, which has shown promising results in traffic prediction. Moreover, Graph Neural Networks have been extended to various variants. For instance, the Diffusion Convolutional Recurrent Neural Network (DCRNN) [21] incorporates a diffusion mechanism for traffic flow prediction, expanding the capabilities of graph convolutional operators. Additionally, in 2019, attention-based Spatio-Temporal Graph Convolutional Networks were proposed [22], which introduced attention mechanisms to enhance the modelling of spatio-temporal dependencies in graph-structured data.

The effectiveness of Graph Neural Networks (GNNs) in traffic prediction has been increasingly demonstrated in recent studies. However, the applicability of these methods is largely dependent on the availability of graph-structured data, which may not be readily accessible in trajectory data. While efforts have been made to construct graphs from sequential data, such as sentences, these approaches encounter challenges when dealing with the distinct spatio-temporal characteristics inherent in trajectory data [13], [23], [24].

B. Transformer

Transformer, a deep learning model introduced by Vaswani et al. [25], has revolutionized the field of natural language processing by addressing sequence transformation tasks. This model leverages self-attention mechanisms, allowing it to selectively attend to specific positions within an input sequence and generate output sequences based on these attentions. A notable advantage of the Transformer is its concurrent processing of sequence positions, which leads to accelerated training and inference compared to traditional recurrent neural networks that rely on sequential computations.

The Transformer's robust sequence encoding capabilities have motivated researchers to explore its application in time

TABLE I
DEFINED SYMBOLS IN GT-TTE

Symbol	Definition
GTE	Abbreviation of Graph Trajectory Enhancement.
Attn	Abbreviation of Attention, and refer in particular to self-attention.
T	A trajectory, which contains a sequence of locations.
T_{meta}	The own properties of one trajectory like departure time or others'.
T_{rec}	A set of GPS points in one trajectory, each GPS point contains the necessary latitude and longitude information as well as some external information.
T_D	The historical trajectory data.
T_{enh}	The enhanced trajectory by GTE.
T_t	The total travel time for prediction.
p	GPS point in one trajectory
m	The max number of nodes in a sub-quadrant (sub-region).
$\mathbf{H}$	The hierarchical graph constructed by Point-region (PR) tree.
d_H	The dimensional size of the embedding vector.
d_m	The embedding size of positional encoding.
d_{model}	The embedding size of trajectory features.
N_H	The number of virtual nodes (divided regions) included in $\mathbf{H}$
E_H	The embedding matrix of $\mathbf{H}$
N_T, N_p	The number of features for the trajectory itself and each GPS point.
$p^r_{\text{feat}}, p_{\text{feat}}$	The features of the region to which each point belongs, constructed from the PR tree, and the features contained in the point itself.
r, r_c	An area and the center of the area.
$h^T_{\text{feat}}, r^T_{\text{feat}}$	Latitude and longitude information for all GPS points of the entire trajectory and the features of the area they belong to.
$A_{\text{reg}}, A_{\text{odr}}$	The Adjacency matrices constructed by region division and sequential information of the trajectory itself.
PE_o, PE_r	Positional encoding and Periodicity encoding.
τ	the maximum period of the time unit.

series prediction tasks. For instance, Li et al. [26] proposed a self-attentive convolutional layer to enhance the Transformer's performance in time series prediction. Xu et al. [27] focused on leveraging the Transformer to model spatial and temporal dependencies for traffic forecasting. Similarly, Cai et al. [28] improved traffic forecasting by incorporating spatial and temporal dependencies while emphasizing the continuity and periodicity of time series.

In the context of TTE, several studies have addressed the problem by proposing frameworks with Transformer. Liu et al. [29] introduced MCT-TTE, an end-to-end framework that focuses on learning spatio-temporal patterns and estimating travel time using both the provided path information and relevant external factors. Jin et al. [30] presented STGNN-TTE, a spatial-temporal graph neural network framework that incorporates a spatial-temporal module to capture real-time traffic conditions and a transformer layer to estimate travel time for individual links and total routes simultaneously. Similarly, Ma et al. [31] proposed a multi-attention-based graph neural network with designed masks and attention heads to learn both global and local spatial travel patterns specifically for bus routes. It is worth noting that these approaches leverage graph structures constructed based on the city's road network, yet they do not explicitly extract spatial information directly from the trajectory data itself.

C. TTE

In the domain of travel time estimation, models of various flavors have been explored over time. Initially, classical time series models like ARIMA [7] were commonly used for this task. As computational intelligence advanced, machine learning techniques such as gradient-boosting regression trees [10] and multiple regression trees gained popularity, allowing for better handling of non-linear data. However, these methods often required substantial feature engineering efforts.

With the advent of big data, deep learning models such as Long Short-Term Memory (LSTM) [32] and Gated Recurrent Units (GRU) [33] gained prominence due to their ability to capture sequential features effectively. To incorporate spatial information, hybrid models combining Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) were proposed [13]–[15].

More recently, attention-based mechanisms have emerged as a significant development in travel time estimation. Transformer-based approaches have been introduced to consider spatio-temporal dependencies and leverage multi-scale static and dynamic structures, leading to improved prediction performance [5]. These approaches leverage attention mechanisms to selectively focus on relevant information and have shown promising results in capturing complex patterns in travel time estimation.

Graph-based methods have also been developed for TTE tasks, as TransTTE, GCT-TTE, and Graph TTE, where the road network structure is used as a fundamental feature of the graph to enhance the geographical information. In these works,the graph are defined as $G = \{V, E\}$, each road in the network is treated as a node and $V = \{v_1, v_2, ..., v_N\}$ is a set of nodes, where N is the number of roads. E is a set of edge and an edge $e_{ij} = (v_i, v_j) \in E$ indicates the vertex $v_i \in V$ links to the vertex $v_j \in V$. The adjacency matrix $A \in R^{N\times N}$ represents the connection relationship of the nodes, which is composed of 0 and 1.

Expanding upon existing studies, we present a novel approach that leverages the transformer architecture to incorporate the inherent graph structure of trajectories. Unlike the prevailing graph structure, our method aims to extract personalized spatial information for individual trajectory points, thereby enhancing the overall semantic understanding of the trajectory. By considering the sequential order of the points and integrating temporal information, we effectively encode the temporal dynamics and capture rich contextual information within the trajectory. This innovative approach, referred to as the Graph Trajectory Enhancement (GTE) Method, enhances the representation and semantic understanding of trajectories

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3417432

for improved analysis and prediction tasks.

III. Preliminary

This subsection presents the trajectory data definition, the TTE and the corresponding notation and problem definitions to facilitate a more comprehensive explanation of the TTE-Task and the final problem objectives in our work. Table I provides the definitions of all symbols used in this study.

a) Definition 1 (Trajectory): A trajectory, denoted as T, comprises n multi-dimensional GPS points (for symbolic definition explicitly, and n is not the same in different trajectories). Each point contains information about its longitude and latitude, and additional external data such as time and distance interval (Chengdu16[1]) or timeID/weekID (Chengdu14/Porto[1]). Additionally, each trajectory may also record its own departure time, total distance, and total travel time, which we refer to as T_{meta}, as well as the content of points records, referred to as T_{rec}. Hence, each trajectory T can be represented as the combination of T_{meta} and T_{rec}. The formulation for trajectory T can be expressed as follows:

$$T = \{T_{\text{meta}}, T_{\text{rec}}\}, \tag{1}$$

$$T_{\text{rec}} = \begin{bmatrix} p^0.\text{lng} & p^0.\text{lat} & \cdots & p^0.\text{ext} \\ p^1.\text{lng} & p^1.\text{lat} & \cdots & p^1.\text{ext} \\ \ldots & \ldots & \ldots & \ldots \\ p^{n-1}.\text{lng} & p^{n-1}.\text{lat} & \cdots & p^{n-1}.\text{ext} \end{bmatrix}, \tag{2}$$

where $p^i.\text{lng}$ and $p^i.\text{lat}$ are the i-th GPS point's longitude and latitude. $p^i.\text{ext}$ is the external information of the i-th GPS point. $T_{\text{meta}} \in \mathbb{R}^{1\times N_T}$ and $T_{\text{rec}} \in \mathbb{R}^{(n)\times N_p}$. N_T represents the cardinality of the feature set within T_{meta}, whereas N_p signifies the cardinality of the feature set within a GPS point encapsulated in T_{rec}.

b) Definition 2 (Hierarchical Graph): A hierarchical graph $\mathbf{H}$ is constructed using historical trajectory data T_D. By splitting and matching all trajectory points into a region hierarchy, we create an abstract hierarchical graph $\mathbf{H} = \{\mathsf{V}\}$, where V is a virtual node of each region. By randomly walking on $\mathbf{H}$, we may obtain the node representation, which also serves as the regional information representation for each trajectory point. Section IV-B1 provides information on constructing a hierarchical graph $\mathbf{H}$.

c) Definition 3 (Trajectory Graph): For any trajectory T, we could construct a trajectory graph $G_T = \{V, E\}$ through the hierarchical graph $\mathbf{H}$. Here, V represents the set of all trajectory GPS points in the specify trajectory T, while E represents a set of edges. In graph G, $e_{ij} = 1$ if $v_i, v_j \in T$ and v_i, v_j both belong to V_k; else, it equals 0. V_k is a virtual node for a region in the hierarchical graph $\mathbf{H}$.

d) Definition 4 (Travel Time Estimation): Travel Time Estimation aims to predict the total amount of time required to travel between two locations using the latitude and longitude information of each GPS point included in a trajectory along with other non-time-related information.

$$T_t = f_{model}(T), \tag{3}$$

[1]The datasets will be introduced in V-A1

where T_t represents the prediction target, specifically the Total Travel Time. The function f_{model} encompasses the GTE (Graph Trajectory Enhancement) module, GCN (Graph Convolutional Network), and Attention block within a unified framework. This framework serves to map the original information contained in trajectories to the target travel time.

In GT-TTE, the process can be delineated as follows: initially, a hierarchical graph $\mathbf{H}$ is constructed using historical trajectory data T_D. This graph is the foundation for obtaining enriched trajectory data T_{enh}, which effectively captures significant local spatial information about the trajectory T. Moreover, graph $\mathbf{H}$ allows the construction of the regional connectivity relations matrix A_{reg}. By integrating the localized enhancement provided by T_{enh} and the global enhancement derived from A_{reg}, the travel time T_t for the trajectory T is estimated. This process can be mathematically represented as follows:

$$\mathbf{H} = f_{\text{GTE}}(T_D), \tag{4}$$

$$T_{\text{enh}}, A_{\text{reg}} = \text{EnhanceData}(\mathbf{H}), \tag{5}$$

$$T_t = f_{\text{GT-TTE}}(f_{\text{Attn}}(T_{\text{enh}}, A_{\text{reg}})), \tag{6}$$

where f_{GTE} denotes the GTE method, and f_{Attn} represents the Self-Attention layers. EnhanceData means that extracting the data from $\mathbf{H}$. The model's objective is to predict the total travel time T_t based on the given trajectory T.

IV. Graph-Transformer for Travel Time Estimation

In this section, we introduce the details of GT-TTE. The model consists of three critical modules: graph-based trajectory enhancement, Graph Convolution Networks, and the attention. Among them, The graph convolution networks is coupled with trajectory enhancement because it is only with the latter that we can apply the former to trajectories.

A. Overall pipeline

As illustrated in Fig. 1, the green part is the process of the GTE module extracting information from original trajectories. Each dataset requires only a single execution of the module, after which the output can be utilized repeatedly. After integrating the long trajectory information, the improved trajectories obtained via GTE are fed into the subsequent GCN and Attn to yield the final prediction results. Specifically speaking, GT-TTE comprises three main components. Firstly, the raw trajectory data is input into the graph-based trajectory enhancement module to obtain enhanced spatial and temporal information. The module generates a graphical representation of the trajectory data that captures the local spatial relationships between each trajectory point. Additionally, the GTE module encodes the sequence of positions in the trajectory, thereby capturing the temporal relationships between GPS points. This approach allows the trajectory data enhanced by the GTE module to be processed in parallel using a graphical convolutional network without truncating the trajectory data. Moreover, GT-TTE integrates an attention mechanism to provide global semantics for the trajectory.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3417432

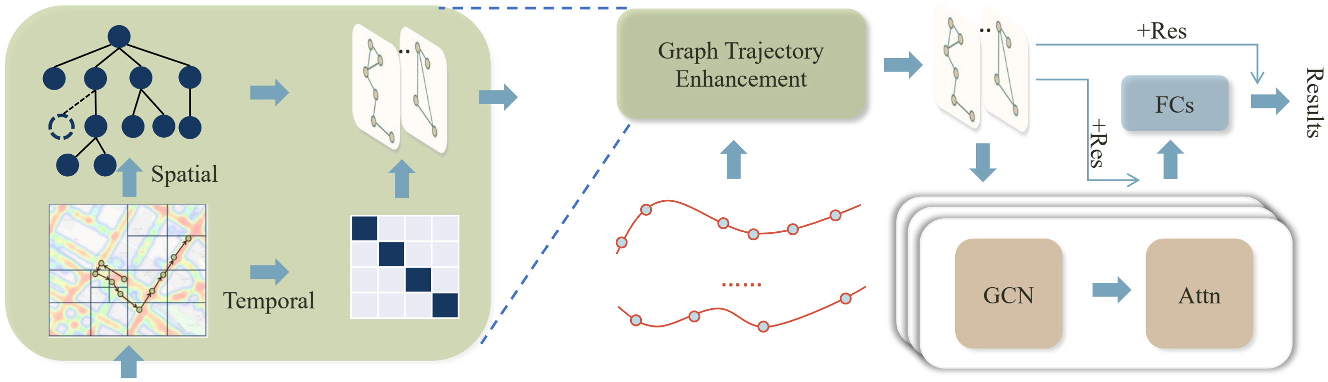


Fig. 1. (Left) The framework of the Graph Trajectory Enhancement (GTE). (Right) The framework of GT-TTE.

B. Graph Trajectory Enhancement

In this work, we employ structured data in the form of a graph to represent the irregular trajectory data. This storage format allows for batch processing of trajectory data of varying lengths without needing to align trajectory lengths. However, graphical modelling of trajectory data is not readily available. In the following subsection, we will first outline the process of modelling trajectory data into graphical data.

1) Spatial Construction:

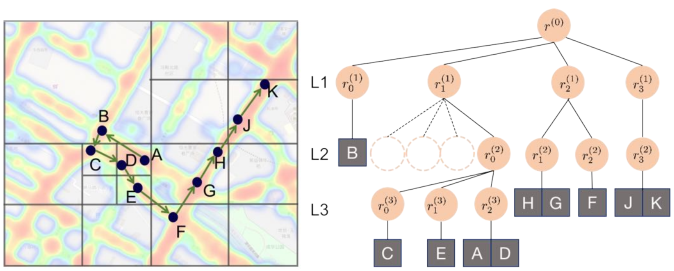


Fig. 2. The hierarchical graph $\mathbf{H}$'s construction process. Divide the region into four quadrants recursively until fewer than m GPS points included in. Each region is abstracted as a virtual node like the orange node in the right part.

a) Structure Construction: We adopt TrajGAT's approach [5] for graph-based data manipulation. To model the trajectory data as a graph, we first use Point-region (PR) quadtree [34] to construct a hierarchical spatial representation of the data, as illustrated in Fig. 2, which displays a trajectory's location and direction information on a map. The PR quadtree algorithm recursively divides the area into four equal quadrants, with each quadrant containing no more than p GPS points. The steps for constructing the PR quadtree are as follows:

- First, the latitude and longitude of all GPS points are extracted from their respective trajectories, and duplicate points are removed (for instance, if the same GPS point exists in both trajectory T_1 and T_2, only one will be saved).
- Divide the region into four quadrants recursively until fewer than m GPS points are under each quadrant.
- Temporarily excluding the real nodes (depicted as light grey squares), we retain the relevant information (such as the longitude and latitude of the central location and the weight and height of each quadrant) about the divided regions (represented by light orange dots, also referred to as virtual nodes) to create the hierarchical partitions denoted by $\mathbf{H}$.
- We consider $\mathbf{H}$ as a hierarchical graph and apply node2vec [35] to capture its tree-like topology. Specifically, we randomly sample a set of paths from $\mathbf{H}$ and learn the embedding vector $e_H \in \mathbb{R}^{1\times d_H}$ for each virtual node by simultaneously exploring various surrounding nodes while retaining as many existing neighbouring nodes as possible. Here, N_H is the number of the virtual nodes (or divided regions) and d_H represents the embedding size.

As depicted in Fig. 2, the entire region is partitioned into 16 grid cells, with each cell containing no more than $m = 2$ GPS points. The resulting embedding matrix $E_H \in \mathbb{R}^{N_H\times d_H}$, where $N_H = 16$. The Structure Construction's process is shown in Alg. IV-B1a. The additional symbols given in the algorithm are redefined for clear demonstration. As mentioned in paper Section III, each Trajectory T contains n GPS points (n is not same in different T). In the process of Graph Trajectory Enhancement (GTE), the input data are the set of unduplicated GPS points $\mathcal{P} = \{p^0_{T_0}, \ldots, p^i_{T_j}, \ldots\}$ extracted from all trajectory data. $p^i_{T_j}$ means the i-th GPS point in the j-th Trajectory. The GPS point include the basical geometry information latitude $p.lat$ and longitude $p.lng$. So, we could get the boundary information of the experimental region $BD = \{min(p.lat), min(p.lng), max(p.lat), max(p.lng)\}$ and then we could define the experimental region by $r_c = \{p_c.lat, p_c.lng, r_c.width, r_c.height\}$, where r_c means the current region which represented by it's center point's p_c location (latitude and longitude), and $r_c.width$ / $r_c.height$ of its' width / height. Then the number of points contained in current region r_c is denoted as $\hat{m}$. The output of the module is a Point-Region QuadTree PRQT represent the whole experimental region which is recursively divided into sub-region(sub-tree) until the leaf nodes(regions) only contains no more than m points.

b) Trajectory Local Semantic Enhancement: Based on the above mentioned procedures, a virtual node representing a sub-area was identified to which each actual node in the

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3417432

Algorithm 1 PRQT

Input:

The points set $\mathcal{P}$ in all Trajectories.

The boundary information BD of current region r_c.

Output:

The Point-Region QuadTree $PRQT$ which leaf nodes(regions) contain less than m GPS points.

if the number of points in current region $r_c.\hat{m} \leq m$ **then**

  return current region $r_c^i = \{p_c.lat, p_c.lng, r_c.width, r_c.height\}$

else

  return $\{$**PRQT**$(Sub\_r_c{}^1)$, **PRQT**$(Sub\_r_c{}^2)$, **PRQT**$(Sub\_r_c{}^3)$, **PRQT**$(Sub\_r_c{}^4)\}$

end if

graph belongs. Such a virtual node stores its region centroid's latitude and longitude information and region width and height information $p^r_{\text{feat}} \in \mathbb{R}^{1\times 4}$, where $p^r_{\text{feat}} \in \mathbb{R}^{1\times 4}$ denotes the GPS point p belonging to the region r and containing features $[r_c.\text{lng}, r_c.\text{lat}, r.\text{width}, r.\text{height}]$, and r_c is the center location of r. Furthermore, we have the embedding vector e_H of virtual nodes (sub-regions) obtained through random walk. Then we concatenate the latitude and longitude information of the factual node with additional information about the belonging region to form the spatial information of the real node as $p_{\text{feat}} = [p.\text{lng}, p.\text{lat}, r_c.\text{lng}, r_c.\text{lat}, r.\text{width}, r.\text{height}]$. For the trajectory T, $h^T_{\text{feat}} = \left[T_{\text{rec}}[:, :2] \| r^T_{\text{feat}}\right] \in \mathbb{R}^{(n+1)\times 6}$, where $\|$ refers to the concatenate spatiation, $r^T_{\text{feat}} \in \mathbb{R}^{(n)\times(d_H+4)}$ is the feature of the regions including the spatial information and embedding vectors. We have also created links between nodes in the same region and since get a region adjacent matrix $A_{\text{reg}} \in \mathbb{A}^{n\times n}$.

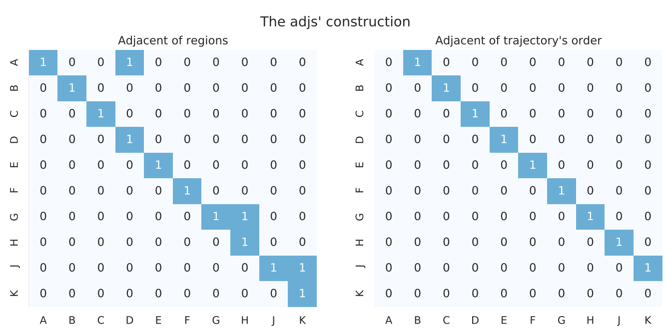


Fig. 3. (Left) The adjacent matrix constructed by the sub-regions after GTE module. (Right) The Adjacent matrix shaped by the order position in trajectories.

Fig. 3 illustrates the two adjacent matrices of the toy trajectory T shown in Fig. 2. The left one is the region adjacent A_{reg}, which is composed by the links between the nodes from the same regions. The right one is an order sparse adjacent A_{odr}, which is obtained by leveraging the inherent sequential information present in the trajectory data. We construct two adjacency matrices that are very sparse, essentially to create a fast-shareable channel in long trajectories, so that the trajectory information is not lost during the transmit process. It assigns all GPS points to the appropriate areas, creating virtual nodes for each one. Facilitating cross-node sharing over short distances through the high-level region to which the sub-region belongs helps mitigate challenges related to information transfer forgetting in lengthy trajectories. As such, even though the area and sequence matrices are extremely sparse, they are essential to GT-TTE for sharing the graph embedding. The sparse adjacency matrices and the graph embedding complement each other.

2) Constructing Temporal Encoding: Two types of temporal encodings are utilized in this study. One is the sequential order of the information contained in one trajectory. The other is the periodicity information included in datasets, such as the day of the week, the time of the day, and so on.

a) Positional Encoding: The order in which each GPS point presents in the trajectory is regarded as its position number $i, i \leq n$. The positional encoding is formulated as follows:

$$PE_o = \begin{cases} \sin\left(i/(1000^{\frac{i}{d_m}})\right) & \text{if } i \text{ is even} \\ \cos\left(i/(1000^{\frac{i}{d_m}})\right) & \text{if } i \text{ is odd} \end{cases}, \tag{7}$$

where d_m is the embedding dimension, and $PE_o \in \mathbb{R}^{n\times d_m}$. Given that the majority of trajectories consist of 100–400 points, the denominator 1000 is sufficient to obtain distinct encodings for each location in the trajectory data instead of the original 10000 as used in [25].

b) Periodicity Encoding: Trajectory data typically incorporates structured temporal information, such as GPS points recorded at specific time intervals throughout the day or on specific days of the week. These temporal patterns exhibit periodicity, regularly occurring within intervals of 24 hours or a week. Exploiting this periodicity provides valuable additional temporal context to the trajectory data. The Periodicity Encoding is computed as

$$PE_r = \sin\frac{2\pi i}{\tau}, \tag{8}$$

where $PE_r \in \mathbb{R}^{\tau}$ is a set of numbers that indicate different moments in the period τ. Symbol τ represents the duration or length of a single period within a given time frame, e.g., $\tau = 24$ in the time of a day or $\tau = 7$ in the day of a week.

Depending on the practical implications, PE_r can be included as the supplementary information $p_t.\text{ext}$ for the trajectory or added to PE_o as the sequential position-time information for the entire trajectory. In our study, we apply the both for local (GTE) and global (attention mechanism) enhancement, respectively.

C. Graph Convolution Network

As previously mentioned, a recursive spatial subdivision technique was utilized to divide the space into sub-quadrants, from which additional spatial information was extracted and integrated into the analysis. We identify the quadrant domain for each GPS point and concatenate the point's latitude, longitude, and quadrant information. Furthermore, intra-quadrant connections have been established for each node, denoted as A_{reg}. However, to facilitate messages passing across different quadrants for obtaining global enhancement, we construct A_{odr}

by leveraging the inherent sequential information embedded in the trajectory data, shown in Fig. 3 (right), enabling us to combine data about neighbour nodes using A_{reg} plus A_{odr}.

In each GCN layer, we denote the input adjacent matrix as $\mathbf{A}$. The graph Laplacian matrix can be computed as

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\hat{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}} \tag{9}$$

where $\mathbf{I}$ is the identity matrix and $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$. $\mathbf{D} = \text{diag}(\sum_j \mathbf{A}_{ij})$ is the diagonal in-degree matrix. For graph convolution operation, we have

$$H^{(l+1)} = \sigma(\mathbf{L}H^{(l)}\theta^{(l)}), \tag{10}$$

where l is the layer index, H^l stands for the output of the l-th layers, θ^l represents the l-th layer's parameters, and $\sigma(\cdot)$ is the sigmoid function, respectively.

D. Self-Attention Mechanism

Self-attention is a specific attention mechanism that enables the modelling of the dependencies among items within the input sequence [25], [36]. It evaluates the relationships between individual elements within a sequence and assigns weights to each element based on its relevance or importance concerning the other elements. This mechanism can be viewed as a self-focusing process, where each element in the sequence pays attention to the other elements to capture the contextual information more effectively, allowing for a comprehensive understanding of the global relationships and dependencies presented in the sequences.

In GT-TTE, we leverage the self-attention mechanism to capture the global relational features among GPS points within a certain trajectory. For a single self-attention layer, we begin with the input data $X_T \in \mathbb{R}^{n \times d_{\text{model}}}$, which represents the embeddings derived from the point records $T_{\text{rec}} \in \mathbb{R}^{n \times N_p}$ included in a single trajectory T (as defined in III-0a). Subsequently, we employ learnable parameters $W_Q, W_K, W_V \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ to obtain the query, key, and value $(Q/K/V)$ matrices:

$$Q = X_T \cdot W_Q, K = X_T \cdot W_K, V = X_T \cdot W_V \tag{11}$$

The attention function is performed as follows [30]:

$$\text{Attention}(Q, K, V) = \text{Softmax}((\frac{QK^\top}{\sqrt{d}}) \cdot V), \tag{12}$$

where d is a normalization factor and its value is consistent with the feature dimension of Q. The query (Q) and key (K) matrices are used to identify the correlation between points. Based on this correlation, a weighted addition is performed on the value (V) matrix. This process aims to reconstruct the features using non-local correlations, thereby learning global contextual features.

On top of it, a multi-head self-attention mechanism is further employed to capture more intricate relationships among GPS points. It divides the input data X_T into multiple heads to focus on different parts of the sequence in parallel and learn diverse representations [37]. The multi-head attention mechanism is formulated as follows [38]:

$$\text{MultiHead}(Q, K, V) = \Big\Vert_{h=1,\ldots,N^h} \text{head}_h W^O, \tag{13}$$

$$\text{head}_h = \text{Attention}(X_T^h \cdot W_Q^h,\ X_T^h \cdot W_K^h,\ X_T^h \cdot W_V^h), \tag{14}$$

where $\text{Multihead}(Q, K, V)$ is the final output of the multi-head self-attention layer that is integrated from the output of a single head^h via a learnable matrix $W^O \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$. N^h is the number of heads. X_T^h is the h-th part of the trajectory data X_T. $W_Q^h, W_K^h, W_V^h \in \mathbb{R}^{\frac{d_{model}}{N^h} \times \frac{d_{model}}{N^h}}$ are weight matrices specific to each attention head.

TABLE II
DATASET DESCRIPTION

Datasets	Chengdu16	Chengdu14	Porto
Travel Time (StD†)	425.16	237.37	173.88
Travel Time Mean	586.80	703.49	595.00
#† of Traj for training	170680	731693	736974
# of Traj for validation	23881	99672	130800
# of Traj for testing	90375	239039	262424
longitude mean	104.08	104.06	-8.62
longitude StD	0.0215	0.0362	0.0096
latitude mean	30.68	30.65	41.16
latitude StD	0.0190	0.0315	0.0061

†: StD is the abbreviation of Standard Deviation, # is the representation of numbers.

V. CASE STUDIES

In this section, a comparative analysis is conducted between the proposed GT-TTE method and the existing state-of-the-art approaches to travel time estimation tasks. This experimental evaluation gives insights into the distinct characteristics and distribution patterns exhibited by the three datasets under consideration. Moreover, a comprehensive analysis is performed to evaluate the impact of the various components comprising GT-TTE, revealing the effectiveness of Graph Trajectory Enhancement (GTE). Finally, experimental evaluations are carried out on trajectories of diverse lengths to showcase the exceptional performance of GT-TTE in accurately estimating full trajectories and its notable predictive capabilities for medium trajectories.

A. Experimental Settings

1) Datasets: We employed three real-world datasets in our experiments. The detail information of the datasets are displayed in Table II.

- Chengdu16: This dataset is collected between Nov 1st, 2016 and Nov 30th, 2016 from taxis in Chengdu, China[2].
- Chengdu14: This dataset is collected between Aug 3rd, 2014 and Aug 29th, 2014 from taxis in Chengdu, China. We process the data in accordance with the previous literature [18][3].
- Porto: This dataset aggregates the trajectories of 442 taxis collected in Porto, Portugal for the entire year from

[2]The dataset could be downloaded after requesting approval via https://outreach.didichuxing.com/app-vue/KDD_CUP_2020.

[3]The datasets could be downloaded from https://drive.google.com/file/d/1KiiSnx5x6f8B-pkkZEk7QYHIHg7I-zp8/view

July 1, 2013 to June 30, 2014. Similarly, we remove all incomplete trajectories and calculate the total travel time of each trajectory, as described in the previous literature [18][3].

2) Baselines: We conducted a comparative analysis involving 10 baseline methods, which can be categorized into two main groups: statistical learning-based methods [39], [40] and deep learning-based methods [13], [18], [41]–[44]. [4]

- AVG [39]. This method estimates the travel time by calculating the historical average of trajectories with the same starting/ending points during the test phase.
- LR [39]. This method trains a linear regression model to represent the relationship between taxi trajectory and travel time based on the location of the origin and destination.
- GBM [39]. This method estimates travel time using gradient boosting decision tree models, which take into account the departure time, day of the week, GPS coordinates, and taxicab geometry.
- TEMP [40]. This method estimates travel time using the average travel time of neighboring trips from a large dataset of historical data.
- WDR [41]. This method uses deep learning techniques to extract handcrafted features from raw trajectory data and incorporates information about road segments to estimate travel time.
- DeepTTE [13]. This method uses 1-D convolution and LSTM networks to estimate travel times from raw GPS trajectories in combination with some other external features.
- STNN [42]. This method uses fully connected neural networks to predict travel time by estimating the distance between an origin and destination GPS coordinate and then combining this prediction with the time of day.
- MURAT [43]. This method utilizes multi-task representation learning to improve the performance of travel time estimation by jointly learning the primary task (i.e., travel time estimation) and other auxiliary tasks (e.g., travel distance).
- Nei-TTE [44]. This method uses GRU to capture features from road network topology and speed interactions and divides the entire trajectory into multiple segments to estimate travel time.
- MetaTTE [18]. The method employs meta-learning to improve the accuracy and generalization of the travel time estimation task for multiple cities.
- DCRNN [21]. A traffic predicting model capturing the spatial dependency with bidirectional random walks on the graph.
- GCT-TTE [45]. The method applies different data modalities capturing different properties of an input path.
- TransTTE [46]. The method applies the Graphomer architecture to accelerate the training process and consider the peculiar properties of road trips.

3) Evaluation metrics: We adopt the following three metrics to evaluate the performance of models:

- Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m}\left|(Y_i - \hat{Y}_i)\right| \tag{15}$$

- Root Mean Squared Error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(Y_i - \hat{Y}_i)^2} \tag{16}$$

- Mean Absolute Percentage Error (MAPE):

$$\mathrm{MAPE} = \frac{100\%}{m}\sum_{i=1}^{m}\left|\frac{\hat{Y}_i - Y_i}{Y_i}\right| \tag{17}$$

where Y_i and $\hat{Y}_i$ are ground truth and predicted speed, and m is the number of samples. MAE and RMSE quantify the absolute size of the difference between the ground truth and the predicted value, whereas MAPE measures the relative magnitude (as a percentage). These metrics with lower values indicate superioty.

4) Configure settings: All experiments are conducted on a Linux server with Intel E5-2620v4 CPUs and GeForce RTX 2080Ti GPUs. All baselines and the proposed model are built with Pytorch 1.7.0 and Python 3.8.3.

5) Parameters settings: In our work, we set the feature embedding dimension of the input to the whole framework as $d_{\text{model}} = 32$, and the embedding size of the position $d_m = 32$. The normalization factor $d = \frac{d_{\text{model}}}{N^h} = 4$, where $N^h = 8$. The max nodes in a sub-quadrant $m = 50$ in Chengdu Dataset and $m = 25$ in Porto. The training epoch is set to 100, and we apply the early stopping strategy as follows. If the optimal performance on the validation phase has not been updated in the last 5 epochs, the model stops training and transitions to the testing phase to obtain the final results.

B. Performance Evaluation

Table III presents a comprehensive comparison of the performance of various baselines across the three datasets. Notably, GT-TTE exhibits superior performance on almost all datasets, particularly MAPE, which offers a more comprehensive representation of a model's efficacy, being less influenced by the mean or extreme values of the dataset. The best-performing method is highlighted in **Bold** and the runner-up is decorated with an underline.

The results consistently demonstrate a universal advantage of deep learning-based approaches over statistical learning-based approaches. This can be attributed to deep learning methods' widespread applicability and effectiveness in leveraging additional semantic information and capturing the inherent sequential dependencies present in the input data. Specifically, deep learning models such as Deep-TTE and Meta-TTE, equipped with LSTM architectures, exhibit superior

[4]The evaluation results for the chengdu14 and porto datasets were obtained from the study [18]. It should be noted that MetaTTE, which was trained on both datasets alternately to improve generalizability originally. To ensure experimental fairness, the results presented in Table III were obtained using the MetaTTE code provided by the authors but trained separately for each dataset.

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT APPROACHES. MAPE IS REPORTED IN PERCENTAGE (%)

	Datasets								
	Chengdu16			Chengdu14			Porto		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
AVG	124.59	234.56	30.25	442.20	8443.60	39.71	182.64	1128.21	26.66
LR	150.36	259.65	32.12	516.23	1204.99	49.09	194.40	279.20	33.90
GBM	129.65	235.25	28.65	454.50	1121.32	41.67	148.53	209.07	24.59
TEMP	128.43	221.41	27.15	334.60	761.05	39.70	174.44	260.81	28.73
WDR	120.48	228.31	29.47	433.99	1024.92	29.74	164.04	244.41	22.84
STNN	115.58	198.27	26.33	427.33	1011.88	30.08	226.30	331.75	35.44
MURAT	108.48	171.95	27.84	396.01	994.95	29.29	165.91	177.83	27.10
Nei-TTE	112.14	168.43	27.14	414.16	1038.71	30.04	106.30	183.03	15.23
DeepTTE	93.56	147.27	13.66	413.09	926.04	24.22	84.29	90.29	14.79
MetaTTE	**35.61**	**70.91**	11.10	235.89	726.13	25.53	1.79	2.50	0.34
DCRNN	38.72	120.65	12.22	270.18	543.90	23.98	1.98	2.49	0.31
GCT-TTE	46.13	73.94	11.31	202.67	458.52	19.32	<u>1.65</u>	<u>2.14</u>	<u>0.24</u>
TransTTE	39.24	71.32	<u>10.86</u>	<u>188.46</u>	<u>320.55</u>	<u>18.79</u>	1.83	2.34	0.26
GT-TTE (Ours)	<u>37.36</u>	<u>104.57</u>	**6.58**	**104.52**	**144.62**	**16.52**	**0.24**	**0.35**	**0.0355**

performance by effectively encoding and utilizing information from the initial segments of the sequences.

Meanwhile, the notable performance advantage of GT-TTE can be attributed to the inclusion of the GTE component, which augments the trajectory with localized spatial information and incorporates the sequential characteristics of the trajectory itself in the analysis. By constructing A_{odr}, GT-TTE enables information to flow between virtual nodes (regions), thereby providing a global-enhanced perspective for trajectory time estimation. The performance improvements achieved by GT-TTE over the runner-up methods are substantial, with approximately 40%, 31%, and 89% improvement observed across the three datasets[5]. Notably, the dataset Porto exhibits a particularly significant improvement with GT-TTE. To gain insights into the factors contributing to this improved performance, a further investigation was conducted on the distribution characteristics of the three datasets.

According the comparison between the graph-based models, we can see that the graph-based model type performs somewhat better than the other models. It is noteworthy that, whereas DCRNN [21], GCT-TTE [45], and TransTTE [46] employ a graph with the road segment connectivity case as an element, the region connectivity case, A_{reg}, is substituted for the graph in our instance without any further information derived from the hierarchical graph $\mathbf{H}$. While graph-based models have advanced significantly, GT-TTE has an advantage in long-distance information transfer and fusion because of the hierarchical graph $\mathbf{H}$ that GTE generated and the trajectories following the augmentation of spatial information from $\mathbf{H}$.

C. Data Analysis

Fig. 4 illustrates the distribution of total travel time, GPS points, and their ratio for three datasets (Chengdu16, Chengdu14 and Porto). (The latter two datasets were treated following the methods described in literature [18], with travel times to be 315–1174 seconds for chengdu14 and 315–945 seconds for Porto.) The first column shows the significantly different distributions of total travel time for the three datasets, suggesting variations in traffic patterns, route choices, or other factors. The second column presents the distribution of the number of GPS points contained in each trajectory, with the Porto dataset exhibiting a relatively trajectories' lengths. The third column is the logarithm of the ratio between the first two columns.

Fig. 4 depicts that the Porto dataset demonstrates a more concentrated distribution of the number of GPS points compared to the chengdu14 and chengdu16 datasets. Additionally, the logarithm of the ratio between the travel time and the number of GPS points, as shown in the third column, indicates a pronounced correlation between the travel time and the number of GPS points for the Porto dataset.

It is noteworthy that the quantity of GPS points per trajectory constitutes concealed information within track data. Most models do not utilize the sequential information of track numbers since it is presumed to be encompassed in track data. However, GT-TTE utilizes a positional encoding method to identify trajectory sequence information, which assists in predicting travel times. This is especially pertinent in the Porto dataset, where the total travel time correlates strongly with the number of trajectory records.

D. Ablation Study

We conducted a series of ablation experiments to evaluate the efficacy of our proposed graph trajectory enhancement (GTE) in improving TTE-Task and the effectiveness of Attention modules. Specifically, we decomposed our approach into the following components for analysis.

- GT-TTE(baseline): Eliminate all components and only utilize the trajectories' features contained in the raw data. The resulting model can be viewed as a fully-connected layer.

[5]Measured by the MAPE metric $\frac{S1-S0}{S1} * 100(\%)$, where $S0$ represents the performance of GT-TTE and $S1$ represents the performance of the runner-up method.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3417432

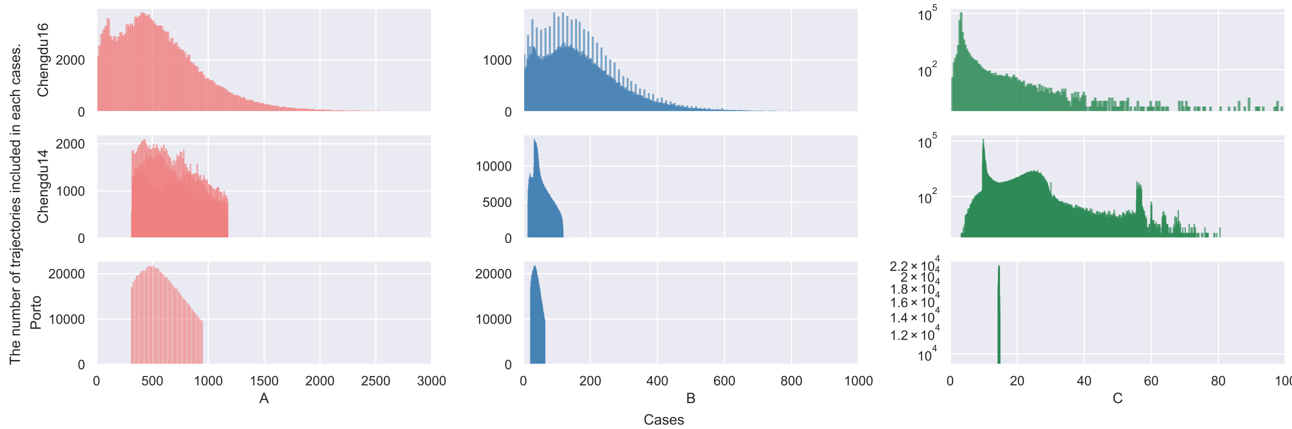


Fig. 4. Data distribution. Each row represents a dataset (Chengdu16, Chengdu14, Porto). **Column A**: the distribution of the total time (label) for trajectories. **Column B**: the distribution of the number of recorded GPS points in a trajectory. **Column C**: the distribution of the logarithm-ratio between the first two columns ($C = \log(\frac{A}{B})$).

- GT-TTE(+Attn): Incorporated an Attention mechanism into the baseline model.
- GT-TTE(+GTE): Incorporated the Contextual Trajectory Augmentation block into the baseline model.
- GT-TTE(ALL): the proposed model include GTE and Attention mechanism.

As Table IV shows, the inclusion of Attention mechanism and GTE are significantly improved the performance of the model compared to the baseline. Notably, our results with underlines indicate that GTE led to a greater improvement in model performance compared to the other methods evaluated. When these two methods are combined, the performance with **Bold** is maximized, yielding optimal results. It should be noted that the extent of improvement varies across different datasets and components. The results presented in Table IV reveal that including the GTE component significantly enhances the performance of the Chengdu16 and Porto datasets. This considerable improvement observed in the Porto dataset may be attributed to its pronounced correlation between travel time and the number of recorded points. In contrast, the Chengdu16 dataset exhibits a wide distribution compared to the Chengdu14 dataset, resulting in limited predictive effectiveness of the GT-TTE(baseline). However, given the smaller number of trajectory entries in the Chengdu16 dataset compared to the others, as indicated in Table II, the incorporation of the GT-TTE component leads to an increase in local spatial information, thus playing a crucial role in improving the model performance through the enhancement of data quality and the re-excavation of spatial relationships.

The Attn component is also crucial in aggregating the contextual information within a single trajectory, allowing for a global perspective. However, when dealing with trajectory data containing more GPS recording points (such as the Chengdu16 and Chengdu14 datasets), the expressive power of the information diminishes more fiercely as it passings through multiple layers. Consequently, relying solely on the global focus provided by the Attn component leads to modest improvements in model prediction compared to the GT-TTE(baseline), but its effectiveness remains constrained.

In contrast, the GT-TTE model, by combining the GTE and Attn components, leverage the strengths of each to overcome their respective limitations and achieve superior performance. The local enhancement of GTE provides detailed spatial information, while the global perspective of Attn ensures a comprehensive understanding of the entire trajectory. By integrating these two components, GT-TTE effectively compensates for their shortcomings and achieves the best overall performance.

E. The Influence of Hyperparameter d_H

d_H is the embedding size of features. In this subsection, we tested the experimental results with d_h varying from 8 to 128. The results are shown in Table. VI.

TABLE VI
EFFECT OF d_H ON PERFORMANCE (MAE / RMSE / MAPE)

d_H	Chengdu16	Chengdu14	Porto
8	40.84/99.93/10.29	104.06/178.90/16.64	1.89/2.36/0.30
16	38.23/98.47/7.23	115.24/190.88/18.07	0.14/0.18/0.02
32	37.36/104.57/6.58	104.52/144.62/16.52	0.24/0.35/0.03
64	87.32/129.35/12.33	170.24/204.88/29.07	39.19/50.18/7.15
128	90.68/138.23/13.68	168.09/202.75/27.12	39.54/53.96/7.18

We empirically examined the impact of the hyperparameter d_h for each dataset. The ideal d_h is 16 on the Porto dataset and 32 on the Chengdu14 and Chengdu16 datasets. This may be because the Porto dataset's distribution is more centralized and the feature representation does not require additional dimensions. We tested the experimental results with d_h varying from 8 to 128. In the Porto dataset, however, good results are still achievable when d_h equals 32. Furthermore, performance on all three datasets declines with increasing d_h, so overall, d_h equal to 32 is a suitable setting.

F. Performance on Trajectories of Different Lengths

In this section, we evaluate the effectiveness of our method on predicting travel times for trajectories with varying lengths.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3417432

TABLE IV
THE PERFORMANCE OF DIFFERENT COMPONENTS IN GT-TTE. MAPE IS REPORTED IN PERCENTAGE (%)

	Datasets								
	Chengdu16			Chengdu14			Porto		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
GT-TTE(baseline)	232.30	347.59	100.82	203.33	240.28	32.24	143.49	172.88	25.41
GT-TTE(+GTE)	47.74	124.35	9.33	136.35	182.94	22.46	0.3337	0.4042	0.0653
GT-TTE(+Attn)	234.05	354.00	80.23	170.64	205.63	27.01	39.29	50.63	7.09
GT-TTE(Ours)	**37.36**	**104.57**	**6.58**	**104.52**	**144.62**	**16.52**	**0.2434**	**0.3508**	**0.0355**

TABLE V
THE PERFORMANCE OF DIFFERENT MODELS ON VARYING LENGTH IN TRAJECTORIES (SHOWN IN MAE / RMSE / MAPE(%))

Seconds Range	DeepTTE	MetaTTE	GT-TTE
Short(<315s)	89.04 / 92.96 / 52.72	196.74 / 283.99 / 59.13	10.44 / 21.29 / 9.26
Medium(315s-1174s)	198.12 / 212.36 / 28.31	147.42 / 182.68 / 25.18	35.92 / 71.65 / 5.42
Long (>1174s)	236.22 / 306.02 / 13.27	219.66 / 324.03 / 17.07	190.41 / 315.99 / 10.63
Total	93.56 / 137.27 / 13.66	35.61 / 70.91 / 11.1	37.36 / 104.57 / 6.5

Specifically, the trajectories were categorized into three distinct groups as regards their total travel time: Short trajectories, characterized by a total duration of fewer than 315 seconds; Medium trajectories, which fell within the range of 315 to 1174 seconds; and Long trajectories, with a total duration exceeding 1174 seconds [13], [18]. Considering the more comprehensive data available in the chengdu16 dataset compared to the other two, we conduct experiments on the chengdu16 dataset and compare our GT-TE with MetaTTE and DeepTTE, demonstrating the second and third highest MAPEs in Table. III, respectively.

Table V exhibits the performance of three models in terms of MAE, RMSE and MAPE, when varying the time of trajectories. It is obciously that DeepTTE and MetaTTE exhibit relatively promising performance only in case of long-trajectory data, while their performance on short and medium trajectories is limited. Additionally, it is noteworthy that their optimal performance can be achieved only when utilizing the entire trajectory dataset. In contrast, GT-TTE demonstrates exceptional predictive performance for trajectories of varying total time length, particularly notable improvements observed for medium trajectories. The prevalence of medium trajectories in everyday life underscores the usefulness of GT-TTE, while its ability to achieve optimal performance across trajectories of varying time lengths highlights its generalizability. The superior performance of GT-TTE across diverse trajectory lengths can be attributed to the GTE module's capability to enrich local information, coupled with the Attn component's handling of the trajectory's global characteristics.

G. Performance on complexity

For evaluating the model's complexity, we employ floating point operations. The table. VII demonstrates that GT-TTE has a limited number of parameters and extremely low FLOPs. It should be noted that we only count the number of parameters and FLOPs during the inference process, the data enhancement method carried out by the GTE module is not included because it can be done just once and then reused repeatedly. We only use the encoder-only mode throughout the inference procedure.The following table displays the number of layers that are included in GT-TTE.

TABLE VII
THE COMPLEXITY AND PARAMETERS OF SEVERAL METHODS.

	FLOPs(M)			Parameters(K)
	Chengdu16	Chengdu14	Porto	
DeepTTE	1532	726	505	295
DCRNN	3728	1453	954	379
GCT-TTE	1011	682	579	342
TransTTE	987	569	344	228
GT-TTE	347	118	87	33

TABLE VIII
THE NUMBER OF MODEL LAYERS.

3*Embedding_linear_layer	
3*Encoder	2*GCN_layer
	1*Attention_layer
	3*Linear_layer
1*Out_linear_layer	

VI. CONCLUSION

In this paper, we propose a Graph-Transformer for Travel Time Estimation (GT-TTE) based on trajectories augmented with spatio-temporal information. Specifically, we introduce the Graph Trajectory Enhancement (GTE) technique to learn the trajectory as a graph, amplify the regional characteristics of the recorded points, and produce a sequence of embedding vectors for the regions to differentiate spatial information and capture attributes in the local domain. Moreover, to preserve the global features of the trajectories, we employ an attention mechanism that allows each trajectory to attend to its own global recording points. Furthermore, we also constructed a regional adjacency matrix based on the GTE method in addition to the spatial features. We incorporated the matrix with the positional encoding to offer the graphical spatio-temporal adjacency data of the trajectory.

To assess the effectiveness of GT-TTE, we conducted a series of comprehensive case studies on three datasets based on real-world scenarios. The simulation results indicate that the GT-TTE approach significantly outperforms the state-of-the-art baselines. A dataset distribution analysis shows that GT-TTE can significantly improve the accuracy of trajectory time prediction by abstracting order and temporal information from the datasets. Additionally, we performed ablation experiments on the GT-TTE approach. The results of these experiments revealed that the proposed GTE augmentation method has a more significant impact on increasing prediction accuracy than the attention mechanism. Lastly, we analyzed the prediction performance of the DeepTTE, MetaTTE, and GT-TTE models across different travel times. The results highlight the effectiveness of GT-TTE in medium trajectory time prediction and its generalization capabilities in predicting the performance of all trajectories. While offering enhanced spatial information, the construction of trajectory graphs remains a complex undertaking. The efficiency and ease with which additional spatio-temporal information can be incorporated play a crucial role in achieving more accurate trajectory time predictions. In our future research, we will delve deeper into this aspect to explore methods that facilitate the quick and effective application of such information.

References

[1] Z. Zhang, H. Wang, Z. Fan, J. Chen, X. Song, and R. Shibasaki, "Gof-tte: Generative online federated learning framework for travel time estimation," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24 107–24 121, 2022.

[2] J. Xu, Y. Gao, C. Liu, L. Zhao, and Z. Ding, "Efficient route search on hierarchical dynamic road networks," *Distributed and Parallel Databases*, vol. 33, no. 2, pp. 227–252, 2015.

[3] Y. Kisialiou, I. Gribkovskaia, and G. Laporte, "The periodic supply vessel planning problem with flexible departure times and coupled vessels," *Computers & Operations Research*, vol. 94, pp. 52–64, 2018.

[4] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, "T-finder: A recommender system for finding passengers and vacant taxis," *IEEE Transactions on knowledge and data engineering*, vol. 25, no. 10, pp. 2390–2403, 2012.

[5] D. Yao, H. Hu, L. Du, G. Cong, S. Han, and J. Bi, "Trajgat: A graph-based long-term dependency modeling approach for trajectory similarity computation," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2275–2285.

[6] Y. Huang, X. Song, S. Zhang, and J. James, "Transfer learning in traffic prediction with graph neural networks," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 3732–3737.

[7] D. Billings and J.-S. Yang, "Application of the arima models to urban roadway travel time prediction-a case study," in *2006 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3. IEEE, 2006, pp. 2529–2534.

[8] A. Guin, "Travel time prediction using a seasonal autoregressive integrated moving average time series model," in *2006 IEEE Intelligent Transportation Systems Conference*. IEEE, 2006, pp. 493–498.

[9] B. Yu, H. Wang, W. Shan, and B. Yao, "Prediction of bus travel time using random forests based on near neighbors," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 4, pp. 333–350, 2018.

[10] B. Gupta, S. Awasthi, R. Gupta, L. Ram, P. Kumar, B. Rohit Prasad, and S. Agarwal, "Taxi travel time prediction using ensemble-based random forest and gradient boosting model," in *Advances in Big Data and Cloud Computing*. Springer, 2018, pp. 63–78.

[11] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[12] T. Robinson, M. Hochberg, and S. Renals, "The use of recurrent neural networks in continuous speech recognition," *Automatic speech and speaker recognition*, pp. 233–258, 1996.

[13] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, "When will you arrive? estimating travel time based on deep neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

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

[15] 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.

[16] 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.

[17] Q. Wang, C. Xu, W. Zhang, and J. Li, "Graphtte: Travel time estimation based on attention-spatiotemporal graphs," *IEEE Signal Processing Letters*, vol. 28, pp. 239–243, 2021.

[18] C. Wang, F. Zhao, H. Zhang, H. Luo, Y. Qin, and Y. Fang, "Fine-grained trajectory-based travel time estimation for multi-city scenarios based on deep meta-learning," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[19] Y. Su, C.-C. J. Kuo *et al.*, "Recurrent neural networks and their memory behavior: a survey," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[20] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

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

[22] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 922–929.

[23] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proceedings of the web conference 2020*, 2020, pp. 2704–2710.

[24] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," *Advances in neural information processing systems*, vol. 32, 2019.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[26] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in neural information processing systems*, vol. 32, 2019.

[27] M. Xu, W. Dai, C. Liu, X. Gao, W. Lin, G.-J. Qi, and H. Xiong, "Spatial-temporal transformer networks for traffic flow forecasting," *arXiv preprint arXiv:2001.02908*, 2020.

[28] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, "Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting," *Transactions in GIS*, vol. 24, no. 3, pp. 736–755, 2020.

[29] F. Liu, J. Yang, M. Li, K. Wang *et al.*, "Mct-tte: Travel time estimation based on transformer and convolution neural networks," *Scientific Programming*, 2022.

[30] G. Jin, M. Wang, J. Zhang, H. Sha, and J. Huang, "Stgnn-tte: Travel time estimation via spatial–temporal graph neural network," *Future Generation Computer Systems*, vol. 126, pp. 70–81, 2022.

[31] J. Ma, J. Chan, S. Rajasegarar, and C. Leckie, "Multi-attention graph neural networks for city-wide bus travel time estimation using limited data," *Expert Systems with Applications*, vol. 202, p. 117057, 2022.

[32] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, 2015.

[33] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," in *International conference on machine learning*. PMLR, 2015, pp. 2067–2075.

[34] H. Samet, "An overview of quadtrees, octrees, and related hierarchical data structures," *Theoretical Foundations of Computer Graphics and CAD*, pp. 51–68, 1988.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3417432

[35] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[36] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, "Graph contextualized self-attention network for session-based recommendation." in *IJCAI*, vol. 19, 2019, pp. 3940–3946.

[37] L. Wang, Z. Song, X. Zhang, C. Wang, G. Zhang, L. Zhu, J. Li, and H. Liu, "Sat-gcn: Self-attention graph convolutional network-based 3d object detection for autonomous driving," *Knowledge-Based Systems*, vol. 259, p. 110080, 2023.

[38] Y. Duan, N. Chen, S. Shen, P. Zhang, Y. Qu, and S. Yu, "Fdsa-stg: Fully dynamic self-attention spatio-temporal graph networks for intelligent traffic flow prediction," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9250–9260, 2022.

[39] W. Lan, Y. Xu, and B. Zhao, "Travel time estimation without road networks: an urban morphological layout representation approach," *arXiv preprint arXiv:1907.03381*, 2019.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

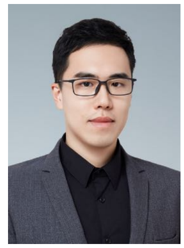
[44] J. Qiu, L. Du, D. Zhang, S. Su, and Z. Tian, "Nei-tte: intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2659–2666, 2019.

[45] V. Mashurov, V. Chopuryan, V. Porvatov, A. Ivanov, and N. Semenova, "Gct-tte: graph convolutional transformer for travel time estimation," *Journal of Big Data*, vol. 11, no. 1, pp. 1–14, 2024.

[46] N. Semenova, V. Porvatov, V. Tishin, A. Sosedka, and V. Zamkovoy, "Logistics, graphs, and transformers: Towards improving travel time estimation," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2022, pp. 589–593.

Yunjie Huang received the B.Eng degree from the Northeastern University at Qinhuangdao, China in 2020 and M.S. degree in Computer Science and Engineering from Southern University of Science and Technology, Shenzhen China in 2023. She is currently pursuing the Ph.D. degree with the The Hong Kong University of Science and Technology (Guangzhou) at Guangzhou, China. Her research interests include Learn to Routing, deep learning in smart city and intelligent transportation systems.

Xiaozhuang Song received the B.Eng. degrees in digital media technology from Beijing University of Posts and Telecommunications, Beijing, China in 2015, and the M.S degree in computer science and technology from the Southern University of Science and Technology, Shenzhen, China in 2022. He is currently pursuing the Ph.D. degree at The Chinese University of Hong Kong, Shenzhen. His research interests include multitask learning, graph neural networks, and intelligent transportation systems.



Shiyao Zhang (S'18-M'20) received the B.S. degree (Hons.) in Electrical and Computer Engineering from Purdue University, West Lafayette, IN, USA, in 2014, the M. S. degree in Electrical Engineering from University of Southern California, Los Angeles, CA, USA, in 2016, and the Ph.D. degree from the University of Hong Kong, Hong Kong. He was a Post-Doctoral Research Fellow with the Academy for Advanced Interdisciplinary Studies, Southern University of Science and Technology from 2020 to 2022. He is currently a Research Assistant Professor with the Research Institute for Trustworthy Autonomous Systems, Southern University of Science and Technology. His research interests include smart cities, smart energy systems, intelligent transportation systems, optimization theory and algorithms, and deep learning applications.

Lei Li is an Assistant Professor at the Hong Kong University of Science and Technology (Guangzhou). He obtained his Ph.D. in 2018 from the University of Queensland under Prof. Xiaofang Zhou's supervision. He obtained his Bachelor's and Master's degrees from Harbin Institute of Technology in 2012 and 2014, respectively. His research interests include spatial and temporal data, graph, and distributed databases.

James Jianqiao Yu (S'11–M'15–SM'20) received the B.Eng. and Ph.D. degree in Electrical and Electronic Engineering from the University of Hong Kong, Pokfulam, Hong Kong, in 2011 and 2015, respectively. He was a post-doctoral fellow at the University of Hong Kong from 2015 to 2018, and was an assistant professor at the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China from 2018 to 2023. He is currently a Lecturer at the Department of Computer Science, the University of York, United Kingdom. His general research interests are in intelligent transportation systems, privacy computing, deep learning, and smart cities. His work is now mainly on spatial-temporal data mining, forecasting and logistics of future transportation systems, and artificial intelligence techniques for industrial applications. He has published over 100 academic papers in top international journals and conferences, and representative papers have been selected as ESI highly cited papers. He was the World's Top 2% Scientists from 2020 to 2023 and of career by Stanford University. He is an Editor of the IET SMART CITIES journal and a Senior Member of IEEE.