TransGTE: a transformer-based model with geographical trajectory embedding for the individual trip destination prediction

Expert Systems With Applications 287 (2025) 128159

Contents lists available at ScienceDirect

Expert Systems With Applications

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

TransGTE: a transformer-based model with geographical trajectory embedding for the individual trip destination prediction

Zhenlin Qin[a], Pengfei Zhang[b], Qi Zhang[c], Kun Gao[d], Zhenliang Ma[a,*]

[a] Transport Planning Division, KTH Royal Institute of Technology, Stockholm, 10044, Sweden
[b] Henan Academy of Sciences, Institute of Physics, Zhengzhou, 450000, Henan, PR China
[c] School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan, 430063, China
[d] Department of Architecture and Civil Engineering, Chalmers University of Technology, Gothenburg, 41296, Sweden

ARTICLE INFO

Keywords:
Destination prediction
Graph convolutional network
Adaptive neural gating mechanism
Geographical trajectory embedding

ABSTRACT

Destination prediction is an essential problem for many location-based applications and services. Although previous works partly solved the sparsity of GPS location data by methods such as discretization and embedding, the problem of properly extracting and utilizing geographical information of trajectories is still unsolved. The paper proposes the TransGTE model, a Transformer-based framework with a novel geographical embedding and fusion mechanism, to adaptively extract and fuse geographical features with trajectories' sequential patterns. TransGTE uses the Graph Convolutional Network (GCN) and Transformer to extract geographical and sequential features and adopts a dynamic gating mechanism to control the weights of sequential and geographical information adaptively. We perform extensive experiments on four taxi trajectory real-world datasets from Porto, Chengdu, Shenzhen and San Francisco, where the TransGTE averagely outperform the best benchmark models by 4.24 %, 2.87 %, 5.91 % and 4.11 % in terms of the Mean Haversine Distance Error. The ablation study validates the effectiveness of the proposed trajectory location representation and dynamic gating mechanism modules used to embed taxi GPS trajectories. Finally, we compare the proposed trajectory embedding with the commonly used transformer-based model, and it highlights the effectiveness of the proposed embedding approach in representing geographical similarities between trajectories. The code for this paper is available at: https://github.com/qzl408011458/TransGTE.

1. Introduction

With the rapid development of intelligent transportation systems, massive trajectory data have been collected from different sources, and one example is the taxi trajectory data from GPS sensors. The taxi GPS trajectories provide the foundation for a spectrum of applications related to urban mobility. Particularly, destination prediction, i.e., predicting the trip destination given a partially realized trajectory of a whole trip up to a certain time, is a fundamental problem supporting many beneficial applications such as POI recommendation (Feng et al., 2015; Yin, Wang, Wang, Chen, & Zhou, 2017), sharing mobility (Hu & Creutzig, 2022), and route recommendations (Cui, Luo, & Wang, 2018; Dai, Yang, Guo, & Ding, 2015).

The destination prediction problem has three major challenges: data sparsity, trajectory sequential patterns, and geographical relationships between trajectories. The data sparsity refers to that vehicle trajectories (e.g., taxi GPS (Moreira-Matias, Gama, Ferreira, Mendes-Moreira, & Damas, 2013) or human mobility trajectories (Zheng, Xie, & Ma, 2010)) are impossible to cover all possible spaces in a city (Wang, Wang, Ku, Cheng, & Guo, 2017; Xue et al., 2015). To extract location-related features from the sparse data, most studies represent locations as grids, i.e., splitting the studied area into grids and assigning GPS coordinates of a trajectory to corresponding grids (Endo, Nishida, Toda, & Sawada, 2017; Manasseh & Sengupta, 2013; Pecher, Hunter, & Fujimoto, 2016), then transforming grids to dense representations through the embedding process to extract context semantics (Zhao et al., 2018). Given the trajectory discretization and embedding, deep recurrent neural networks (such as LSTM (Ebel, Gol, Lingenfelder, & Vogelsang, 2020; Li, Cui, Zhang, Liu, & Song, 2021)) are commonly used to model sequential patterns (e.g., regularity and sequential dependencies). Such modeling pipeline achieves a good performance in representing the temporal information. However, it has limited capability in capturing dynamic geographical relationships between trajectories depending on trip contexts. For example, assuming that two GPS trajectories close in geography are converted to a discrete format (e.g., grid sequence), they may have entirely different representations but actually close distanceswhen they are assigned to the grids

[*] Corresponding author.
E-mail address: zhema@kth.se (Z. Ma).

https://doi.org/10.1016/j.eswa.2025.128159
Received 2 October 2024; Received in revised form 21 April 2025; Accepted 11 May 2025
Available online 14 May 2025

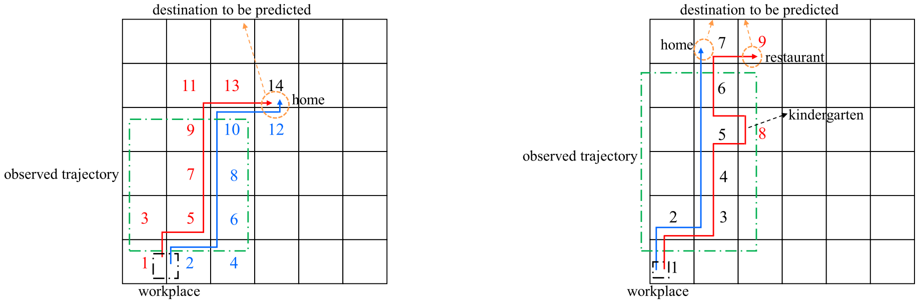


Fig. 1. Example of the different utility of geographical information. The trajectories in Fig. 1(a) have pretty different representations but are similar in reality, so geographical similarity positively affects the prediction in this case. In Fig. 1(b), the difference in visited grids 5 and 8 results in different destinations, in which case incorporating the geographical similarity of grids 5 and 8 has a negative effect on the prediction.

using distinct indices. The conventional representation model only considers the sequence information of the trajectory grids and thus treats these trajectories as totally different travel patterns, which may eventually deteriorate the destination prediction performance.

Modeling the geographical relationships between trajectories is challenging in the context of the destination prediction problem. First, it is nontrivial to model the geographic proximity between grids (i.e., GPS locations), which is not directly observed from the grid sequence. Most models (Ebel et al., 2020; Zhao et al., 2018) are based on NLP technologies (Jatnika, Bijaksana, & Suryani, 2019; Jin, Zhang, & Liu, 2018). They can mine statistical correlations of GPS locations from the context but cannot effectively model the geographic proximity, such as neighboring relationships between grids for the two trajectories as shown in Fig. 1(a). Second, indiscriminately using geographical similarity could mislead the prediction model training without considering critical sections of the studied trajectory. For example, Fig. 1(b) shows that grids 5 and 8 are adjacent, however, they should not be treated as similar based on the trip context. The visited grid 8 indicates picking up a child from a kindergarten, which most likely ends up at a different destination (eating outside by visiting grid 9) instead of directly going home (visiting grid 7). In this case, it is misleading to incorporate geographical similarity between grids 5 and 8, i.e., considering grids 5 and 8 as the same pattern based on their close distance. In summary, the geographical features should be represented and modeled in a dynamic manner in the destination prediction depending on trip contexts.

To address these gaps and challenges in taxi trajectory destination prediction, we propose a novel deep learning-based destination prediction model, i.e., TransGTE, which uses Graph Convolutional Network (GCN) models to extract geographical features and Transformer models to extract sequential features. We also propose a dynamic neural gating mechanism to adaptively/dynamically fuse sequential and geographical features by controlling the utilization of geographical features depending on trip contexts in predicting trip destinations. Key contributions include:

- Propose a Transformer-based trip destination prediction model (TransGTE) to predict the trip destination coordinates given the partially realized GPS trajectory up to the prediction time. It simultaneously considers sequential and geographical similarities of trajectories.
- Propose a context-aware trajectory embedding model that uses the GCN to capture the neighboring information of a given trajectory and also a gating model to dynamically fuse geographical and sequential features depending on the trip contexts extracted by the GCN and Transformer models.
- Conduct systematic experiments to validate the model performance in predicting the destination coordinates using four real-world datasets and comparing with state-of-art benchmark models. Ablation studies are also performed to understand model component contributions and illustrate the proposed GPS trajectory embedding method in representing trajectory similarities.

The remainder of the paper is organized as follows: Section 2 reviews the related approaches in the literature. Section 3 defines the studied problem and proposes the destination prediction methodology, including representing GPS trajectory, modeling sequential patterns, and predicting the trip destination. Case studies using open-source benchmark datasets are presented in Section 3.2. The final section concludes the main findings and discusses future work.

2. Related work

The studied problem focuses on the trip destination prediction using GPS trajectories. The relevant studies in the literature include the individual mobility prediction and trip destination prediction, which are reviewed separately in the paper. Also, we focus on reviewing studies using GPS trajectory data from the modeling perspective. For a detailed review of individual mobility and trip destination prediction problem, please refer to Ma and Zhang (2022).

2.1. Individual mobility prediction

Individual mobility prediction is a task to predict the next location (or/and the time) by mining mobility patterns from individual travel records. Early studies used Markov models (Gambs, Killijian, & Del Prado Cortez, 2012; Lu, Wetter, Bharti, Tatem, & Bengtsson, 2013; Thiagarajan et al., 2009) to model location transition probabilities, i.e., estimating a candidate location's probability that an individual will visit next. Recently, various deep learning-based prediction models have been proposed to tackle the task, and Recurrent Neural Networks(RNN) is a commonly used model. For example, Liu, Wu, Wang, and Tan (2016) extended the RNN by using the time-specific and distance-specific transition matrices to model local temporal and spatial contexts in each RNN layer. Jiang et al. (2018) utilized the urban ROI labels and constructed a deep-sequence learning model with RNN to predict the next possible destination. Most recently, the long short-term memory (LSTM) and gated recurrent unit (GRU) have been used for human mobility prediction (Chen et al., 2020; Yang, Sun, Zhao, Liu, & Chang, 2017). The attention mechanisms are commonly used in the time sequence prediction task. For example, Feng et al. (2018) proposed a prediction model DeepMove based on the seq2seq framework to capture the multi-level periodicity for mobility prediction from lengthy and sparse trajectories. Following that, Chen et al. (2020) incorporated the GCN model into the DeepMove to capture the spatial dependence of individual mobility. Compared to statistical models, deep learning models tend to achieve a better performance in individual mobility prediction. Recently, Zhang, Koutsopoulos, and Ma (2023) formulates a different problem and proposes a DeepTrip model to predict the next trip information with arbitrary prediction times (the time when the prediction is made). It also

Table 1
Comparison between the most related methods and TransGTE.

Model	Sequential pattern podel	Location feature extraction	Main input type	Prediction category
MLP (De Brébisson, Simon, Auvolat, Vincent, & Bengio, 2015)	Fully connected netowrk	–	Raw GPS	Coordinate
Multi-Input (Ebel et al., 2020) LSTM	LSTM	Embedding of k-d tree grid index	GPS grid	Coordinate
T-CONV (Lv, Sun, Li, & Moreira-Matias, 2019)	Multi-layer CNN	CNN	Raw GPS	Coordinate
TALL (Zhao et al., 2018)	Bi-LSTM	Embedding of $k \times k$ grid index, $k \in \mathbb{N}^*$	GPS grid	Grid index
GPS-embedding BiLSTM (Liao et al., 2022)	Bi-LSTM	lnglat and Quadtree grid index	GPS grid and Grid-lnglat	Coordinate
DeepMove (Feng et al., 2018)	GRU	Embedding	Check-in data	Location index
MobTCast (Xue et al., 2021)	Transformer	Embedding	Check-in data	Location index
MHSA (Hong et al., 2023)	Transformer	Embedding	Check-in data	Location index
TransGTE	Transformer	GCN-based GPS embedding with a gating mechanism	GPS grid	Coordinate

proposes a novel overlapped embedding method to represent continuous travel attributes capturing simultaneously the categorical and numerical feature information. Moreover, several studies in individual's next location prediction adopt a Transformer backbone to capture mobility features from check-in data (Hong, Zhang, Schindler, & Raubal, 2023; Xue, Salim, Ren, & Oliver, 2021), which shows the potential of the Transformer model applied to mobility-related tasks.

2.2. Destination prediction

Given a partially realized trajectory as a query, the destination prediction task is to predict the location where the passenger will go. Most studies model the spatiotemporal patterns of a given GPS trajectory to match other trajectories in the database with similar patterns and known destinations to infer the destination. These methods partition the space of all trajectories into discrete grids (Krumm & Horvitz, 2006; Wei, Zheng, & Peng, 2012) and assign each GPS point win index or id. However, they tend to have low prediction accuracy due to the data sparsity issue from the partition. Some improved methods are proposed to solve the sparsity problem, which considers the difference between destinations (Li, Li, Gong, Zhang, & Yin, 2016) or split trajectories into sub-trajectories to link geographically close locations (Xue et al., 2015). Also, some studies apply clustering algorithms to extract centralized points with highly visited frequencies. For example, Alvarez-Garcia, Ortega, Gonzalez-Abril, and Velasco (2010) extracted important spatial points by clustering and used the HMM model to model the relationship between these points. Yang, Xu, Xu, Zheng, and Chen (2014) identified the specific features of stay points where people stay a certain time for some activities using a variant of the DBSCAN clustering algorithm and then used the ordered Markov Model to predict the next locations. Besse, Guillouet, Loubes, and Royer (2018) used the Gaussian mixture model to cluster historical trajectories into several clusters and assigned a new trajectory to the cluster that it most likely belongs to. The final destination is predicted by using the characteristics extracted from these clusters.

In recent years, deep learning techniques have been developed rapidly and applied to the destination prediction task. The deep learning-based model proposed by De Brébisson et al. (2015) performs the best in the ECML/PKDD 15 competition. It feeds the first k points from the origin and the last k points close to the destination of a given query trajectory into Multi-Layer Perceptron (MLP). Zhang, Zhao, Zheng, and Li (2020) used raw GPS data and applied an ensemble learning approach for destination prediction. Some studies used CNN to extract spatial features in predicting the destination, such as Lv, Sun, Li, and Moreira-Matias (2020) and Zhang, Zhang, Liang, and Ozioko (2018). These models transform GPS trajectories into multi-resolution images and extract multi-scale spatial features by CNN. However, the CNN model can not differentiate trajectories with similar shapes and long distances due to its translation invariance (Le et al., 2010). RNN-based models are also reported in predicting the trajectory destination.

To extract rich semantics of locations, the deep learning based approaches usually transform sparse GPS coordinates into discrete grids, and then obtain low-dense vectors by embedding the grids. For example, Ebel et al. (2020) adopted a k-d tree-based space discretization to map GPS locations to discrete regions and used the LSTM to model the sequential pattern of trajectories. Endo et al. (2017) used a grid-based discretization method to convert trajectories into a grid space and used an RNN encoder-decoder model to predict the visiting probabilities of candidate locations for moving objects. Zhao et al. (2018) also adopted a grid-based discretization and proposed an LSTM-based model named 'TALL' in predicting destinations for exploring meaningful mobility patterns with different spatial grid granularities. Liao et al. (2022) hierarchically partitioned the city into grids and used the 'Grid-lnglat' (using the centric coordinate to represent a GPS point) and 'Quadtree' (a method of GPS discretization with multiple spatial scales) embedding to represent raw GPS trajectories. Then, the attention-based dual BiLSTMs neural network is used to model the relationship between the heading destination and the bidirectional sequential context of visited locations. Besides, Transformer-based models also are developed for mobility-related prediction tasks, like origin-destination demand prediction (Huang et al., 2023; Li et al., 2024; Lin, He, Liu, Gao, & Qu, 2023), traffic flow prediction (Tian, Wang, Hu, & Ma, 2023). Such models are usually combined with other spatial feature extraction method, such as adding road network information, to extract the spatio-temporal patterns in the observed traffic data. However, most studies used cluster-based methods to narrow down destination candidates when the prediction task aims to output GPS coordinates of the destination (De Brébisson et al., 2015; Ebel et al., 2020,?; Liao et al., 2022; Lv et al., 2020; Zhang et al., 2018). The selected clustering algorithm may impact the model prediction performance and thus requires significant efforts to tune clustering parameters which tend to be subjective and overfitting.

In summary, deep learning based methods outperform conventional methods in the mobility related prediction tasks. We summarize the main features of the methods closely related to our study in Table 1. It shows that the recurrent network structures with GPS discretization are commonly used in existing studies. The key reason to use grid embedding (not raw GPS locations) is due to the sparseness of the GPS data in both time and space while a huge continuous learning space in the model. The grid embedding can significantly narrow the learning space for an efficient model training (Ebel et al., 2020; Liao et al., 2022; Zhao et al., 2018) and surpass the methods directly using the raw data (De Brébisson et al., 2015). Although existing works achieved a good performance in mobility prediction, very limited studies were found in effectively tackling with the dilemma of how to control the extraction of geographical information to adapt to different prediction scenarios like the example shown in Fig. 1. The study absorbs some existing techniques (such as GPS discretization) for agile modeling and proposes a

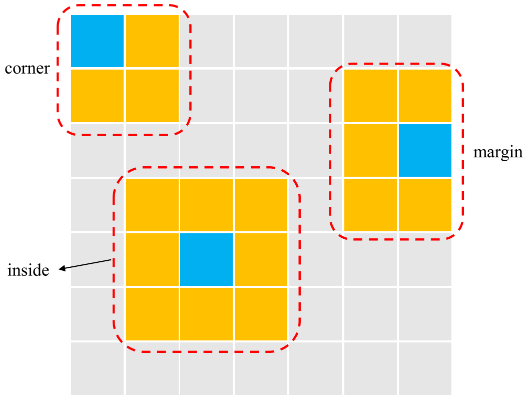


Fig. 2. Examples of grids' neighboring relationships. There are mainly three types of neighboring relationships given that the (blue) grids are located at the corner, margin, and inside of the grid network. Each blue grid is enclosed by the surrounding orange grids, counted as the number of the blue grid's edges.

Transformer-based trip destination prediction model to realize dynamical extraction and fusion of the features of geographical information and sequential patterns.

3. Methodology

3.1. Problem definition

To formally define the problem, we first introduce the following definitions.

Definition 1. Taxi GPS Trajectory. A complete taxi GPS trajectory is represented as $S^m = \{l_1, l_2, \dots, l_m\}$, where the i-th GPS point $l_i = (lng_i, lat_i)$ includes longitude (lng_i), latitude (lat_i). A partial trajectory of S^m is defined as $S^{m_r} = \{l_1, l_2, \dots, l_{m_r}\}$, where $r \in [0.1, 1)$ is the completion ratio of partial trajectory, $m_r = round(r \times m)$ and $round(\cdot)$ the operation to obtain an integer.

Definition 2. Temporal Attributes. Given a taxi GPS trajectory $S^m = \{l_1, l_2, \dots, l_m\}$, its corresponding time logs are $\{t_1, t_2, \dots, t_m\}$. Each time log t_i consists of a specific time point and date, such as $(11:00, 17/10/2013)$. To represent the temporal pattern of the trajectory, we use the departure time t_1's hour h and week w (transformed by the date of t_1) as the temporal attributes fed into the prediction model.

Definition 3. Grids. We partition the city into $Z = q \times k$ grids and assign a unique index to each grid $L \in [1, 2, \dots, Z]$. Then a partial taxi trajectory can be transformed into a grid trajectory $S_L^{m_r} = \{L_1, L_2, \dots, L_{m_r}\}$, where $L_i \in \mathbb{N}^*$ is the i-th grid with GPS point $l_i(lng_i, lat_i)$.

Definition 4. Grid Network. The grid network is represented as an unweighted graph $G = (V_G, E_G, A)$, where $V_G = \{v_1, v_2, \dots, v_Z\}$ is the set of grids, E_G the collection of edges, and $A \in \mathbb{R}^{Z \times Z}$ the adjacency matrix. There are three types of neighboring relationships in the grid network, including corner, margin, and inside as shown in Fig. 2, in which the blue grids have 3, 5, and 8 edges, respectively, linking to neighboring grids (orange ones). For each grid pair of (L_i, L_j), $A(i, j)$ equals 1 if there is an edge between them; 0 otherwise.

Definition 5. Trajectory Location Vector. Given a partial trajectory S^{m_r}, the corresponding Trajectory Location Vector TLV is constructed from its grid trajectory $S_L^{m_r}$: $TLV = [\sum_{i=1}^{m_r} O(L_i)]$, where $O(\cdot)$ is the one-hot encoding (Harris & Harris, 2012) transforming a grid index L_i into a vector $O(L_i) \in \mathbb{R}^{1 \times Z}$. For example, Fig. 3 shows that, given a grid network with $L \in [1, 2, \dots, 9]$ and a partial trajectory $S_L^{m_r} = \{1, 1, 1, 2, 5, 5, 6\}, TLV = [3, 1, 0, 0, 2, 1, 0, 0, 0], TLV \in \mathbb{R}^{1 \times Z}$ maps the trajectory $S_L^{m_r}$ into Grid Network.

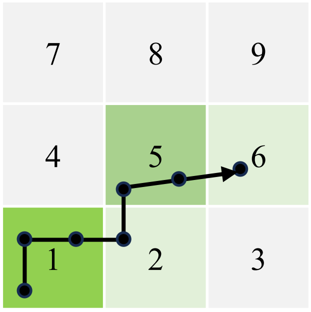


Fig. 3. An example of the Trajectory Location Vector TLV. The green grids represent the visited grids in the trajectory $S_L^{m_r} = \{1, 1, 1, 2, 5, 5, 6\}$, and the gray grids denote other grids out of the trajectory. The green color shade is positively related to how many times the corresponding grid appears in the trajectory. By the Definition 5, TLV equals to $[3, 1, 0, 0, 2, 1, 0, 0, 0]$ representing where the trajectory is in Grid Network.

The studied problem is to predict a taxi passenger's destination in a trip, formally defined as: Given a grid network G, a partial trajectory S^{m_r} and its temporal attributes t, w, predict the trip destination location $l_{\hat{y}}(lng_{\hat{y}}, lat_{\hat{y}})$.

3.2. Methodology framework

Fig. 4 shows the proposed TransGTE framework. It consists of three major modules: the Geographical Feature Extraction, the Sequential Pattern Model and the Prediction Module. The Geographical Feature Extraction module transforms a taxi GPS trajectory $S^m = \{l_1, l_2, \dots, l_m\}$ into a grid trajectory $S_L^{m_r} = \{L_1, L_2, \dots, L_{m_r}\}$, embeds each GPS point of the grid trajectory. Additionally, the corresponding temporal attributes are embedded into dense vectors to extract the trajectory's temporal patterns. Graph Convolutional Network (GCN) takes Trajectory Location Vector TLV and adjacency matrix A as inputs to extract the geographical features V_L^G. And sequential features are extracted through the Sequential Pattern model. The Adaptive Neural Fusion Gate (ANFG) links the sequential and geographical features, and dynamically fuses them to generate a context-aware trajectory representation. Finally, the Prediction Module takes inputs of the results from Sequential Pattern Model and temporal embedding vectors, and outputs the predicted destinations.

Compared with the recurrent network structures applied in previous works, we used the Transformer as the Sequential Pattern Model which is proven having better capacity to model most of time series prediction tasks (Xue & Salim, 2021; Zhou et al., 2021) and NLP tasks (Devlin, Chang, Lee, & Toutanova, 2018; Wang et al., 2019) given its self-attention mechanism. Also, conceptually the GPS trajectory patterns exhibit characteristics of both time series and text-like semantic context (after GPS discretization). It is promising and reasonable to model the problem using the Transformer structure to capture the sequential and semantic pattern features of GPS trajectories. However, the prediction model can still not be aware of the geographical information. To the best of our knowledge, we firstly applied GCN in capturing the geographical features of GPS trajectory in trip destination prediction. GCN is commonly employed in other traffic prediction tasks (Sun, Jiang, Lam, & He, 2022; Zhao et al., 2020) like traffic flow prediction to capture neighboring flow information of nodes of road network since it constructs a filter transferring the operations on a graph to the Fourier domain to aggregating spatial features between nodes using its first-order neighborhood information. Different from such applications, GCN in our study only captures the neighboring information of the given GPS trajectory, not the network nodes. Besides, the previous methods in traffic flow prediction usually used fully-connected networks to directly fuse spatial/geographical features (extracted by GCN) and sequential features (the hidden vectors in RNN models) from the historical traffic volumes

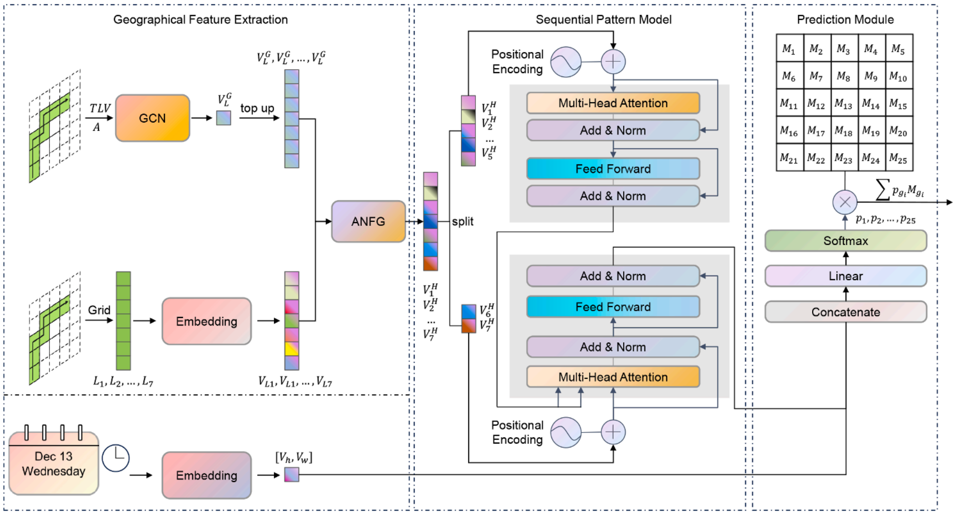

Geographical Feature Extraction
Sequential Pattern Model
Prediction Module
TLV
A
GCN
V_L^G
top up
$V_L^G, V_L^G, \ldots, V_L^G$
ANFG
Grid
Embedding
$L_1, L_2, \ldots, L_7$
$V_{L1}, V_{L1}, \ldots, V_{L7}$
split
V_1^H V_2^H … V_7^H
V_1^H V_2^H … V_5^H
V_6^H V_7^H
Positional Encoding
Multi-Head Attention
Add & Norm
Feed Forward
Add & Norm
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Positional Encoding
Dec 13 Wednesday
Embedding
$[V_h, V_w]$
M_1 M_2 M_3 M_4 M_5
M_6 M_7 M_8 M_9 M_{10}
M_{11} M_{12} M_{13} M_{14} M_{15}
M_{16} M_{17} M_{18} M_{19} M_{20}
M_{21} M_{22} M_{23} M_{24} M_{25}
$\sum p_{g_i} M_{g_i}$
$p_1, p_2, \ldots, p_{25}$
Softmax
Linear
Concatenate

Fig. 4. The overview of the TransGTE model.

(without any parameterized representation like embedding). This way may cause confusion of the two features since they are both extracted from (GPS trajectory's) grid embeddings by GCN and Transformer in our model. Therefore, it is necessary to use the newly developed ANFG module to eliminate the confusion, which is inspired by the GRU model with the modification of supporting the parallel inputs including embeddings of unit/grid in a given trajectory and its neighbors, instead of recursive inputs in the original structure. It benefits the prediction model to automatically adapt to all kinds of prediction scenarios by learning effective ratios of sequential and geographical features given different prediction contexts (i.e., already observed GPS trajectories).

3.3. Grid and time embedding

We first transform the GPS trajectory S^{m_r} into grid trajectory $S_L^{m_r}$. Additionally, we incorporate the time of day (in hour) t and day of week w of the trip departure time as extra temporal attributes (Definition 2).

Inspired by the word2vec (Mikolov, Chen, Corrado, & Dean, 2013), we transform the grid label L, time of day t (represented in hour), and day of week w into one-hot vectors $O(L)$, $O(t)$ and $O(w)$, and then convert them into low-dimensional dense representations by multiplying a transformation matrix:

$$V_L = O(L) \times M_L, \tag{1}$$

$$V_h = O(t) \times M_h, \tag{2}$$

$$V_w = O(w) \times M_w, \tag{3}$$

where M_L, M_t and M_w are the learnable transformation matrices of L, t and w, respectively. Then high-dimensional vectors $O(L)$, $O(t)$, and $O(w)$ are embedded into low-dimensional dense vectors V_L, V_t, and V_w, which contain more informative features.

3.4. Sequential pattern model

The sequential features, i.e., the sequential dependency between visiting locations, are important information for the destination prediction. The deep recurrent neural networks like LSTM (Ebel et al., 2020) and Seq2seq (Sutskever, Vinyals, & Le, 2014) are widely applied in various sequential modeling tasks. Besides, several recent studies (Trivedi, Silverstein, Strubell, Shenoy, & Iyyer, 2021; Xue et al., 2021; Yan, Zhao, Song, Yu, & Dong, 2023) adopt the Transformer model to mine sequential features from mobility data and achieve desirable performance. Compared with the recurrent neural networks, the Transformer uses a multi-head attention mechanism, one of self-attention mechanisms, to process the time series data, which is proven to be more efficient in many time-series learning tasks (Wen et al., 2022). The study adopts the Transformer (Vaswani et al., 2017) to extract the sequential patterns, as shown in Fig. 4. We divide the trajectory $\mathbf{S}$ into two parts, i.e., $\mathbf{S}_O$ and $\mathbf{S}_D$, where $\mathbf{S}_O$ is the part close to the origin and $\mathbf{S}_D$ close to the destination. The motivation is that the trajectory's first and last portions may have different impacts on the destination prediction (Lv et al., 2020), and thus their corresponding dependencies should be modeled separately. The sequential feature extraction process is formulated as follows:

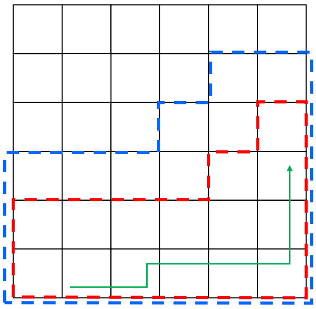


Fig. 5. An example of extracting geographical information by GCN. Given a taxi trajectory (green), the 1-layer GCN can capture information within the red box, and the 2-layer GCN can capture the geographical information within the blue box.

$$\mathbf{S}'_O, \mathbf{S}'_D = \Theta(\mathbf{S}_O, \mathbf{S}_D), \tag{4}$$

$$\mathbf{C} = \Gamma(\mathbf{S}'_O, \mathbf{S}'_D), \tag{5}$$

where $\mathbf{C}$ is context feature of the trajectory $\mathbf{S}$ (the last vector of the decoder's outputs), $\Gamma(\cdot)$ denotes the sequence processing operation with $\mathbf{S}'_O$ and $\mathbf{S}'_D$ as inputs of its encoder and decoder, and $\Theta(\cdot)$ is the position encoding function (PE).

Note that conceptually, the order of visited locations is critical for the studied trajectory-based destination prediction problem which is similar to the importance of the word order in a sentence. The structure of the encoder and decoder in $\Gamma(\cdot)$ is the same as the vanilla Transformer (Vaswani et al., 2017), which uses multi-head attention modules and position-wise feed-forward networks. However, the multi-head attention modules lack ability in directly modeling sequential order information like RNN or convolution structures applied in previous works. We introduce the position encoding function $\Theta(\cdot)$, which is a standard trigonometric transformation and the same as the vanilla Transformer. Through Eqs. 4 and 5, the Sequential Pattern Model accomplishes ex-

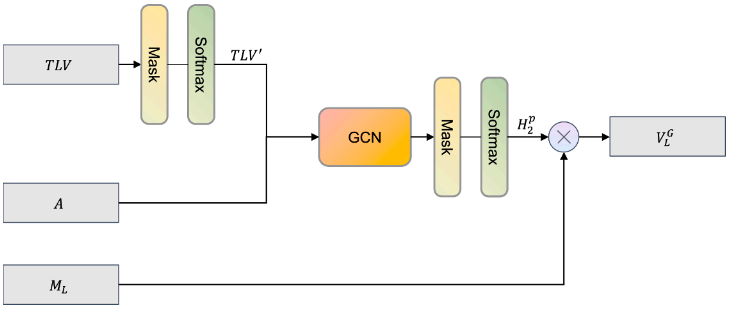


Fig. 6. The computation operation of the GCN.

tracting the context feature **C** of sequential patterns from the trajectory **S**, whose representation is generated by the ANFG in Section 3.6.

3.5. Geographical embedding

The geographically close trajectories may share several same or adjacent trajectory sections. Capturing the geographical information, contributing to the awareness of spatial proximity between trajectories or their passing route segments, is essential for precise destination prediction. In other words, vehicle trips are highly likely to visit the same destinations through these passing route segments.

We employ the GCN to extract geographical information. The GCN can model the geographical relationships between visited grids of the vehicle trajectory and their neighbors (Fig. 5). Different from other traffic prediction studies (Sun et al., 2022; Zhao et al., 2020) that apply the GCN to aggregate the neighboring information of each node in the whole road network, we use the GCN to aggregate the neighboring information of a given taxi trajectory in the local grid network since the information of grids far away from it may confuse the model in the geographical similarity of trajectories. The related operations to filter such redundant information are realized by the $mask(\cdot)$ in Eqs. 6 and 8.

As shown in Fig. 6, there are 3 variables involved in the calculation related to GCN, i.e., the trajectory location vector TLV, the adjacency matrix A, and L's transformation matrix M_L. The operation is:

$$TLV' = softmax(mask(TLV)), \tag{6}$$

$$H_{i+1} = tanh(\tilde{D}^{\frac{1}{2}} \hat{A} \tilde{D}^{\frac{1}{2}} H_i W_i), \tag{7}$$

where $mask(\cdot)$ sets all zero-value cells in TLV to a large negative number, and TLV' is the normalized vector of TLV. H_i is the i-th GCN layer's output and TLV' is the first GCN layer's input. $\hat{A} = A + I_n$ is the matrix with the added self-connection, where I_n is the identity matrix. $\tilde{D}$ is the degree matrix of $\hat{A}$ and W_i the i-th GCN layer's parameters to be trained. In this study, we use a 2-layer GCN model referring to other spatiotemporal time series studies (Li, Yu, Shahabi, & Liu, 2017; Liang, Zhao, & Sun, 2022; Zhao et al., 2020), which means that 2-hop geographical information of each grid is considered (Fig. 7). Then we transform the GCN output obtained by Eq. 7 to a probability distribution:

$$H_2^p = softmax(mask(H_2)), \tag{8}$$

where H_2^p is the percentage of geographical information that the visited grids of $S_H^{m_r}$ aggregate from neighboring grids. And H_2 is the geographical information captured by the 2-layer GCN model. To obtain the geographical features, we embed H_2^p into the semantic space of grids:

$$V_L^G = H_2^p \times M_L, \tag{9}$$

where V_L^G represents the geographical dependence of $S_H^{m_r}$ within a 2-hop distance.

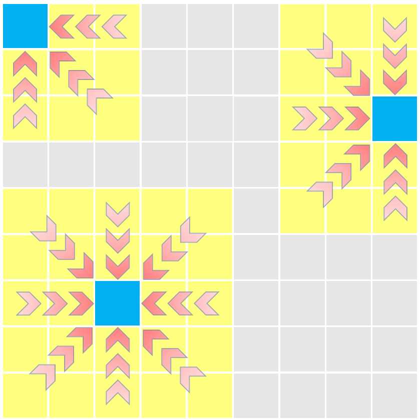


Fig. 7. The example of the 2-hop geographical dependence. The 2-layer GCN model can make each grid (blue) aggregate the information of its neighboring grids (yellow) within the 2-hop distance.

3.6. Adaptive neural fusion gate and trajectory embedding

To automatically determine the extent of geographical feature utilization, we propose a gating mechanism that adaptively fuses the sequential and geographical features.

$$\gamma_i = \sigma(W_\gamma[V_{L_i}, V_{L_i}^G] + b_{\gamma_i}), \tag{10}$$

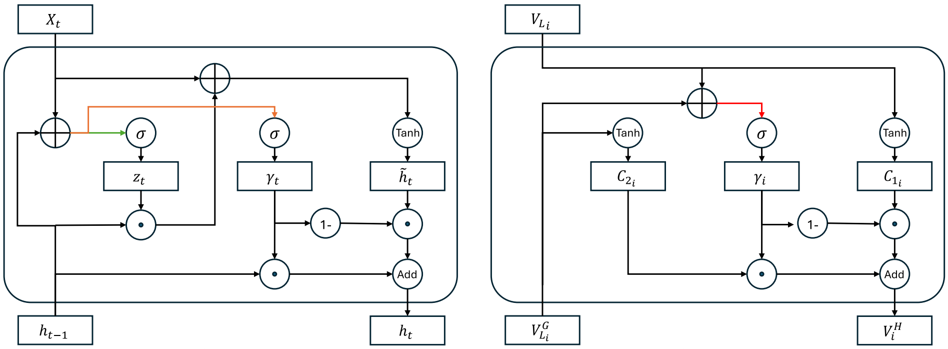
$$C_{1_i} = tanh(W_{C_{1_i}} V_{L_i} + b_{C_{1_i}}), \tag{11}$$

$$C_{2_i} = tanh(W_{C_{2_i}} V_{L_i}^G + b_{C_{2_i}}), \tag{12}$$

$$V_i^H = (1 - \gamma_i) * C_{1_i} + \gamma_i * C_{2_i}, \tag{13}$$

where γ_i is the gating coefficient matrix controlling the utility weights of geographical features, and $\sigma(\cdot)$ is the sigmoid function. C_{1_i} and C_{2_i} denote the content extracted from V_{L_i} and $V_{L_i}^G$. V_i^H is the embedding vector of grid L_i's hybrid features.

The design of the ANFG is inspired by the GRU model mechanism (Chung, Gulcehre, Cho, & Bengio, 2014). As shown in Fig. 8, the GRU model accepts information of the current state X_t and previous hidden state h_{t-1} while the ANFG model accepts two embedding vectors V_{L_i} and $V_{L_i}^G$ representing different features of GPS trajectories. Then both models use the similar operations of information filtering (Eq. 13). The gating mechanism's effectiveness in filtering redundant sequential information is adequately validated in time series learning tasks (Mahmoud & Mohammed, 2021). Thus, it has a high potential for merging the information of sequence and geography in our studied task. However, different from the GRU model, the ANFG model uses forward neural networks to update V_{L_i} and $V_{L_i}^G$ in Eqs. 11 and 12 respectively, in which suitable representations of C_{1_i} and C_{2_i} are learned from data.



(a) GRU (b) ANFG

Fig. 8. The structures of GRU and ANFG. The ⊙, ⊕, and Add respectively denote Hadamard product, vector concatenation and vector sum. σ and $Tanh$ represent forward neural networks with activation functions of sigmoid and hyperbolic tangent.

Based on Eqs. 1 to 6, each L_i is transformed into a hybrid embedding vector V_i^H. The grid trajectory $S_L^{m_r}$ is embedded into $S_H^{m_r} = \{V_1^H, V_2^H, \dots, V_{m_r}^H\}$, which serves as the input of Eq. 4.

3.7. Prediction module

To predict GPS coordinates of the destination location, it is essential to obtain the coordinates of the destination candidates (Besse et al., 2018; De Brébisson et al., 2015; Lv et al., 2020). Existing approaches cluster the destination coordinates of the taxi GPS trajectories and use the clustering centers as the destination candidates. To simplify this procedure, we obtain the destination candidates as follows. Given a grid g_i, and a set $S_{g_i} = \{d_k \| k \in \{1, 2, \dots, K\}, d_k = (lng_k, lat_k)\}$, which contains all the trajectory destinations located at g_i, where d_k is the k^{th} trajectory destination, K is the number of destinations located at g_i, and (lng_k, lat_k) denotes the longitude and latitude of destination d_k. If $S_{g_i} \neq \emptyset$, we calculate g_i's centroid coordinate as:

$$M_{g_i} = (lng_{g_i}, lat_{g_i}), \tag{14}$$

$$lng_{g_i} = \frac{1}{K}\sum_{k=1}^{K} lng_k, \tag{15}$$

$$lat_{g_i} = \frac{1}{K}\sum_{k=1}^{K} lat_k, \tag{16}$$

where M_{g_i} is the centroid of grid g_i. (lng_{g_i}, lat_{g_i}) are the longitude and latitude of M_{g_i}, respectively. For the case that $S_{g_i} = \emptyset$, we set the centroid coordinate of g_i as the geographical center. Compared to other approaches using clustering to obtain the centroid matrix, our method greatly reduces the computation time on clustering hyper-parameter tuning.

The prediction module takes the decoder's output obtained in Eq. 5 as the input of a fully-connected network with a softmax function. It outputs the grids' probability distribution $P = [p_{g_1}, p_{g_2}, \dots, p_{g_N}]$. The destination coordinate is calculated as:

$$P = softmax(W_p(\mathbf{C} \oplus V_h \oplus V_w) + b_p), \tag{17}$$

$$lng_{\hat{y}} = \frac{1}{N}\sum_{i=1}^{N} p_{g_i} lng_{g_i}, \tag{18}$$

$$lat_{\hat{y}} = \frac{1}{N}\sum_{i=1}^{N} p_{g_i} lat_{g_i}, \tag{19}$$

where ⊕ is the concatenation operation to link the three features, including context of sequential patterns and temporal patterns of departure time's hour and week. Then Eq. 17 uses a fully-connected network to fuse them to output a probability distribution P of destination candidates. By assigning the probabilities to the S_{g_i} and computing the weighted sum in Eqs. 18 and 19, we obtain the coordinate of the predicted destination $l_{\hat{y}} = (lng_{\hat{y}}, lat_{\hat{y}})$, where N is the number of grids.

Table 2
The statistics of datasets.

Dataset	Porto	Chengdu	Shenzhen	San Francisco
Taxi number	442	about 26,000	14,453	About 500
Processed samples	656,171	591,426	250,570	173,057
Grid granularity	50 × 50	70 × 70	70 × 70	60 × 60
Grid area (m^2)	233 × 105	246 × 203	441 × 252	232 × 193

To train the model, we use the Mean Squared Error (MSE) to measure the difference between the predicted $l_{\hat{y}} = (lng_{\hat{y}}, lat_{\hat{y}})$ and the true $l_y = (lng_y, lat_y)$ destinations:

$$MSE = \frac{1}{2N_\pi}\sum_{\pi\in\Pi}[(lng_{\hat{y}_\pi} - lng_{y_\pi})^2 + (lat_{\hat{y}_\pi} - lat_{y_\pi})^2], \tag{20}$$

where Π is the train set with a size of N_π.

4. Experiments

4.1. Data description

We use the real-world taxi trajectory datasets from four cities to evaluate the proposed model, i.e., Porto[1], Chengdu, Shenzhen[2] (Zhang, Zhao, Zhang, & He, 2015) and San Francisco[3] (Piorkowski, Sarafijanovic-Djukic, & Grossglauser, 2009). The GPS trajectories are produced by taxi orders with anonymous passengers. We first remove the possibly invalid trips under 5 minutes that may be generated by passengers' order cancellations. Since the original datasets are too large to train models, we randomly select the trajectories in the municipality of each city. Each dataset is split into a training set, validation set, and test set with a ratio of 7:1:2. The processed datasets and split settings are adapted to training and test for all the models used in the Experiments section. The final statistics of these datasets are presented in Table 2. All the datasets are used to validate the TransGTE model performance compared to other benchmark models in Section 4.5 and only the Porto dataset is utilized in Section 4.6 and 4.7.

4.2. Evaluation metrics

We use the Mean Haversine Distance Error (the average difference between predicted and true coordinates) to evaluate model performance, which was also used in the ECML/PKDD 15 competition as the evaluation metric. It measures distances between two points on a

[1] https://www.kaggle.com/competitions/pkdd-15-predict-taxi-service-trajectory-i/data.
[2] https://people.cs.rutgers.edu/~dz220/data.html.
[3] https://ieee-dataport.org/open-access/crawdad-epflmobility.

sphere based on their latitudes and longitudes.

$$HD(y_1, y_2) = 2 \cdot R \cdot arctan(\sqrt{\frac{\alpha}{1-\alpha}}), \tag{21}$$

$$\alpha = sin^2(\frac{\phi_2 - \phi_1}{2}) + \cos(\phi_1)cos(\phi_2)sin^2(\frac{\lambda_2 - \lambda_1}{2}), \tag{22}$$

where ϕ is the latitude, λ is the longitude, and $R = 6371km$ denotes the earth radius. The unit of $HD(y_1, y_2)$ is km in Eq. 21, which is replaced with meter (m) presented in the Experiment section for better understanding.

4.3. Baseline

To validate the TransGTE model, we compare it with 9 benchmark models in two categories (i.e., cluster-based model and grid-based model):

- **MLP** (De Brébisson et al., 2015): a cluster-based model that adopts the Multi-Layer Perceptrons framework with inputs of the first and last 5 points of raw GPS trajectories. It was the winning model in the taxi destination prediction competition (ECML/PKDD 15).
- **Multi-Input LSTM** (Ebel et al., 2020): a cluster-based model that maps GPS locations to discrete regions and embed them into dense vectors, and then transforms GPS trajectories into vector sequences. It merges the vector sequences and discrete variables as inputs of LSTM.
- **T-CONV** (Lv et al., 2019): a cluster-based model that transforms trajectories into two-dimensional images, and uses a convolutional neural network (CNN) model to capture multi-scale patterns for the precise destination prediction.
- **GPS-embedding BiLSTM** (Liao et al., 2022) a cluster-based model that uses two GPS embedding methods to convert trajectories into embedding sequences, and adopts an attention-based dual BiLSTMs neural network to capture sequential features.
- **TALL** (Zhao et al., 2018): a grid-based model that uses bi-LSTM to model sequence and gives more attention to meaningful locations having strong correlations with the destination using the attention mechanism. It was originally designed to predict which grid the destination is in.
- **Seq2seq-GCN** (Chen et al., 2020): It is applied in the check-in location prediction task. It adopts the encoder of seq2seq framework to generate the hidden state and cell state of the historical trajectories. The GCN is used to generate graph embeddings of the location topology graph. Finally, it predicts future check-in locations by aggregated the temporal dependence and graph embeddings in the decoder. It is a state-of-the-art model in mobility prediction tasks.
- **DeepMove** (Feng et al., 2018): It develops a multi-modal embedding recurrent neural network to capture the complicated sequential transitions and uses a historical attention model with two mechanisms to capture the multi-level periodicity in a principle way.
- **MobTCast** (Xue et al., 2021): It first uses the Transformer encoder to capture mobility features with both the history POI sequence and semantic information. The embeddings of mobility attributes like POI location are concatenated as the input of encoder.
- **MHSA** (Hong et al., 2023): It utilizes the Transformer encoder to learn location transition patterns from historical location visits, their visit time and activity duration, as well as their surrounding land use functions, to infer an individual’s next location. The embeddings of mobility attributes like POI location are fused by adding operation as the input of encoder.

4.4. Experimental settings

In the training step, we randomly initialize all parameters of the TransGTE. We set 250 training epochs in total and the early stop point when the model has no improvement in the validation set in 10 epochs. We use Adam as the optimizer with a 0.0001 learning rate and set the batch size as 64. The above training settings are applied to all the baseline models and TransGTE. The main hyper-parameters related to the model structure are shown in Table 3. All the model training and experimental evaluations are conducted on a workstation with Intel Xeon(R) CPU E5-2650 v3 @2.30 GHz, Nvidia 1080ti GPU, and 64-GB memory. And the framework of TransGTE is mainly realized by Pytorch 1.6 in Python 3.7 virtual environment, which also can be easily deployed to other devices like mobile phone by employing the corresponding Python libraries.

Table 3
Hyper-parameter setting of model structure.

Hyper-parameter name	Value
h embedding size	16
w embedding size	16
Grid embedding size	256
GCN hidden size	4
Multi-head size of encoder & decoder	8
Layers of encoder & decoder	1
Feedforward neuron size of encoder & decoder	1024

Besides, the completion ratio of the partial trajectory r is an essential factor in the experiment. The prediction models can extract more information in the trajectories with a higher completion ratio, which is also closer to the destinations. To explore the influence of the completion ratio, we set r as 0.1, 0.3, 0.5, 0.7, and 0.9 to get various prediction scenarios. rend to the training loss. Therefore, the TransGTE model as well as the ANFG can converge in training. Due to the complexity of mathematical proof, we show the model convergence performance numerically (common practice in deep learning studies, such as GRU model in Bahdanau, Cho, and Bengio (2014)). The loss and ANFG gradient are representatively visualized in Fig. 9 given $r = 0.7$ on the Porto dataset. In Fig. 9a, we found that the model achieved the best performance in validation set with about 10 epochs. Then training loss and validation loss tend to converge after about 100 epochs. Fig. 9b shows the L2 norm of ANFG gradient in training, which has a similar trend to the training loss. These results show that the TransGTE model and the training algorithm can converge well in training.

Compared to the baseline modes, the TransGTE model has a more complex model structure that may lead to a longer inference time for the practical deployment. From the structure of Fig. 4, there are almost linear matrix operations like embedding, ANFG, and Sequential Pattern models (i.e., Transformer). The most time cost stems from the GCN computation that has not been adequately optimized in current literature. Even so, the TransGTE model only has 25.37 MB parameters (shown in Table 3). Given the completion ratio $r = 0.9$, it takes about 0.02, 0.06, 0.03 and 0.02 seconds of inference time on average sample batch for Porto, Chengdu, Shenzhen and San Francisco, respectively. It can be optimized further using practical deployment technologies and strategies, such as model compression (Choudhary, Mishra, Goswami, & Sarangapani, 2020) and knowledge distillation (Gou, Yu, Maybank, & Tao, 2021).

4.5. Performance comparison

Table 4 summarizes the performance comparison results in Mean Haversine Distance Error with different trajectory completion ratios on the Porto. Generally, grid-based models outperform cluster-based models by significant margins, which shows the effectiveness of the developed grid-based prediction module with less tuning efforts for clustering. In grid-based benchmark models, the MHSA achieves the best performance. Compared with the MHSA, the TransGTE model improves the prediction performance by 2.38 %, 2.31 %, 4.92 %, 5.95 %, and 5.62 % for different r (4.24 % improvement in average). The models using attention mechanisms (TransGTE, mHSA, MobTcast, DeepMove, Seq2seq-

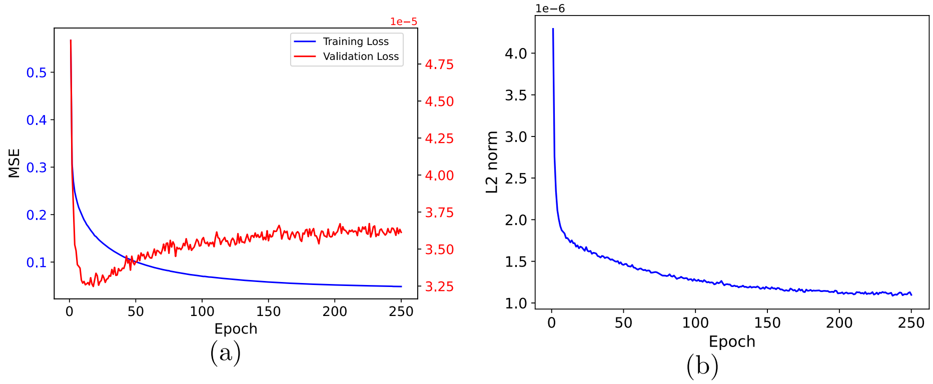


Fig. 9. The visualization of training process given $r = 0.7$. The left subfigure plots training loss and validation loss (MSE). The right subfigure plots the L2 norm of ANFG gradient in training.

Table 4
The Mean Haversine Distance Error (m) of different models in terms of r for the Porto dataset.The bold and underline numbers respectively denote the best and the second best performance given the same r. The marks are adapted to all the performance comparison tables.

Category	Model	r				
		0.1	0.3	0.5	0.7	0.9
Cluster-based	MLP	2991	2180	1363	854	758
	Multi-input LSTM	2687	1915	1248	809	652
	T-CONV	2820	2074	1348	850	666
	GPS-embedding BiLSTM	2848	2036	1086	560	304
Grid-based	TALL	1809	1364	953	616	277
	Seq2seq-GCN	1797	1315	926	587	260
	Deepmove	1795	1310	919	580	237
	MobTCast	1787	1306	917	572	237
	MHSA	1804	1294	913	571	231
	TransGTE	**1761**	**1264**	**868**	**537**	**218**

Table 5
The Mean Haversine Distance Error (m) of grid-based models in terms of r for three different datasets.

Dataset	r	TALL	Seq2seq-GCN	Deepmove	MobTCast	MHSA	TransGTE
Chengdu	0.1	2007	1999	1982	1981	2001	**1959**
	0.3	1542	1511	1492	1518	1532	**1470**
	0.5	1064	1062	1037	1056	1071	**1002**
	0.7	660	654	629	648	660	**602**
	0.9	251	241	221	234	238	**212**
Shenzhen	0.1	4039	4153	4039	3999	4036	**3975**
	0.3	3102	3005	2965	2943	2959	**2822**
	0.5	2185	2109	2072	2041	2032	**1882**
	0.7	1455	1373	1305	1319	1315	**1221**
	0.9	744	644	623	632	618	**563**
San Francisco	0.1	2476	2459	2405	2417	2427	**2402**
	0.3	2002	2076	1997	1990	1975	**1877**
	0.5	1516	1528	1465	1490	1414	**1358**
	0.7	1015	1097	1018	1051	1006	**952**
	0.9	747	668	585	584	573	**543**

GCN, TALL, and GPS-embedding BiLSTM) have better performance than the non-attentive model (Multi-Input LSTM), especially in the long sequence dataset (given a large r). The Seq2seq-GCN also uses geographical information, but it performs worse than the TransGTE model since it cannot adaptively control the utility of geographical information. The Transformer-based models MobTCast and MHSA have superiority in capturing sequential features than most of other baseline models from the results. However, they are still insensitive to geographical features that lead to falling behind the TransGTE.

The cluster-based models applied in the Porto dataset all use the same clustering centers for destination candidates provided by the MLP (De Brébisson et al., 2015). Moreover, there are short of details about clustering methods adopted by these models to obtain the clustering centers for other datasets. Therefore, we only test the grid-based models on the remaining datasets for further validation. As shown in Table 5, the TransGTE averagely outperforms the best baseline models Deepmove on the Chengdu by about 2.87 %. Moreover, the TransGTE respectively achieves a best performance on the Shenzhen and San Francisco by 5.91 % and 4.11 % on average compared to the best benchmark MHSA. Therefore, the results from Tables 4 and 5 show the state-of-the-art performance of TransGTE and imply that it can adapt to different city scales and urban layouts.

4.6. Ablation analysis

We conduct an ablation analysis to explore the effectiveness of the proposed ANFG mechanism and the constructed trajectory location vector TLV. In the ablation studies, all the ablation models use the same hyperparameters and structures in the sequential pattern model, which are validated on the Porto dataset. The detailed settings of these models are presented in Table 6.

Table 6
Ablation module settings.

Model	Input	Architecture description
(1)Transformer	S^{m_r}	It removes the GCN and ANFG in Fig. 4, and is the same as a vanilla Transformer with encoder and decoder.
(2)Transformer + NRV + GCN + ANFG	S^{m_r}	It replaces the TLV with a normal random vector (NRV) as GCN's input in Fig. 4.
(3)Transformer + TLV + FCN + ANFG	S^{m_r}, TLV	It is obtained by replacing the GCN in Fig. 4 with a fully-connected network (FCN).
(4)Transformer + TLV + CNN + ANFG	S^{m_r}, TLV	It replaces the GCN in SFER in Fig. 4 with a 1D convolutional neural network (1D-CNN).
(5)Transformer + TLV + GCN + ADD	S^{m_r}, A, TLV	It replaces the ANFG in Fig. 4 with a non-parametric additive operation.
(6)TransGTE	S^{m_r}, A, TLV	The overall TransGTE model is presented.
(7)TransGTE without TEV	S^{m_r}, A, TLV	It removes the temporal embedding vectors (TEV) produced by the pipeline at the bottom of Fig. 4.

Table 7
Ablation studies evaluated by Mean Haversine Distance Error (m).

Model	r				
	0.1	0.3	0.5	0.7	0.9
(1)Transformer	1781	1286	904	564	227
(2)Transformer + NRV + GCN + ANFG	1775	1280	881	545	220
(3)Transformer + TLV + FCN + ANFG	1780	1282	888	558	224
(4)Transformer + TLV + CNN + ANFG	1782	1270	880	557	219
(5)Transformer + TLV + GCN + ADD	1772	1278	890	557	226
(6)TransGTE	**1761**	**1264**	**868**	**537**	**218**
(7)TransGTE without TEV	1789	1276	884	548	227

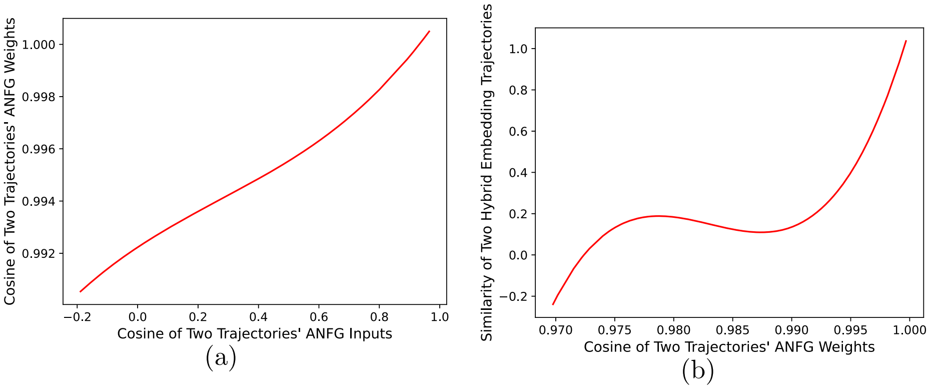


Fig. 10. The left subfigure describes the relationship between ANFG inputs and weights, and the right one describes the relationship between ANFG weights and trajectory representations.

Table 7 shows the ablation results. Firstly, comparing models (1) with (2)-(6), replacing any part in the framework of Fig. 4 can improve the performance, implying these models, except model (1), are able to capture geographical information to different levels of extent. Specifically, the models with the TLV obtain improvement from the results of (1) and (3)-(6). When using a normal random vector (NRV) to replace it, we find that the model performance decreases compared to models (2) and (6), which means that the TLV is more efficient than NRV in representing the geographical features. As shown in Figs. 3 and 5, the main reason is that the GCN with TLV as input only focuses on the trajectories' neighboring information, while NRV may comprise long-distance geographical information that is not significant to destination prediction. Through models (3), (4), and (6), we compare three ways to extract geographical features from the TLV. The results show that the GCN achieves better performance than other extractors, which implies that GCN has an advantage in capturing geographical features. From the results of models (5) and (6), we observe that our gating mechanism combined with GCN has superiority compared to the other two combinations. It suggests that the ability to merge sequential and geographical features adaptively using the gating mechanism would benefit the prediction performance. Moreover, the model without temporal embedding vectors has significant performance decline from the results of (6) and (7). It means that temporal information about trips is also essential to the trajectory destination prediction.

4.7. Interpretation of GPS trajectory representation

To illustrate the advantage of our proposed method in modeling GPS trajectory's geographical similarity, we compare the TransGTE and Transformer models by visualizing their predictions and measuring the similarity of their embedding trajectories. Inspired by the definition of semantic similarity between sentences in NLP (Gomaa & Fahmy, 2013), we use a cosine measurement (Orkphol & Yang, 2019; Si, Zheng, Zhou, & Zhang, 2019) to compute the similarity between two embedding trajectories. To simplify the calculation, we use the mean of a given embedding trajectory $\mathbf{S} = \{V_1, V_2, \ldots, V_{m_r}\}$ like the hybrid embedding trajectory $S_H^{m_r} = \{V_1^H, V_2^H, \ldots, V_{m_r}^H\}$ in the Section 3.5. The calculation is as follows:

$$\overline{V} = \frac{1}{m_r}\sum_{i=1}^{m_r} V_i, \tag{23}$$

$$sim(\mathbf{S}_1, \mathbf{S}_2) = \frac{\overline{V}_1 \overline{V}_2}{\|\overline{V}_1\| \|\overline{V}_2\|}, \tag{24}$$

where $\overline{V}$ is the mean of the embedding trajectory $\mathbf{S}$, and $sim(\mathbf{S}_1, \mathbf{S}_2)$ the similarity of the two embedding trajectories.

We first investigate the relationship between the ANFG weights and different trajectory data distributions. We randomly select a GPS trajectory from test set as the frame of reference and other 5000 trajectories as object of reference. Then the 5000 trajectories can be located by the first one. Since the TransGTE uses trajectory embeddings instead of original GPS trajectories, the embeddings of 5001 trajectories represent the trajectory data distributions. The differences of the data distributions can be measured by the similarity between frame and object trajectories that are represented by the ANFG input $[V_{L_i}, V_{L_i}^G]$ in Eq. 10. The ANFG dynamic adaptations can be measured by the cosine of ANFG weights (the γ_i in Eq. 10) between frame and object trajectories. Here, tensors of the ANFG inputs and weights are arithmetically averaged along the sequence dimension to fit different trajectory lengths. The above computational process of tensor similarity is similar to Eqs. 23 and 24 where the V_i is replaced with the corresponding tensors. Through cubic curve data fitting, Fig. 10a shows a trend that two trajectories' ANFG weights become more similar as the cosine of two ANFG inputs increases. The cosine of two trajectories' ANFG weights consistently has large values over 0.97 because the geographical features are aggregated by the sequential features in the GCN. Although the sequential features have a small weight in Eq. 13, the ANFG model can still adjust its weights under different trajectory data distributions to obtain different GPS trajectory representations as shown in Fig. 10b.

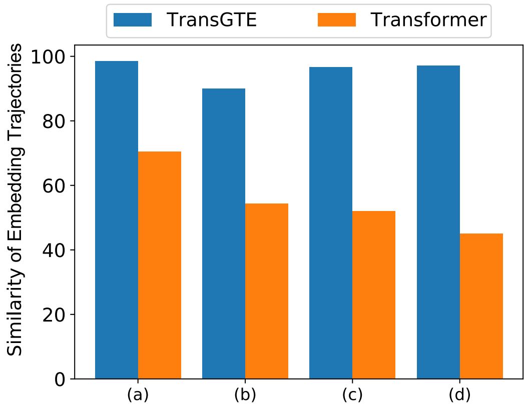


Fig. 11. The similarity of embedding trajectories in the 4 cases.

To illustrate if capturing trajectories' geographical similarity can effectively improve destination prediction, we select 4 typical case examples in which each pair of trajectories are geographically close in different scenarios: (a) two trajectories have the same origin and destination grids; (b) two trajectories share the same origin and destination grids but have partly different route segments; (c) two trajectories have the same destination and share a large part of route segments, but they start from adjacent grids with a 1-hop distance; (d) two trajectories have the same destination and share a large part of route segments, but they start from adjacent grids with a 2-hop distance. Besides, we use the Transformer as

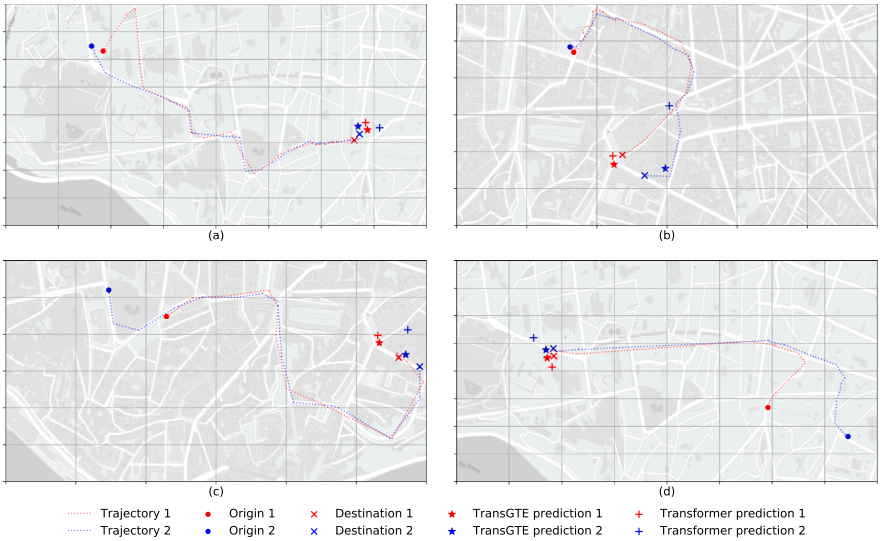


Fig. 12. The visualization of two models' prediction in the 4 cases.

the control group (model settings in Table 6), which can capture rich sequential patterns but poor geographical features. Our proposed TranGTE and the Transformer models are both pre-trained with a completion rate of $r = 0.9$ for the experiment.

The results are shown in Figs. 11 and 12. Fig. 12(a) shows that the two trajectories are mainly different in the first several grids. The TransGTE has precise destination predictions while the Transformer has a significant prediction deviation from the true destination of Trajectory 2. Quantitatively, Fig. 11(a) shows that the TransGTE model predicts that the two trajectories are 98.57 % similar while Transformer's assessment is 70.41 %. It implies that the TransGTE can capture branch roads relationship (i.e., the geographically close branch routes may merge into the major roads). Fig. 12(b) illustrates that one of the trajectories changes to a branch road near the destination grid. In this case, the Transformer is uncertain about the two trajectories' similarity (54.40 % in Fig. 11(b)), but the TransGTE still precisely captures the similarity (89.94 % in Fig. 11(b)) since it is aware of the neighboring grids' location information. In Fig. 12(c), the two trajectories have adjacent origins with a 1-grid distance. Compared with TransGTE, the Transformer has a major error in predicting Trajectory2′s destination because it just partially captures major roads' similarity (52.04 % in Fig. 11(c)) while the TransGTE captures origins' proximity (96.72 % in Fig. 11(c)). Fig. 12(d) shows the case that the origins are in about a 2-grid distance. The Transformer can not accurately predict one of the true destinations (Destination 2), and it has an extremely low assessment of the two trajectories' similarity (44.97 % in Fig. 11(d)). It implies that a long distance between the origins has a negative impact on capturing major roads' similarity using the Transformer. The TransGTE evaluates two trajectories to be 97.17 % in their similarity (Fig. 11(d)) since it can capture 2-grid locations efficiently by 2-hop geographic-aware GCN that filters the redundant sequential information of branches before entering the major road by the gating mechanism. In summary, the TransGTE has competitive advantages in representing GPS trajectories' geographical similarity for a better prediction performance compared to the commonly used sequential pattern models, such as the Transformer.

5. Conclusion

This paper proposes a novel deep learning model, TransGTE, for the trajectory-based individual trip destination prediction task. The TransGTE model uses the GCN and Transformer to extract geographical and sequential features of GPS trajectories. The TransGTE adopts a gate mechanism to dynamically fuse these two types of features and outputs GPS trajectory embeddings. Also, the TransGTE uses a grid-based prediction module to predict the destination coordinates (rather than the cluster-based methods) to reduce the computation complexity and cost.

Four real-world taxi trajectory datasets from different cities including Porto, Chengdu, Shenzhen and San Francisco are used to validate the model performance. We compared the TransGTE with representative baseline models including models using the sequential modeling (e.g, LSTM and attention mechanism) and spatial feature extraction frameworks (e.g., CNN and GCN). The results show that the TransGTE respectively outperforms the best benchmark models by 4.24 %, 2.87 %, 5.91 % and 4.11 % on the Porto, Chengdu, Shenzhen and San Francisco in terms of the Mean Haversine Distance Error. The ablation study results show that the combination of GCN and ANFG modules achieves the best performance. Finally, we numerically analyze ANFG's adaptive mechanism and visualize the similarity of embedding trajectories in cases with geographically proximate paths. This comparative analysis elucidates how our method utilizes effective GPS trajectory representations to capture geographical similarities and explains why these similarities enhance predictive accuracy.

Future work may explore improving the model prediction performance by incorporating more mobility-related data or changing conditions contributing to the prediction task, e.g., city POI data, weather or event information. Given that urban environments, such as high-rise buildings, can significantly impact the quality of GPS data during collection, future study can include addressing these biases and examining their potential effects on the model's predictions.

Data availability

The dataset used is open source with link provided in the manuscrip in the case study section.

Credit contribution statement

The authors confirm contribution to the paper as follows: study conception and design: Zhenlin Qin, Pengfei Zhang, Zhenliang Ma; data collection: Qi Zhang, Zhenliang Ma; analysis and interpretation of results: Zhenlin Qin, Pengfei Zhang, Qi Zhang, Kun Gao, Zhenliang Ma; draft manuscript: Zhenlin Qin, Pengfei Zhang, Zhenliang Ma; revise manuscript: Zhenliang Ma, Pengfei Zhang. All authors reviewed the results and approved the final version of the manuscript.

Declaration of competing interest

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

Acknowledgements

This work was supported by Natural Science Foundation of Henan (242300421456), the Key Technologies R&D Programme of Henan Province (242102210063), and KTH TRENoP and Digital Futures in Sweden.

Supplementary material

Supplementary material associated with this article can be found, in the online version at 10.1016/j.eswa.2025.128159.

References

Alvarez-Garcia, J. A., Ortega, J. A., Gonzalez-Abril, L., & Velasco, F. (2010). Trip destination prediction based on past GPS log using a hidden markov model. *Expert Systems with Applications*, *37*. https://doi.org/10.1016/j.eswa.2010.05.070

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Besse, P. C., Guillouet, B., Loubes, J. M., & Royer, F. (2018). Destination prediction by trajectory distribution-based model. *IEEE Transactions on Intelligent Transportation Systems*, *19*, 2470–2481. https://doi.org/10.1109/TITS.2017.2749413

Chen, J., Li, J., Ahmed, M., Pang, J., Lu, M., & Sun, X. (2020). Next location prediction with a graph convolutional network based on a seq2seq framework. *KSII Transactions on Internet and Information Systems*, *14*. https://doi.org/10.3837/tiis.2020.05.003

Choudhary, T., Mishra, V., Goswami, A., & Sarangapani, J. (2020). A comprehensive survey on model compression and acceleration. *Artificial Intelligence Review*, *53*, 5113–5155.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *International Conference on Machine Learning.* http://arxiv.org/abs/1412.3555.

Cui, G., Luo, J., & Wang, X. (2018). Personalized travel route recommendation using collaborative filtering based on GPS trajectories. *International Journal of Digital Earth*, *11*. https://doi.org/10.1080/17538947.2017.1326535

Dai, J., Yang, B., Guo, C., & Ding, Z. (2015). Personalized route recommendation using big trajectory data. In *Proceedings - international conference on data engineering*. (*2015-May*). https://doi.org/10.1109/ICDE.2015.7113313

De Brébisson, A., Simon, É., Auvolat, A., Vincent, P., & Bengio, Y. (2015). Artificial neural networks applied to taxi destination predictiond. In *Ceur workshop proceedings*. (*1526*).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Ebel, P., Gol, I. E., Lingenfelder, C., & Vogelsang, A. (2020). Destination prediction based on partial trajectory data. In *IEEE intelligent vehicles symposium, proceedings*. https://doi.org/10.1109/IV47402.2020.9304734

Endo, Y., Nishida, K., Toda, H., & Sawada, H. (2017). Predicting destinations from partial trajectories using recurrent neural network. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 160–172). Springer.

Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A., & Jin, D. (2018). DeepMove: Predicting human mobility with attentional recurrent networks. In *The web conference 2018 - proceedings of the world wide web conference, WWW 2018*. https://doi.org/10.1145/3178876.3186058

Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y. M., & Yuan, Q. (2015). Personalized ranking metric embedding for next new POI recommendation. In *Ijcai international joint conference on artificial intelligence*. (*2015-January*).

Gambs, S., Killijian, M. O., & Del Prado Cortez, M. N. (2012). Next place prediction using mobility Markov chains. In *Proceedings of the 1st workshop on measurement, privacy, and mobility, MPM'12*. https://doi.org/10.1145/2181196.2181199

Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. *International Journal of Computer Applications*, *68*. https://doi.org/10.5120/11638-7118

Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, *129*(6), 1789–1819.

Harris, D. M., & Harris, S. L. (2012). Digital design and computer architecture, 2nd edition. Morgan Kaufmann. https://doi.org/10.1016/C2011-0-04377-6

Hong, Y., Zhang, Y., Schindler, K., & Raubal, M. (2023). Context-aware multi-head self-attentional neural network model for next location prediction. *Transportation Research Part C: Emerging Technologies*, *156*, 104315.

Hu, J.-W., & Creutzig, F. (2022). A systematic review on shared mobility in china. *International Journal of Sustainable Transportation*, *16*(4), 374–389.

Huang, B., Ruan, K., Yu, W., Xiao, J., Xie, R., & Huang, J. (2023). Odformer: Spatial–temporal transformers for long sequence origin–destination matrix forecasting against cross application scenario. *Expert Systems with Applications*, *222*, 119835.

Jatnika, D., Bijaksana, M. A., & Suryani, A. A. (2019). Word2vec model analysis for semantic similarities in English words. In *Procedia computer science* (pp. 160–167). (*vol. 157*). https://doi.org/10.1016/j.procs.2019.08.153

Jiang, R., Song, X., Fan, Z., Xia, T., Chen, Q., Chen, Q., & Shibasaki, R. (2018). Deep ROI-based modeling for urban human mobility prediction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *2*, 1–29. https://doi.org/10.1145/3191746

Jin, X., Zhang, S., & Liu, J. (2018). Word semantic similarity calculation based on word2vec. In *ICCAIS 2018 - 7th international conference on control, automation and information sciences*. https://doi.org/10.1109/ICCAIS.2018.8570612

Krumm, J., & Horvitz, E. (2006). Predestination: Inferring destinations from partial trajectories. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*. (*vol. 4206 LNCS*). https://doi.org/10.1007/11853565_15

Le, Q. V., Ngiam, J., Chen, Z., Chia, D., Koh, P. W., & Ng, A. Y. (2010). Tiled convolutional neural networks. In *Advances in neural information processing systems 23: 24th annual conference on neural information processing systems 2010, NIPS 2010*.

Li, C., Geng, M., Chen, Y., Cai, Z., Zhu, Z., & Chen, X. M. (2024). Demand forecasting and predictability identification of ride-sourcing via bidirectional spatial-temporal transformer neural processes. *Transportation Research Part C: Emerging Technologies*, *158*, 104427.

Li, X., Li, M., Gong, Y. J., Zhang, X. L., & Yin, J. (2016). T-desp: Destination prediction based on big trajectory data. *IEEE Transactions on Intelligent Transportation Systems*, *17*. https://doi.org/10.1109/TITS.2016.2518685

Li, Y., Cui, S., Zhang, L., Liu, B., & Song, D. (2021). Taxi destination prediction with deep spatial-temporal features. In *2021 IEEE 3rd international conference on communications, information system and computer engineering, CISCE 2021*. https://doi.org/10.1109/CISCE52179.2021.9445931

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926.

Liang, Y., Zhao, Z., & Sun, L. (2022). Memory-augmented dynamic graph convolution networks for traffic data imputation with diverse missing patterns. *Transportation Research Part C: Emerging Technologies*, *143*, 103826.

Liao, C., Chen, C., Xiang, C., Huang, H., Xie, H., & Guo, S. (2022). Taxi-passenger's destination prediction via GPS embedding and attention-based biLSTM model. *IEEE Transactions on Intelligent Transportation Systems*, *23*. https://doi.org/10.1109/TITS.2020.3044943

Lin, H., He, Y., Liu, Y., Gao, K., & Qu, X. (2023). Deep demand prediction: An enhanced conformer model with cold-start adaptation for origin–destination ride-hailing demand prediction. *IEEE Intelligent Transportation Systems Magazine*. *16*(3), 111–124

Liu, Q., Wu, S., Wang, L., & Tan, T. (2016). Predicting the next location: A recurrent model with spatial and temporal contexts. In *Thirtieth AAAI conference on artificial intelligence*.

Lu, X., Wetter, E., Bharti, N., Tatem, A. J., & Bengtsson, L. (2013). Approaching the limit of predictability in human mobility. *Scientific Reports*, *3*. https://doi.org/10.1038/srep02923

Lv, J., Sun, Q., Li, Q., & Moreira-Matias, L. (2019). Multi-scale and multi-scope convolutional neural networks for destination prediction of trajectories. *IEEE Transactions on Intelligent Transportation Systems*, *21*(8), 3184–3195.

Lv, J., Sun, Q., Li, Q., & Moreira-Matias, L. (2020). Multi-scale and multi-scope convolutional neural networks for destination prediction of trajectories. *IEEE Transactions on Intelligent Transportation Systems*, *21*. https://doi.org/10.1109/TITS.2019.2924903

Ma, Z., & Zhang, P. (2022). Individual mobility prediction review: Data, problem, method and application. *Multimodal Transportation*, *1*(1), 100002.

Mahmoud, A., & Mohammed, A. (2021). A survey on deep learning for time-series forecasting. *Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges*, (pp. 365–392).

Manasseh, C., & Sengupta, R. (2013). Predicting driver destination using machine learning techniques. In *Ieee conference on intelligent transportation systems, proceedings, itsc*. https://doi.org/10.1109/ITSC.2013.6728224

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *1st international conference on learning representations, ICLR 2013 - workshop track proceedings*.

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., & Damas, L. (2013). Predicting taxi-passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, *14*. https://doi.org/10.1109/TITS.2013.2262376

Orkphol, K., & Yang, W. (2019). Word sense disambiguation using cosine similarity collaborates withword2vec and wordnet. *Future Internet*, *11*. https://doi.org/10.3390/fi11050114

Pecher, P., Hunter, M., & Fujimoto, R. (2016). Data-driven vehicle trajectory prediction. In *Sigsim-pads 2016 - proceedings of the 2016 annual acm conference on principles of advanced discrete simulation*. https://doi.org/10.1145/2901378.2901407

Piorkowski, M., Sarafijanovic-Djukic, N., & Grossglauser, M. (2009). epfl/mobility. In https://doi.org/10.15783/c7j010.

Si, S., Zheng, W., Zhou, L., & Zhang, M. (2019). Sentence similarity computation in question answering robot. In *Journal of physics: Conference series*. (*1237*). https://doi.org/10.1088/1742-6596/1237/2/022093

Sun, Y., Jiang, G., Lam, S. K., & He, P. (2022). Learning traffic network embeddings for predicting congestion propagation. *IEEE Transactions on Intelligent Transportation Systems*, *23*. https://doi.org/10.1109/TITS.2021.3105445

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112). (*4*).

Thiagarajan, A., Ravindranath, L., LaCurts, K., Madden, S., Balakrishnan, H., Toledo, S., & Eriksson, J. (2009). Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM conference on embedded networked sensor systems* SenSys '09 (p. 85–98). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/1644038.1644048. https://doi.org/10.1145/1644038.1644048

Tian, R., Wang, C., Hu, J., & Ma, Z. (2023). Multi-scale spatial-temporal aware transformer for traffic prediction. *Information Sciences*, *648*, 119557.

Trivedi, A., Silverstein, K., Strubell, E., Shenoy, P., & Iyyer, M. (2021). Wifimod: Transformer-based indoor human mobility modeling using passive sensing. In *Proceedings of 2021 4th ACM SIGCAS conference on computing and sustainable societies, COMPASS 2021*. https://doi.org/10.1145/3460112.3471951

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L.,& Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*. (*2017-December*).

Wang, L., Wang, M., Ku, T., Cheng, Y., & Guo, X. (2017). A hybrid model towards moving route prediction under data sparsity. In *20th international conference on information fusion, fusion 2017 - proceedings*. https://doi.org/10.23919/ICIF.2017.8009862

Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D. F., & Chao, L. S. (2019). Learning deep transformer models for machine translation. arXiv preprint arXiv:1906.01787.

Wei, L. Y., Zheng, Y., & Peng, W. C. (2012). Constructing popular routes from uncertain trajectories. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*. https://doi.org/10.1145/2339530.2339562

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., & Sun, L. (2022). Transformers in time series: A survey. arXiv preprint arXiv:2202.07125.

Xue, A. Y., Qi, J., Xie, X., Zhang, R., Huang, J., & Li, Y. (2015). Solving the data sparsity problem in destination prediction. *VLDB Journal*, *24*. https://doi.org/10.1007/s00778-014-0369-7

Xue, H., Salim, F., Ren, Y., & Oliver, N. (2021). MobTCast: Leveraging auxiliary trajectory forecasting for human mobility prediction. *Advances in Neural Information Processing Systems*, *34*, 30380–30391.

Xue, H., & Salim, F. D. (2021). Termcast: Temporal relation modeling for effective urban flow forecasting. In *Advances in knowledge discovery and data mining: 25th pacific-asia conference, PAKDD 2021, virtual event, may 11–14, 2021, proceedings, part i* (pp. 741–753). Springer.

Yan, B., Zhao, G., Song, L., Yu, Y., & Dong, J. (2023). PreCLN: Pretrained-based contrastive learning network for vehicle trajectory prediction. *World Wide Web*, *26*(4), 1853–1875.

Yang, C., Sun, M., Zhao, W. X., Liu, Z., & Chang, E. Y., et al. (2017). A neural network approach to jointly modeling social networks and mobile trajectories. *ACM Transactions on Information Systems*, *35*. https://doi.org/10.1145/3041658

Yang, J., Xu, J., Xu, M., Zheng, N., & Chen, Y. (2014). Predicting next location using a variable order markov model. In *Proceedings of the 5th ACM SIGSPATIAL international workshop on geostreaming, IWGS 2014*. https://doi.org/10.1145/2676552.2676557

Yin, H., Wang, W., Wang, H., Chen, L., & Zhou, X. (2017). Spatial-aware hierarchical collaborative deep learning for POI recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *29*. https://doi.org/10.1109/TKDE.2017.2741484

Zhang, D., Zhao, J., Zhang, F., & He, T. (2015). UrbanCPS: A cyber-physical system based on multi-source big infrastructure data for heterogeneous model integration. In *Proceedings of the ACM/IEEE sixth international conference on cyber-physical systems* (pp. 238–247).

Zhang, L., Zhang, G., Liang, Z., & Ozioko, E. F. (2018). Multi-features taxi destination prediction with frequency domain processing. *PLoS ONE*, *13*. https://doi.org/10.1371/journal.pone.0194629

Zhang, P., Koutsopoulos, H. N., & Ma, Z. (2023). Deeptrip: A deep learning model for the individual next trip prediction with arbitrary prediction times. *IEEE Transactions on Intelligent Transportation Systems*, (pp. 1–14). https://doi.org/10.1109/TITS.2023.3252043

Zhang, X., Zhao, Z., Zheng, Y., & Li, J. (2020). Prediction of taxi destinations using a novel data embedding method and ensemble learning. *IEEE Transactions on Intelligent Transportation Systems*, *21*. https://doi.org/10.1109/TITS.2018.2888587

Zhao, J., Xu, J., Zhou, R., Zhao, P., Liu, C., & Zhu, F. (2018). On prediction of user destination by sub-trajectory understanding: A deep learning based approach. In *International conference on information and knowledge management, proceedings*. https://doi.org/10.1145/3269206.3271708

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., & Li, H. (2020). T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, *21*, 3848–3858. https://doi.org/10.1109/TITS.2019.2935152

Zheng, Y., Xie, X., & Ma, W. (2010). Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Engineering Bulletin*, *33*.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-Series forecasting. In *35th AAAI conference on artificial intelligence, AAAI 2021*. (*12B*). https://doi.org/10.1609/aaai.v35i12.17325