Neural Networks 186 (2025) 107290

Contents lists available at ScienceDirect

Neural Networks

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

Full Length Article

GeM: Gaussian embeddings with Multi-hop graph transfer for next POI recommendation

Wenqian Mu [a,1], Jiyuan Liu [b,1], Yongshun Gong [a,*], Ji Zhong [c], Wei Liu [d], Haoliang Sun [a], Xiushan Nie [e], Yilong Yin [a,*], Yu Zheng [f]

[a] *School of Software, Shandong University, Jinan, China*
[b] *School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China*
[c] *Shandong Yunhai Guochuang Cloud Computing Equipment Industry Innovation Co., Ltd., Shandong, China*
[d] *School of Computer Science, University of Technology Sydney, Australia*
[e] *School of Computer Science and Technology, Shandong Jianzhu University, Jinan, China*
[f] *JD Intelligent Cities Research, China*

ARTICLE INFO

Keywords:
Next POI recommendation
Gaussian embeddings
Spatio-temporal learning
Graph representation

ABSTRACT

Next Point-of-Interest (POI) recommendation is crucial in location-based applications, analyzing user behavior patterns from historical trajectories. Existing studies usually use graph structures and attention mechanisms for sequential prediction with single fixed points. However, existing work based on the Markov chain hypothesis neglects dependencies of multi-hop transfers between POIs, which is a common pattern of user behaviors. To address these limitations, we propose GeM, a unified framework that effectively employs Gaussian distribution and Multi-hop graph relation to capture movement patterns and simulate user travel decisions, considering user preference and objective factors simultaneously. At the subjective module, Gaussian embeddings with Mahalanobis distance are exploited to make the embedded space non-flat and stable, which enables the expression of asymmetric relations, while the objective module also mines graph information and multi-hop dependency through a global trajectory graph, reflecting POI associations with user movement. Besides, matrix factorization is used to learn user-POI interaction. By combining both modules, we get a more accurate representation of user behavior patterns. Extensive experiments conducted on two real-world datasets show that our model outperforms the compared state-of-the-art methods.

1. Introduction

The rapid growth of location-based social networks (LBSNs) has facilitated the development of next point-of-interest (POI) recommendation services such as Foursquare, Gowalla, and Yelp, which play an essential role in urban computing by leveraging spatial–temporal data to enhance users' urban experiences and decision-making (Gong, Dong, Zhang and Chen, 2023; Gong, Li et al., 2023; Guo, Lin, Li, Chen, & Wan, 2019). The large volume of historical check-in data is of great value, which can help users explore attractive places by capturing their behavioral patterns. In brief, users may visit POIs out of objective necessity (e.g. visiting a company after leaving a metro station) or out of subjective preference (e.g. choosing a restaurant near the company at lunchtime). Most existing approaches cast the next POI recommendation problem as a sequential prediction task via spatio-temporal learning. They leveraged a sequential model, e.g., recurrent neural network (Thaipisutikul & Chen, 2024; Wang, Yuan, Yang, Peng, & Jiang, 2024; Wu, Li, Zhao, & Qian, 2020) and transformer (He, Zhou, Luo, Gao, & Wen, 2023; Kong, Chen, Li, Bi, & Shen, 2024; Xie & Chen, 2023), to capture the users' movement patterns. Recent advanced studies involved additional information to improve the performance, which demonstrated that composition diagram structures, e.g., POI category (Dong, Meng, & Zhang, 2021; Sun, Zheng, Lu, Zhu, & Zhang, 2024), and the relationship diagram between users (Huang, Ma, Dong, Foutz, & Li, 2022; Rao et al., 2022), are significant to recommend next POIs.

Despite these successes, several challenges still have not been fully addressed.

I. Embedding problem caused by fixed points. Most studies on model sequential and graphic structure based on user trajectories and POI information with fixed points, which have been proven to be powerful, still carry some significant limitations. Point vectors are typically

[*] Corresponding authors.
E-mail addresses: ysgong@sdu.edu.cn (Y. Gong), ylyin@sdu.edu.cn (Y. Yin).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.neunet.2025.107290
Received 28 October 2024; Received in revised form 12 January 2025; Accepted 13 February 2025
Available online 22 February 2025

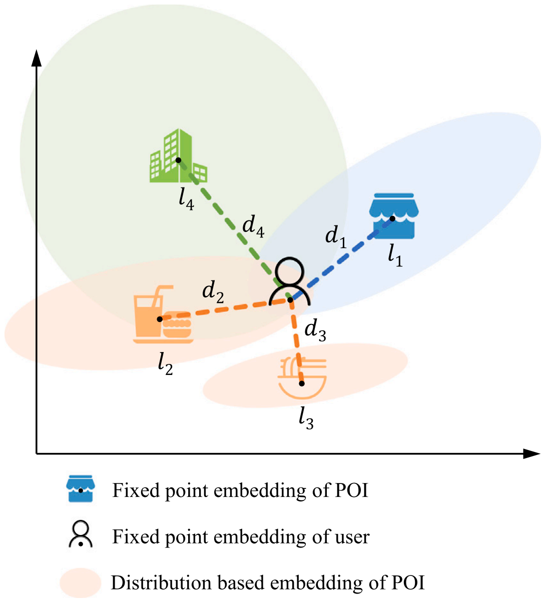
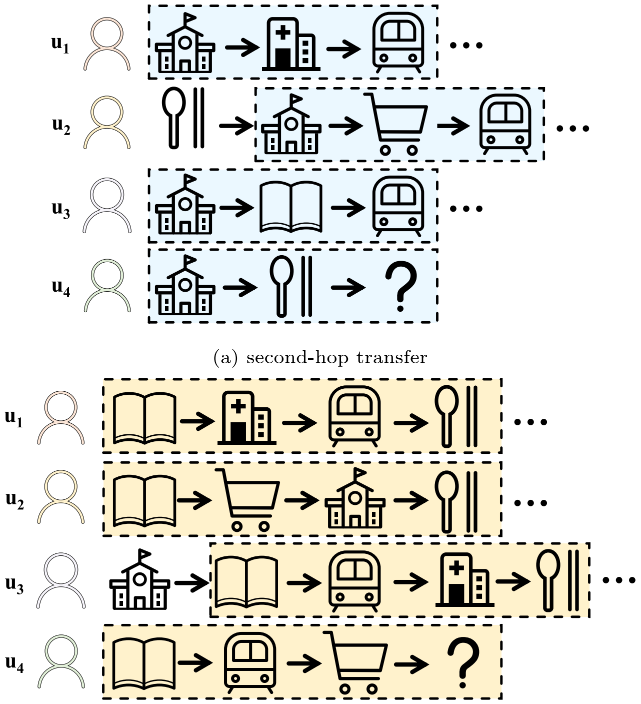

l_4
d_4
d_1
l_1
d_2
d_3
l_2
l_3
Fixed point embedding of POI
Fixed point embedding of user
Distribution based embedding of POI

Fig. 1. POI fixed-point embedding makes some inaccurate recommendations due to the inadequate representation of POI attributes while distribution-based embedding can handle embedding with asymmetry well.



(a) second-hop transfer

(b) third-hop transfer

Fig. 2. An example of local POIs transfer showing second-hop and third-hop convergence, the figure shows mutil-hop transfers of several POIs in a local region.

compared by dot products, cosine distance, or Euclidean distance, none of which provide for asymmetric comparisons between objects (as is necessary to represent contains or partially related, which is common between POIs).

We illustrate an example in Fig. 1 to explain why the fixed point embedding does not handle the POI recommendation problem well. Consider a scenario where user u is selecting a lunch spot, with locations l_1 a convenience store, l_2 fast food outlets, l_3 a Chinese restaurant, and l_4 a shopping center represented by different colors corresponding to their POI categories. d_i represents the distance between the fixed point of location l_i and user u, and $d_3 < d_1 = d_2 < d_4$. According to previous user behavior patterns, l_1 and l_2 are often chosen as the next point to visit, thus u's position in the embedding space (mapped into a 2D map) may locate in the middle of l_1 and l_2. Due to l_3 having the same category as l_2, their fixed points are close. When recommendations are generated using traditional Euclidean distance, user u is more likely to be recommended l_3 because it is the closest to u's embedding in terms of distance d_3, as opposed to l_1 or l_2, which are farther from the user's embedding with distances d_1 and d_2, respectively. Another case is that l_4 is less likely to be recommended although it aligns well with the interests of the user which has an inclusive relationship with l_1 and l_2. However, if the locations are embedded within a distribution, the recommendation outcome changes significantly. In this case, user u would be directed toward l_1, l_2, and l_4 while l_3 would not be recommended since it is spatially farthest from user u's position within this distribution. This demonstrates how fixed point embedding can obscure the diverse characteristics of POIs, resulting in misleading and inaccurate recommendation outcomes.

II. Transfer problem caused by multi-hop dependencies. When considering POI transfers at the objective level of need, previous work incorporated collaborative signals (Li, Chen, Luo, Yin, & Huang, 2021) to learn POI embedding or constructed Markov transition matrix for adding global information (Yang, Liu, & Zhao, 2022), but neglected the direct multi-hop transition information from the last POIs simply and effectively.

As shown in Fig. 2, given the first three users $u_{1,2,3}$'s trajectory segments and the current trajectory of u_4, it can be inferred from Fig. 2(a) that after leaving school people tend to visit nearby POIs and eventually underground stations. However, it is difficult to capture the multi-hop transfer pattern in the graph with a traditional transition matrix, which obeys the Markov chain hypothesis. In this way, prediction is based on the last POI restaurant, so the school may be recommended as the next POI because there is a restaurant → school link of u_2. If this simple second-hop transfer relationship is explicitly captured (e.g., the intrinsic second-hop relationship between school, nearby locations, and underground station in the blue box in Fig. 2(a)), the underground station can be recommended to user u_4 based on the penultimate POI school. And the same mode applied in Fig. 2(b) makes it possible to capture third-hop relationship, between the library, nearby locations, and the restaurant, as illustrated in the yellow box.

III. Lack of unified framework. In reality, people often make POI access decisions based on subjective preferences or objective needs, such as choosing a restaurant or visiting an office after leaving a metro station. Previous work has always modeled these two important factors together, ignoring the benefits of modeling them independently. Thus, it is difficult to accurately capture these two complementary patterns of user behavior. So, there is a lack of accurate representation of separation modeling in a unified framework.

To alleviate the above-mentioned problems, we make efforts on the following three aspects:

For **problem I**, we adopt a novel learning principle different from previous fixed point based models. Specifically, our framework first learns a Gaussian distribution for each POI, which well captures the asymmetric relationship between POIs and the users' cognitive asymmetry, due to the centrality, smoothness, and distributional properties of Gaussian distribution. Then, to interconnect the users' historical trajectory with the POI distribution, we adopt the Mahalanobis distance between the embedded points of the users' trajectory and the Gaussian distribution as the conditional probability distribution of the current user's choice, which accurately models the relationship between POIs and users' choice in the form of a probabilistic model. Thus we can obtain a latent Gaussian space that fully represents the conditional probability distribution of users' choices with POIs.

For **problem II**, we construct a global trajectory flow graph that summarizes the trajectories between POIs, on which we capture the collaboration information, and then yield a basic transfer matrix using a self-attention layer, in which node interactions are sufficiently carried out. To deal with our proposed multi-hop transfer problem, we employ matrix multiplication on the basic transfer matrix, obtaining the hth hop transfer matrix and distribution through the corresponding inverse hth POIs. Then we aggregate these distributions to get the final hth hop transfer distribution, which can leverage the information of the last h POIs of a user's historical trajectory for accurate recommendations.

For **problem III**, we model the users' choice factors independently into two modules, which capture user behavioral patterns more accurately. In the first module, a self-attention layer and LSTM are used to pool the user's historical trajectories. Then the computation of conditional distributions is done in a Gaussian space using the Mahalanobis distance, which fully represents the users' subjective preferences. The second module introduces graph information and multi-hop transfer matrices in the FPMC (Rendle, Freudenthaler, & Schmidt-Thieme, 2010) framework, which provides more dimensional information and is capable of capturing multi-hop dependencies. As we mentioned above, the hth hop transfer distribution represents the intrinsic linkage of POIs with multi-hop dependencies. Then modeling user-POI interactions by multiplying user and POI matrices. Using the above two interaction distributions, we can well obtain the results of the second module from another complementary perspective. Eventually, the two modules converge to capture the behavioral pattern factors of users in a unified framework thus recommending the next POI more accurately.

In summary, our contributions are as follows:

- We explicitly model the Gaussian distribution of POIs, which is more effective than traditional embedding of fixed points, and employ the Mahalanobis distance to measure the conditional probability distribution of the user's next POI.
- We discover the multi-hop transfer case in POI objective linkage and use a self-attention layer to obtain a basic transition matrix, then adaptively obtain the hth hop transfer distribution which captures hth hop dependencies that ordinary Markov chains do not inherit.
- We propose GeM, a unified framework that explicitly models the subjective and objective factors that influence user choice separately, to fully capture user behavior patterns in a complementary way.
- Extensive experiments were conducted on two real datasets to evaluate the proposed model. The results show that the recommended performance of GeM is significantly better than the state-of-the-art baseline.

2. Related works

2.1. Next POI recommendation

Spatio-temporal learning has been widely applied in tasks such as POI recommendation (Long, Chen, Nguyen, & Yin, 2023), urban traffic prediction (Li et al., 2024; Yu et al., 2025), and weather forecasting (Gong et al., 2024). POI recommendation can effectively uncover user behavior patterns and provide support for personalized services. Early work on POI recommendations focused on feature engineering and traditional (non-deep learning) machine learning-based approaches. For example, due to the high sparsity of user sign-in data, matrix factorization (MF) (Koren, Bell, & Volinsky, 2009) based methods have been investigated for better POI recommendation modeling. There is also work (Rendle et al., 2010) using a matrix factorization method embedded in personalized Markov chains (FPMC) that combines the advantages of matrix factorization and Markov chains based on the last items. In general, these earlier approaches are quite limited in their ability to model sequential data compared to deep neural network models.

On another line of research, RNN-based approaches dominate the next POI recommendation field due to the powerful ability to learn the information from sequence data, proposing a variant of RNN to capture temporal dynamics and sequence correlation (Zhang, Sun, Wu et al., 2022; Zhao et al., 2020). STRNN (Liu, Wu, Wang, & Tan, 2016) improves the performance of the model by modeling the local spatio-temporal context of each layer with a spatio-temporal interval transfer matrix. PLSPL (Wu et al., 2020) proposes a new approach of personalized long- and short-term preference learning to learn the specific preferences of each user. TiSASRec (Li, Wang, & McAuley, 2020) uses a self-attention mechanism approach that considers not only the absolute position of items, but also the time interval between items in a sequence. Flashback (Yang, Fankhauser, Rosso, & Cudre-Mauroux, 2020) explicitly uses spatio-temporal context to search for past hidden states with high predictive power to model sparse user movement tracking. Time-LSTM (Zhu et al., 2017) proposes to add temporal gates to the LSTM structure to better accommodate spatio-temporal effects. LSTPM (Sun et al., 2020) constructs a geo-dilated RNN that aggregates locations visited recently, but only for short-term preference. HOPE (Sun et al., 2021) applies LSTM to model short-term preferences for out-of-town visits, while asymmetric-SVD is used for long-term preference modeling.

Some work has also introduced attention mechanisms to the next POI recommendation problem, and Deepmove (Feng et al., 2018) proposes a GRU cyclic layer and an attention layer to capture the multilevel periodicity of densely embedded trajectories. Recently, with the good performance of transformer architecture (Vaswani et al., 2017) in various tasks, more works use the attention mechanism completely to obtain better recommendation results, STAN (Luo, Liu, & Liu, 2021) uses a two-layer attention mechanism and spatio-temporal interval embedding to capture the point-to-point inter-sequence correlation. C^{2}DREIF (Ni, Cai, & Jiang, 2024) employs a top-down transformer to accurately extract user intents within each domain, taking into account the user's long-term and short-term preferences. TADSAM (Li & Liang, 2022) dynamically uses time windows on the attention mechanism to achieve time-aware recommendations. Wang, Wang, and Lu (2023) constructed the long- and short-term attention mechanism to better capture the user's long-term and short-term preferences. MSDCCL (Zhu, Li, Liu, & Luo, 2024) constructs a soft-level denoising module by carrying out a soft signal denoising strategy based on the attention mechanism for the recommender. CART (Zhang, Sun, Zhang, Wu and Xia, 2022) introduces context-level and relation-level attention in recommender and uses conversation modules, realizing more accurate next POI recommendations with uncertain check-ins.

All these studies ignore the good performance brought by combining the self-attention mechanism and the traditional RNN model in a spatially embedded approach to the next POI recommendation problem.

2.2. Graphs in location-based recommendation

Considering the graph characteristics of the geographical location of POI data, and the natural suitability of the process of user-POI interaction to be described by a graph structure, many works have enhanced the overall effect of the model by using graph information as auxiliary information.

Early work such as GE (Xie et al., 2016) used graphs more as auxiliary structural information to achieve better embedding. And with the good performance of GCN (Kipf & Welling, 2016) in handling graph-structured data, more work (An et al., 2024; Han et al., 2021; Zhang, Liu, Zhou, & Chu, 2021) uses data beyond POI and user trajectories to exploit various graph information, here we focus on models that only use POI data itself to build graphs for learning (Fu et al., 2024; Lim et al., 2022; Wang et al., 2021). DRAN (Wang, Zhu, Liu and Wang, 2022) constructs distance and transfer graphs of node embedding based on distance and transfer graphs and augments the attention network using the decentred representation. ASGNN (Wang et al., 2021) uses graph-to-node embedding with personalized hierarchical attention networks to capture users' long-term and short-term preferences. HMT-GRN (Lim et al., 2022) uses graph embedding followed by hierarchical beam search for different regions and POI distributions. SGRec (Li et al., 2021) proposes a sequence-to-graph enhancement method to obtain collaborative information across sequences and designs a multi-task learning scheme to obtain better recommendation results. KGCN (Wang, Sun, Fang, Yang and Wang, 2022) introduces knowledge graphs (TKGs) with temporal information in POI recommendations and learns user preferences for TKGs by dynamically capturing spatio-temporal neighborhoods. MVKC (Yang, Wu, Zhang, Wang, & Nie, 2024) generates augmented views by fusing the augmented knowledge graph and interaction graph, and extracts graph features in a self-supervised manner, effectively alleviating the sparsity of user-item interaction graphs. ARNPP-GAT (Liang & Zhang, 2020) combines an attention-based recurrent neural point process with a graph attention network. GCRec (Wu, He, Zhang, Wang, & Ye, 2022) employs a linear graph convolution module to capture users' long-term information and designs a user-specific capsule module and a position-aware gating module to capture sequential patterns of users. A recent work, GETNext (Yang et al., 2022) uses a global trajectory flow graph to aggregate information, such as the category of POI, to capture users' collaboration signals. However, previous work has never considered the existence of realistic multi-hop convergence situations and ignored the significant impact of the penultimate POI in the graph structure on the next POI. Nor did it incorporate graph information into the overall structure of the matrix factorization. In this respect, our model is the first work to exploit the multi-hop convergence property of graph-based recommendations.

2.3. Gaussian embedding in recommendation

In recent years, due to the good representation property of Gaussian distribution, some researchers applied Gaussian embedding to learn vocabulary (Qian, Feng, Wen, & Chua, 2021; Vilnis & McCallum, 2014), knowledge graph (He, Liu, Ji, & Zhao, 2015; Xie, Zhou, Liu, & Huang, 2023; Xu, Nayyeri, Alkhoury, Yazdi, & Lehmann, 2020) and nodes in graph (Bojchevski & Günnemann, 2017). This explicit Gaussian distribution embedding works well for the vocabulary and graph structure that exists in nature.

Similarly, the application of Gaussian distribution to the embedding of recommended items in recommendation systems works well. For example, GER (Dos Santos, Piwowarski, & Gallinari, 2017) represents Gaussian embedding of users or items, resulting in better collaborative filtering results. geRec (Jiang, Yang, Xiao, & Shen, 2020) replacing the representation of a low-dimensional space with a Gaussian distribution to represent users and items instead of the original embedding of a single immobile point, can fit the uncertain preferences exhibited by some users. STOSA (Fan et al., 2022) implements a stochastic embedding of Gaussian distribution of items, and uses a Wasserstein distance-based self-attention module to measure the similarity between items in a sequence with uncertain signals for recommendation. Meanwhile, many works using the VAE (Kingma & Welling, 2013) framework for similar recommendation problems inevitably have Gaussian distribution (Song et al., 2019; Xu et al., 2022; Zhou et al., 2023).

However, the above work has not been well applied to the next POI recommendation problem, and the good nature of Gaussian distribution embedding for POI has not been well studied.

3. Problem formulation

This section introduces several critical concepts in next POI recommendation. Let $\boldsymbol{U} = \{u_1, u_2, \ldots, u_U\}$ denotes a set of users, $\boldsymbol{L} = \{l_1, l_2, \ldots, l_L\}$ denotes a set of POIs (e.g., restaurants and hotels), and $\boldsymbol{T} = \{t_1, t_2, \ldots, t_T\}$ denotes a set of timestamps. Each POI $\in \boldsymbol{L}$ is represented by a tuple $p = (\text{lat, lon})$ for latitude and longitude. Related symbol descriptions used in this study are summarized in Table 1.

Table 1
Symbol description.

Symbols	Descriptions
$\boldsymbol{U}, \boldsymbol{L}, \boldsymbol{T}$	The set of user, POI and time slot
$r = (u, l, t)$	check-in
$p_i = (lat_i, lon_i)$	GPS coordinates of location i
m_i	The original trajectory length of u_i
n	The maximum length of trajectory
$seq\left(u_i\right)$	Fixed-length sequence of user u_i
$\mathcal{G}_t = \left(\mathcal{V}_t, \mathcal{E}_t, \boldsymbol{A}_t\right)$	Graph of global trajectory
$g_{l_k} = \mathcal{N}(\boldsymbol{\mu}_{l_k}, \boldsymbol{\Sigma}_{l_k})$	Gaussian distribution of POI l_k
d	Embedding dim
$\boldsymbol{\Phi}'$	Basic transfer matrix
$\boldsymbol{\Phi}_h$	hth hop transfer matrix

Definition 1 (*Historical Check-ins*)**.** The trajectory of user u_i is a time-ordered check-in sequence. Each check-in r_k of a user u_i trajectory is indicated by a tuple $\left(u_i, l_k, t_k\right)$, where l_k is the position and t_k is the timestamp. Each user $u_i \in \boldsymbol{U}$ has a trajectory sequence $traj\left(u_i\right) = \left\{r_1, r_2, \ldots, r_{m_i}\right\}$, m_i is the current length of the trajectory of u_i.

To facilitate a uniform training model, we convert each trajectory into a fixed-length sequence $seq\left(u_i\right) = \left\{r_1, r_2, \ldots, r_n\right\}$, with n as the maximum length we consider. If $n < m_i$, we only consider the n most recent check-ins. If $n > m_i$, unused positions are interpolated with zeros. These positions will be masked and removed during the calculation.

Definition 2 (*Trajectory Spatio-Temporal Relation Matrix*)**.** Here we follow the definition of STAN (Luo et al., 2021) to use time interval and geographic distance. We denote the spatial distance between the ith visited GPS location and the jth visited GPS location as $\Delta^s_{ij} = haversine\left(p_i, p_j\right)$. The time interval between the ith visit and the jth visit is expressed as $\Delta^t_{ij} = \left|t_i - t_j\right|$. Specifically, the trajectory spatial relationship matrix $\boldsymbol{\Delta s} \in \mathbb{R}^{n \times n}$ and the trajectory temporal relationship matrix $\boldsymbol{\Delta t} \in \mathbb{R}^{n \times n}$ are separately denoted as follows:

$$\boldsymbol{\Delta s} = \begin{bmatrix} \Delta^s_{11} & \Delta^s_{12} & \cdots & \Delta^s_{1n} \\ \Delta^s_{21} & \Delta^s_{22} & \cdots & \Delta^s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta^s_{n1} & \Delta^s_{n2} & \cdots & \Delta^s_{nn} \end{bmatrix}, \tag{1}$$

$$\boldsymbol{\Delta t} = \begin{bmatrix} \Delta^t_{11} & \Delta^t_{12} & \cdots & \Delta^t_{1n} \\ \Delta^t_{21} & \Delta^t_{22} & \cdots & \Delta^t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta^t_{n1} & \Delta^t_{n2} & \cdots & \Delta^t_{nn} \end{bmatrix}. \tag{2}$$

Definition 3 (*Global Transfer Graph*)**.** The POI transfer-based relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$ is a weighted graph where $\mathcal{V}$, $\mathcal{E}$ and are the sets of nodes and edges, respectively. $\boldsymbol{A} \in \mathbb{R}^{L \times L}$ is a weighted adjacency matrix, and $\boldsymbol{A}(i, j) = freq\left(l_i, l_j\right)$ describes a direct transformation relation between l_i and l_j, where $freq\left(l_i, l_j\right)$ represents the frequency of transitions between l_i and l_j for all users.

Definition 4 (*hth Hop Transfer Matrix*)**.** hth Hop Transfer Matrix is a lookup table that shows each POI's hth hop transfer probability to next POI. Given a POI segment of user u_i's trajectory $traj(u_i) = (l_1,\ l_2,\ \ldots,\ l_m)$, where l_m is the last visit of user u, the hth hop transfer matrix $\boldsymbol{\Phi}_h$ represents the transfer guidance of last hth visit l_{m-h+1} to the next POI. In particular, the ordinary Markov matrix is equivalent to a first-hop transfer matrix $\boldsymbol{\Phi}_1$, and $\boldsymbol{\Phi}_1[l_m]$ (the l_mth row of $\boldsymbol{\Phi}_1$) represents the transfer probability of the last POI l_m to the next POI l_{m+1}.

Definition 5 (*Mahalanobis Distance*)**.** Mahalanobis distance measures the distance between the point $\boldsymbol{x}$ and the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

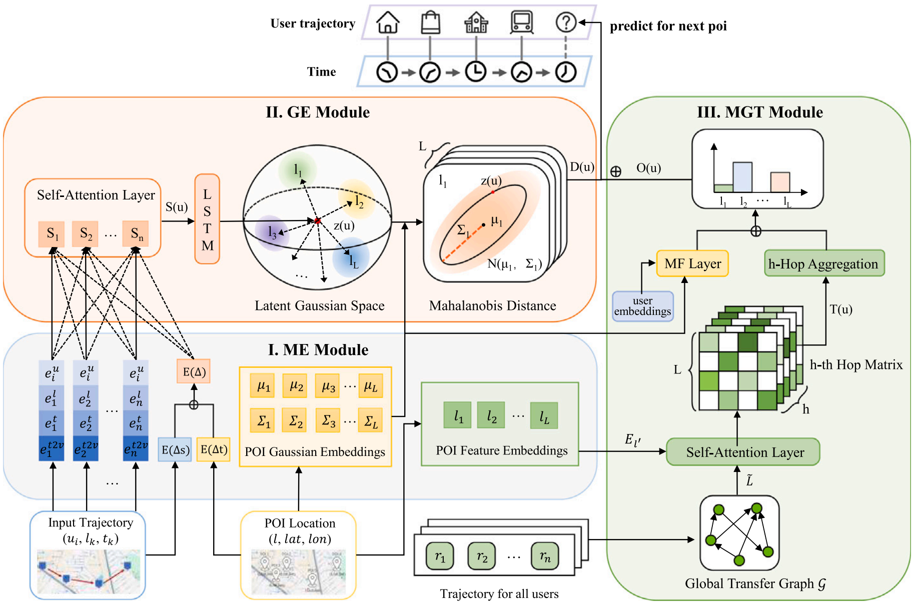


Fig. 3. The architecture of the proposed GeM model. (I. ME Module represents the MultiModal Embedding Module, II. GE Module is the Gaussian Embedding Module and III. MGT Module constructs the Multi-hop Graph Transfer Module.)

as:

$$\mathrm{Ma\_dst}(\boldsymbol{x}, \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})) = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}. \tag{3}$$

Definition 6 (*Mobility Prediction*)**.** The goal of the next POI recommendation is to provide a list of possible POIs that users tend to visit next by learning the specified user trajectory and the historical check-in history of all users.

More formally, information of a specific user u_i is given as a set of historical trajectories $\left\{traj\left(u_i\right)\right\}_{u_i \in U}$, location candidates $\boldsymbol{L} = \left\{l_1, l_2, \ldots, l_L\right\}$ and current trajectory $traj\left(u_i\right) = \left\{r_1, r_2, \ldots, r_{m_i}\right\}$. With the computed temporal relationship matrix $\boldsymbol{\Delta s}$, $\boldsymbol{\Delta t}$, and the constructed global transfer graph $\mathcal{G}$, we aim to predict the POIs most likely to be visited by users u_i in the future, i.e., $r_{m_i+1} \in l_{pred}$, where l_{pred} is the predicted POI list.

4. Our approach: GeM

4.1. Model structure overview

Fig. 3 illustrates the overall design of our proposed GeM model. This model incorporates several key components.

(1) **MultiModal Embedding Module**

We use the ME module (part I in Fig. 3), to embed and fuse the user trajectory information in a multimodal way. The GPS information is also used to embed the POI into two vectors to obtain a representation of its Gaussian distribution, in preparation for the downstream module.

(2) **Gaussian Embedding Module**

We construct a GE Module (part II in Fig. 3) with a self-attention layer and LSTM to obtain sequential embedding anchors of the users' historical trajectories, placed in the same space with POI Gaussian embedding, which represents the users' preference for POIs and the Gaussian distribution of POIs. It is worth noting that we directly adopt the Mahalanobis distance as the conditional probability distribution of the next POI.

(3) **Multi-hop Graph Transfer Module**

We construct the MGT Module (part III in Fig. 3), first defining a global trajectory graph to reflect the basic POI static transfer frequencies, which goes through the following three modules:

- A Graph Encoder Layer, the embedded POIs are pooled to get the graph information initially, and then the self-attention layer obtains the basic transfer matrix.
- A Multi-hop Transfer Layer that performs matrix computation to obtain the multi-hop distribution through the basic transfer matrix, and adaptively aggregates the two distributions to obtain the multi-hop transfer distribution.
- A Matrix Factorization Layer that employs the users and POIs embedding matrix for Matrix multiplication, and adds the multi-hop transfer distribution to obtain the objective transfer results.

At last, aggregating the results of Gaussian Embedding Module and Multi-hop Graph Transfer Module yields the final recommended POI distribution.

4.2. MultiModal embedding module

4.2.1. User trajectory embedding layer

A multimodal embedding module is used to encode user, location, and time into low-dimensional latent embedding, denoted as $\boldsymbol{e}_u, \boldsymbol{e}_l, \boldsymbol{e}_t \in \mathbb{R}^d$, respectively. Here, two types of temporal embedding are used. The continuous timestamp is divided by $7 \times 24 = 168$ h for ordinary embedding, which captures the time feature in a week. Meanwhile, dividing the continuous timestamps by 24 h and using the time2vec (Kazemi et al., 2019) module for periodic embedding works well to obtain regular features within a day. These two temporal discretizations can represent the exact time of day and within a week, and the two temporal dimensions complement each other well in capturing long and short periodic time information.

The embedding of an hour t in a day, denoted $\boldsymbol{e}_{t2v}$, is a vector of length d. The ith element is defined as

$$\boldsymbol{e}_{t2v}[i] = \begin{cases} \omega_i t + \varphi_i, & \text{if } i = 0 \\ \sin\left(\omega_i t + \varphi_i\right), & \text{if } 1 \le i \le d-1, \end{cases} \tag{4}$$

where ω and φ are learnable parameters. A sin activation function is used to capture the periodic behaviors.

Thus, the input dimensions of the embedding $\boldsymbol{e}_u$, $\boldsymbol{e}_l$ and $\boldsymbol{e}_t$ are U, L and 168 respectively, while the $\boldsymbol{e}_{t2v}$ input is a discrete timestamp of 24. The output of the user trajectory embedding layer for each check-in r is the sum $\boldsymbol{e}_r = \boldsymbol{e}_u + \boldsymbol{e}_l + \boldsymbol{e}_t + \boldsymbol{e}_{t2v} \in \mathbb{R}^d$. For the embedding of each user sequence $seq\left(u_i\right) = \left\{r_1, r_2, \ldots, r_n\right\}$, we denote as $\boldsymbol{E}(\boldsymbol{u}_i) = \left\{\boldsymbol{e}^{r_1}, \boldsymbol{e}^{r_2}, \ldots, \boldsymbol{e}^{r_n}\right\} \in \mathbb{R}^{n \times d}$.

4.2.2. Spatio-temporal embedding layer

The spatio-temporal embedding layer is used to densely represent spatial and temporal differences in terms of hours and hundreds of meters as the base unit, respectively. This layer is multiplied by a unit embedding vector $\boldsymbol{e}_{\Delta s}$ and $\boldsymbol{e}_{\Delta t}$ by the spatial and temporal intervals, respectively. In this layer, we get the dense representations of spatial and temporal relations within sequences. The spatial and temporal differences are embedded in $\boldsymbol{e}^{\Delta s}_{ij}, \boldsymbol{e}^{\Delta t}_{ij} \in \mathbb{R}^d$:

$$\begin{cases} \boldsymbol{e}^{\Delta s}_{ij} = \boldsymbol{\Delta}^{s}_{ij} \times \boldsymbol{e}_{\Delta s} \\ \boldsymbol{e}^{\Delta t}_{ij} = \boldsymbol{\Delta}^{t}_{ij} \times \boldsymbol{e}_{\Delta t}. \end{cases} \tag{5}$$

Similar to previous work on STAN (Luo et al., 2021), we use another interpolated embedding layer:

$$\begin{cases} \boldsymbol{e}^{\Delta s}_{ij} = \dfrac{\boldsymbol{e}^{inf}_{\Delta s}(Upper(\Delta s) - \Delta s) + \boldsymbol{e}^{sup}_{\Delta s}(\Delta s - Lower(\Delta s))}{Upper(\Delta s) - Lower(\Delta s)} \\ \boldsymbol{e}^{\Delta t}_{ij} = \dfrac{\boldsymbol{e}^{inf}_{\Delta t}(Upper(\Delta t) - \Delta t) + \boldsymbol{e}^{sup}_{\Delta t}(\Delta t - Lower(\Delta t))}{Upper(\Delta t) - Lower(\Delta t)}, \end{cases} \tag{6}$$

where Δs represents temporal difference between check-in i and j, $Upper(\Delta s)$ and $Lower(\Delta s)$ represents the upper bound and the lower bound of the interval to which Δs belongs, respectively, and temporal difference is the same.

The final spatial and temporal embeddings are added together to create a matrix of spatio-temporal relationships for the trajectory:

$$\boldsymbol{E}(\boldsymbol{\Delta}) = \mathrm{Sum}(\boldsymbol{E}(\boldsymbol{\Delta s})) + \mathrm{Sum}(\boldsymbol{E}(\boldsymbol{\Delta t})) \in \mathbb{R}^{n\times n}, \tag{7}$$

where $\boldsymbol{E}(\boldsymbol{\Delta s}), \boldsymbol{E}(\boldsymbol{\Delta t}) \in \mathbb{R}^{n\times n\times d}$, which are the embeddings of $\boldsymbol{\Delta s}$ and $\boldsymbol{\Delta t}$, respectively.

4.2.3. Gaussian embedding layer

In this layer, we aim to embed POIs into a Gaussian POI space where all POIs are represented as Gaussian distribution $g = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

More specifically, to limit the complexity of the model and reduce the computational overhead, we assume that the embedding dimensions are uncorrelated (Bojchevski & Günnemann, 2017). Thus $\boldsymbol{\Sigma}$ is considered as a diagonal covariance matrix and can be further represented as a vector, so we have two vectors $\boldsymbol{\mu}_{l_k}, \hat{\boldsymbol{\Sigma}}_{l_k} \in \mathbb{R}^d$. To reduce the model size and alleviate overfitting, the embedding of $\boldsymbol{\mu}_l$ shares parameters with the user trajectory embedding layer. Then, we have the Gaussian mean vector and variance vector of the POI embedding:

$$\boldsymbol{\mu}_l = \boldsymbol{e}_l, \boldsymbol{\Sigma}_l = \mathrm{ELU}\left(\hat{\boldsymbol{\Sigma}}_l\right) + \boldsymbol{1}, \tag{8}$$

where $\boldsymbol{1} \in \mathbb{R}^d$ is an array with a padding value of 1 and ELU is an exponential linear unit. They are both used to ensure that each element of the variance vector is non-negative. Thus, we obtain a Gaussian embedding of each candidate POI $g_l = \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$, where $l \in \boldsymbol{L}$.

4.3. Gaussian embedding module

We assume that there exists a latent space $\boldsymbol{Z} \in \mathbb{R}^d$ in which there exist L distributions about POIs. $\boldsymbol{z}(\boldsymbol{u})$ is learned from the user u's historical trajectory, which satisfies the following conditional probability distribution with respect to the distribution of POIs in the space:

$$p(l|seq(u)) \propto p(l|\boldsymbol{z}(\boldsymbol{u})) \qquad s.t. \quad l \in [1, L]. \tag{9}$$

This is the distribution of recommendations that we aspire to obtain through Gaussian space, thus we will solve the problem from both the perspective of modeling $\boldsymbol{z}$ and capturing the conditional probability relationship between $\boldsymbol{z}$ and $\boldsymbol{p}$.

4.3.1. Latent embedding layer

We propose a self-attention aggregation layer to explicitly add spatio-temporal factor intervals to the interactions. This layer aims to aggregate the intrinsic importance of the users' historical trajectory, providing updated information about the visited nodes for the sequence embedding.

Given a user with a non-padding length m of the embedding trajectory matrix $\boldsymbol{E}(\boldsymbol{u})$ and the spatio-temporal relationship matrix $\boldsymbol{E}(\boldsymbol{\Delta})$, the layer first constructs a mask matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$, with the upper left element $\mathbb{R}^{m\times m}$ is 1 while the other elements are 0. The layer then computes a new sequence $\boldsymbol{S}(\boldsymbol{u}) \in \mathbb{R}^{n\times d}$ as an internal aggregated representation of the user trajectory, after linear projection by different parameter matrices $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d\times d}$ as:

$$\boldsymbol{S}(\boldsymbol{u}) = \mathrm{Attention}(\boldsymbol{E}(\boldsymbol{u})\boldsymbol{W}_Q, \boldsymbol{E}(\boldsymbol{u})\boldsymbol{W}_K, \boldsymbol{E}(\boldsymbol{u})\boldsymbol{W}_V, \boldsymbol{E}(\boldsymbol{\Delta}), \boldsymbol{M}) \tag{10}$$

with

$$\mathrm{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, \boldsymbol{\Delta}, \boldsymbol{M}) = \left(\boldsymbol{M} * \mathrm{Softmax}(\frac{\boldsymbol{QK}^T + \boldsymbol{\Delta}}{\sqrt{d}})\right)\boldsymbol{V}. \tag{11}$$

After obtaining a new sequence $\boldsymbol{S}(\boldsymbol{u})$ that aggregates internal trajectories and spatio-temporal information, the attention sequence is naturally embedded flexibly at a point in latent space by a recurrent neural network (LSTM). So we model the latent vector $\boldsymbol{z}$ of the user u's history trajectory by the following equation:

$$\boldsymbol{z}(\boldsymbol{u}) = \mathrm{LSTM}(\boldsymbol{S}(\boldsymbol{u})), \tag{12}$$

where $\boldsymbol{S}(\boldsymbol{u})$ is of length m (trajectory embeddings after that are masked to 0).

4.3.2. Distance selection layer

After successfully obtaining the latent vector $\boldsymbol{z}(\boldsymbol{u}) \in \mathbb{R}^d$, we assume that the embedding of the POI l satisfies the Gaussian distribution of d dimension $g_l = \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$, so we can compute the conditional probability distribution $p(l|\boldsymbol{z}(\boldsymbol{u}))$ by the Mahalanobis distance as:

$$\begin{aligned} &p(l|\boldsymbol{z}(\boldsymbol{u})) \propto \text{-Ma\_dst}(\boldsymbol{z}(\boldsymbol{u}), \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)) \\ &= -\sqrt{(\boldsymbol{z}(\boldsymbol{u}) - \boldsymbol{\mu}_l)^{\mathrm{T}} \boldsymbol{\Sigma}_l{}^{-1}(\boldsymbol{z}(\boldsymbol{u}) - \boldsymbol{\mu}_l)} \\ &s.t. \quad l \in [1, L], \end{aligned} \tag{13}$$

where $\boldsymbol{z}(\boldsymbol{u})$ represents the dynamic embedding of various information from the user's previous trajectories, $\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l$ represent the mean and variance, respectively.

Finally, the obtained conditional probability distribution of the next POI is as:

$$\begin{aligned} &D(u) = \text{-Ma\_dst}(\boldsymbol{z}(\boldsymbol{u}), \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)) \\ &s.t. \quad l \in [1, L]. \end{aligned} \tag{14}$$

It is also worth noting that the probability of selecting a POI in this module is negatively correlated with distance. In other words, what one expects is to draw the next POIs closer in space so that POIs with similar concepts can be close to each other. Calculating the Mahalanobis distance in this way can well avoid the spatial collapse problem that occurs during the training process, better stabilize the embedding, and recommend training by paying more attention to the relative distance while ensuring that the dimensions are relatively independent.

4.4. Multi-hop graph transfer module

Unlike Gaussian Embedding Module which captures the multimodal embedding of dynamic sequences, in this module, we want to construct a static transfer distribution with the benefit of graph information. We make the following assumptions based on previous work (Rendle et al.,

2010) on matrix factorization:

$$p\,(l|seq(u)) \propto \sum_{i\in[1,m]} p(l|l_{i-th}) + p(l|u) \quad s.t. \quad l \in [1, L], \tag{15}$$

where l_{ith} denotes the ith visit POI in the user's trajectory.

In this hypothesis, the static transfer distribution is related to the interrelationships between the visited POIs and the connection of users to the POIs.

4.4.1. Graph encoder layer

Given a global transfer graph $\mathcal{G}$, we wish to learn the transferability information relations between POIs through the topological information structure of the graph, for which we first use the POI Feature Embedding to obtain the embedding of POI nodes as $\boldsymbol{E}_{l'}$. Since we wish to obtain objective POI intrinsic links through the graph structure, the POIs are embedded here in a separate graph representation space and do not share parameters with the POI embedding of the multimodal embedding layer.

We capture the dynamic transfer distribution between POIs explicitly through a self-attention mechanism, where $\boldsymbol{W}_Q$ reflects the active transfer queries of POIs to other POIs and $\boldsymbol{W}_K$ reflects the passive key values of POIs themselves. This basic attention mechanism is able to model the dynamic transfer of POIs to each other, i.e. the transfer attention matrix consisting of the edge attention scores between all of the POIs, which is presented by Eq. (16).

$$\begin{aligned}\boldsymbol{\Phi}' &= \text{Attention}(\boldsymbol{E}_{l'}\boldsymbol{W}_Q, \boldsymbol{E}_{l'}\boldsymbol{W}_K)\\ &= \text{Softmax}\left(\frac{\boldsymbol{QK}^T}{\sqrt{d}}\right)\end{aligned} \tag{16}$$

The obtained transfer attention matrix reflects the objective intrinsic connections between POIs and can reflect to some extent the general pattern of transfer within POIs, providing an objective view of the users' recommendation results from the POIs themselves. It is worth looking at this simple attention mechanism in terms of matrix factorization, which models the multiplication of item matrices in FPMC.

After this, we aggregate information from the global static transfer map in a simple and efficient way. Due to the excellent results of the spectral GCN in characterizing the overall graph space relations, we use the following operations for the calculation of the Laplace matrix.

In particular, let $\boldsymbol{A} \in \mathbb{R}^{L\times L}$ denote the adjacency matrix of $\mathcal{G}$ and we first compute the normalized matrix $\tilde{\boldsymbol{L}}$ as:

$$\tilde{\boldsymbol{L}} = (\boldsymbol{D} + \boldsymbol{I}_N)^{-1}(\boldsymbol{A} + \boldsymbol{I}_N). \tag{17}$$

In the above equation, $\boldsymbol{D}$ is the degree matrix, and $\boldsymbol{I}_N$ is the unit matrix of $\mathcal{G}$.

In the next step, we aim to get a deeper extraction of graph information. We combine the obtained POI attention scores with graph information. Using the calculated POI basic transfer matrix to weight the normalized Laplace matrix $\tilde{\boldsymbol{L}}$ to obtain the first-hop transfer distribution as:

$$\boldsymbol{\Phi}_1 = \boldsymbol{\Phi}' \odot (\exp(\tilde{\boldsymbol{L}})) \in \mathbb{R}^{N\times N}, \tag{18}$$

where ⊙ denotes elemental multiplication. We transfer the range of the normalized matrix $\tilde{\boldsymbol{L}}$ from [0,1] to [1,e], hoping to follow the principle of enlarging larger original elements while avoiding the appearance of zero values.

4.4.2. Multi-hop transfer layer

Considering the adequate integration of multi-hop information into the hth hop transfer matrix, the following equation is proposed:

$$\boldsymbol{\Phi}_h = (\boldsymbol{\Phi}_1)^h. \tag{19}$$

Naturally, once we have the hth hop transfer matrix $\boldsymbol{\Phi}_h$, we can obtain an objective POI transfer distribution $T(u)$ based on the last m POIs in the user's historical trajectory $traj(u) = (l_1, l_2, \ldots, l_m)$ as:

$$\begin{aligned}T(u) &= \sum_{i\in[1,m]} p(l|l_{i-th})\\ &= \boldsymbol{\Phi}_1[l_m] + \boldsymbol{\Phi}_2[l_{m-1}] + \cdots + \boldsymbol{\Phi}_m[l_1],\end{aligned} \tag{20}$$

where m is the length of the current trajectory, and $\boldsymbol{\Phi}[l_m]$ means the l_mth row of $\boldsymbol{\Phi}$.

However, we do not need to use the multi-hop distribution of all historical trajectories. We only need to consider the transfer distribution of the last h hops, which is substituted by h-hop below. Also, the formula for h hop is as follows:

$$T(u) = \sum_{h=1}^{n} \boldsymbol{\Phi}_h[l_{m-h+1}]. \tag{21}$$

It explicitly describes the multi-hop convergence phenomenon and incorporates the key inverse visit information, which reflects the POI-POI interaction. Considering the varying visitation frequencies of different POIs, we use $\boldsymbol{w}$ to adaptively adjust the weight of each POI in multi-hop transfers. By learning the multi-hop suitability attributes of different POIs, the adaptive aggregation of the transfer distribution is achieved for each hop. The final multi-hop transfer distribution is

$$T(u) = \sum_{h=1}^{n} \boldsymbol{w}_h\boldsymbol{\Phi}_h[l_{m-h+1}], \tag{22}$$

where $\boldsymbol{w}_h \in \mathbb{R}^{1\times L}$ adaptively adjusts the attention scores of different POIs for each h-hop transfer.

4.4.3. Matrix factorization layer

We then model the objective transfer distribution by a method similar to FPMC (Rendle et al., 2010), obtaining the following equation:

$$\begin{aligned}p\,(l|seq(u)) &\propto \sum_{h=1}^{n} p(l|l_{\{m-h+1\}-th}) + p(l|u)\\ &= T(u) + p(l|u)\\ &s.t. \quad l \in [1, L].\end{aligned} \tag{23}$$

Concerning the second term of the equation, to reduce the number of model parameters, the user and POI embedding parameters share with multimodal embedding layer as:

$$p(l|u) \propto \boldsymbol{e}_u\boldsymbol{e}_l. \tag{24}$$

Finally, the product of the user and POI matrices, which reflects the user-POI interaction, is added to obtain the objective transfer distribution $O(u)$ as:

$$O(u) = \lambda_1 * T(u) + \boldsymbol{e}_u\boldsymbol{e}_l. \tag{25}$$

At the same time, we can look at the resultant formula for the objective aspect from the perspective of matrix factorization and Markov chain, changing to an adaptive inverse last h POIs aggregation instead of only based on the last one POI. In addition, a self-attention mechanism that incorporates a global graph structure is used, rather than a pure matrix embedding, as in traditional methods.

4.5. Add layer & Optimization

Finally, the recommendation result $D(u)$ is obtained for the distance selection perspective in the latent Gaussian space, which reflects the user's preferences for POI. The recommendation result $O(u)$ is also obtained for the objective perspective of POI, using the multi-hop transfer attention mechanism and matrix factorization, which captures the intrinsic linkage of POIs and the collaborative information of users. To obtain the final result $Prob(u)$, the two components are aggregated

Table 2
Basic dataset statistics.

Datasets	NYC	TKY
#user	1083	2293
#POI	5135	7873
#check-in	147 938	447 570
Avg #visit per POI	28.809	56.848
Avg #visit per user	136.600	195.190

as follows:

$$Prob(u) = p\left(l_k|seq(u)\right) = \text{Softmax}(D(u) + \lambda_2 * O(u)), \tag{26}$$

where λ_2 is the set aggregation hyper-parameter that controls the recommended percentage of distance selection and objective POI aspects.

We now have the final aggregated POI recommendation results, using cross entropy as the loss function for the next POI recommendation:

$$\mathcal{L} = -\sum_{u\in U}\sum_{m\in H(u)}\sum_{l=1}^{L} \hat{Prob}(u)_{m,l}\log(Prob(u)_{m,l}), \tag{27}$$

where $H(u)$ is the current trajectory length of the user and $Prob(u)_{m,l}$ is the recommendation result for POI l with the current trajectory length m of user u. $\hat{Prob}(u)_{m,l}$ is an indicator equal to 1 if POI l is ground truth and 0 otherwise.

5. Experiments

In this section, we first give a brief introduction to the datasets, the baseline models, and the evaluation metrics. Then, we compare the experimental results of GeM with other baselines. Finally, we explore the impacts of different hyper-parameter settings on the performance of GeM.

5.1. Experimental settings

5.1.1. Datasets

We evaluate our proposed GeM model on two real datasets **NYC** and **TKY** (Yang, Zhang, Zheng, & Yu, 2014). The number of users, locations, and check-ins for each dataset is shown in Table 2. NYC and TKY are both from the Foursquare dataset, which collected user check-ins from April 2012 to February 2013.

In our experiments, we use the original raw dataset, with each record containing the user, POI, GPS coordinates, and timestamp. For the two datasets, most of the data is retained (even though there will be some noise and inactive users). Much previous work has used sliced trajectories with fixed length windows or maximum time intervals, however, this may prevent the model from learning long-term dependencies and lead to a model that focuses more on trajectory-ness than recommendation-ness. For each user with m sign-ins, we divide a dataset into training, validation, and testing datasets. The datasets are formed as order sequences by timestamps, where the first $[1, m-2)$ sequences are for training; the $(m-2)$th is for validations; the $(m-1)$th is for test; the $(2, m-2)$th, $(m-1)$th, mth sequences are set as the labels of the three parts respectively.

5.1.2. Baseline models

We compared our GeM with the following baselines:

- MF (Koren et al., 2009) is a classical approach in recommendation problems that learns potential representations of users and POIs through matrix decomposition.
- FPMC (Rendle et al., 2010) combines matrix decomposition and Markov chains to model user's long-term preferences and sequential behavior.
- LSTM (Hochreiter & Schmidhuber, 1997) is a variant of the RNN model that employs both short-term and long-term sequential patterns.
- ST-RNN (Liu et al., 2016) is a variant of the RNN model adopting the time distance transition matrix to capture user's and sequential patterns.
- LSTPM (Sun et al., 2020) is a sequential recommendation model that combines long-term and short-term features.
- PLSPL (Wu et al., 2020) learns different users' long-term preferences by attention mechanism and short-term preference with LSTM and combines them through two linear layers.
- SASRec (Kang & McAuley, 2018) is a recommendation model based on a self-attention mechanism.
- STAN (Luo et al., 2021) captures peer-to-peer interactions between non-adjacent check-ins using spatio-temporal information from the self-attention layer.
- GETNext (Yang et al., 2022) builds a global trajectory flow graph and a new Transformer model to better exploit a wide range of collaborative signals.
- DCHL (Lai et al., 2024) employs a multi-view disentangled hypergraph learning framework by integrating collaborative, transitional, and geographical information through adaptive fusion and contrastive learning.

5.1.3. Evaluation matrices

Following the previous works, we employ two widely used evaluation metrics of ranking evaluation, including Recall of top-k (k ∈ 1, 2, 5, 10, 20) and mean inverse rank (MRR). Recall@k indicates whether the true POI appears in the POI recommended by top-k. Since Recall@k treats top-k recommendations as an unordered list and ignores the ranking of correct predictions, we use MRR, which measures the recommendation order of correctly recommended POIs in a ranked list of results. Given a dataset with m samples (trajectories), metrics can be formulated as follows:

$$Recall@k = \frac{1}{m}\sum_{i=1}^{m} 1(rank \le k), \tag{28}$$

$$MRR = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{rank}. \tag{29}$$

Each metric is calculated 10 times and averaged.

5.1.4. Training details & Hyper-parameters

We developed our model using the PyTorch framework and experimented on the following hardware platforms (CPU: Intel(R) Xeon(R) Gold 5218R, GPU: NVIDIA GeForce RTX 4090). The key hyper-parameter settings in our model are listed below, the embedding dimension of POI, time, and user are all $d = 50$ for NYC, while $d = 80$ for TKY. The multi-hop graph transfer module converts the input node features to 50-dim. In addition, we use an Adam (Kingma, 2014) optimizer with a 1e−3 learning rate and a 5e−4 weight decay rate, and we apply early stopping when the validation loss does not decrease over 5 epochs. As for the hyper-parameter λ_1 and λ_2 controlling the aggregation of different modules, we set them to 1.0 and 0.6 respectively. At the same time, we set the number of hops h about multi-hop transfer to 10. We use the same settings in two datasets and run each model for 50 epochs. Details of the stability experiments on the hyper-parameter design are shown in Section 5.4.

5.2. Experimental results

A comparison of the performance between our model and the baseline on the two datasets is shown in Tables 3 and 4. We report the top-1, top-2, top-5, top-10, top-20 Recall and MRR.

In terms of the dataset, the difficulty of the next POI recommendation problem depends on the number of POIs, the number of users, and the sparsity of the check-ins, which is also clear from the performance of the models in the two datasets. Learning long-term trajectories becomes relatively straightforward due to the dividing dataset

Table 3
Recommendation performance comparison with baselines on NYC.

	NYC					
	Recall@1	Recall@2	Recall@5	Recall@10	Recall@20	MRR
MF	0.0372	0.061	0.1049	0.1578	0.2383	0.0798
FPMC	0.1065	0.1345	0.2131	0.2962	0.3326	0.1712
LSTM	0.1316	0.1567	0.2724	0.3313	0.3592	0.1883
ST-RNN	0.1461	0.1683	0.2818	0.3537	0.4072	0.2143
LSTPM	0.1597	0.2419	0.3619	0.4228	0.4690	0.2488
PLSPL	0.1731	0.2642	0.3638	0.4423	0.5201	0.2756
SASRec	0.1975	0.2936	0.3965	0.5180	0.5761	0.3019
STAN	0.2081	0.2937	0.4015	0.5224	0.5934	0.3045
GETNext	0.2178	0.3495	0.4674	0.5841	0.6271	0.3496
DCHL	0.2679	0.3404	0.4408	0.4976	0.6177	0.3750
GeM(ours)	**0.2946**	**0.4084**	**0.6026**	**0.6875**	**0.7142**	**0.4224**
Improvement	9.96%	16.76%	28.92%	17.70%	13.88%	12.64%

Table 4
Recommendation performance comparison with baselines on TKY.

	TKY					
	Recall@1	Recall@2	Recall@5	Recall@10	Recall@20	MRR
MF	0.0264	0.0450	0.0732	0.1295	0.1879	0.0862
FPMC	0.0823	0.1346	0.2015	0.2735	0.3403	0.1345
LSTM	0.1298	0.1514	0.2613	0.3068	0.3414	0.1843
ST-RNN	0.1329	0.1532	0.2607	0.3025	0.3678	0.2091
LSTPM	0.1386	0.2079	0.2772	0.3465	0.4059	0.2149
PLSPL	0.1405	0.2271	0.3021	0.3708	0.4274	0.2232
SASRec	0.1451	0.2294	0.3276	0.3825	0.4236	0.2244
STAN	0.1584	0.2475	0.3663	0.4257	0.4851	0.2543
GETNext	0.1801	0.2825	0.4090	0.4753	0.5352	0.2820
DCHL	0.1490	0.2125	0.3283	0.4188	0.5493	0.2852
GeM(ours)	**0.2128**	**0.3131**	**0.4355**	**0.5026**	**0.5540**	**0.3134**
Improvement	18.15%	10.37%	6.47%	5.74%	0.85%	9.88%

approach we take, which results in the majority of the user trajectories being exposed to the models. Nevertheless, we can see that our model significantly outperforms the baselines by large margins. For instance, on the NYC dataset, our model achieves improvements of 9.96%, 16.76%, 28.92%, and 17.70% on Recall@1, Recall@2, Recall@5, and Recall@10, respectively, and outperforms baselines by 12.64% on MRR. Similar trends are observed on the TKY dataset.

Moreover, among all the baseline models, non-deep models such as MF have a relatively good performance due to the way we handle the data making the problem relatively simple. It is also not surprising that RNN models such as LSTM, LSTPM, etc. also perform relatively well. However, models with attention mechanisms, SASRec and STAN, perform better than traditional models due to the inclusion of attention mechanisms that capture spatio-temporal correlation among non-successive visits. The model GETNext performs well in Recall@1, Recall@2, Recall@5, and Recall@10, using a transformer with a graph structure that well aggregates global information to predict the user's next movement point, while DCHL performs better in Recall@20 and MRR, capturing higher-order collaborative signals. It is to be noted that we do not use any semantic information to construct the graph structure, as no other information is used in other baselines of the comparison, for example, in GETNext we remove the POI category information.

5.3. Ablation study

We conduct an ablation study to evaluate the impact of each proposed component on the model's final performance on the NYC and TKY datasets. Particularly, seven experiments are conducted:

w/o-MGT(GE): Using only the Gaussian Embedding module.

w/o-GE(MGT): Using only the Multi-hop Graph Transfer module.

w/o-Multi-hop: A variant of the Multi-hop Graph Transfer module, removing the Multi-hop Transfer mechanism from it.

w/o-Self-attention: A variant of the Multi-hop Graph Transfer module, removing the Self-attention mechanism in Graph Encoder Layer and replacing it with a learning mapping matrix $\boldsymbol{W}$, similar to the design in GETNext (Yang et al., 2022).

w/o-MF: A variant of the Multi-hop Graph Transfer module, removing the Matrix Factorization layer.

w/o-Gaussian: A variant of the Gaussian Embedding module, removing the Gaussian distribution assumption and using Euclidean space for embedding.

Ours: The complete GeM model.

In each experiment, the settings of the remaining components were kept unchanged, with only one or two components being removed or replaced. The results are shown in Tables 5 and 6.

The best performance is achieved for the complete model. For the remaining components, the results reveal that removing the Multi-hop Graph Transfer (MGT) module results in a noticeable decline in performance, especially for metrics like Recall@5 and Recall@20. This suggests that the MGT module plays a crucial role in expanding the recommendation breadth. In contrast, the model shows a larger improvement in Recall@1 and Recall@2 when using only the Gaussian Embedding module. From the results of using the two modules MGT and GE respectively, it is clear that the overall modeling is not a simple distributional fusion, but rather exploits the complementary nature of subjective and objective factors to comprehensively capture user behavioral patterns. Within the MGT module, the most important aspect of the multi-hop graph transfer network is the introduction of the multi-hop mechanism, the removal of which decreases various metrics by 4% to 22% on NYC and 13% to 22% on TKY. When Gaussian embedding is applied instead of Euclidean embedding, the accuracy of recommendations is significantly improved, with 2% to 12% improvements across all metrics. The above results suggest that the Gaussian distribution assumption improves recommendation accuracy, while the Multi-hop Graph Transfer module greatly improves the effectiveness of the model on a variety of breadth recommendations, with the two modules complementarily providing different information. All these components ultimately lead to better overall recommendation performance for GeM.

5.4. Stability study

We further qualitatively analyze the impact of the weight of multi-hop transfer, embedding dimension, the number of multi-hop, and various distance measurements in GeM.

5.4.1. Impacts of λ_1 and λ_2

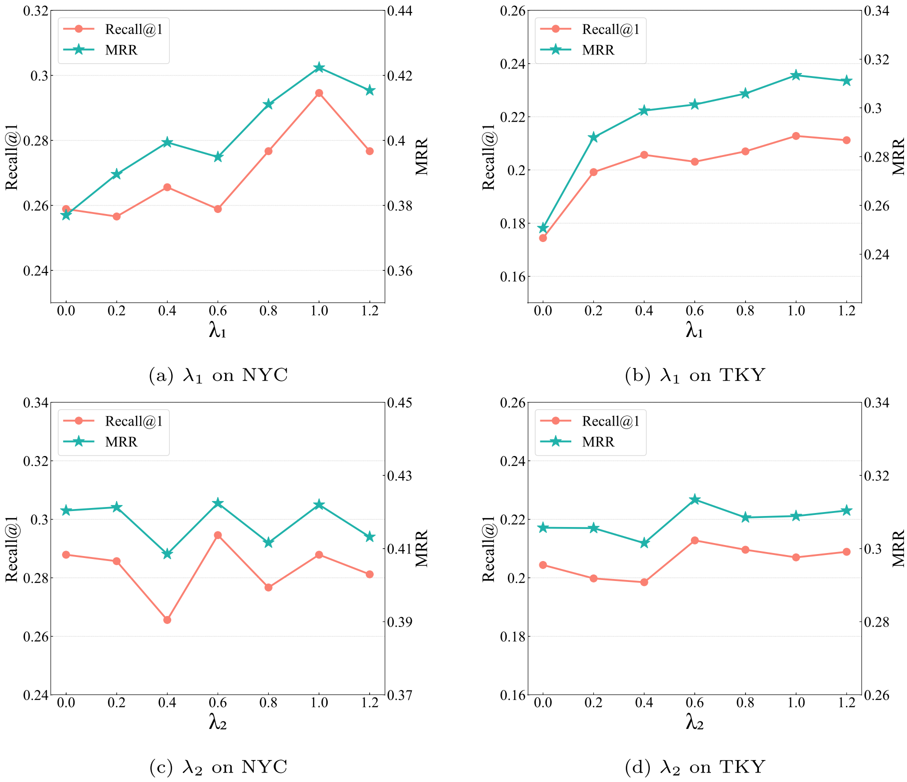
To investigate the impact of the weights of global multi-hop transfer λ_1 and the subjective-objective module aggregation λ_2, we conduct experiments by varying λ_1 and λ_2 within the range $\{0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2\}$. As illustrated in Fig. 4, the performance of our model is influenced by these two parameters on both the NYC and TKY datasets. For λ_1, increasing its value enhances both Recall@1 and MRR up to $\lambda_1 = 1.0$, where the model achieves the best results, with Recall@1 and MRR peaking at approximately 0.30 and 0.42 on the NYC dataset, and around 0.22 and 0.32 on the TKY dataset, respectively. Beyond this point, further increasing λ_1 leads to a slight decline in performance, likely due to overemphasizing global multi-hop transfer. For λ_2, the results are more stable across the tested range, with the optimal performance observed at $\lambda_2 = 0.6$, where both Recall@1 and MRR reach their highest values on both datasets. These findings indicate that $\lambda_1 = 1.0$ and $\lambda_2 = 0.6$ provide the best balance for leveraging global multi-hop transfer and subjective–objective module aggregation, ensuring robust performance across datasets.

Table 5
Ablation study: Comparing the full model with six variants on NYC.

	Recall@1	Recall@2	Recall@5	Recall@10	Recall@20	MRR
w/o-GE(MGT)	0.2589	0.3995	0.5223	0.5803	0.6227	0.3770
w/o-MGT(GE)	0.2879	0.4107	0.5959	0.6808	0.7087	0.4164
w/o-Multi-hop	0.2433	0.3906	0.5245	0.5915	0.6361	0.3691
w/o-Self-attention	0.2633	0.4017	0.5558	0.6093	0.6428	0.3880
w/o-MF	0.2924	0.4040	0.6026	0.6785	0.7098	0.4221
w/o-Gaussian	0.2901	0.3995	0.5535	0.6138	0.6629	0.4016
GeM(ours)	**0.2946**	**0.4084**	**0.6026**	**0.6875**	**0.7142**	**0.4224**

Table 6
Ablation study: Comparing the full model with six variants on TKY.

	Recall@1	Recall@2	Recall@5	Recall@10	Recall@20	MRR
w/o-GE(MGT)	0.1744	0.2539	0.3307	0.3854	0.4361	0.2507
w/o-MGT(GE)	0.2037	0.2968	0.4270	0.5026	0.5377	0.3029
w/o-Multi-hop	0.1875	0.2623	0.3619	0.4140	0.4765	0.2680
w/o-Self-attention	0.1868	0.2688	0.3665	0.4225	0.4733	0.2694
w/o-MF	0.2083	0.3098	0.4231	0.4993	0.5364	0.3068
w/o-Gaussian	0.2018	0.3066	0.4257	0.4863	0.5325	0.3028
GeM(ours)	**0.2128**	**0.3118**	**0.4355**	**0.5026**	**0.5540**	**0.3134**



(a) λ_1 on NYC

(b) λ_1 on TKY

(c) λ_2 on NYC

(d) λ_2 on TKY

Fig. 4. Impact of different value on λ_1 and λ_2.

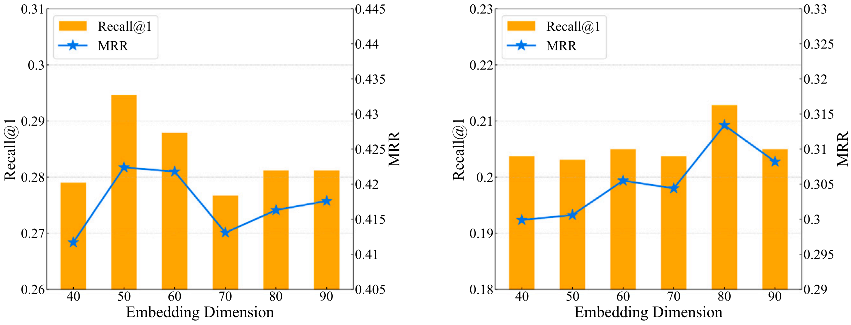
5.4.2. Impact of embedding dimension

We changed the dimensionality of the embedding d from 40 to 90 with step 10 and conducted separate experimental analyses on the two used datasets, with Recall@1 and MRR for metrics. Fig. 5 shows that $d = 50$ is the optimal dimensionality for both trajectory and spatio-temporal embedding on NYC dataset while the module performs best when $d = 80$ on TKY. In general, the recommendation performance of our model is highly sensitive to the embedding dimension's hyper-parameters, likely because the Gaussian embedding approach causes dimensional changes to alter the overall spatial relative positions.

5.4.3. Impact of multi-hop

We conducted experiments to evaluate the impact of varying hop counts on model performance, exploring configurations from 1-hop to 15-hop. As illustrated in Fig. 6, Recall@1 and MRR reach their lowest values at 1-hop, underscoring the model's limited ability to capture transfer regularities when provided with minimal contextual information. As the number of hops increases, both metrics exhibit substantial improvement, peaking at 10 hops. This suggests that a 10-hop range is optimal for capturing the prevalent and meaningful transfer patterns on the datasets.

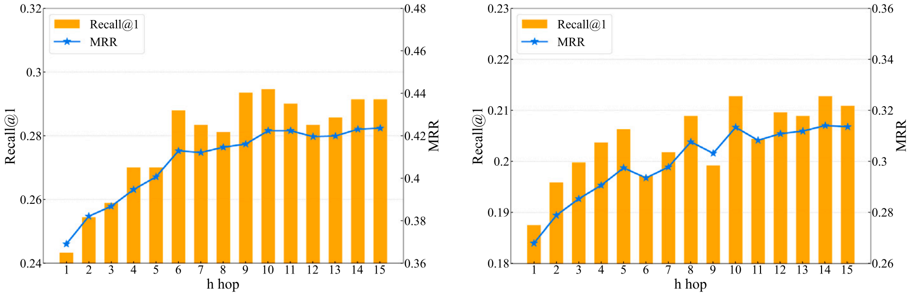
Beyond 10 hops, performance either plateaus or begins to decline slightly. This decline can be attributed to the diminishing relevance of distant interactions, as longer-range dependencies are likely less frequent or more ambiguous in user behavior. Additionally, increasing the hop count beyond this point introduces higher computational and memory overheads, which may outweigh the marginal benefits of incorporating longer-range patterns.



(a) NYC

(b) TKY

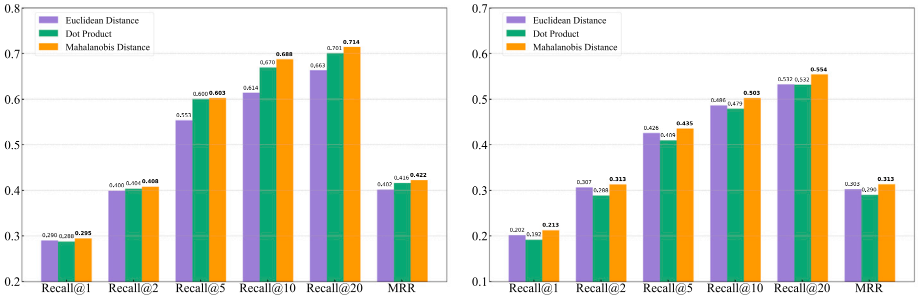
Fig. 5. Impact of different embedding dimensions.



(a) NYC

(b) TKY

Fig. 6. Impact of different multi-hop.



(a) NYC

(b) TKY

Fig. 7. Impact of different distance measurements.

The results reveal that transfer regularities within a 10-hop range are both prominent and consistent across the NYC and TKY datasets. In contrast, patterns involving longer trajectories are less common and contribute minimally to model improvement. This aligns with the observation that user behavior tends to exhibit stronger correlations within shorter, more localized interaction paths, while longer dependencies often include noise or weak connections that dilute their contribution to performance.

5.4.4. *Impact of distance measurements*

We adapt three distance measurements: Euclidean distance, dot product, and Mahalanobis distance, to investigate the impact on GeM as shown in Fig. 7. On both two datasets, the Mahalanobis distance achieves notably higher scores than other metrics across multiple recall points and MRR. Specifically, on NYC dataset, surpasses Euclidean and dot product metrics, with a significant recall advantage, particularly at higher ranks such as Recall@10 and Recall@20. Similarly, on TKY dataset, the Mahalanobis distance leads in almost every recall level, with an evident performance improvement, underscoring its robustness. The observed superior performance of the Mahalanobis distance is due to its capability to probabilistically model the relationship between POIs and user choices, effectively capturing data variance and correlations, which enables GeM to achieve enhanced recall across various levels.

6. Conclusions and future work

In this work, we present GeM, the first POI recommendation model that exploits Gaussian embedding and multi-hop graph transfer mechanisms. We introduce Gaussian distribution assumptions to capture the asymmetrical relation of POIs, and Mahalanobis distance is adopted to calculate the conditional probability distribution for recommendation. Furthermore, based on the global trajectory graph we constructed, we adopt a self-attention layer to obtain a basic transfer matrix, and the final recommendation based on the last ten POIs is obtained through a multi-hop aggregation layer, thus enhancing the learning of the objective connections inherent in POI. Matrix factorization is also used to obtain objective transfer distributions of POI about users. Finally, both subjective and objective aspects are integrated to produce complementary recommendations. We conducted a comprehensive ablation study and stability study in the experimental section to demonstrate the role of each component on recall and MRR metrics. The effect of variations in hyper-parameters on model performance was analyzed. Through a series of experiments on two real datasets, our model significantly outperformed all current state-of-the-art models on the two datasets.

In future work, we will further explore obtaining Gaussian distribution with additional information, for example, by introducing family social relations into the distribution features. And construct various graph structures to enrich the information representation of multi-hop transfers.

CRediT authorship contribution statement

Wenqian Mu: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Data curation, Conceptualization. **Jiyuan Liu:** Writing – original draft, Methodology, Investigation, Conceptualization. **Yongshun Gong:** Supervision, Project administration, Funding acquisition. **Ji Zhong:** Project administration. **Wei Liu:** Supervision, Project administration. **Haoliang Sun:** Supervision, Project administration. **Xiushan Nie:** Supervision. **Yilong Yin:** Supervision, Project administration, Funding acquisition. **Yu Zheng:** Supervision, Project administration.

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.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62476154, 62202270), and the Major Basic Research Project of Natural Science Foundation of Shandong Province (ZR2024ZD03), and the Shandong Excellent Young Scientists Fund (Oversea) (2022HWYQ-044), and the Taishan Scholar Project of Shandong Province (tsqn202306066), and the Natural Science Foundation of Shandong Province (ZR2021QF034)

Data availability

Data will be made available on request.

References

An, Y., Li, Z., Liu, W., Sun, H., Chen, M., Lu, W., et al. (2024). Spatio-temporal graph normalizing flow for probabilistic traffic prediction. In *Proceedings of the 33rd ACM international conference on information and knowledge management* (pp. 45–55).

Bojchevski, A., & Günnemann, S. (2017). Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. arXiv preprint arXiv:1707.03815.

Dong, Z., Meng, X., & Zhang, Y. (2021). Exploiting category-level multiple characteristics for POI recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *35*(2), 1488–1501.

Dos Santos, L., Piwowarski, B., & Gallinari, P. (2017). Gaussian embeddings for collaborative filtering. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 1065–1068).

Fan, Z., Liu, Z., Wang, Y., Wang, A., Nazari, Z., Zheng, L., et al. (2022). Sequential recommendation via stochastic self-attention. In *Proceedings of the ACM web conference 2022* (pp. 2036–2047).

Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A., et al. (2018). Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference* (pp. 1459–1468).

Fu, J., Gao, R., Yu, Y., Wu, J., Li, J., Liu, D., et al. (2024). Contrastive graph learning long and short-term interests for POI recommendation. *Expert Systems with Applications*, *238*, Article 121931.

Gong, Y., Dong, X., Zhang, J., & Chen, M. (2023). Latent evolution model for change point detection in time-varying networks. *Information Sciences*, *646*, Article 119376.

Gong, Y., He, T., Chen, M., Wang, B., Nie, L., & Yin, Y. (2024). Spatio-temporal enhanced contrastive and contextual learning for weather forecasting. *IEEE Transactions on Knowledge and Data Engineering*.

Gong, Y., Li, Z., Liu, W., Lu, X., Liu, X., Tsang, I. W., et al. (2023). Missingness-pattern-adaptive learning with incomplete data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(9), 11053–11066.

Guo, S., Lin, Y., Li, S., Chen, Z., & Wan, H. (2019). Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting. *IEEE Transactions on Intelligent Transportation Systems*, *20*(10), 3913–3926.

Han, P., Shang, S., Sun, A., Zhao, P., Zheng, K., & Zhang, X. (2021). Point-of-interest recommendation with global and local context. *IEEE Transactions on Knowledge and Data Engineering*, *34*(11), 5484–5495.

He, S., Liu, K., Ji, G., & Zhao, J. (2015). Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM international on conference on information and knowledge management* (pp. 623–632).

He, Y., Zhou, W., Luo, F., Gao, M., & Wen, J. (2023). Feature-based POI grouping with transformer for next point of interest recommendation. *Applied Soft Computing*, *147*, Article 110754.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Huang, Z., Ma, J., Dong, Y., Foutz, N. Z., & Li, J. (2022). Empowering next POI recommendation with multi-relational modeling. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 2034–2038).

Jiang, J., Yang, D., Xiao, Y., & Shen, C. (2020). Convolutional gaussian embeddings for personalized recommendation with uncertainty. arXiv preprint arXiv:2006.10932.

Kang, W.-C., & McAuley, J. (2018). Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining* (pp. 197–206). IEEE.

Kazemi, S. M., Goel, R., Eghbali, S., Ramanan, J., Sahota, J., Thakur, S., et al. (2019). Time2vec: Learning a vector representation of time. arXiv preprint arXiv:1907.05321.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Kong, X., Chen, Z., Li, J., Bi, J., & Shen, G. (2024). KGNext: Knowledge-graph-enhanced transformer for next POI recommendation with uncertain check-ins. *IEEE Transactions on Computational Social Systems*.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Lai, Y., Su, Y., Wei, L., He, T., Wang, H., Chen, G., et al. (2024). Disentangled contrastive hypergraph learning for next POI recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval* (pp. 1452–1462).

Li, Y., Chen, T., Luo, Y., Yin, H., & Huang, Z. (2021). Discovering collaborative signals for next POI recommendation with iterative Seq2Graph augmentation. arXiv preprint arXiv:2106.15814.

Li, X., Gong, Y., Liu, W., Yin, Y., Zheng, Y., & Nie, L. (2024). Dual-track spatio-temporal learning for urban flow prediction with adaptive normalization. *Artificial Intelligence*, *328*, Article 104065.

Li, P. L. Y. G. X., & Liang, Y. Z. Y. (2022). TADSAM: A time-aware dynamic self-attention model for next point-of-interest recommendation.

Li, J., Wang, Y., & McAuley, J. (2020). Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining* (pp. 322–330).

Liang, W., & Zhang, W. (2020). Learning social relations and spatiotemporal trajectories for next check-in inference. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(4), 1789–1799.

Lim, N., Hooi, B., Ng, S.-K., Goh, Y. L., Weng, R., & Tan, R. (2022). Hierarchical multi-task graph recurrent network for next poi recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*.

Liu, Q., Wu, S., Wang, L., & Tan, T. (2016). Predicting the next location: A recurrent model with spatial and temporal contexts. *Vol. 30*, In *Proceedings of the AAAI conference on artificial intelligence*.

Long, J., Chen, T., Nguyen, Q. V. H., & Yin, H. (2023). Decentralized collaborative learning framework for next POI recommendation. *ACM Transactions on Information Systems*, *41*(3), 1–25.

Luo, Y., Liu, Q., & Liu, Z. (2021). Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the web conference 2021* (pp. 2177–2185).

Ni, R., Cai, W., & Jiang, Y. (2024). Contrastive cross-domain sequential recommendation via emphasized intention features. *Neural Networks*, *179*, Article 106488.

Qian, C., Feng, F., Wen, L., & Chua, T.-S. (2021). Conceptualized and contextualized gaussian embedding. *Vol. 35*, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 13683–13691).

Rao, X., Chen, L., Liu, Y., Shang, S., Yao, B., & Han, P. (2022). Graph-flashback network for next location recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1463–1471).

Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2010). Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on world wide web* (pp. 811–820).

Song, J., Shen, H., Ou, Z., Zhang, J., Xiao, T., & Liang, S. (2019). ISLF: Interest shift and latent factors combination model for session-based recommendation. In *IJCAI* (pp. 5765–5771).

Sun, K., Qian, T., Chen, T., Liang, Y., Nguyen, Q. V. H., & Yin, H. (2020). Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. *Vol. 34*, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 214–221).

Sun, H., Xu, J., Zhou, R., Chen, W., Zhao, L., & Liu, C. (2021). HOPE: a hybrid deep neural model for out-of-town next POI recommendation. *World Wide Web*, *24*(5), 1749–1768.

Sun, L., Zheng, Y., Lu, R., Zhu, H., & Zhang, Y. (2024). Towards privacy-preserving category-aware POI recommendation over encrypted LBSN data. *Information Sciences*, *662*, Article 120253.

Thaipisutikul, T., & Chen, Y.-N. (2024). An improved deep sequential model for context-aware POI recommendation. *Multimedia Tools and Applications*, *83*(1), 1643–1668.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Vilnis, L., & McCallum, A. (2014). Word representations via gaussian embedding. arXiv preprint arXiv:1412.6623.

Wang, X., Sun, G., Fang, X., Yang, J., & Wang, S. (2022). Modeling spatio-temporal neighbourhood for personalized point-of-interest recommendation. In *Proceedings of the thirty-first international joint conference on artificial intelligence*. International Joint Conferences on Artificial Intelligence Organization.

Wang, K., Wang, X., & Lu, X. (2023). POI recommendation method using LSTM-attention in LBSN considering privacy protection. *Complex & Intelligent Systems*, *9*(3), 2801–2812.

Wang, D., Wang, X., Xiang, Z., Yu, D., Deng, S., & Xu, G. (2021). Attentive sequential model based on graph neural network for next poi recommendation. *World Wide Web*, *24*(6), 2161–2184.

Wang, C., Yuan, M., Yang, Y., Peng, K., & Jiang, H. (2024). Revisiting long-and short-term preference learning for next POI recommendation with hierarchical LSTM. *IEEE Transactions on Mobile Computing*.

Wang, Z., Zhu, Y., Liu, H., & Wang, C. (2022). Learning graph-based disentangled representations for next poi recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 1154–1163).

Wu, B., He, X., Zhang, Q., Wang, M., & Ye, Y. (2022). Gcrec: Graph-augmented capsule network for next-item recommendation. *IEEE Transactions on Neural Networks and Learning Systems*.

Wu, Y., Li, K., Zhao, G., & Qian, X. (2020). Personalized long-and short-term preference learning for next POI recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *34*(4), 1944–1957.

Xie, J., & Chen, Z. (2023). Hierarchical transformer with spatio-temporal context aggregation for next point-of-interest recommendation. *ACM Transactions on Information Systems*, *42*(2), 1–30.

Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., & Wang, S. (2016). Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM international on conference on information and knowledge management* (pp. 15–24).

Xie, P., Zhou, G., Liu, J., & Huang, J. X. (2023). Incorporating global–local neighbors with Gaussian mixture embedding for few-shot knowledge graph completion. *Expert Systems with Applications*, *234*, Article 121086.

Xu, C., Nayyeri, M., Alkhoury, F., Yazdi, H., & Lehmann, J. (2020). Temporal knowledge graph completion based on time series gaussian embedding. In *The semantic web–ISWC 2020: 19th international semantic web conference, Athens, Greece, November 2–6, 2020, proceedings, part I 19* (pp. 654–671). Springer.

Xu, X., Yang, C., Yu, Q., Fang, Z., Wang, J., Fan, C., et al. (2022). Alleviating cold-start problem in CTR prediction with a variational embedding learning framework. In *Proceedings of the ACM web conference 2022* (pp. 27–35).

Yang, D., Fankhauser, B., Rosso, P., & Cudre-Mauroux, P. (2020). Location prediction over sparse user mobility traces using rnns. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence* (pp. 2184–2190).

Yang, S., Liu, J., & Zhao, K. (2022). GETNext: trajectory flow map enhanced transformer for next POI recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 1144–1153).

Yang, X.-J., Wu, Y.-H., Zhang, Z.-H., Wang, J., & Nie, F.-P. (2024). Multitype view of knowledge contrastive learning for recommendation. *Neural Networks*, Article 106690.

Yang, D., Zhang, D., Zheng, V. W., & Yu, Z. (2014). Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *45*(1), 129–142.

Yu, P., Zhang, X., Gong, Y., Zhang, J., Sun, H., Zhang, J., et al. (2025). Enhancing origin–destination flow prediction via bi-directional spatio-temporal inference and interconnected feature evolution. *Expert Systems with Applications*, *264*, Article 125679.

Zhang, J., Liu, X., Zhou, X., & Chu, X. (2021). Leveraging graph neural networks for point-of-interest recommendations. *Neurocomputing*, *462*, 1–13.

Zhang, L., Sun, Z., Wu, Z., Zhang, J., Ong, Y., & Qu, X. (2022). Next point-of-interest recommendation with inferring multi-step future preferences. In *Proceedings of the 31st international joint conference on artificial intelligence* (pp. 3751–3757).

Zhang, L., Sun, Z., Zhang, J., Wu, Y., & Xia, Y. (2022). Conversation-based adaptive relational translation method for next poi recommendation with uncertain check-ins. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhao, P., Luo, A., Liu, Y., Xu, J., Li, Z., Zhuang, F., et al. (2020). Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *34*(5), 2512–2524.

Zhou, F., Qian, T., Mo, Y., Cheng, Z., Xiao, C., Wu, J., et al. (2023). Uncertainty-aware heterogeneous representation learning in POI recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Zhu, Y., Li, H., Liao, Y., Wang, B., Guan, Z., Liu, H., et al. (2017). What to do next: Modeling user behaviors by time-LSTM. *Vol. 17*, In *IJCAI* (pp. 3602–3608).

Zhu, X., Li, L., Liu, W., & Luo, X. (2024). Multi-level sequence denoising with cross-signal contrastive learning for sequential recommendation. *Neural Networks*, *179*, Article 106480.