Information Processing and Management 60 (2023) 103335

Contents lists available at ScienceDirect

Information Processing and Management

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

Improving the spatial–temporal aware attention network with dynamic trajectory graph learning for next Point-Of-Interest recommendation

Gang Cao, Shengmin Cui, Inwhee Joe *

Department of Computer Science, Hanyang University, Seoul 04673, South Korea

ARTICLE INFO

Keywords:
Point-Of-Interest
Attention mechanism
Graph convolution
Dynamic user preference modeling

ABSTRACT

Next Point-Of-Interest (POI) recommendation aim to predict users' next visits by mining their movement patterns. Existing works attempt to extract spatial–temporal relationships from historical check-ins; however, the following critical factors have not been adequately considered: (1) structured features implied in trajectory that reflect individual visit tendency; (2) collaborative signals from other users and (3) dynamic user preference. To this end, we jointly take into full consideration the graph-structured information as well as sequential effects of user trajectory sequences and propose the Trajectory Graph enhanced Spatial–Temporal aware Attention Network (TGSTAN). Given the general preference among users and the shifts of individual interests over time, we present a novel trajectory-aware dynamic graph convolution network module (TDGCN) to facilitate the capturing of local spatial correlations. Specifically, TDGCN dynamically adjusts the normalized adjacency matrix of the trajectory graph by element-wise multiplication with self-attentive POI representations. The local trajectory graph is generated from the same training batch to reflect real-time and collaborative signals, while also following causality. Moreover, we explicitly integrate spatial–temporal interval information with bilinear interpolation to comprehensively attach relative proximity to attention mechanism when capturing long-term dependence. Extensive experiments on three real-world Location-Based Social Networks datasets (Foursquare_TKY, Weeplaces and Gowalla_CA) demonstrate that the proposed TGSTAN consistently outperforms the existing state-of-the-art baselines with an average of 8.18%, 6.59%, and 9.60% improvement on the three datasets, respectively.

1. Introduction

With the rapid growth of Location-Based Social Networks (LBSNs) service providers such as Gowalla, Yelp and Foursquare, sharing check-ins, comments and tips when visiting points of interest (POIs) on social networking platforms has become a prevalent way to socialize among users in recent years (Islam, Mohammad, Das, & Ali, 2022). Consequently, the accumulation of rich user check-in data benefits the POI recommendation systems, which aims to model users' visit preferences and predict the most plausible POI on next movements. In addition to its applications in mobility prediction and route planning, from a business perspective, it also helps to implement more appropriate advertising strategies (Jiang, Qian, Shen, Fu, & Mei, 2015; Yang, Liu, & Zhao, 2022).

Since users' historical trajectories have a profound influence on their current or future behavior patterns, sequential effects play a decisive role in the performance of POI recommendations. How to effectively mine the dependency of historical sequences has

* Corresponding author.
E-mail addresses: cg1106@hanyang.ac.kr (G. Cao), shengmincui@hanyang.ac.kr (S. Cui), iwjoe@hanyang.ac.kr (I. Joe).

https://doi.org/10.1016/j.ipm.2023.103335
Received 9 October 2022; Received in revised form 10 February 2023; Accepted 24 February 2023
Available online 7 March 2023

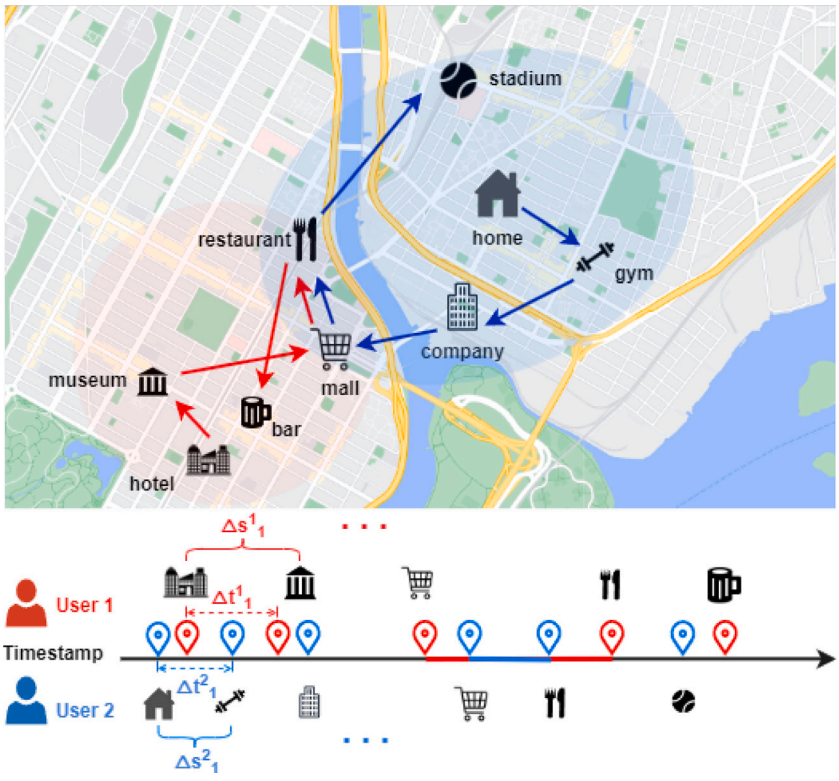


Fig. 1. Two trajectory examples in New York of Weeplaces dataset.

become the focus of many studies (Cheng, Yang, Lyu, & King, 2013; Christoforidis, Kefalas, Papadopoulos, & Manolopoulos, 2021; Islam et al., 2022; Wang, Jiang, Xu, Wang, & Yang, 2022; Yang et al., 2022). Early studies mainly adopt Markov chains (Ye, Zhu, & Cheng, 2013; Zhang, Chow, & Li, 2014) and matrix factorization (Gao, Tang, Hu, & Liu, 2013; Koren, Bell, & Volinsky, 2009; Rendle, Gantner, Freudenthaler, & Schmidt-Thieme, 2011) to model sequential transitions for conventional POI recommendation, which treat users' behavior patterns as static. These traditional methods ignore the fact that user preferences change from time to time and lack the ability to handle sparse sequential data, gradually replaced by neural network-based approaches with the development of deep learning (Luo, Liu, & Liu, 2021; Wang et al., 2022). The pioneer work STRNN (Liu, Wu, Wang, & Tan, 2016) explicitly incorporates temporal and geographic contextual information into RNN models due to its good ability to handle time-series data. Numerous works such as Hidasi, Karatzoglou, Baltrunas, and Tikk (2015), Zhu et al. (2017), Li, Shen, and Zhu (2018), Yang, Sun, Zhao, Liu, and Chang (2017), Zhao et al. (2019) and Feng et al. (2018) extend the LSTM or GRU models to enhance the ability to capture both long-term and short-term dependencies by introducing dedicated spatial and temporal gates to control the flow of contextual information (Lian, Wu, Ge, Xie, & Chen, 2020). Later on, since Self-Attention network (SAN) (Vaswani et al., 2017) show its remarkable potential in handling sequential tasks, SAN-based models such as SASRec (Kang & McAuley, 2018), TiSASRec (Li, Wang, & McAuley, 2020) quickly surpass CNN or RNN-based approaches as the state-of-the-art backbone for sequential recommendation (Wang et al., 2022). Some recent SAN-based works has attempted to further improve the performance of next POI recommendation by introducing hierarchical grids to efficiently exploit geographic information (Lian et al., 2020), considering non-adjacent locations and non-consecutive visits (Luo et al., 2021), and explicitly reflecting spatial and temporal proximity with interval information (Wang et al., 2022).

Previous SAN-based works have adequately explored the temporal and spatial information as well as sequential effects, however, there are still several issues to be resolved. (1) The spatial correlations among POIs (i.e. structural information) are not effectively leveraged, whereas spatial preferences of users can be inferred from the graph-structured check-in sequences. Zhao, Zhang et al. (2020), Yang, Fankhauser, Rosso, and Cudre-Mauroux (2020) and Lian et al. (2020) implicitly capture spatial clustering phenomenon by delineating spatial regions through hierarchical grids, which focus only on adjacent locations and are not capable of reflecting global spatial distances (Luo et al., 2021). Besides, Luo et al. (2021) and Wang et al. (2022) generalize POIs as points without spatial association and reflects proximity only by time interval and geographic distance, which focuses too much on interval information and ignores structural features. A specific example of the trajectories of two users in same day is shown in Fig. 1 and we can observe the spatial aggregation of users' trajectories with their residences, implying non-trivial structural information among POIs to be uncovered. (2) Collaborative signals among users are not properly considered in existing SAN-based recommenders. People tend to be influenced by oral transmission from crowds or commercial propaganda and are hence attracted to specific real-life

scenarios (e.g., mall promotions, influential sport events or performances), which manifests as overlap of users' trajectory segments within a certain range of time periods (both users visit the same mall and restaurant consecutively in Fig. 1). To better exploit these collaborative signals, a recent work GETNext (Yang et al., 2022) designs a user-agnostic global trajectory flow map constructed from historical check-in records of all users and then embedded by a multi-layer Graph Convolution Network (GCN) to represent global transitions among POIs. However, we find that this approach of using global graph to generate POI embeddings is likely to cause information leakage problem. In order to effectively mine the sequential nature of the data, the length of the input sequence is dynamically increased during model training. More specifically, assuming that the current training step is i, the subsequence from subscript 1 to i is the input, and the $i+1$-th POI as label, while the data after the label should be invisible to the model. Apparently, GETNext embeds global POIs prior to training in an attempt to capture generic movement patterns, but ignores causality. (3) The dynamics of personal preferences are still not sufficiently modeled. As illustrated in Yin, Cui, Chen, Hu, and Zhou (2015), users' preferences change over time due to shifting interests. Previous approaches simply treat this characteristic as a sequential prediction task rather than dynamically tracking users' latest preferences in real time.

In order to alleviate the above-mentioned shortcomings, we have done the following innovative work that distinguishes this research from the existing studies. For the sequential effects aspect, we utilize a novel bilinear interpolation solution to integrate the global nature of the spatial–temporal interval while alleviating the data sparsity problem. Then, we inject the above interpolation's embedding into the attention mechanism to facilitate the capture of long-term dependencies. In addition, it is applied to explicitly realign the positional encoding in both temporal and spatial dimensions separately, aiming to enhance the relative proximity of the POI sequence representation from a global perspective. For the structured feature aspect, we introduce graph learning to enhance the modeling of spatial correlations among neighboring POIs. A novel Trajectory-aware Dynamic GCN (TDGCN) module is proposed to exploit user preference from both the POI sequences and trajectory graph simultaneously. We first construct a local trajectory graph based on the current training batch rather than a global graph, so as to absorb the collaboration signals from other users while minimizing the probability of information leakage. The POI representation is then processed by the self-attention mechanism and element-wise multiplied with the normalized adjacency matrix of the trajectory graph for reflecting dynamics. We integrate the above innovative methods into SAN and propose the Trajectory Graph Enhanced Spatial–Temporal Aware Attention Network (TGSTAN for short), which leads to a more stable and reliable performance for next POI recommendation.

Our main contributions can be summarized as follows:

(1) We propose a novel time-sensitive TDGCN module to mine the dynamics of user preferences. TDGCN extracts implied structural feature and absorbs collaborative signals concurrently in a lightweight manner, while minimizing the probability of information leakage.
(2) We apply a new approach to discretize the spatial–temporal information to alleviate the data sparsity problem while comprehensively considering the influence between non-adjacent locations.
(3) We newly design an enhanced positional encoding method for attention mechanism to emphasize relative spatial–temporal proximity.
(4) We propose a novel unified framework called TGSTAN, which first introduces graph learning to SAN-based methods. TGSTAN comprehensively consider spatial–temporal effects, long-term and short-term sequential dependencies, and dynamic user preferences from global and local perspectives to provide more comprehensive and interpretable recommendations.
(5) We demonstrate that the proposed framework consistently outperforms existing state-of-the-art methods significantly. Experimental results also reveal the effectiveness of each key component, as well as the stability and robustness of the proposed model.

The remainder of this paper is organized as follows. Sections 2 and 3 review related works and provide preliminaries for the next POI recommendation, respectively. Then Section 4 details the implementation of our proposed TGSTAN framework. The experimental results and analysis are discussed in Section 5. Finally, we conclude this paper and give future outlook in Section 6.

2. Related work

In this section, we briefly introduce existing representative studies related to our work, including conventional POI recommendation, next POI recommendation and graph-based location recommendation. Then we state the difference between our method and the existing studies.

2.1. Conventional POI recommendation

POI recommendation has been extensively studied over the past years. Early studies default user preference is static and build latent factor models by migrating common methods used in other sequential recommendation tasks (Zhang, Sun, Zhang, Kloeden, & Klanner, 2020), which mainly include Markov chains (Ye et al., 2013; Zhang et al., 2014) and matrix factorization (Gao et al., 2013; Koren et al., 2009; Rendle et al., 2011; Shi, Larson, & Hanjalic, 2014; Zhao, Zhao, Yang, Lyu, & King, 2016). For instance, Zhang et al. (2014) mine sequential patterns by proposed Additive Markov Chain with a location transition graph. Moreover, based on personalized Markov chains (FPMC) proposed by Rendle, Freudenthaler, and Schmidt-Thieme (2010), Cheng et al. (2013) linearly combine Markov chains and matrix factorization method to model personalized movement transition by considering region localization constraint. A recent work (Zhao, Lou, Qian, & Hou, 2020) further improves general matrix factorization algorithm by fusing sentimental attributes with spatial context and introducing a sentiment similarity measure between POIs. In general,

early Markov-based approaches attempt to exploit sequential information between consecutive check-ins to learn the transition probabilities of user movement. But trapped by the sparsity of sequential data, they perform poorly in modeling intermittent visits (Luo et al., 2021). In addition to above methods, collaborative filtering (CF) is also a widely adopted approach in early POI recommendation studies. Li, Ge, Hong, and Zhu (2016), Ye, Yin, and Lee (2010), Ye, Yin, Lee, and Lee (2011), Zhang et al. (2014) employ the user-based CF method that incorporates social influences from friends into the modeling of user preferences. Similarly, Yuan, Cong, Ma, Sun, and Thalmann (2013) perform CF-based recommendation by leveraging the temporal behavior and geographic influence of other users on POIs. Since the CF-based method is generally based on similarity metrics among users (i.e., common historical check-in behavior), the scarcity of check-in information is likely to prevent the similarity from being accurately measured. In addition, the cold-start problem is likewise an inherent drawback of the CF-based approach (Qiao, Luo, Li, Tian, & Ma, 2020).

2.2. Next POI recommendation

With the development of information technology and the accumulation of location-based data, deep learning and advanced embedding methods have garnered attention in recent years in next POI recommendation (Wang et al., 2022; Yang et al., 2022). STRNN (Liu et al., 2016) extends RNN to capture spatial and temporal cyclic effects by constructing two specific transition matrices, where the time and distance intervals between consecutive visits are explicitly represented. Likewise, STGN (Zhao et al., 2019) incorporates time and distance gates into the basic LSTM unit for capturing long-term and short-term preferences. A similar work DeepMove (Feng et al., 2018) employ a recurrent GRU layer and introduce a Historical Attention Module to capture periodicity of human mobility in multi-level. Meanwhile, as the self-attention mechanism (Vaswani et al., 2017) has shown its excellent ability in capturing the long-range dependencies of trajectory sequences, attention-based recommendation models have been proposed one after another. Wu, Li, Zhao, and Qian (2020) combine attention mechanism and LSTM into a unified model to capture users' long-term and short-term preferences respectively. For sequential recommendation task, the pioneering work SASRec (Kang & McAuley, 2018) firstly introduce the self-attention to identify relevant items by leveraging the user's recent interactions. And on the basis of SASRec, TiSASRec (Li et al., 2020) further models relative time intervals as well as absolute positions between interactions explicitly to capture spatial–temporal patterns, then assigns different weights to each item for future interaction prediction with the time-aware self-attention mechanism. To more fully exploit geographic information and capture spatial clustering phenomenon, GeoSAN (Lian et al., 2020) uses a hierarchical map gridding approach to represent GPS coordinates of POIs and encodes them with a self-attention based geography encoder. By taking into account the correlations between non-adjacent locations in non-consecutive visits, STAN (Luo et al., 2021) explicitly construct a trajectory spatial–temporal relation matrix and propose a bi-attention architecture that firstly aggregates key relevant locations and then recalls the most plausible target among candidates with consideration of personalized item frequency (PIF). The most recent state-of-the-art work STiSAN (Wang et al., 2022), propose two lightweight approaches, Time Aware Position Encoder and Interval Aware Attention Block, adding dynamic positional encoding under temporal constraints to enhance sequence representations and focus on capturing spatial relationship with the modified attention layer.

In summary, the above attention-based models attempt to reflect the spatial–temporal relativity of user preferences by explicitly computing the relative temporal and geographic intervals, and then assigning different weights to each POI of the implicit latent representation with attention mechanism. However, all models focus too exclusively on the interval information between check-in records, while overlooking the potential effects of structural features between POIs on spatial transitions to a large extent.

2.3. Graph-based location recommendation

The inherent properties of check-in data reflect the relationship between users and POIs, and the graph structure constructed based on the user–poi relationship can effectively reflect spatial dependencies. Consequently, graph-based methods have gained attention in recent years as an alternative paradigm for POI recommendation (Islam et al., 2022). Informed by the bipartite Session-based Temporal Graph (Xiang et al., 2010), Yuan, Cong, and Sun (2014) propose Geographical–Temporal influences Aware Graph (GTAG) with both exploiting temporal and geographical influences that firstly introduce graph-based approach into POI recommendation task. The tripartite graph GTAG exploit user, POI and session nodes to represent a check-in record and reflecting temporal proximity by adjusting weights of edges. However, the challenge of data sparsity tends to cause graph size explosion due to the introduction of session nodes (Yang et al., 2022). The graph-based embedding model GE (Xie et al., 2016) utilizes four bipartite relational graphs to model spatial and temporal influences as well as sequential and semantic effects, respectively, which are then jointly embedded into a shared low-dimensional space. Then GE makes recommendations based on the similarity of the user's query embedding and the unvisited POI. Another two similar works are JLGE (Christoforidis, Kefalas, Papadopoulos, & Manolopoulos, 2018) and UP2VEC (Qiao et al., 2020). The main difference is that while both use joint representation learning, JLGE constructs two unbipartite (user–user and POI–POI) to better provide personalized recommendations for each user, while UP2VEC focuses on the heterogeneous nature of LBSNs. Moreover, STGCN (Han et al., 2020) fuses the spatial–temporal context information with the proposed user record multigraph and pioneers the application of GCN to POI recommendation.

Note that all the above graph-based methods are only applicable to the conventional POI recommendation scenario rather than sequential recommendation. For next POI recommendation, only few existing graph-based studies have been done. SGRec (Li, Chen, Yin, & Huang, 2021) introduces the Seq2Graph augmentation method for learning POI embeddings, aiming to exploit the collaborative signals of neighbor POI nodes. GNN is endowed with category awareness to learn denser sequential dependencies, thus dealing with the data sparsity problem of POI-wise interactions. Lim et al. (2020) extend Graph Attention Network (Veličković et al.,

Table 1
Notations and corresponding descriptions.

Notations	Descriptions
U, P, T	user, POI and timestamp set
S^u	user u's historical trajectory sequence
n	index of the last check-in(previously unvisited) as target
l	maximum window length for splitting trajectory sequence
d	embedding dimension for latent representations
N	number of stacked encoder blocks
E'	embedding representation of POI with GPS grid encoding
R	spatial–temporal relation matrix
$\Delta t, \Delta s$	time and geography interval
E^A	bilinear interpolation embedding of M
pos	POI's position index
P^T, P^S	temporal and spatial positional embedding
Z	final context embedding
M, Y	mask matrix and output of Interval aware attention layer
D, A, I	degree matrix, adjacency matrix and identity matrix of trajectory graph
$\tilde{H}^{(1)}$	output of Trajectory aware Dynamic GCN
$F^{(N)}$	output of the N stacked encoder blocks
K	number of negative samples
C	candidates embedding
O	output of Target aware decoder
$\hat{y}_{i,j}$	predicted score over POI j at step i
T'	temperature factor in loss function
w	weight for negative samples in loss function

2017) for Next POI Recommendation by representing spatial, temporal and preference factors in POI–POI graphs while neglecting the sequential effect of user trajectories. Instead of capturing local transition patterns with randomly sampling neighbor nodes to augment like SGRec does, the most recent work GETNext (Yang et al., 2022) constructs a unified trajectory flow map that manifest global transition patterns of all POIs to reveal generic movements of users in a global view. The vectorized POI representation of is learned from the trajectory flow graph using a multi-layer GCN and then fused with user, category and time embeddings to be injected into the modified Transformer model. However, since the user-agnostic trajectory flow map is generated based on all check-in records of all users, we doubt that GENext whether guarantee that future information will not be leaked during each training step.

2.4. Summary

Conventional Markov-based or CF-based approaches suffer from data sparsity and cold-start problems, while SAN-based models proposed in recent years mainly focus on mining the sequential effects by encoding time intervals and geographic distances separately. In addition, most previous graph-based approaches do not sufficiently consider the real-time dynamics of users' movement preferences in sequential recommendation. This study differs from existing work in that our proposed method firstly takes temporal as well as spatial influence into account and dynamically mines the structured features implied in user trajectories, in a unified framework. Take a step further, the present approach attempts to fuse the effect of collaborative signals on user preferences, which is still under-studied in deep learning-based sequential recommendation.

3. Preliminaries

In this section, we providing basic term definitions and problem formulation for Next POI recommendation problem. We denote the set of user, POI and timestamp as $U = \{u_1, u_2, \dots, u_{|U|}\}$, $P = \{p_1, p_2, \dots, p_{|P|}\}$ and $T = \{t_1, t_2, \dots, t_{|T|}\}$, respectively. The notations used in this paper and their corresponding meanings are organized in Table 1.

Definition 1. POI: A point of interest (POI) is a specific spatial item associated with a geographic location that a user is likely to visit. Each poi $p \in P$ is represented by a tuple $\langle l, lat, lng \rangle$, i.e., the location id of POI with its associated latitude and longitude coordinates.

Definition 2. Check-In: A user's check-in activity is denoted as a tuple $c = \langle u, p, t \rangle \in |U| \times |P| \times |T|$, which indicates that user u visits poi p at timestamp t.

Definition 3. Historical Trajectory: All check-in records of user u sorted in chronological order forms his/her historical trajectory $S^u = \{c_1^u, c_2^u, \dots, c_{|S^u|}^u\}$ where c_i^u is the ith check-in record of user u.

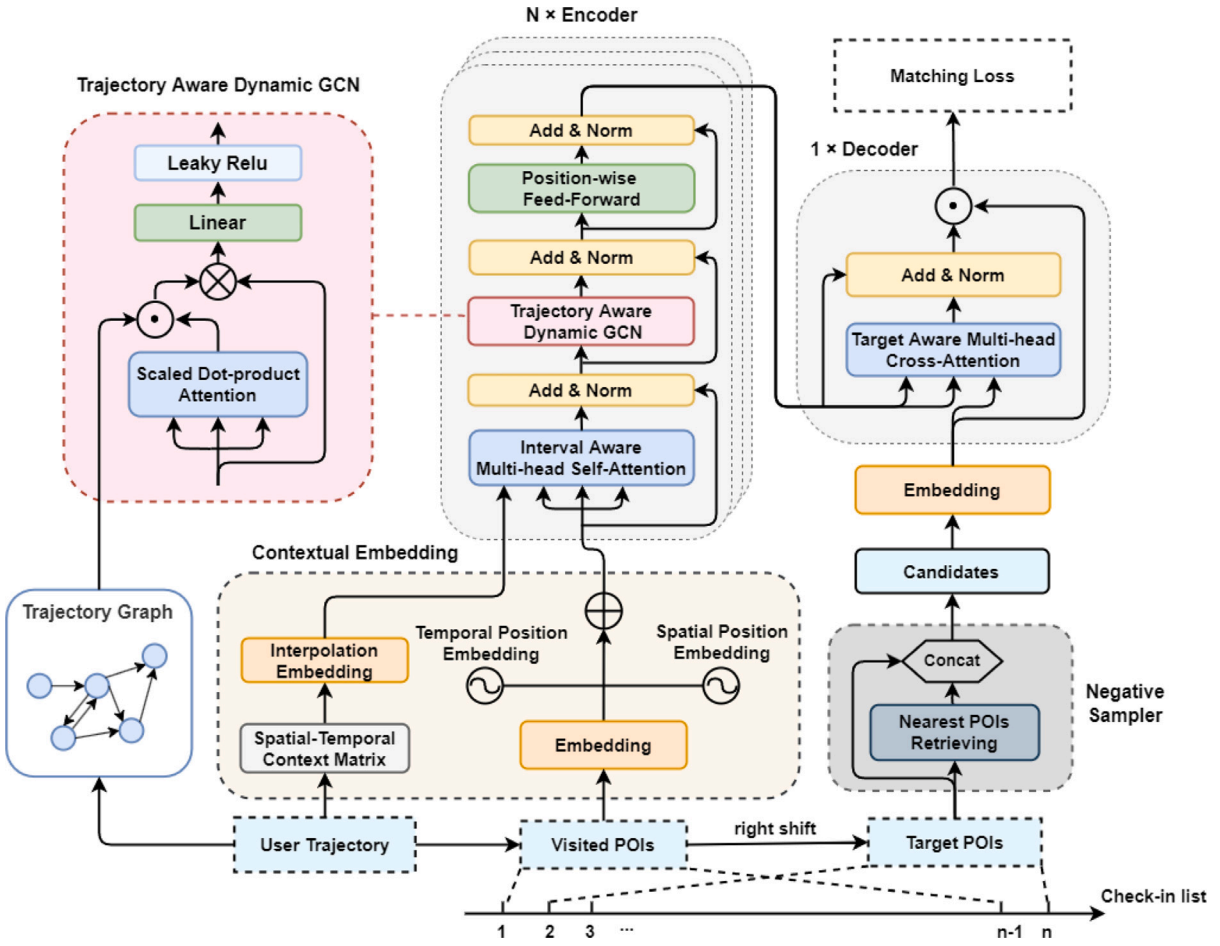


Fig. 2. The architecture of the proposed TGSTAN: the chronological POI sequence is first processed by the contextual embedding layer to obtain a dense representation and fed to the Encoder. Attention layer integrates an interpolation embedding of the contextual matrix to achieve spatial–temporal interval awareness. The local Trajectory Graph is generated from the same training batch of check-ins, which can capture the collaborative signals while preventing information leakage. The trajectory-aware GCN module incorporates attention mechanisms to capture dynamic user preferences in real time. The output of encoder and the candidate embedding generated by the distance-aware sampler are input to the decoder to finally obtain a list of recommended POIs with different weights.

Definition 4. Trajectory Graph: The trajectory graph of check-in sequences is a weighted directed graph, described as $G = (V, E, W)$. V stands for the set of nodes (POIs) and E is the set of weighted edges connecting adjacent nodes. That is, for two nodes $v_1, v_2 \in V$, there exists an edge $e^{(v_1,v_2)} \in E$ between them only if they are adjacent (i.e.,visited consecutively in trajectory). And the weight $w_{e^{(v_1,v_2)}} \in W$ indicates the total number of times this edge appears in the whole trajectories of sequences.

3.1. Problem formulation

The goal of next POI recommendation is to offer user a list of ranked POIs that user is inclined to visit at next timestamp, by modeling user's preference from historical trajectory. Given the user's visited check-in sequence $S^u = c_1^u \to c_2^u \to \cdots \to c_m^u$ where $c_i^u = \langle u, p_i, t_i \rangle$, we aim at predicting the next POI p_{m+1} that are most probably visited at next timestamp t_{m+1}. In order to follow the causal condition that no future data is used to predict future data (Luo et al., 2021), We define the last previously unvisited POI c_n^u in the whole historical trajectory $\{c_1^u, c_2^u, \ldots, c_{|S^u|}^u\}$ as the target POI label for predicting in validation stage, which is invisible to the model training process. We takes the check-in sequence $\{c_1^u, c_2^u, \ldots, c_{n-2}^u\}$ that as source and the subscript right-shifted POI sequence $\{c_2^u, c_3^u, \ldots, c_{n-1}^u\}$ as target during the training process. For each step $i \in \{1, 2, \ldots, n-2\}$, sequence $c_1^u \to c_2^u \to \cdots \to c_i^u$ is used as input and the goal is to predict the $i+1$-th visited POI.

4. Proposed framework

Fig. 2 shows the overall architecture of our proposed model Trajectory Graph Enhanced Spatial–Temporal Aware Attention Network (TGSTAN). TGSTAN consists of three major components: (1) a contextual embedding module that encoding POI sequence

and spatial–temporal proximity in multi-level to obtain denser latent representations; (2) a POI sequential dependency learning Encoder block stacked with an Interval-aware Attention layer, a Trajectory-aware Dynamic Graph Convolution Network module and a Feed-forward network; (3) a Target-aware Attention Decoder (Wang et al., 2022) that extracting user's travel trend preference from weighted POIs representations and ranking a list of candidates for recommendation. More details will be elaborated in the subsequent sections.

4.1. Contextual embedding module

4.1.1. POI sequence embedding

Given that trajectories $S^u = \{c_1^u, c_2^u, \dots, c_{|S^u|}^u\}$ of each user $u \in U$ is not consistent in length, we first divide all trajectory inputs in a uniform length with a fixed-length window. The specific division method is elaborated in Section 5.1.2. Suppose the POI embedding matrix is $E \in \mathbb{R}^{l \times d}$, where $l \in \mathbb{R}$ is the maximum POI sequence length and $d \in \mathbb{R}$ is the embedding dimension. Moreover, in order to exploit geographical information for each POI more efficiently, we employ the map gridding method and geography encoder proposed in Lian et al. (2020) with original implementation,[1] aiming to generate the grid addressing keys of POIs and to encode nearby grids in similar representations, respectively. Then we concatenate POI embedding and the embedded geographic encoding to obtain the enhanced POI latent representation $E' \in \mathbb{R}^{l \times d'}$, where $d' = 2d$.

4.1.2. Spatial–temporal context matrix embedding

We propose a novel bilinear interpolation method to discretize the entire spatial–temporal context, aiming to further alleviate the sparsity problem of check-in data. Luo et al. (2021) point out that the gridding-based encoding method used in Lian et al. (2020) ignores the explicit modeling of time intervals and spatial distances, which mainly aggregates adjacent locations resulting in insufficient ability to capture non-adjacent spatial dependency. To compensate for this, we explicitly build a spatial–temporal relation matrix aiming at reflecting the spatial–temporal relative proximity as well as long-term dependency. For a pair of check-ins c_i^u and c_j^u $(j < i)$ of user u's historical trajectory S^u, the time interval and geographical distance are calculated as $\Delta_{ij}^t = |t_i - t_j|$ and $\Delta_{ij}^s = Haversine(g_i, g_j)$ where g_i represents the GPS coordinate information of the ith POI p_i. To be specific, the spatial–temporal relation matrix $R \in \mathbb{R}^{l \times l \times 2}$ (1) is shown as below:

$$R = \begin{bmatrix} \Delta_{11}^{t,s} & 0 & \dots & 0 \\ \Delta_{21}^{t,s} & \Delta_{22}^{t,s} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_{l1}^{t,s} & \Delta_{l2}^{t,s} & \dots & \Delta_{ll}^{t,s} \end{bmatrix} \tag{1}$$

Note that R is a lower triangular matrix for the sake of preventing information leakage (Kang & McAuley, 2018; Lian et al., 2020; Wang et al., 2022), which ensures that check-in records after the ith subscript are not accessible to the model training at the step i.

Since the spatial–temporal relation matrix R is generated by hundreds of thousands of check-in records, a proper encoding method is needed for preventing data sparsity problem. Inspired by the linear interpolation approach in Liu et al. (2016) and Liu, Wu, and Wang (2017), we introduce a novel bilinear interpolation method to model the impact of continuous spatial–temporal context. The upper and lower bound unit embedding vectors e_{up} and e_{low} are used to represent the explicit intervals via the bilinear interpolation, which are dense representations avoiding a sparse relation encoding. The motivation of this approach is to integrate the relative proximity information in both time and space dimensions from a global perspective, rather than simply adding the temporal and spatial interpolation embeddings together like Luo et al. (2021). The bilinear interpolation embedding is denoted as $E^\Delta \in \mathbb{R}^{l \times l \times d}$ (2), where tu, tl, su, sl represent the upper bound and lower bound of time intervals and geographic distances, respectively.

$$E^\Delta = \frac{e_{up}^{\Delta t}(tu - \Delta t)(su - \Delta s) + e_{low}^{\Delta t}(\Delta t - tl)(su - \Delta s) + e_{up}^{\Delta s}(tu - \Delta t)(\Delta s - sl) + e_{low}^{\Delta s}(\Delta t - tl)(\Delta s - sl)}{(tu - tl)(su - sl)} \tag{2}$$

4.1.3. Interval-scaled positional encoding

We propose the Interval-Scaled Positional Encoding method as an alternative to the traditional approach. The positional encoding of each item in POI sequence is globally reconstructed based on the interpolation embedding of the corresponding dimension, which enhances the attachment of importance to the relative spatial–temporal proximity. The original self-attention mechanism (Vaswani et al., 2017) uses weighted sum function to establish dependencies between inputs and outputs. Nevertheless, attention is not inherently able to model the position of elements in a sequence while the order information is crucial in the recommendation task, hence the need for a positional encoding to capture relative positions. Sine and cosine functions (3) are used to encode positions into d dimension space (Vaswani et al., 2017) where $i \in \{1, 2, \dots, d/2\}$ and pos represents the POI's position. And based on this, we further adopt an interval-scaled temporal positional encoding method informed by Wang et al. (2022), in order to enhance the representation of the relative time proximity without losing the absolute order information. The dynamically adjusted position pos_k is calculated as (4):

$$\begin{cases} P_{(pos,2i)}^T = sin\left(pos/10000^{2i/d'}\right) \\ P_{(pos,2i+1)}^T = cos\left(pos/10000^{2i+1/d'}\right) \end{cases} \tag{3}$$

[1] https://github.com/libertyeagle/GeoSAN

$$pos_k = pos_{k-1} + \frac{E^{\Delta t_{k-1,k}}}{\overline{E^{\Delta t}}} + 1 \tag{4}$$

where $E^{\Delta t_{k-1,k}}$ is the interpolation embedding of time interval between the kth POI's position and the previous position adjacent to it. $\overline{E^{\Delta t}} = \frac{1}{l-1}\sum_{k=2}^{l} E^{\Delta t_{k-1,k}}$ is an average normalization factor to balance the timestamp distribution with variability that exists among different users. The extra "1" is added to ensure that even for an extremely small time interval, its encoding is also distinguishable for model.

We substitute the original method of explicitly computing time interval (Wang et al., 2022) with the embedding of its linear interpolation, which is superior to reflect the global temporal context. Considering that time interval and geographical distance have same properties in the constructed Spatial–Temporal Context Matrix, we further perform a spatial positional encoding on the basis of geographic distance in a similar way. Then we inject the temporal positional encoding $P^T \in \mathbb{R}^{l\times d'}$ and spatial positional encoding $P^S \in \mathbb{R}^{l\times d'}$ into the embedded POI sequence and we obtain the final contextual embedding $Z \in \mathbb{R}^{l\times d'}$ as (5):

$$Z = \begin{bmatrix} E'_1 + P^T_1 + P^S_1 \\ E'_2 + P^T_2 + P^S_2 \\ \vdots \\ E'_l + P^T_l + P^S_l \end{bmatrix} \tag{5}$$

4.2. Encoder

As shown in Fig. 2, the encoder block consists of an Interval Aware Multi-head Self-Attention layer, a Trajectory Aware Dynamic GCN module and a Position-wise Feed-Forward Network with each of the above components performing the residual connection and layer normalization. The details about the implementation of each part are illustrated below.

4.2.1. Interval aware self-attention layer

The multi-head self-attention mechanism firstly proposed in Vaswani et al. (2017) has achieved great success in all kinds of NLP tasks due to the full parallelism and ability to capture long-term dependencies. We adopt this mechanism, which has also been widely used in sequence modeling, by further modifying the attention function to incorporate the spatial–temporal context matrix above, aiming at prompting model to focus on local POI spatial information and enhancing the interpretability for recommendation. Formally calculated as (6),

$$Attention(Q, K, V, E^{\Delta}, M) = softmax\Big(\big(\frac{QK^T}{\sqrt{d_0}} + E^{\Delta}\big) * M\Big)V \tag{6}$$

where $d_0 = d'/h$, h is the number of attention heads and $\sqrt{d_0}$ is a scale factor to avoid the vanishing gradient caused by the normalization of too large dot-product (Cheng, Dong, & Lapata, 2016). $Q, K, V \in \mathbb{R}^{l\times d'}$ are query, key and value vector of sequence representation, where $Q = K = V$ in self-attention. E^{Δ} represents the output of the interpolation embedding of the Spatial–Temporal Context Matrix Δ. In order to blind future data additionally, a mask matrix $M \in \mathbb{R}^{l\times l}$ with the same shape as the attention map $\frac{QK^T}{\sqrt{d_0}} \in \mathbb{R}^{l\times l}$, whose upper triangular elements are filled with "$-\infty$", is multiplied element by element (Luo et al., 2021; Wang et al., 2022). The attention function (6) calculates the correlation weights between queries and corresponding keys and assigns it to each value, summed weights as the attentive results. The multi-head self-attention mechanism ensures this process performed in parallel by projecting vectors into different representation subspaces as (7). Finally, all the attention heads are concatenated and further mapped to get the final output. This interval aware self-attention layer takes the output of contextual embedding module Z, E^{Δ} and mask M as input (8):

$$head_i = Attention\big(QW_i^Q, KW_i^K, VW_i^V, E^{\Delta}, M\big), i \in \{1, 2, \ldots, h_0\} \tag{7}$$

$$Y = MultiHead\big(Z, Z, Z, E^{\Delta}, M\big) = Concat\big(head_1, head_2, \ldots, head_{h_0}\big)W^o \tag{8}$$

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d'\times d_0}$ are the corresponding linear projection matrices of the head i and $W^o \in \mathbb{R}^{d_0\times d'}$ denotes the final output projection matrix.

4.2.2. Trajectory aware dynamic GCN

Our intuition is that users' trajectories not only reveal their routes of movement, i.e., in addition to the explicit temporal and spatial interval features that can be used to modeling, users' distinct travel tendencies are hidden among POIs to be explored. Consequently, in order to further exploit the spatial proximity from of structured POIs, we employ the spectral Graph Convolution Network (GCN) (Kipf & Welling, 2016), which is capable of mining the unstructured patterns hidden in topological information of graphs. Given the Trajectory Graph G and let A represents the adjacency matrix of G. Firstly, we need to construct the normalized Laplacian matrix L of the adjacency matrix, and since G is a directed graph, it is calculated as follows (Yang et al., 2022):

$$L = \big(D + I\big)^{-1}\big(A + I\big) \tag{9}$$

where D, I are the degree matrix and the identity matrix of G, respectively. And the layer-wise propagation rule of GCN is defined as (10):

$$H^{(i)} = \sigma\left(LH^{(i-1)}W^{(i)}\right) \tag{10}$$

where $H^{(i-1)}$ represents the input node representations of ith layer for $i > 0$, which is the output of the previous layer as well. $W^{(i)}$ is the linear transformation matrix and σ is the non-linear activation function. Specifically, GCN learn each node's representation by aggregating the information of its neighboring nodes to generate an intermediate representation firstly. And then after performing the linear projection and non-linear activation, all nodes are updated with information from spatial aggregation.

However, the weights assigned by the original GCN to different neighbors are identical, a shortcoming that limits the model's ability to capture the relevance of POI spatial information. Moreover, the POIs in the sequence injected into the encoder are in chronological order, while the graph convolution operation in (10) is not time sensitive. In other words, weight matrix L corresponding to Trajectory Graph G that contains the interactive relationship among POIs is invariant regardless of the timestamp the time at which the training step is now performed. But considering realistic scenarios, the POIs that a user is inclined to visit at different periods are significantly affected by time (e.g., restaurants are generally only visited during meal times). Accordingly, the constant L might result in the model failing to capture the correlation features between local POIs and the dynamic spatial information under time-varying conditions if we directly adopting the original GCN.

Inspired by Guo, Lin, Wan, Li, and Cong (2021), we propose a novel Trajectory Aware Dynamic GCN (TDGCN) on the basis of spectral GCN (Kipf & Welling, 2016) to address the above issues. A self-attention operation is integrated to adjust the correlation weights among POIs dynamically so that TDGCN is able to learn the dynamic characteristics of POI information across spatial dimension. In our case, we dynamically generate the local trajectory graph of the batch data based on the current training step and then pass it into the TDGCN module, instead of generating a global graph containing all POIs of training set before training as in Yang et al. (2022). One reason for this is conducive to better focus on modeling local trends of users and generate personalized representations. And the other is to prevent information leakage effectively. As shown in Fig. 2, TDGCN takes the output of Interval Aware self-attention layer Y and the normalized Laplacian matrix $\tilde{L}$ (9) computed from the local trajectory graph $\tilde{G}$ of current batch of training data as input. TDGCN is conducted as (12):

$$\tilde{S} = softmax\left(\frac{\tilde{H}^{(i-1)}\tilde{H}^{(i-1)^T}}{\sqrt{d'}}\right) \tag{11}$$

$$\tilde{H}^{(i)} = TDGCN\left(\tilde{L}, \tilde{H}^{(i-1)}\right) = \sigma\left(\left(\tilde{L} \odot \tilde{S}\right)\tilde{H}^{(i-1)}W^{(i)}\right) \tag{12}$$

where $\tilde{L}, \tilde{S} \in \mathbb{R}^{l\times l}$, $\tilde{S}$ is the self-attentive result and $\odot$ represents dot-product operation. Let $\tilde{H}^{(0)} = Y \in \mathbb{R}^{l\times d'}$ be the initial input node representation and σ is a Leaky Relu with leaky rate 0.2 for non-linearity.

TDGCN performs only single-layer graph convolution operation, since the multi-layer graph convolution tried in our experiments did not bring performance improvement. Intuitively, the specially modified single-layer graph convolution is able to sufficiently aggregate spatial neighbor information. The mutual weights among POI nodes are dynamically updated at each training step by dot-producting the self-attentive POI representations $\tilde{S}$ with the weight matrix $\tilde{L}$. After normalization by the softmax function, the larger the element in $\tilde{S}$ is, the stronger the spatial correlation between its corresponding two POIs.

4.2.3. Feed-forward network

In addition to multi-head self-attention layer and DGCN module, we also employ a fully-connected feed-forward network in our decoder block, which is used to endow the representation with non-linear capability as well as to integrate the impact of interactions between different dimensional features (Chen, Zhao, Zhu, Zhuo, & Qian, 2022; Vaswani et al., 2017). The position-wise Feed-forward network contains two distinct linear transformations and a Relu activation function between them (13):

$$F = FFN\left(\tilde{H}^{(1)}\right) = \left(\sigma\left(\tilde{H}^{(1)}W_1 + b_1\right)\right)W_2 + b_2 \tag{13}$$

where $w_1 \in \mathbb{R}^{d'\times d_i}$ and $w_2 \in \mathbb{R}^{d_i\times d'}$ are the two linear transformation matrices, d_i is the dimension of inner-layer and we set $d_i = 4d'$. $b_1, b_2 \in \mathbb{R}^{1\times d'}$ are bias parameters to be learned and $F \in \mathbb{R}^{n\times d'}$ is the output of feed-forward network.

4.2.4. Residual connection & layer normalization

As in Fig. 2, we stacked N encoder blocks, which facilitates capturing hierarchical features of sequence representation. However, such a deep network with multi-layer is prone to gradient disappearance and network degradation. So we further conduct residual connection (He, Zhang, Ren, & Sun, 2016) and layer normalization (Ba, Kiros, & Hinton, 2016) operation to improve model stability and speed up the training process (Vaswani et al., 2017). Suppose x as the input of each component (14):

$$x = LayerNorm\left(x + Sublayer\left(LayerNorm(x)\right)\right) \tag{14}$$

where $SubLayer(\cdot)$ refers to one of the attention layer,dynamic GCN layer and feed forward layer. The normalization performed as (15):

$$LayerNorm(x) = \alpha \odot \frac{x-\mu}{\sqrt{\sigma^2+\epsilon}} + \beta \tag{15}$$

where $\odot$ represents element-wise product, μ, σ stand for the mean and standard deviation of input x, and α, β, ϵ are the parameters for scaling and bias to be learned. Moreover, according to Vaswani et al. (2017), we also adopt dropout operations in each module to avoid overfitting led by rapidly growing number of parameters.

4.3. Target aware cross-attention decoder

According to Lian et al. (2018) and Zhou et al. (2018), many previous attention-based works directly match the output of the attention module with the candidate set, with the consequence that the recommendations are generally sub-optimal. We denote the output of the N encoder blocks as $F^{(N)}$, and adopt a target-aware multi-head cross-attention decoder (TCAD) following Lian et al. (2020), Wang et al. (2022) and Rashed, Elsayed, and Schmidt-Thieme (2022), aiming at enhancing the representations of weighted POIs in user trajectory with respect to candidates and predict the likelihood scores for each candidate according to user preferences. TCAD takes the embedding of candidates as query input while feeding the normalized POI representations $F^{(N)}$ into the decoder block as the keys and values in (16):

$$O = TCAD\left(F^{(N)}|C\right) = Concat\left(Attention\left(CW_h^Q, F^{(N)}W'W_h^Q, F^{(N)}W_h^V\right)\right)_{h=1:h_0} W^o \tag{16}$$

where $C \in \mathbb{R}^{l' \times d'}$ represents the candidates embedding, and $l' = l \times (1 + K)$ since the candidate set consists of the target POI and k negative samples, which is embedded in the same way as in Section 4.1.1. $W' \in \mathbb{R}^{l \times l'}$ is the linear transformation matrix that is used to project queries and keys into latent representations with same dimension. In contrast to (8), spatial–temporal context matrix is no longer added in self-attention operation, but the aforementioned mask is still required to satisfy the causality constraint.

Given the updated representations of user preference $O_i \in \mathbb{R}^{1 \times d'}$ at step i, the probability of each POI candidate to be the next visit is calculated as (17):

$$\hat{\boldsymbol{y}}_{\boldsymbol{i},\boldsymbol{j}} = Sum\left(O_i \odot C_j\right) \tag{17}$$

where $\odot$ is the inner production and $C_j \in \mathbb{R}^{1 \times d'}$ represents the jth POI in the candidates. $Sum(\cdot)$ function does a weighted sum operation on the last dimension of the tensor with the aim of converting the dimension of candidate scores $\hat{\boldsymbol{Y}} \in \mathbb{R}^{(1+K) \times d'}$ to be $\mathbb{R}^{1+K}$.

4.4. Optimization

Kang and McAuley (2018) and Li et al. (2020) use the binary cross-entropy loss to optimize their sequential models, which is not efficient due to the unbalanced ratio of positive and negative samples. Only one negative sample selected randomly, meanwhile a large number of informative negative samples are directly ignored, which might lead to a small gradient when updating loss and slow down the training process. Luo et al. (2021) balance the informativeness and training efficiency by setting a hyperparameter for the number of negative samples, but simply performing random sampling still lacks stability in the effective use of information. According to Lian et al. (2020), the geographic distance information between POIs can be further exploited for more effective negative sampling. When a user u visits POI p_i at timestamp t_i, then the POI that is closer to p_i has a higher probability of being a potential location for the next visit than the POI that is far from p_i empirically. Therefore we pre-retrieve K' previously unvisited POIs nearest to the target POI $\hat{p}$, which contain more effective information than the general ones. Then, we randomly select K POIs among them as negative samples for optimization. We introduce the weighted binary cross-entropy loss function with importance sampling as (18) proposed by Lian et al. (2020) to optimize our model:

$$\mathcal{L} = -\sum_{S^u \in S} \sum_{i=1}^{l} \left(log\sigma\left(\hat{\boldsymbol{y}}_{\boldsymbol{i},\hat{\boldsymbol{p}}_i}\right) + \sum_{k=1}^{K} w_k log\left(1 - \sigma\left(\hat{\boldsymbol{y}}_{\boldsymbol{i},\boldsymbol{k}}\right)\right)\right) \tag{18}$$

$$w_k = \frac{exp\left(\frac{\hat{\boldsymbol{y}}_{\boldsymbol{i},\boldsymbol{k}}}{T'} - ln\tilde{Q}\left(k|i\right)\right)}{\sum_{k=1}^{K} exp\left(\frac{\hat{\boldsymbol{y}}_{\boldsymbol{i},\boldsymbol{k}}}{T'} - ln\tilde{Q}\left(k|i\right)\right)} \tag{19}$$

where w_k represents the weight of the kth negative sample and S is the historical trajectories of all users for training. In (19), T' is the temperature factor used to control the divergence of the negative samples' probability distribution from the uniform distribution. $\tilde{Q}(k|i)$ denotes the unnormalized probability of proposal distribution and $ln\tilde{Q}(k|i)$ is an approximated term to compute normalization in the probability efficiently (Lian et al., 2020). This specialized loss function makes the more informative negative sample contribute more to the gradient, thus speeding up training process.

5. Experiments

5.1. Experimental details

5.1.1. Datasets

We follow recent works (Lian et al., 2020; Luo et al., 2021; Wang et al., 2022; Yang et al., 2022) and select the following three real-world datasets of LBSNs to evaluate our proposed model, which are publicly available and widely used in LBSN recommendation studies:

(1) **Foursquare_TKY**: The Foursquare dataset (Yang, Zhang, Zheng, & Yu, 2014) covers long-term check-in data within New York City and Tokyo from Apr. 2012 to Feb. 2013. We follow choose the data of Tokyo for our experiment and named it "Foursquare_TKY", which contains 573703 check-ins.

Table 2
Datasets statistics (after data pre-processing).

Dataset	Foursquare_TKY	Weeplaces	Gowalla_CA
#user	2263	1357	4107
#POI	7873	18344	13051
#check-in	443732	650576	321203
sparsity	97.51%	97.39%	99.40%
mean traj. length	196.1	479.4	78.2

(2) **Weeplaces**[2]: The Weeplaces dataset is collected from Weeplaces, a location-based service website that aims to visualize users' check-in activity. It now integrates with the APIs of other LBSN services, such as Foursquare, Gowalla and Facebook Places. All of the crawled data including user check-in history, user friendship and location profile is initially generated in Foursquare.

(3) **Gowalla_CA** (Yuan et al., 2013): Gowalla_CA is extracted from the original Gowalla dataset provided by Cho, Myers, and Leskovec (2011) and contains 736148 check-ins within California and Nevada from Feb. 2009 to Oct. 2010 (Yuan et al., 2013).

To ensure the compatibility of the model across different datasets, although certain datasets provide additional information (e.g. POI categories, user relationships, etc.), we only use the original raw datasets where each check-in record contains only the user, POI, GPS coordinates and timestamp information. In the data pre-processing stage, we eliminate unpopular POIs with less than 10 visits and exclude those inactive users with fewer than 20 check-in records to ensure the quality of the data used for training. Key statistics of three datasets after pre-processing are shown in Table 2.

5.1.2. Training strategy

In order to maximize the utilization of check-in sequences in datasets during training, we adopt the following sliced partition strategy with a fixed-length window to split the dataset into train/validation set according to Lian et al. (2020) and Wang et al. (2022). User trajectories are sorted by chronological order and we retrieve the last previously **unvisited** check-in for each user, where the POI is in the future status for each of previous trajectories, denoted as the nth check-in. For the trajectory $\{c_1^u, c_2^u, \dots, c_n^u\}$ of user u,we use $\{c_1^u, c_2^u, \dots, c_{n-1}^u\}$ check-ins for training and $\{c_{n-l}^u, c_{n-l+1}^u, \dots, c_n^u\}$ for validation (the nth check-in as the target label and the most recent previous l check-ins to target as source input). We set the maximum window length $l = 100$ as previous works (Lian et al., 2020; Luo et al., 2021; Wang et al., 2022) do. Longer sequences will be split into non-overlapping sub-sequences of length l from right to left. For shorter part, we repeatedly pad zeros to the left until the sequence length is equal to l in order not to interfere with the gradient updating.

5.2. Evaluation metrics

For effective evaluation, we select 100 previously unvisited POIs nearest to the target POI as negative samples, which together with the target POIs form a plausible candidate list of length 101 for each user's historical trajectory. We choose Hit Rate (HR) and Normalized Discounted Cumulative Gain (NDCG) to evaluate the recommendation performance of the model. HR@k indicates the rate of the target label hits in the top-k probability samples, calculated as (20),

$$HR@k = \frac{\Sigma_{|Valid|}|C_k \cap label|}{|Valid|} \tag{20}$$

where $Valid$ and C_k represents the validation set and the ranked candidates at a cutoff k, respectively. NDCG performs logarithmic discounting based on rank, emphasizing the importance of positions in the recommendation list (Liang, Charlin, McInerney, & Blei, 2016), as formulated in (21),

$$NDCG@k = \frac{1}{I}\sum_{i=1}^{k}\frac{2^{|C_i \cap label|}-1}{log_2(i+1)} \tag{21}$$

where C_i means the top ith ranked sample of candidates and I is the normalization factor "Ideal DCG", which equals to the maximum possible value of $DCG@k$. We report the above two metrics at cutoff $k = 5$ and 10.

5.3. Baseline models

We choose several state-of-the-art models proposed recently as baselines for comparison to evaluate the effectiveness of our proposed approach:

- ST-RNN (Liu et al., 2016): This method extends RNN to model local spatial–temporal features by incorporating time and distance-specific transition matrices.

[2] https://www.yongliu.org/datasets.html

- SASRec (Kang & McAuley, 2018): This method is a classic sequential recommendation framework that introduce self-attention mechanisms to focus on the "relevant" items from historical actions.
- STGN (Zhao et al., 2019): This method designs the time and distance gates in addition to LSTM for long and short term sequential user's preference learning.
- TiSASRec (Li et al., 2020): This method proposes a time-aware self-attention layer that explicitly modeling the relative time intervals for future interactions predicting.
- GeoSAN (Lian et al., 2020): This method adopts a hierarchical gridding representation of GPS coordinates to discretize spatial information and feed it into the proposed self-attention based geography aware encoder.
- STAN (Luo et al., 2021): The method is a state-of-the-art model for next location recommendation. A bi-layer attention framework is proposed to capture the potential features between non-adjacent POIs and non-contiguous check-ins by explicitly exploiting spatial–temporal interval information.
- STiSAN (Wang et al., 2022): The method utilizes a novel time aware position encoder and aggregates the spatial–temporal relation matrix into self-attention mechanisms that outperforms existing state-of-the-art models without requiring much computational burden.

5.4. Experimental settings

We implement our TGSTAN on the Pytorch 1.12.0 and conduct all experiments on the hardware platform with AMD Ryzen 7 3700X 8-Core Processor and Nvidia GeForce RTX 2080Ti GPU. The hyperparameters settings of our model are listed as follows. The enhanced POI representation is concatenated with two $d = 256$ vectors of POI embedding and geographic encoding. The attention head number h_0 is set as 2 so as to make each attention head handle a 128-dim latent representation as the same as Wang et al. (2022). We stack $N = 4$ encoder blocks and pre-retrieve $K' = 1000$ nearest POIs to the target and randomly sample $K = 10$ and 100 negative samples for training and evaluation, respectively. For Foursquare _TKY and Gowalla_CA datasets, the temperature T' and dropout rate are 1.0 and 0.7, while set as 100.0 and 0.3 for Weeplaces. Moreover, we adopt the Adam optimizer with a learning rate of 1e-3 and weight decay rate of 5e-4. The number of training epochs is 35 for each dataset. All baseline models are implemented using the official source code and optimal hyperparameters provided in original papers.

5.5. Recommendation performance

The experimental results of our proposed TGSTAN and baselines are as shown in Table 3. From our observations we can draw the following conclusions:

(1) It is apparent that our TGSTAN outperforms all baseline models by a substantial margin on all three datasets. In terms of HR@5 and NDCG@5 evaluation metrics, we achieve an average performance improvement of up to 7.95% and 10.77% over the best baseline model STiSAN, indicating that TGSTAN provides more relevant and reliable recommendation results for head part of recommendation list.
(2) Generally speaking, since the RNN-based STRNN and LSTM-based STGN are not able to capture the long-term dependencies between POIs, they perform significantly worse than models based on attention mechanisms like SASRec,TiSASRec, etc., despite the consideration of temporal and spatial intervals for modeling. Based on this, the performances of attention-based models are further improved by adequately exploiting the geographic information and interval relationships. Due to the fully considered spatial–temporal matrix embedding, STAN slightly outperforms GeoSAN on the Foursquare_TKY dataset, while GeoSAN's dedicated hierarchical geo-coding approach makes it perform better to some extent on Weeplaces and Gowalla_CA in case of distance-based sampling strategy and importance sampling optimization. STiSAN explicitly considers the spatial–temporal interval information among POIs with two lightweight approaches, time-aware position encoder and interval-aware attention layer, achieving the second best performance.
(3) We observe that all models have roughly similar performance on the Foursuqare_TKY and Weeplaces datasets, but all show a significant decline on the Gowalla_CA dataset. Referring to Table 2, we empirically determine that the main reason for this is that the sparsity of datasets directly affects the recommendation performance of the model. The impact of different data sparsity levels on performance is further discussed in Section 5.8.3.

5.6. Ablation study

To evaluate the effectiveness of each component of our proposed framework, ablation study is conducted in this section. The structure of TGSTAN maintains the same, except that a particular component is removed in each experiments. The following experimental variants are considered: (1) without temporal positional encoding; (2) without spatial positional encoding; (3) without interpolation embedding of spatial–temporal context matrix in attention layer; (4) POI embedding without geographic encoding; (5) without Trajectory-aware Dynamic GCN; (6) without Target-aware Cross-attention Decoder; (7) fusing POI embedding with user embedding and timestamp embedding.

The results of ablation experiments is shown as Table 4 and analyzed as follows:

Table 3
Performance comparison. The best performance scores are in bold and second scores are underlined.

	Foursquare_TKY				Weeplaces				Gowalla_CA			
	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
STRNN	0.1872	0.1217	0.2835	0.1689	0.1753	0.1064	0.2581	0.1745	0.1303	0.0927	0.2124	0.1174
STGN	0.1986	0.1346	0.2803	0.1827	0.1739	0.1169	0.2705	0.1994	0.1575	0.1083	0.2435	0.1301
SASRec	0.2964	0.2321	0.3563	0.2805	0.2965	0.2187	0.4048	0.2574	0.1928	0.1304	0.2770	0.1586
TiSASRec	0.3109	0.2457	0.3738	0.2932	0.3078	0.2358	0.4412	0.2839	0.2015	0.1343	0.2874	0.1647
GeoSAN	0.3728	0.2711	0.4842	0.3058	0.3578	0.2659	0.4712	0.3086	0.2486	0.1552	0.3283	0.2073
STAN	0.3905	0.3048	0.4738	0.3373	0.3424	0.2472	0.4496	0.2941	0.2349	0.1474	0.3037	0.1867
STiSAN	0.4326	0.3414	0.5332	0.3780	0.4297	0.3459	0.5465	0.3842	0.2785	0.2023	0.3675	0.2336
Ours	**0.4702**	**0.3753**	**0.5696**	**0.4074**	**0.4606**	**0.3775**	**0.5643**	**0.4102**	**0.3007**	**0.2291**	**0.3925**	**0.2588**
Improvement	8.69%	9.93%	6.83%	7.78%	7.19%	9.14%	3.26%	6.77%	7.97%	13.25%	6.80%	10.79%

We report the average performance of each model over five runs and perform a two-sample t-test between each baseline and our model. The test results denote that the reported scores are statistically significant (p-value < 0.01), rejecting the H_0 hypothesis.

Table 4
Results of ablation study. Bold indicates the best result.

	Foursquare_TKY				Weeplaces				Gowalla_CA			
	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
Original	**0.4702**	**0.3753**	**0.5696**	**0.4074**	0.4606	0.3775	0.5623	0.4102	**0.3007**	**0.2291**	**0.3925**	**0.2588**
$-P^T$	0.4569	0.3747	0.5320	0.3988	0.4480	0.3744	0.5372	0.4030	0.2681	0.2095	0.3511	0.2362
$-P^S$	0.4459	0.3615	0.5289	0.3882	0.4503	0.3750	0.5453	0.4058	0.2720	0.2109	0.3514	0.2367
$-E^{\Delta}$	0.4543	0.3653	0.5528	0.3972	0.4525	0.3733	0.5505	0.4051	0.2888	0.2267	0.3752	0.2545
$-GE$	0.4282	0.3418	0.5091	0.3677	0.4215	0.3398	0.5284	0.3742	0.2600	0.1940	0.3572	0.2253
$-TDGCN$	0.4591	0.3665	0.5625	0.3997	0.4451	0.3621	0.5343	0.3905	0.2817	0.2185	0.3733	0.2479
$-TCAD$	0.4634	0.3727	0.5625	0.4020	**0.4823**	**0.3983**	**0.5891**	**0.4325**	0.2895	0.2278	0.3750	0.2550
$+E^{U\&T}$	0.4556	0.3666	0.5484	0.3966	0.4451	0.3621	0.5343	0.3905	0.2978	0.2289	0.3876	0.2579

(1) GE is the component that has the greatest impact on the accuracy of recommendation, while each of the other components also contributes to varying degrees to the final performance. Given the drawbacks that GE only aggregates adjacent POIs and is not capable of reflecting spatial distance, the bilinear interpolation of the global spatial–temporal context matrix is introduced as a supplement for further improvement. It can be seen that the performance of model decreases by 2.95%, 2.10%, 4.41% on three datasets in terms of HR@10 when E^{Δ} is not summed with the attention map. This indicates that the interpolation embedding can effectively reflect the spatial–temporal proximity between each check-in of the historical trajectory, which enables the attention mechanism to achieve a more reasonable and more accurate weight assignment.

(2) TDGCN is proved to be effective in mining structural information and learning spatial aggregation. Without the TDGCN, the performance of the model deteriorated by 1.89%, 4.80%, 4.21% in terms of NDCG@10 on three datasets, respectively. Thanks to the consideration of collaborative signals and real-time dynamic modeling of user preferences, TDGCN is able to capture more adequately the spatial correlations between relevant POIs even if GE and E^{Δ} perform well enough. Moreover, removing TDGCN leads to greater performance penalty on the Weeplaces and Gowalla_CA. The check-ins in Foursquare_TKY are collected from relatively centralized geographic areas, with data in Foursquare_TKY constrained in 2,000 km^2, whereas POIs in Gowalla_CA spread over 400,000 km^2 across California and Nevada (Yang et al., 2022). We infer that the E^{Δ} for global embedding does not effectively reflect the spatial aggregation effects among composite regions under actual geographic conditions, which is rectified by TDGCN.

(3) TCAD is only applicable to specific cases. Introducing the TCAD into model brings some degree of performance improvement on the Foursquare_TKY and Gowalla_CA datasets, but suppresses accuracy of the model on Weeplaces. We speculate that this may be due to the lack of spatial–temporal relationship between candidates and current POI in TCAD. Intuitively, the effect of this module is negatively correlated with the average length of the check-in sequences as shown in Table 2. Recall that an attention matching layer similar to TCAD is introduced in STAN (Luo et al., 2021) to recall the most plausible candidates, with the difference that STAN explicitly builds a matrix of spatial–temporal relationships based on all global POI candidates beforehand. While considering the PIF information, however, this approach consumes a lot of computational and spatial resources. As a compromise, the distance-based sampling strategy is used to simulate distance-aware POI transitions, aiming to reduce computational complexity while satisfying the PIF to some extent.

(4) We follow Luo et al. (2021) to encode check-ins comprehensively by fusing embeddings of user, POI and timestamp, where continuous timestamp is mapped into a representation of 168 dimensions (168 h units per week) to reflect the periodicity in weeks. However, the fused embeddings does not lead to a performance improvement. One potential reason is that the additional user and temporal embeddings introduce a deviation from the last POI of input sequence in the embedding space when matching with the candidates.

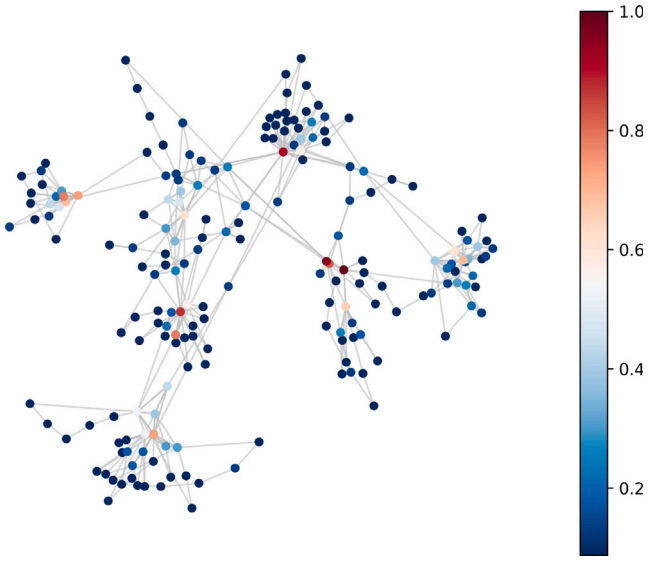


Fig. 3. Trajectory graph visualization on Foursquare_TKY dataset. (The edge weights, directions and self-loops are omitted for better visual presentation).

5.7. Inspecting the TDGCN

5.7.1. Visualization & interpretability of trajectory graph

We randomly select a trajectory graph generated during model training on the Foursquare_TKY dataset for visualization and conduct a case study. Fig. 3 shows the spatial aggregation relationships among all POIs of this graph and check-in counts (after normalization) for each POI. The graph is constructed with a batch size of 8 and contains 195 nodes and 429 edges representing the visited POIs and the consecutive check-in relationships between the corresponding two POIs, respectively. The average degree of the trajectory graph is 4.40, which means that with a general view of the local trajectory graph, user has 4.40 potential movement options in the current state. In addition, the average weight of the edges is 1.43, indicating that a certain edge corresponding to the consecutive visits between two POIs appears in the graph on average 1.43 times. We can observe that the trajectory graph shows a significant spatial clustering phenomenon (with a mean clustering coefficient of 0.275), and the POIs with more check-ins tend to be distributed in the core of the sub-clusters of the graph to assume the role of location transitions. As collaborative signals, such POIs are endowed with greater weight in enhancing the POI sequence representation and play a greater role in recommendation. Thus, the trajectory graph visualization experiment suggests that the structural information implied in the check-in sequences is worth mining.

5.7.2. Influence of TDGCN

To clarify to what extent our TDGCN module can actually promote more accurate recommendations, we further conduct the following experiments to verify the advantages of TDGCN. First, we remove all components that process spatial information (including GE, P^S, E^Δ and TCAD) to avoid potential redundancy with the effects of TDGCN. At this point our model is almost equivalent to a vanilla multi-head self-attention network except that it still contains the P^T and the distance-aware negative sampler. We then tested the following three sets of experiments on each of the three datasets: (1) pure attention mechanism (denoted as w/o GCN); (2) attention mechanism & Spectral GCN in (10) (denoted as GCN); (3) attention mechanism & TDGCN. The experimental results are shown in Fig. 4.

We can clearly see that the model incorporating TDGCN consistently performs best in the three comparisons, where the introduction of TDGCN brings an average improvement of 14.81% compared to pure SAN without GCN, and 3.99% ahead of regular GCN. This proves that taking into account the structural information of user trajectories and the collaborative signals is indeed effective in improving the model performance, and also shows that our TDGCN is more effective in modeling dynamic user preferences because of its time-sensitive nature. Referring to the Table 3, even with the removal of several components, our improved SAN due to graph learning is still very competitive with other baselines, with performance only moderately lagging behind STiSAN. Overall, TDGCN is an easily extendable and effective approach to capture spatial POI correlations.

5.8. Stability study

5.8.1. Embedding dimension

We vary the embedding dimension d for POI sequence and its geographic encoding from 64 to 320 with a step 64. The result is shown in Fig. 5. The optimal d is 256 for Foursquare_TKY and Weeplaces datasets and 192 for Gowalla_CA. Generally, our

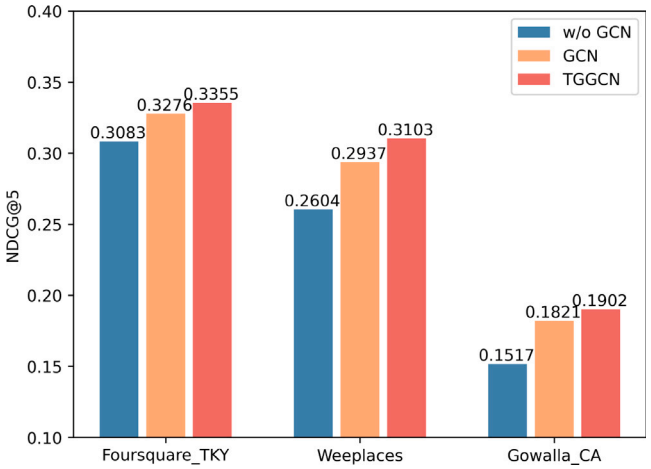
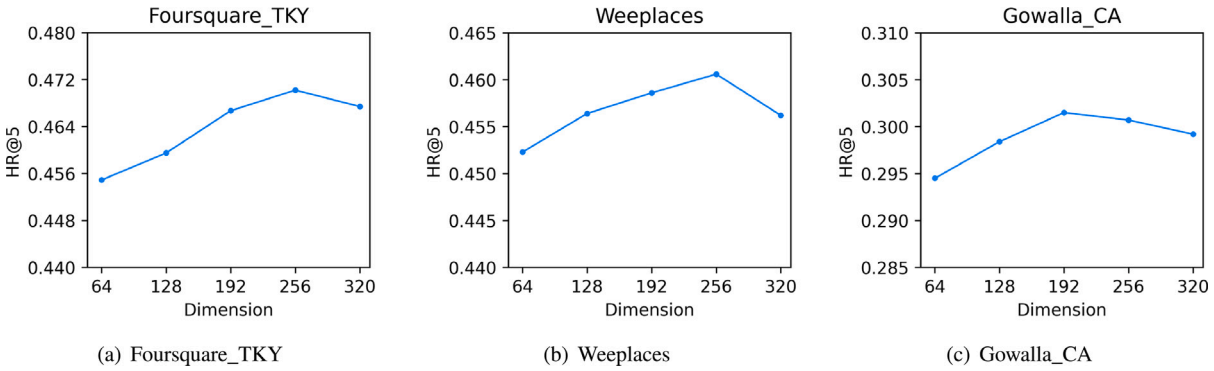


Fig. 4. Influence of graph learning on model performance on three datasets.



(a) Foursquare_TKY (b) Weeplaces (c) Gowalla_CA

Fig. 5. Impact of embedding dimension on model recommendation performance on three datasets.

proposed TGSTAN is insensitive to changes in the embedding dimension. In the case of HR@5, the recommended performance of the model varies only in the range of 3.254%, 1.80% and 2.32% relative to their respective peak performance, which are acceptable fluctuations. In addition to the embedded POI sequence, the model also embeds its hierarchical map gridding representation. This geo-coding implies actual geographic relationships and is associated with the scaling of the map. Intuitively, this spatially discretized encoding is not able to fully exploit its ability to reflect the spatial relationships between POIs when the embedding dimension is not appropriate, resulting in a slight degradation of the model performance.

5.8.2. Number of negative samples

A crucial hyper-parameter for our model is the number of samples of the negative sampler K, who determines the efficiency of the model optimization process. A too small K leads to a loss function that tends to a binary cross-entropy loss and does not take advantage of informative samples, while a too large K leads to an increase in computational effort when calculating the gradient. We set K = [1,5,10,15,20,25] and conduct a series of experiments that other hyper-parameters follow settings as in Section 5.4. As results shown in Fig. 6, for the three datasets, the optimal K for the distance-aware negative sampler is 10,20 and 5, respectively. We observe that the number of negative samples has a significant impact on the improvement of the recommendation performance, which proves the effectiveness of our sampling strategy that emphasizes distance information. Moreover, the statistics in Table 2 reveal that the volume of the dataset is positively correlated with the optimal sample size to some extent.

5.8.3. Threshold for cold user & POI

By adjusting the thresholds to eliminate different levels of inactive users and unpopular POIs, we obtain 5 sets of data with sparsity ranging from 68.31% to 97.39% on the Weeplaces dataset, and the processed data statistics are shown Table 5. we compare our TGSTAN with three recent strong baseline models GeoSAN, STAN, and STiSAN to analyze the stability of the model's recommendation performance as data sparsity varies. The results are shown in Fig. 7. From the figure, we can see that TGSTAN still performs significantly better than the other three models over all sparsity levels. In addition, the performance of all models first increases as the dataset becomes progressively denser and then tends to decrease when a critical value is reached. Clearly, the

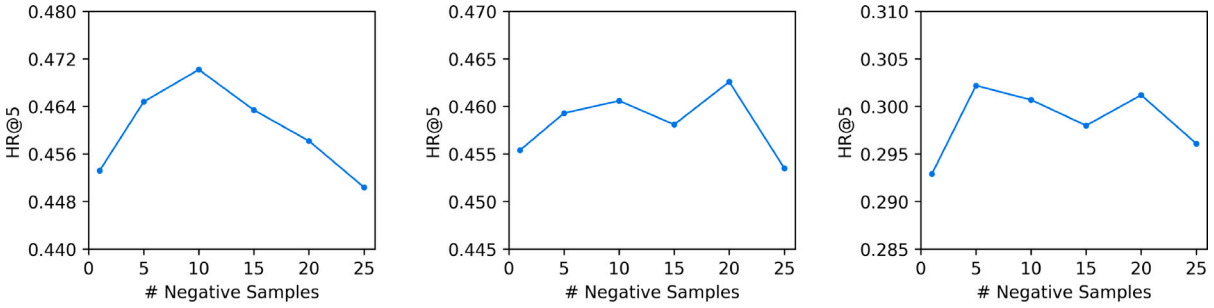


Fig. 6. Impact of the number of negative samples on model recommendation performance on three datasets.

Table 5
Statistics of Weeplaces under different cold user & POI thresholds.

Dataset	Weeplaces				
inactive user threshold	10	20	40	60	80
unpopular POI threshold	20	40	80	120	160
#user	1357	995	532	278	98
#POI	18344	8733	3903	2305	1552
#check-in	650576	465145	240681	126467	48201
mean traj. length	479.4	467.5	452.4	454.9	491.8
sparsity	97.39%	94.65%	88.41%	80.26%	68.31%

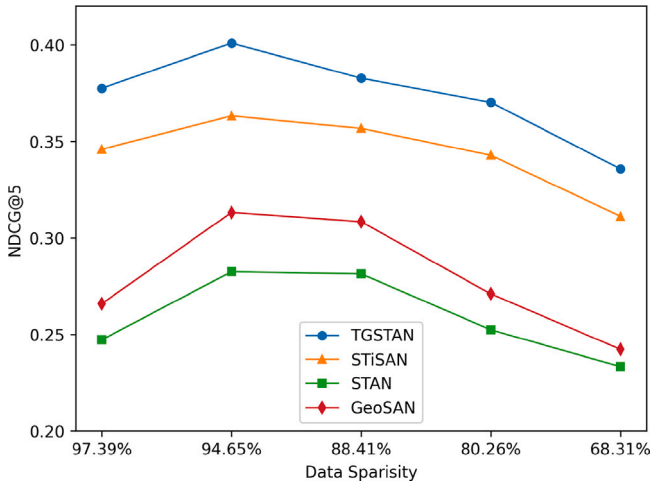


Fig. 7. Impact of different sparsity levels on model recommendation performance on Weeplaces.

most challenging cold-start problem for recommender systems is mitigated when users with shorter trajectories are not considered. However, with increasing thresholds, a large number of users and POIs are filtered out, resulting in a degradation of performance due to the model not having enough training instances to adequately fit the next POI recommendation problem. Overall, our model performs well against a large number of cold users & POIs (e.g. case of 97.39% sparsity) and has good stability with data sparsity greater than 80% (e.g., performance degrades by only 1.93% at sparsity level of 80.26%).

5.9. Complexity analysis

In this section, we analyze the space and time complexity of the key components to characterize the computational efficiency of our proposed model compared to the state-of-the-art model.

5.9.1. Space complexity

In the Contextual Embedding Module, the parameters to be learned are the embedding of global POIs $O(|P|d)$, the upper and lower bounds of the Spatial–Temporal Context Matrix in two dimensions, which equals to $O(4d)$. The Encoder consists of Multi-head

Attention layer, TDGCN, Feed-forward Network and Layer Normalization, contributing to the learnable parameters with $O(4d'^2)$, $O(d'^2)$, $O(8d'^2)$ and $O(6d')$, respectively. Similarly, the number of parameters in the Target Aware Decoder is $O(4d'^2+2d')$. Therefore, the overall space complexity (total number of parameters) of the TGSTAN model is equivalent to $O(|P|d + d'^2 + d')$. Compared to STiSAN (Wang et al., 2022), the Multi-head Attention mechanism we adopt brings additional $O(4d'^2)$ parameters in linear projection matrices of queries, keys, values and final attention output. Besides, it is acceptable to introduce $O(4d + d'^2)$ additional parameters only because of the newly proposed interpolation embedding and TDGCN.

5.9.2. Time complexity

The computational complexity of the TGSTAN mainly contains $O(l^2)$ for bilinear interpolation embedding, $O(l)$ for Interval-scaled Positional Encoding, $O((l^2 + l)d')$ for the Attention layer, $O(|E|d' + l^2d')$ for TDGCN, and $O(ld'^2)$ for Feed-forward Network, respectively. Our additional positional encoding for spatial distance introduces a complexity of $O(l)$, which is negligible compared to $O(l^2d)$ for the general self-attention operation. In addition, the two terms in $O(|E|d' + l^2d')$ of TDGCN denote the graph convolution and spatial attention operations, respectively, where $|E|$ represents the number of edges in the local trajectory graph generated by the current batch.

To better quantify the time complexity of our proposed model, we conduct a time cost experiment on the Foursquare_TKY dataset under the conditions of Section 5.4. Both training and validation batch size are set to 16. The running time of TGSTAN is around 0.215 s per batch on the GPU during the training process and 0.034 s per batch during the validation stage. In comparison, the state-of-the-art model STiSAN (Wang et al., 2022) performs with training and validation times of 0.179 and 0.022 s per batch, respectively, under the same conditions.

5.10. Further discussion

5.10.1. Theoretical & practical implication

The theoretical implications of this study is that we propose a unified framework that combines graph learning and SAN for the next POI recommendation task. A comprehensive consideration of spatial–temporal effects, long-term and short-term dependencies, collaborative signals and the dynamics of user preference leads to more accurate and interpretative recommendations. A novel bilinear interpolation is introduced to refine the positional encoding and augment the attention mechanism for capturing sequential effects. Furthermore, we incorporate graph learning method to capture the spatial clustering relations of POIs, which reflect the implied movement tendencies as well as dynamics.

We further discuss the practical implications of this study below. First, in Section 5.8.3, we illustrate the superior performance of our model compared to previous methods under varying levels of sparse data and cold-start situations, which are two crucial challenges typically faced by various recommender systems. We believe that this model can provide users with relatively reliable POI recommendation services in practical applications, even in the case of sparse data and cold starts. Moreover, our TGSTASN can be extended to be applied in travel service and business promotion, providing route planning solutions and more accurate advertising placement respectively.

5.10.2. Limitations

Information leakage: We mention in Section 2.3 that GETNext (Yang et al., 2022) generates a global trajectory flow map to obtain POI embeddings, which inevitably has a high probability of leading to information leakage during the training iterations. Instead, we use the check-in data from the same batch to construct a local trajectory graph that reflects the dynamics and is lightweight. We do not claim that our approach completely avoids information leakage, but makes the probability of information leakage much lower (from the full amount of POIs in the dataset to the number within a batch). Ji, Sun, Zhang, and Li (2020) indicate that in offline evaluation, not using a global timeline to partition the dataset causes information leakage. However, the proposed timeline scheme has some contradictions with batch-learning recommendation models, i.e., non-incremental learning methods require retraining and it is difficult to decide the number of historical interactions to be used for retraining. An inappropriate retraining strategy may lead to recommendation models that fail to capture dynamic changes in user preferences or forget the long-term preferences of users (Ji et al., 2020). In addition, this approach based on global timeline partitioning may be strongly influenced by the degree of balance in the distribution of user data in the dataset, which we argue is detrimental to personalized recommendations. In summary, to the best of our knowledge, deep learning methods for POI recommendation considering collaborative signals are under-researched for how to avoid information leakage altogether. We build the trajectory graph by batch and recall only the POIs that users have not currently visited to constitute the candidate set as an attempt, but more fine-grained solutions to the above limitations are yet to be explored.

Model complexity: This study focuses on improving the recommendation performance by considering the collaborative signals, structural features between POIs and dynamic preferences by incorporating several innovative approaches including TDGCN, bilinear interpolation and Interval-scaled Position Encoding into existing models. However, another limitation of our proposed model is that the time and space complexity is somewhat higher than that of the state-of-the-art baseline model STiSAN (Wang et al., 2022). How to reduce model size and computational burden to speed up the training process while maintaining the recommended performance is an issue that needs to be addressed in future work.

6. Conclusion & future work

In this paper, we propose an improving Spatial–Temporal Attention Network based on local-level dynamic trajectory graph learning. Specifically, we represent spatial–temporal relative proximity explicitly and incorporate it into the attention layer and its position encoding. In this way, the attention mechanism captures the long-term dependencies of POI sequences more precisely and is able to reflect correlations of non-adjacent locations as well. In addition, we propose a novel dynamic trajectory-aware GCN to update the weights of the graph in real time and learn the user's personalized local spatial relationships. We conduct comparison experiments with the baselines on three real-world datasets and the results show that our proposed TGSTAN surpasses the existing state-of-the-art approaches by a large margin. Ablation study and hyperparameter tuning demonstrate the necessity of each component in TGSTAN in terms of performance improvement and model stability.

For future work, we will further consider social attributes in studying the influence of collaborative factors on user's preferences. In addition, explicitly distinguishing user movement patterns under specific time periods (e.g., weekdays, weekends and holidays) is a potential direction for improvement. And how to make the model lighter by introducing methods such as model compression and pruning is also in need of refinement.

CRediT authorship contribution statement

Gang Cao: Conceptualization, Methodology, Software, Data curation, Writing – original draft. **Shengmin Cui:** Methodology, Writing – review & editing. **Inwhee Joe:** Supervision.

Data availability

All datasets used in this study are public datasets and cited.

Acknowledgments

This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency, South Korea grant funded by the Ministry of Culture, Sports and Tourism in 2022 (Project name: Customized tourism through AI-based tourist situation recognition and tourism information curation development of itinerary recommendation platform technology, Project Number: R2022020116).

References

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *Stat*, *1050*, 21.

Chen, G., Zhao, G., Zhu, L., Zhuo, Z., & Qian, X. (2022). Combining non-sampling and self-attention for sequential recommendation. *Information Processing & Management*, *59*(2), Article 102814.

Cheng, J., Dong, L., & Lapata, M. (2016). Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on empirical methods in natural language processing* (pp. 551–561).

Cheng, C., Yang, H., Lyu, M. R., & King, I. (2013). Where you like to go next: Successive point-of-interest recommendation. In *Twenty-third international joint conference on artificial intelligence*.

Cho, E., Myers, S. A., & Leskovec, J. (2011). Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International conference on knowledge discovery and data mining* (pp. 1082–1090).

Christoforidis, G., Kefalas, P., Papadopoulos, A., & Manolopoulos, Y. (2018). Recommendation of points-of-interest using graph embeddings. In *2018 IEEE 5th International conference on data science and advanced analytics* (pp. 31–40). IEEE.

Christoforidis, G., Kefalas, P., Papadopoulos, A. N., & Manolopoulos, Y. (2021). RELINE: point-of-interest recommendations using multiple network embeddings. *Knowledge and Information Systems*, *63*(4), 791–817.

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).

Gao, H., Tang, J., Hu, X., & Liu, H. (2013). Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM Conference on recommender systems* (pp. 93–100).

Guo, S., Lin, Y., Wan, H., Li, X., & Cong, G. (2021). Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge & Data Engineering*, (01), 1.

Han, H., Zhang, M., Hou, M., Zhang, F., Wang, Z., Chen, E., et al. (2020). STGCN: a spatial-temporal aware graph learning method for poi recommendation. In *2020 IEEE International conference on data mining* (pp. 1052–1057). IEEE.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on computer vision and pattern recognition* (pp. 770–778).

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939.

Islam, M. A., Mohammad, M. M., Das, S. S. S., & Ali, M. E. (2022). A survey on deep learning based point-of-interest (POI) recommendations. *Neurocomputing*, *472*, 306–325.

Ji, Y., Sun, A., Zhang, J., & Li, C. (2020). A critical study on data leakage in recommender system offline evaluation. arXiv preprint arXiv:2010.11060.

Jiang, S., Qian, X., Shen, J., Fu, Y., & Mei, T. (2015). Author topic model-based collaborative filtering for personalized poi recommendations. *IEEE Transactions on Multimedia*, *17*(6), 907–918.

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

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

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

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

Li, H., Ge, Y., Hong, R., & Zhu, H. (2016). Point-of-interest recommendations: Learning potential check-ins from friends. In *Proceedings of the 22nd ACM SIGKDD International conference on knowledge discovery and data mining* (pp. 975–984).
Li, R., Shen, Y., & Zhu, Y. (2018). Next point-of-interest recommendation with temporal and multi-level context attention. In *2018 IEEE International conference on data mining* (pp. 1110–1115). IEEE.
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).
Lian, D., Wu, Y., Ge, Y., Xie, X., & Chen, E. (2020). Geography-aware sequential location recommendation. In *Proceedings of the 26th ACM SIGKDD International conference on knowledge discovery & data mining* (pp. 2009–2019).
Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., & Sun, G. (2018). Xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International conference on knowledge discovery & data mining* (pp. 1754–1763).
Liang, D., Charlin, L., McInerney, J., & Blei, D. M. (2016). Modeling user exposure in recommendation. In *Proceedings of the 25th International conference on world wide web* (pp. 951–961).
Lim, N., Hooi, B., Ng, S.-K., Wang, X., Goh, Y. L., Weng, R., et al. (2020). STP-UDGAT: Spatial-temporal-preference user dimensional graph attention network for next POI recommendation. In *Proceedings of the 29th ACM International conference on information & knowledge management* (pp. 845–854).
Liu, Q., Wu, S., & Wang, L. (2017). Multi-behavioral sequential prediction with recurrent log-bilinear model. *IEEE Transactions on Knowledge and Data Engineering*, *29*(6), 1254–1267. http://dx.doi.org/10.1109/TKDE.2017.2661760.
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*.
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).
Qiao, Y., Luo, X., Li, C., Tian, H., & Ma, J. (2020). Heterogeneous graph-based joint representation learning for users and POIs in location-based social network. *Information Processing & Management*, *57*(2), Article 102151.
Rashed, A., Elsayed, S., & Schmidt-Thieme, L. (2022). CARCA: Context and attribute-aware next-item recommendation via cross-attention. arXiv preprint arXiv:2204.06519.
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).
Rendle, S., Gantner, Z., Freudenthaler, C., & Schmidt-Thieme, L. (2011). Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th International ACM SIGIR Conference on research and development in information retrieval* (pp. 635–644).
Shi, Y., Larson, M., & Hanjalic, A. (2014). Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys*, *47*(1), 1–45.
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*.
Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.
Wang, E., Jiang, Y., Xu, Y., Wang, L., & Yang, Y. (2022). Spatial-temporal interval aware sequential POI recommendation. In *2022 IEEE 38th International conference on data engineering* (pp. 2086–2098). IEEE.
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.
Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., et al. (2010). Temporal recommendation on graphs via long-and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD International conference on knowledge discovery and data mining* (pp. 723–732).
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).
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, C., Sun, M., Zhao, W. X., Liu, Z., & Chang, E. Y. (2017). A neural network approach to jointly modeling social networks and mobile trajectories. *ACM Transactions on Information Systems (TOIS)*, *35*(4), 1–28.
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.
Ye, M., Yin, P., & Lee, W.-C. (2010). Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems* (pp. 458–461).
Ye, M., Yin, P., Lee, W.-C., & Lee, D.-L. (2011). Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on research and development in information retrieval* (pp. 325–334).
Ye, J., Zhu, Z., & Cheng, H. (2013). What's your next move: User activity prediction in location-based social networks. In *Proceedings of the 2013 SIAM International conference on data mining* (pp. 171–179). SIAM.
Yin, H., Cui, B., Chen, L., Hu, Z., & Zhou, X. (2015). Dynamic user modeling in social media systems. *ACM Transactions on Information Systems (TOIS)*, *33*(3), 1–44.
Yuan, Q., Cong, G., Ma, Z., Sun, A., & Thalmann, N. M. (2013). Time-aware point-of-interest recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on research and development in information retrieval* (pp. 363–372).
Yuan, Q., Cong, G., & Sun, A. (2014). Graph-based point-of-interest recommendation with geographical and temporal influences. In *Proceedings of the 23rd ACM International conference on conference on information and knowledge management* (pp. 659–668).
Zhang, J.-D., Chow, C.-Y., & Li, Y. (2014). Lore: Exploiting sequential influence for location recommendations. In *Proceedings of the 22nd ACM SIGSPATIAL International conference on advances in geographic information systems* (pp. 103–112).
Zhang, L., Sun, Z., Zhang, J., Kloeden, H., & Klanner, F. (2020). Modeling hierarchical category transition for next POI recommendation with uncertain check-ins. *Information Sciences*, *515*, 169–190.
Zhao, G., Lou, P., Qian, X., & Hou, X. (2020). Personalized location recommendation by fusing sentimental and spatial context. *Knowledge-Based Systems*, *196*, Article 105849.
Zhao, K., Zhang, Y., Yin, H., Wang, J., Zheng, K., Zhou, X., et al. (2020). Discovering subsequence patterns for next POI recommendation. In *IJCAI* (pp. 3216–3222).
Zhao, S., Zhao, T., Yang, H., Lyu, M. R., & King, I. (2016). STELLAR: Spatial-temporal latent ranking for successive point-of-interest recommendation. In *Thirtieth AAAI Conference on artificial intelligence*.
Zhao, P., Zhu, H., Liu, Y., Xu, J., Li, Z., Zhuang, F., et al. (2019). Where to go next: A spatio-temporal gated network for next poi recommendation. In *Proceedings of the AAAI Conference on artificial intelligence* (p. 5877).
Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., et al. (2018). Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International conference on knowledge discovery & data mining* (pp. 1059–1068).
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.. *17*, In *IJCAI* (pp. 3602–3608).