Learning Graph-based Disentangled Representations for Next POI Recommendation

Zhaobo Wang
w19990112@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Yanmin Zhu*
yzhu@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Haobing Liu
liuhaobing@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Chunyang Wang
wangchy@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

ABSTRACT

Next Point-of-Interest (POI) recommendation plays a critical role in many location-based applications as it provides personalized suggestions on attractive destinations for users. Since users' next movement is highly related to the historical visits, sequential methods such as recurrent neural networks are widely used in this task for modeling check-in behaviors. However, existing methods mainly focus on modeling the sequential regularity of check-in sequences but pay little attention to the intrinsic characteristics of POIs, neglecting the entanglement of the diverse influence stemming from different aspects of POIs. In this paper, we propose a novel Disentangled Representation-enhanced Attention Network (DRAN) for next POI recommendation, which leverages the disentangled representations to explicitly model different aspects and corresponding influence for representing a POI more precisely. Specifically, we first design a propagation rule to learn graph-based disentangled representations by refining two types of POI relation graphs, making full use of the distance-based and transition-based influence for representation learning. Then, we extend the attention architecture to aggregate personalized spatio-temporal information for modeling dynamic user preferences on the next timestamp, while maintaining the different components of disentangled representations independent. Extensive experiments on two real-world datasets demonstrate the superior performance of our model to state-of-the-art approaches. Further studies confirm the effectiveness of DRAN in representation disentanglement.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Point-of-Interest, Next POI Recommendation, Disentangled Representation Learning, Graph Convolution Networks

ACM Reference Format:
Zhaobo Wang, Yanmin Zhu, Haobing Liu, and Chunyang Wang. 2022. Learning Graph-based Disentangled Representations for Next POI Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477495.3532012

1 INTRODUCTION

Point-of-Interest (POI) recommender systems lie at the heart of rapidly expanding location-based social networks (LBSNs), due to their success in alleviating the information overload problem by recommending attractive places to users [14, 22]. One of the most distinguished features of POI recommendation is that geographical influence of POIs have a marked impact on users' decisions [5, 30, 36]. For example, users prefer to visit nearby POIs owing to the distance restriction. As a prominent natural extension of general POI recommendation, next POI recommendation attracts increasing attention from both academic and industrial fields [6, 27, 37]. The key functionality of next POI recommendation is to help users explore the most appropriate destination at a specific time point, which requires capturing user dynamic preferences by mining users' historical check-in sequences and geographical information of POIs.

Assuming that users' next movement is highly related to previous check-in sequences, the majority of existing approaches resort to sequential models for human mobility patterns modeling. Traditional methods [3, 26] adopt Markov Chains to capture the influence of previous visit behaviors on users' next decisions. Due to the strong capacity for handling sequential data, Recurrent Neural Networks (RNNs) based models have become the dominating solutions in this field [4, 7, 21, 38]. In addition, since the spatial effect is of crucial importance to next POI recommendation, spatial and temporal information has been leveraged to learn latent representations of both POIs and users. The most popular scheme is to directly incorporate spatial or temporal intervals between two successive visits into recurrent hidden states [13, 39]. Other attempts turn to extend RNNs with additional gate mechanisms for capturing spatio-temporal factors within sequences [27, 41]. Furthermore, recent state-of-the-art methods [18, 23] aim to employ the

[*] Corresponding Author.

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07...$15.00
https://doi.org/10.1145/3477495.3532012

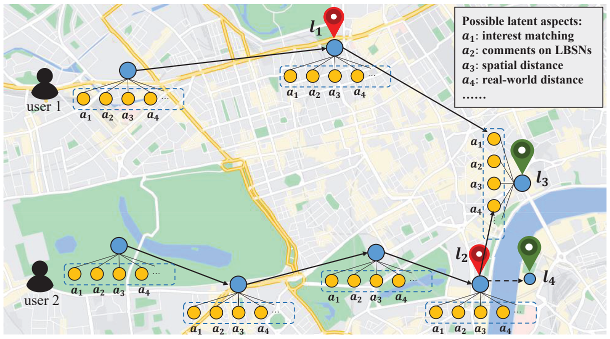


Figure 1: An illustration of diverse aspects of POIs. Red pushpins point out the current location of two users, while green pushpins point out the candidate POIs.

self-attention mechanism [29] for sequences modeling. To further utilize the geographical information, these methods propose to enhance the modeling process by explicitly considering the spatial relation matrices or collaborative signals of check-in sequences.

Despite effectiveness, the entangled nature of geographical relations is ignored by the aforementioned works. We argue that the prior methods of incorporating geographical information fail to model the complex influence of POIs. (1) First, the diverse latent influence of POI on users’ transition are not disentangled effectively. Generally, the characteristics of POIs stem from several aspects, while different aspects may have distinct influence. It is common for dissimilar users to visit the same POI driven by the distinct aspects of POIs. Figure 1 shows an example. u_1’s visit with POI l_3 due to the personal interests in the features of l_3 (e.g., nice surroundings of l_3), even though l_3 is far from her current location. On the other hand, u_2 focuses on the function of l_3 (e.g., restaurant), and the transition is mainly motivated by the short travel distance between her current location l_2 and l_3. However, the entanglement of such different aspects within POIs is extensively overlooked by the black-box representation modeling process utilized by existing methods, leading to the low expressiveness and non-robustness of POI representations. (2) Second, the distance-based influence of POIs are also oversimplified in most prior methods. In Figure 1, l_4 and l_3 are similar sites, while l_4 is closer to l_2 from the spatial distance perspective. However, u_2 may have less interest in l_4 because of the physical barrier (i.e., the river) between l_2 and l_4. Therefore, the distance-based influence of POIs also contain multiple different aspects, it is inappropriate to only represent the distance-based influence of POIs with spatial intervals or holistic representations.

To address the limitations mentioned above, we focus on augmenting the sequence modeling scheme with POI representations that process strong expressiveness for latent aspects and corresponding influence. To this end, we propose a novel Disentangled Representation-enhanced Attention Network (DRAN) for next POI recommendation. Instead of directly modeling the entangled influence within the holistic POI embeddings, we first propose to explicitly disentangle the representation of each POI into multiple independent components, each of which is utilized to represent an individual aspect and corresponding influence. Specifically, we propose a novel Disentangled Graph Convolution Network (DGCN) to learn two types of disentangled representations, i.e., transition-based and distance-based disentangled representations, to address the first and second aforementioned limitations. Moreover, DGCN makes full use of the graph structure of POIs by refining global POI relation graphs, which avoids only considering spatial contexts within check-in sequences. Subsequently, we extend the self-attention mechanism for modeling user dynamic preferences with the consideration of personalized spatio-temporal information, while maintaining each component of the disentangled representation independent. Empirical results show that our method outperforms state-of-the-art baselines on real-world LBSN datasets, and our strategy of learning disentangled representations is demonstrated to be effective.

The main contributions of our work are summarized as follows:

- We propose to explicitly disentangle the multiple latent aspects of POIs for capturing corresponding influence. A novel DGCN is proposed to achieve this goal. To the best of our knowledge, this is the first attempt in next POI recommendation to handle the complex correlation between POIs with disentangled representation learning.
- We propose DRAN, which focuses on fully utilizing the global-level influence of POIs and personal-level dynamic preferences for next POI recommendation tasks.
- Extensive experiments on two real-world LBSN datasets are conducted to evaluate the proposed model. Experimental results demonstrate that our model not only consistently outperforms state-of-the-art baselines, but also has the effective capability for representation disentanglement.

2 RELATED WORK

In this section, we briefly review the recent literature along the following lines of fields: (1) next POI recommendation, (2) graph convolution network-based recommendation, and (3) disentangled representation learning.

2.1 Next POI Recommendation

Next POI recommendation aims to recommend for users the future visit according to check-in data. Early attempts apply Markov Chain-based methods [8, 26] to estimate the transition probability between POIs. Due to the high sparsity of check-in data, Matrix Factorization [25] is adopted to learn dense embeddings of both users and POIs. Further extensions explicitly introduce auxiliary factors, such as regional restriction [3] and temporal information [19], to endow embeddings with better expressiveness. However, these methods fail to model high-order sequential patterns, since they mainly learn transition regularity between successive visits.

Another research line achieves massive improvement by employing deep learning-based methods. Due to the strong capability of learning informative representations of POIs from sequential data, RNN-based methods become dominant in the field of next POI recommendation. STRNN [21] extends RNN with time-specific and distance-specific matrices to capture temporal and spatial influence. HST-LSTM [13] directly utilizes spatio-temporal intervals between successive visits in the LSTM with a hierarchical structure. Time-LSTM [41] equips the LSTM unit with additional time gates for capturing user short-term preferences. STGN further exploits the gate mechanism of LSTM by adding spatio-temporal gates. LSTPM [27] jointly models user long- and short-term preferences via LSTM, and designs a geo-dilated RNN to capture geographical relations for short-term preference. Moreover, the more recent works explore new methods of leveraging the self-attention mechanism. SGRec [18] proposes an attention-based model to aggregate POI representations learned by sequence-to-graph augmentation. STAN [23] enhances the attention method by explicitly utilizing spatial and temporal matrices.

Overall, all existing methods represent each POI as a holistic representation, thus suffering from the disadvantage of neglecting the complex factors behind POIs. In contrast, our work explicitly disentangles POI representations into several independent components, which enables the proposed DRAN to model different aspects of POIs individually.

2.2 Graph Convolution Network-based Recommendation

GCN [12] has been attracting considerable attention, thanks to its high efficiency in handling graph-structured data. To extract useful information from correlated items across users, user-item bipartite graphs are constructed to depict the high-order connectivity relationship among items [9, 32]. Besides, GCN is also applied to other types of graphs in recommendation tasks. For instance, user social graphs [34] and knowledge graphs [31] are utilized to enrich the expressiveness of learned representations.

Generally, POI data are naturally suitable to be depicted by a graph structure, since each POI denotes a real place with a geographical location. Early studies [30, 35] mainly apply graph embedding methods that neglect the high-order connectivity among POIs. Recent works [18, 20] attempt to use node-wise graph attention networks for POI representation learning, which may overlook the geographical influence involved in the graph structure of POIs. More importantly, all of the graph-based methods in this field neglect the diverse latent aspects of POIs and their corresponding influence on user behaviors.

2.3 Disentangled Representation Learning

Disentangled representation learning aims to explicitly identify and separate explanatory factors of variations behind data [2]. Such representations with multiple independent components not only prove to own better expressiveness and robustness, but also provide additional interpretability for semantic data used in various fields [1, 10]. With regard to recommendation tasks, Macrid-VAE [24] proposes to decompose user interactions from both macro and micro levels via variational auto-encoders. DGCF [33] devises intent-aware graphs to discover different user intents over items. CIGCN [40] extends GCN to maintain dimensions independent. However, all of the existing works focus on the user-item relationship. These works ignore the underlying item-item relationship and are insufficient to explore different user intentions within sequential data. Furthermore, works like CIGCN and Macrid-VAE may struggle with excessive trivial intentions since they distinguish and keep each dimension of representation irrelevant. In this paper, we first construct POI relation graphs to capture the rich influence of POIs, and then leverage the novel DGCN to disentangle different components of representations appropriately.

3 PRELIMINARIES

In this section, we first formally formulate the next POI recommendation task, and then briefly introduce the graph convolution operation and disentangled representations of POIs.

3.1 Problem Formulation

We consider a typical next POI recommendation scenario with a POI set $\mathcal{L} = \{l_1, l_2, ..., l_N\}$ ($|\mathcal{L}| = N$) and a user set $\mathcal{U} = \{u_1, u_2, ..., u_M\}$ ($|\mathcal{U}| = M$). Each POI $l \in \mathcal{L}$ is a spatial site, which is associated with unique geographical coordinate ($longitude, latitude$) tuple, i.e., (lon, lat). The historical check-in sequence of user $u \in \mathcal{U}$ is denoted as $H(u) = \{(l_i^u, t_{l_i^u}) | i = 1, 2, \cdots, m\}$, where each tuple $(l_i^u, t_{l_i^u})$ indicates user u visited POI l_i^u at timestamp $t_{l_i^u}$. Given a target user u, the task of next POI recommendation aims to recommend a list of POIs that u may be interested in at the next timestamp, which is formally defined as:

Input: Users $\mathcal{U}$, POIs $\mathcal{L}$ and users' check-in sequences H.

Output: A ranked POI list that user u would be interested at the next timestamp.

3.2 Graph Convolution

Graph convolution operations could be viewed as a node representation learning method, which update node representations through information aggregation of neighbor nodes. It is initially defined in the Fourier domain. For an undirected graph $G = (V, E, \boldsymbol{A})$ where V is the set of nodes and $|V| = n$, E is the set of edges and $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is a weighted adjacency matrix, the widely used GCN [12] adopts renormalized adjacency matrix $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ and updates m-dimensional node representations $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ based on its propagation rule, which is parameterized as

$$X^{update} = (\boldsymbol{D} + \boldsymbol{I})^{-1/2} \hat{\boldsymbol{A}} (\boldsymbol{D} + \boldsymbol{I})^{-1/2} \boldsymbol{X} \Theta = \tilde{\boldsymbol{A}} \boldsymbol{X} \Theta, \tag{1}$$

where Θ is the trainable weight matrix, $\boldsymbol{D}$ is the degree matrix.

3.3 Disentangled Representation

The disentangled representation $\boldsymbol{x} \in \mathbb{R}^d$ of POI l is expected to be composed of several relatively independent components, which enable it to depict different aspects of POIs. Assuming that K aspects need to be disentangled, the representation $\boldsymbol{x}_i$ of POI l_i is defined as

$$\boldsymbol{x}_i = (\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \cdots, \boldsymbol{x}_{i_K}), \tag{2}$$

where $\boldsymbol{x}_{i_j} \in \mathbb{R}^{\frac{d}{K}}$ denotes the representation of the j-th aspect of POI l_i. We refer to $\boldsymbol{x}_{i_j}$ as a component of the disentangled representation x_i in our paper. It is worthwhile to mention that $\boldsymbol{x}_{i_j}$ is supposed to be independent of $\boldsymbol{x}_{i_k}$ if $j \neq k$ for representing distinct aspects and avoiding redundancy.

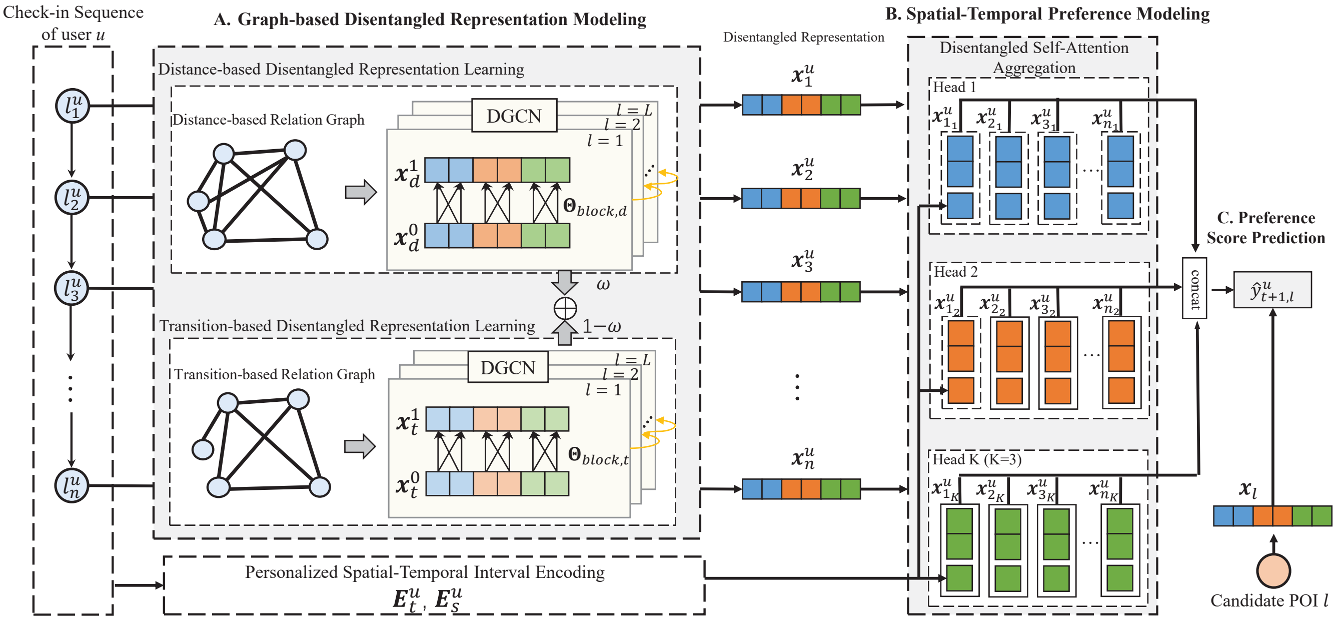

Check-in Sequence of user u
l_1^u
l_2^u
l_3^u
l_n^u
A. Graph-based Disentangled Representation Modeling
Distance-based Disentangled Representation Learning
Distance-based Relation Graph
DGCN
$l = L$
$l = 2$
$l = 1$
x_d^1
x_d^0
$\Theta_{block,d}$
ω
$1-\omega$
Transition-based Disentangled Representation Learning
Transition-based Relation Graph
DGCN
$l = L$
$l = 2$
$l = 1$
x_t^1
x_t^0
$\Theta_{block,t}$
Personalized Spatial-Temporal Interval Encoding
E_t^u, E_s^u
Disentangled Representation
x_1^u
x_2^u
x_3^u
x_n^u
B. Spatial-Temporal Preference Modeling
Disentangled Self-Attention Aggregation
Head 1
$x_{1_1}^u$ $x_{2_1}^u$ $x_{3_1}^u$ $x_{n_1}^u$
Head 2
$x_{1_2}^u$ $x_{2_2}^u$ $x_{3_2}^u$ $x_{n_2}^u$
Head K (K=3)
$x_{1_K}^u$ $x_{2_K}^u$ $x_{3_K}^u$ $x_{n_K}^u$
concat
C. Preference Score Prediction
$\hat{y}_{t+1,l}^u$
x_l
Candidate POI l

Figure 2: The overall architecture of DRAN. Here, we assume that there are three latent aspects for POIs. DRAN utilizes DGCN to learn the graph-based disentangled representation for each POI, and then aggregates multiple useful information based on check-in sequences for user preference estimation.

4 PROPOSED METHOD

In this section, we present the details of our proposed DRAN model. The overall architecture of DRAN is illustrated in Figure 2. It consists of three key components: (1) graph-based disentangled representation modeling module (part **A** in Figure 2), which learns disentangled representations of POIs utilizing the novel proposed DGCN layer; (2) user spatial-temporal preference modeling module (part **B** in Figure 2), which integrates relevant spatial and temporal information to model user historical check-in sequence; (3) prediction and optimization module (part **C** in Figure 2), which estimates user preferences and optimizes all trainable parameters.

4.1 Graph-based Disentangled Representation Modeling

4.1.1 POI Relation Graph Construction. We argue that the representation learning strategies utilized in prior work are insufficient to depict the complex influence of POIs, since they ignore intrinsic characteristics of POIs. Hence, we adopt two global POI relation graphs to guide the representation learning. Generally, There are two types of relations among POIs, i.e., distance and transition relations. We construct corresponding graphs respectively.

Distance-based relation graph $G_d = (V_d, E_d, \boldsymbol{A}_d)$ is an undirected graph, where V represents the set of POI nodes and $|V_d| = N$, E_d represents the set of edges in G_d. $\boldsymbol{A}_d \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix. $\boldsymbol{A}_d(i,j) = 1$ if the physical distance between POI l_i and l_j is less than threshold $\Delta d = 2km$ and $\boldsymbol{A}_d(i,j) = 0$ otherwise.

Transition-based relation graph $G_t = (V_t, E_t, \boldsymbol{A}_t)$ is a weighted graph, where V_t and E_t are the sets of nodes and edges respectively. Each entry $\boldsymbol{A}_t(i,j) = freq(l_i, l_j)$ of $\boldsymbol{A}_t$ describes the direct transition relation between l_i and l_j, where $freq(l_i, l_j)$ represents the transition frequency between l_i and l_j among all users.

4.1.2 Disentangled Representation Propagation. We aim to leverage the graph convolution operation to distill useful information from POI relation graphs for learning disentangled representations, while keeping the different components of representations independent. Considering the propagation rule of GCN in Equation (1), the n-th dimension $\boldsymbol{X}^{update}(m,n)$ of $\boldsymbol{x}_m^{update}$ after the convolution operation is calculated as

$$\boldsymbol{X}^{update}(m,n) = \sum_{i=1}^{N} \boldsymbol{A}(m,i)(\sum_{j=1}^{N} \boldsymbol{X}(i,j)\boldsymbol{\Theta}(j,n)), \quad (3)$$

this equation demonstrates that each dimension of $\boldsymbol{x}_m$ is strongly entangled with all other dimensions, which limits the expressivity of representations for representing different aspects of POIs. To tackle this problem, we propose to only interrelate different dimensions of the same component. Specifically, each dimension in k-th component $\boldsymbol{x}_{m_k}$ is calculated as

$$\boldsymbol{X}^{update}(m,n) = \sum_{i=1}^{N} \boldsymbol{A}(m,i)(\sum_{j=index(\boldsymbol{x}_{m_k}(1))}^{index(\boldsymbol{x}_{m_k}(fin))} \boldsymbol{X}(i,j)\boldsymbol{\Theta}(j,n)), \quad (4)$$

where $index(\boldsymbol{x}_{m_k}(1)), index(\boldsymbol{x}_{m_k}(fin))$ represent the index of the first and final dimension of component $\boldsymbol{x}_{m_k}$ respectively. To realize Equation (4), we generalize the holistic weight matrix $\boldsymbol{\Theta}$ in Equation (1) to a block diagonal weight matrix $\boldsymbol{\Theta}_{block}$. Therefore, the propagation rule of our proposed DGCN is

$$X^{update} = \tilde{A}X\Theta_{block}, \tag{5}$$

with

$$\Theta_{block} = diag(\Theta_1, \Theta_2, \cdots, \Theta_K), \Theta_i \in \mathbb{R}^{\frac{d}{K} \times \frac{d}{K}}, \tag{6}$$

which ensures that the updated component $\boldsymbol{x}_{i_j}^{update}$ is only calculated by the j-th component of $\boldsymbol{x}_i$ and its neighbors during the propagation process. It is worthwhile to emphasize that DGCN adopts a multi-dimensional vector rather than a single dimensional feature to represent the aspect of nodes, which improves the expressivity of vectors and avoids depicting excessive trivial aspects.

We employ our novel proposed DGCN to acquire two types of disentangled representations, i.e., the distance-based disentangled representation $\boldsymbol{x}_d \in \boldsymbol{X}_d$ and transition-based disentangled representation $\boldsymbol{x}_t \in \boldsymbol{X}_t$, based on POI relation graphs. We also ensure that both $\boldsymbol{x}_d$ and $\boldsymbol{x}_t$ contain K components and explain the reasons in Section 4.1.3. Moreover, we stack multiple DGCN layers and introduce activation functions to obtain contributions of different order neighbors as

$$\begin{cases} \boldsymbol{X}_d^{(l+1)} = tanh(\tilde{A}\boldsymbol{X}_d^l \Theta_{block,d}^l) , \\ \boldsymbol{X}_t^{(l+1)} = tanh(\tilde{A}\boldsymbol{X}_t^l \Theta_{block,t}^l) , \end{cases} \tag{7}$$

where $\boldsymbol{X}_d^l$ and $\boldsymbol{X}_t^l$ denote the sets of distance-based and transition-based representations on the l-th layer respectively, $\boldsymbol{X}_d^0$ and $\boldsymbol{X}_t^0$ are the trainable POI embeddings. $\Theta_{block,d}^l$ and $\Theta_{block,t}^l$ represent layer-specific parameters with $\boldsymbol{\Theta_{block}}$ form.

4.1.3 Representation Aggregation. Having applied L DGCN layers on POI relation graphs, we leverage the layer aggregation strategy to combine representations from different layers. Specifically, sum-pooling is adopted as our aggregation function, formulated as

$$\begin{cases} \boldsymbol{X}_d = Sum(\boldsymbol{X}_d^0, \boldsymbol{X}_d^1, \cdots, \boldsymbol{X}_d^L), \\ \boldsymbol{X}_t = Sum(\boldsymbol{X}_t^0, \boldsymbol{X}_t^1, \cdots, \boldsymbol{X}_t^L), \end{cases} \tag{8}$$

where Sum denotes the sum-pooling function. In this way, we not only maintain the influential signals from different order neighbors, but also alleviate the over-smoothing problem [16] caused by the increase of convolution layers. To be mentioned, since the sum operation is element-wise, the independence of different components is maintained after aggregation.

When properly integrating these two types of relations for modeling K different components of disentangled representations, a preliminary idea is to generate I transition-based components and $K-I$ distance-based components, then concatenates them together as the final representation. However, such an idea may be unsuitable because of two reasons. First, the unnecessary hyperparameter I needs to be introduced to control the ratio of these two types of components, which may also render us unable to retain all influence. Second, in the real world, each aspect of POIs originally contains different types of influence, only corresponding an aspect to one special type of influence (i.e., the transition-based or distance-based influence) may limit the expressivity of representations. Hence, we generate K components for $\boldsymbol{x}_t$ and $\boldsymbol{x}_d$ respectively, and get the final disentangled representation $\boldsymbol{x}_i \in \boldsymbol{X}_{fin}$ of POI l_i with the weighted sum of $\boldsymbol{x}_{d,i} \in \boldsymbol{X}_d$ and $\boldsymbol{x}_{t,i} \in \boldsymbol{X}_t$ as

$$\begin{cases} \boldsymbol{x}_i = (\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \cdots, \boldsymbol{x}_{i_K}), \\ \boldsymbol{x}_{i_j} = \omega_j \boldsymbol{x}_{d,i_j} + (1 - \omega_j)\boldsymbol{x}_{t,i_j}, \end{cases} \tag{9}$$

where $\omega_j \in \mathbb{R}$ is a trainable parameter to control the contribution of different parts for the final representation, $\boldsymbol{x}_{d,i_j}, \boldsymbol{x}_{t,i_j}$ is the j-th components of $\boldsymbol{x}_{d,i}$ and $\boldsymbol{x}_{t,i}$. In this way, The corresponding influence of $\boldsymbol{x}_i$ are automatically determined by the model, i.e., $\boldsymbol{x}_i$ mainly reflect the distance-based influence if ω_j approaches 1 and 0 otherwise.

4.2 User Spatial-Temporal Preference Modeling

With the disentangled representations $\boldsymbol{X}_{fin}$, we encode the check-in sequence $H(u) = \{(l_i^u, t_{l_i^u}) | i = 1, 2, \cdots, m\}$ of user u into a fix-length sequence $S(u) = \{\boldsymbol{x}_1^u, \boldsymbol{x}_2^u, \cdots, \boldsymbol{x}_n^u\}$ for modeling user dynamic preferences, where $\boldsymbol{x}_i^u \in \boldsymbol{X}_{fin}$. If $n > m$, we add padding to the left until the length is n. If $n < m$, we truncate the sequence and only consider the recent n records.

4.2.1 Personalized Spatial-Temporal Interval Encoding. Empirically, user preferences for individual POIs are heavily limited by spatial constraints, i.e., users prefer to visit a nearby POI rather than a distant POI. In addition, historical visited POIs are always associated with user current preferences. Therefore, we explicitly model spatial and temporal relations between POIs to enhance the effectiveness of our model. Considering that each user has his/her own check-in preference, we adopt relative length of the interval to model spatial and temporal relations within $S(u)$. More formally, the temporal interval matrix $\boldsymbol{M}_t^u \in \mathbb{N}^{n \times n}$ and spatial interval matrix $\boldsymbol{M}_s^u \in \mathbb{N}^{n \times n}$ of u is given as

$$\boldsymbol{M}_t^u = \begin{bmatrix} t_{1,1}^u & t_{1,2}^u & \cdots & t_{1,n}^u \\ t_{2,1}^u & t_{2,2}^u & \cdots & t_{2,n}^u \\ \cdots & \cdots & \cdots & \cdots \\ t_{n,1}^u & t_{n,2}^u & \cdots & t_{n,n}^u \end{bmatrix} \quad \boldsymbol{M}_s^u = \begin{bmatrix} s_{1,1}^u & s_{1,2}^u & \cdots & s_{1,n}^u \\ s_{2,1}^u & s_{2,2}^u & \cdots & s_{2,n}^u \\ \cdots & \cdots & \cdots & \cdots \\ s_{n,1}^u & s_{n,2}^u & \cdots & s_{n,n}^u \end{bmatrix}, \tag{10}$$

where $t_{i,j}^u = \left\lfloor \frac{|t_i - t_j|}{t_{min}^u} \right\rfloor$ represents the relative length of the temporal interval between i-th and j-th visited POIs, t_{min}^u denotes the minimum temporal interval (except for 0) of user u. Similarly to $t_{i,j}^u$, $s_{i,j}^u = \left\lfloor \frac{haversine(l_i^u, l_j^u)}{s_{min}^u} \right\rfloor$ denotes the relative length of the spatial interval between l_i^u and l_j^u. Assuming that over large intervals may be redundant for the preference modeling, we clip the maximum interval of $\boldsymbol{M}_t^u$ and $\boldsymbol{M}_s^u$ to thresholds Δt and Δs respectively, i.e., $t_{i,j}^u = min(t_{i,j}^u, \Delta t)$ and $s_{i,j}^u = min(s_{i,j}^u, \Delta s)$. Besides, we pad $t_{i,j}^u$ and $s_{i,j}^u$ to zero if $\boldsymbol{x}_i^u$ or $\boldsymbol{x}_j^u$ is the padding.

To generate the dense representations of spatial and temporal relations within sequences, embedding operations are used to project the scaled intervals to d-dimensional vectors. After retrieval, interval matrices $\boldsymbol{M}_t^u$ and $\boldsymbol{M}_s^u$ are transformed to embedding matrices

$E^u_t \in \mathbb{R}^{n\times n\times d}$ and $E^u_s \in \mathbb{R}^{n\times n\times d}$ as

$$E^u_t = \begin{bmatrix} e^{u,t}_{1,1} & e^{u,t}_{1,2} & \cdots & e^{u,t}_{1,n} \\ e^{u,t}_{2,1} & e^{u,t}_{2,2} & \cdots & e^{u,t}_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ e^{u,t}_{n,1} & e^{u,t}_{n,2} & \cdots & e^{u,t}_{n,n} \end{bmatrix} \quad E^u_s = \begin{bmatrix} e^{u,s}_{1,1} & e^{u,s}_{1,2} & \cdots & e^{u,s}_{1,n} \\ e^{u,s}_{2,1} & e^{u,s}_{2,2} & \cdots & e^{u,s}_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ e^{u,s}_{n,1} & e^{u,s}_{n,2} & \cdots & e^{u,s}_{n,n} \end{bmatrix}, \tag{11}$$

where $e^u_{i,j} \in \mathbb{R}^d$ represents the corresponding vector of $t^u_{i,j}$ or $s^u_{i,j}$. We also apply a constant zero vector as embedding for zero value.

4.2.2 Disentangled Self-Attention Aggregation. To capture the multi-level regularity of check-in sequences, we propose an extension to relative position self-attention to incorporate different relations among POIs in a sequence. However, the disentangled representation $x \in X_{fin}$ is composed of several independent components, directly applying self-attention aggregation would undermine their independence. To tackle this problem, we separate each component into an individual attention head during the aggregation process, and then gather them up. More formally, given the input sequence $S(u)$, we firstly generate K sequences which only contain an individual component of x^u_i as

$$S(u)_k = \{x^u_{1_k}, x^u_{2_k}, \cdots, x^u_{n_k}\}, \tag{12}$$

where $1 \leq k \leq K$, $x^u_{i_k} \in \mathbb{R}^{\frac{d}{K}}$ denotes the k-th component of x^u_i. We also split the entries of E^u_t and E^u_t into K chunks, i.e., $e^u_{i,j} = (e^u_{i,j,1}, e^u_{i,j,2}, \cdots, e^u_{i,j,K})$, to ensure $e^u_{i,j,k}$ and $x^u_{i_k}$ have the same dimension. Then, a new sequence $R(u)_k = \{r^u_{1_k}, r^u_{2_k}, \cdots, r^u_{n_k}\}$ is generated within the k-th attention head and the element $r^u_{i_k}$ is calculated by a weighted sum of relevant visited POIs as

$$r^u_{i_k} = \sum_{j=1}^{i} \alpha_{ij}(x^u_{j_k}W^V + e^{u,t}_{i,j,k} + e^{u,s}_{i,j,k} + p_j) \tag{13}$$

where $W^V \in \mathbb{R}^{\frac{d}{K}\times\frac{d}{K}}$ is a projection matrix, $p_j \in \mathbb{R}^{\frac{d}{K}}$ is a position embedding for the j-th offset position. Future POIs are masked due to the causality. α_{ij} represents the weight coefficient that is computed via the softmax function as

$$\begin{cases} \alpha_{ij} = \dfrac{\exp a_{ij}}{\sum_{m=1}^{i} \exp a_{im}}, \\ a_{ij} = \dfrac{x^u_{i_k}W^Q(x^u_{j_k}W^K + e^{u,t}_{i,j,k} + e^{u,s}_{i,j,k} + p_j)^T}{\sqrt{\frac{d}{K}}}, \end{cases} \tag{14}$$

where $W^Q \in \mathbb{R}^{\frac{d}{K}\times\frac{d}{K}}, W^K \in \mathbb{R}^{\frac{d}{K}\times\frac{d}{K}}$ are projection matrices. It should be mentioned that the main purpose of splitting $e^u_{i,j}$ is dimension reduction, an alternative method is to project $e^u_{i,j}$ to a $\mathbb{R}^{\frac{d}{K}}$ dimensional vector. However, we found out the two methods have similar performances, and thus we do not utilize the second method to avoid excessive parameters.

As suggested in [11, 15], feed-forward networks are applied after each self-attention layer, which endows the model with nonlinearity and encodes the interactions between dimensions as

$$z^u_{i_k} = FFN(r^u_{i_k}) = ReLU(r^u_{i_k}W_{1_k} + b_{1_k})W_{2_k} + b_{2_k}, \tag{15}$$

where $W_{1_k}, W_{2_k} \in \mathbb{R}^{\frac{d}{K}\times\frac{d}{K}}$ and $b_{1_k}, b_{2_k} \in \mathbb{R}^{\frac{d}{K}}$. Moreover, dropout regularization, layer normalization and residual connection are leveraged to alleviate the overfitting problem and speed up the training process. It should be noted that different parameter matrices and biases (i.e., W and b in Equation (15)) are used in different attention heads, which prevents the potential information entanglement among different components.

After stacking several attention blocks, we get the output sequence $Z(u)_k = \{z^u_{1_k}, z^u_{2_k}, \cdots, z^u_{n_k}\}$. Since each element $z^u_{i_k}$ of $Z(u)_k$ is a combined representation of the k-th aspect of POIs, positions, spatial and temporal relations, it could be viewed as the representation of user preference on the k-th aspect of POIs at step i. To completely consider the influence of each component, We integrate all the K output sequences into a holistic sequence $Z(u)$ to represent the whole check-in behaviors of user u as

$$Z(u) = Concat(Z(u)_1, Z(u)_2, \cdots, Z(u)_K) = (z^u_1, z^u_2, \cdots, z^u_n), \tag{16}$$

where $Concat(\cdot)$ denotes the concatenation operation and $z^u_i = [z^u_{i_1}, z^u_{i_2}, \cdots, z^u_{n_K}]$. For each element $z^u_i \in \mathbb{R}^d$, it not only denotes the holistic representation of the user preference at step i, but also includes several independent components to depict user interests in different aspects of POIs.

4.2.3 User Preference Estimation. So far, we generate the preference sequence $Z(u)$ of user u. To estimate the user preference on the next visited POI, we calculate the preference score for each candidate with the softmax function. Specifically, at timestamp $t+1$, the preference score $\hat{y}^u_{t+1,l}$ of POI l is computed as

$$\hat{y}^u_{t+1,l} = \frac{exp(z^u_t x^T_l)}{\sum_{i=1}^{\mathbb{L}} exp(z^u_t x^T_i)} \tag{17}$$

where $z^u_t \in Z(u)$ denotes the user preference at timestamp t, $x \in X_{fin}$ denotes the disentangled representation of l. Note that the numerator of Equation (17) can be rewritten as

$$exp(z^u_t x^T_l) = exp(\sum_{k=1}^{K}\sum_{i=1}^{\frac{d}{K}} z^u_{t_k}(i) \cdot x_k(i))), \tag{18}$$

where $z^u_{t_k}(i), x_k(i)$ represent the i-th dimension of $z^u_{t_k}$ and x_k respectively. Hence, the preference score could also be viewed as the integrated preference of different components via sum operation.

4.3 Model Optimization

Given the training samples, we apply the cross-entropy loss function to optimize all parameters as

$$\mathbb{J} = -\sum_{u\in\mathcal{U}}\sum_{i\in H(u)}\sum_{j=1}^{N} y^u_{i,j}\log(\hat{y}^u_{i,j}) + \lambda||\Theta||_2 \tag{19}$$

where $y^u_{i,j}$ is an indicator that is equal to 1 if l_j is the ground truth and 0 otherwise, $||\Theta||_2$ represents the $L2$ regularization of all parameters for preventing overfitting. Although additional regularization like cosine distance [28] proves to be useful in keeping different components independent [33]. Since it is not a design for user preference estimation, we do not adopt it to avoid the potential degradation of model performance. Moreover, our empirical

Table 1: Statistic of Datasets.

Dataset	#User	#POI	Avg.# visit per POI	#Density
Foursquare	2,291	3,777	47.71	0.011
Gowalla	10,136	23,797	18.91	0.0013

results also demonstrate that our model is sufficient to effectively disentangle different aspects of POIs.

5 EXPERIMENT

In this section, we present our empirical results to evaluate the effectiveness of DRAN.

5.1 Experimental Setup

5.1.1 Datasets. We evaluate our proposed model on two real-world LSBN datasets, which are collected from Foursquare[1] and Gowalla[2] and have been widely used in prior works. Foursquare dataset includes check-in data from August 2010 to July 2011 within Singapore. Gowalla dataset includes check-in data from February 2009 to October 2010 within California and Nevada. Following previous works, we filter out inactive users with fewer than 5 records and POIs with fewer than 5 visitors. The statistics of datasets are summarized in Table 1. For each dataset, we sort the check-in sequence of each user in chronological order. The first 80% check-ins of each user are taken as the training set and the last 10% are taken as the test set, the remaining 10% are taken as the validation set.

5.1.2 Baselines. We compare DRAN with following representative methods for next POI recommendation, covering classic methods (MF and FPMC), RNN-based methods (TMCA, STGN and LSTPM), attention-based method (STAN) and GNN-based method (SGRec).

- **MF**[25]: A classical method based on Matrix Factorization over the user-POI matrix.
- **FPMC**[26]: Such a method combines Markov Chains and Matrix Factorization to model location transitions.
- **TMCA**[17]: This is an LSTM-based method that incorporates multiple kinds of contexts for next POI recommendation. For fairness of comparison, we remove the POI categorical context since no other methods use it.
- **STGN**[39]: It is a state-of-the-art method that integrates both spatial and temporal intervals between successive check-ins by extending LSTM with time and distance gates.
- **LSTPM**[27]: This is a state-of-the-art LSTM-based method, which captures long-term and short-term preferences with a nonlocal network and a geo-dilated RNN respectively.
- **STAN**[23]: A state-of-the-art method based on the attention mechanism. This method uses spatio-temporal effects for aggregating locations, but it only considers spatio-temporal correlation within check-in sequences.
- **SGRec**[18]: It is a state-of-the-art GNN-based method that uses Seq2Graph augmentation for capturing collaborative signals from POIs. We also remove the POI categorical context for fairness comparison.

5.1.3 Evaluation Metrics. Following the previous works, we employ two widely used evaluation metrics of ranking evaluation, including Recall at a cutoff top $\mathbb{k}$ (Recall@$\mathbb{k}$) and Normalized Discounted Cumulative Gain at a cutoff top $\mathbb{k}$ (NDCG@$\mathbb{k}$). For the sake of the comprehensive evaluation, we retrieve all the unvisited POIs of each user to the target POI as negative candidates. We report Recall and NDCG with the popular $\mathbb{k} \in \{2, 5, 10\}$ in this paper. Each metric is calculated 10 times and averaged.

5.1.4 Parameter Settings. For a fair comparison, we set the dimension of POI embedding to 64 for all methods. A grid search is performed to confirm the optimal settings of other common parameters, including the learning rate in $\{1e^{-1}, 1e^{-2}, 1e^{-3}, 1e^{-4}\}$, the dropout rate in $\{0, 0.1, 0.2, 0.3, 0.4\}$, the coefficient λ of $L2$ regularization in $\{0, 1e^{-1}, 1e^{-2}, 1e^{-3}, 1e^{-4}\}$. In our model, We use the Adam Optimizer with default betas, the learning rate of $1e^{-3}$, the dropout rate of 0.4 for Foursquare and 0.2 for Gowalla, the coefficient λ of $1e^{-3}$ for Foursquare and 0 for Gowalla, the number of components of 4, the maximum length of sequences of 100, the distance and time thresholds of 256, the number of DGCN layers of 3. We analyze the influence of key parameters in Section 5.4.

5.2 Performance Comparison

Table 2 presents the empirical results of all methods on the two datasets. We conduct a T-test with the p-value of 0.01 to ensure that the improvement of DRAN is statistically significant. From the results, we have following observations:

- Our proposed DRAN consistently achieves the best performance in terms of every metric on both Foursquare and Gowalla datasets. For example, the performance gains provided by DRAN over the best baselines are 7.32% and 5.24% in terms of Recall@10 on the two datasets respectively. This validates the effectiveness of DRAN. We credit the improvement to following reasons. (1) DRAN explicitly disentangle the complex influence of POIs on users by independently modeling multiple latent aspects of POIs, which improves the expressivity of POI representations. (2) On the basis of global POI relation graphs, DGCN captures two types of disentangled representations, endowing it exploits the distance-based and transition-based influence among all POIs. (3) By designing the appropriate self-attention module (i.e., disentangled self-attention), avoiding the component entanglement in the sequence modeling part.
- Deep learning-based methods perform better than conventional methods, due to their demonstrated capability of learning POI representations from check-in sequences. Among deep learning-based methods, RNN-based methods (TMCA, STGN and LSTPM) are less competitive than STAN. It is reasonable since these RNN-based methods only integrate spatio-temporal information between successive visits into the RNN framework, while STAN leverages the self-attention mechanism to capture spatio-temporal correlation among non-successive visits. However, the learning process of these methods is constrained on the POIs within the check-in sequences, which easily leads to insufficient modeling of POI representations.

[1] https://sites.google.com/site/yangdingqi/home
[2] http://snap.stanford.edu/data/loc-gowalla.html

Table 2: Performance comparison with baselines. Bold scores are the best results for each metric, while the second best scores are underlined. $*$ represents significance level p-value< 0.01 of comparing DRAN with the best baseline.

	Foursquare						Gowalla					
	Recall@2	Recall@5	Recall@10	NDCG@2	NDCG@5	NDCG@10	Recall@2	Recall@5	Recall@10	NDCG@2	NDCG@5	NDCG@10
MF	0.1457	0.1909	0.2356	0.1223	0.1484	0.1632	0.0975	0.1404	0.1778	0.0821	0.1013	0.1170
FPMC	0.1789	0.2492	0.2966	0.1553	0.1865	0.2174	0.1021	0.1489	0.1863	0.0904	0.1072	0.1216
TMCA	0.2042	0.2761	0.3325	0.1857	0.2192	0.2453	0.1284	0.1893	0.2401	0.1074	0.1332	0.1563
STGN	0.2002	0.2684	0.3261	0.1865	0.2132	0.2362	0.1178	0.1814	0.2365	0.1057	0.1378	0.1529
LSTPM	0.2364	0.2966	0.3609	0.2289	0.2404	0.2611	0.1452	0.2043	0.2576	0.1255	0.1532	0.1811
SGRec	0.2964	0.3511	0.3975	0.2809	0.3053	0.3205	0.2122	0.2666	0.3127	0.198	0.2224	0.2375
STAN	0.3257	0.3730	0.4204	0.3121	0.3367	0.3526	0.1873	0.2461	0.2973	0.1711	0.1977	0.2154
DRAN	**0.3551***	**0.4092***	**0.4512***	**0.3389***	**0.3631***	**0.3775***	**0.2288***	**0.2832***	**0.3291***	**0.2145***	**0.2384***	**0.2535***
Improvement	9.02%	9.70%	7.32%	8.58%	7.84%	7.06%	7.82%	6.22%	5.24%	8.33%	7.19%	6.73%

- STAN has better performance on Foursquare while it is worse than SGRec on Gowalla. It is mainly caused by the difficulty of learning informative POI representations only from sequences in datasets with low data density (0.0013 of Gowalla). In contrast, SGRec learns POI representations by aggregating useful information from neighbors, which alleviates the sparsity problem. Such observation is consistent with SGRec [18]. This further demonstrates the effectiveness of considering POI correlations across check-in sequences. However, both of the two state-of-the-art methods neglect the complex influence and the multiple latent aspects of POIs, leading to suboptimal representations.
- We also observe that the overall improvement of DRAN on Foursquare is more than that on Gowalla. It is not a surprise since the average visit per POI of Foursquare is much higher than that of Gowalla. In other words, different visits of the same POI in Foursquare are more likely to be inspired by dissimilar aspects. This further verifies the significance of learning disentangled POI representations for next POI recommendation.

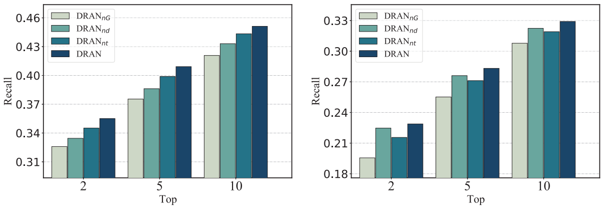
5.3 Ablation Study

To verify the effectiveness of the main designs in our method, we conduct an ablation study. We denote DRAN as the base model and remove different components to obtain three variants as follows.

- DRAN$_{nG}$: This variant removes the whole DGCN layers and generates POI representations by random initialization.
- DRAN$_{nd}$: This variant removes the distance-based relation graph and only considers the transition-based influence.
- DRAN$_{nt}$: This variant removes the transition-based relation graph and only considers the distance-based influence.

Figure 3 shows the results of the ablation study. From the results, we have the following findings. First, adding graph-based disentangled POI representations dramatically boosts the model performance. After removing DGCN which are utilized to learn disentangled representations, DRAN$_{nG}$ suffers from a considerable decrease in performance compared with DRAN. This indicates that the latent aspects of POIs and corresponding influence are of vital use, and need to be sufficiently modeled. Second, both the distance-based influence and transition-based influence prove to be useful. By dropping POI relation graphs, DRAN$_{nd}$ and DRAN$_{nt}$ both obtain low performance on the two datasets. This implies that the intrinsic characteristics of POIs do contain multiple types of influence, and DRAN can explicitly capture these influence from POI relation graphs effectively. Third, the characteristics of different datasets are distinct. Jointly analyzing the results across datasets, we notice that DRAN$_{nd}$ outperforms DRAN$_{nt}$ on Gowalla but performs poorly on Foursquare. One possible reason is that the importance of influence is dissimilar in different datasets. Such observation verifies that it is not suitable to fix a certain number of components for a special type of influence, and reconfirms the necessity of our representation integration strategy (cf. Section 4.1.3).

Figure 3: Performance comparison of different variants

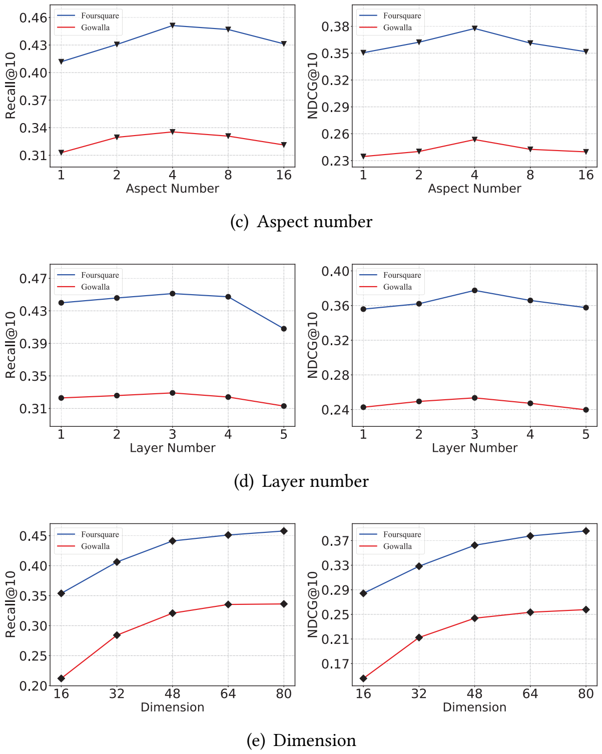


(a) Foursquare (b) Gowalla

5.4 Hyperparameter Study

We investigate the impact of three key hyperparameters on DRAN, including the number of components K, representation dimension d and the number of DGCN layers L. Figure 4 shows the results.

5.4.1 Effect of Component Number. To further verify whether explicitly modeling diverse latent aspects of POIs can lead to performance improvement, we search the component number K in the range of $\{1, 2, 4, 8, 16\}$. Specifically, DRAN gets the worst performance when $K = 1$, i.e., represents each POI in a uniform manner. By increasing K from 1 to 4, we can observe the substantial increase of performance on both datasets, which demonstrates the effectiveness of disentangling different aspects of POIs once again. we also notice the performance decreases when $K = 16$. This may



(c) Aspect number

(d) Layer number

(e) Dimension

Figure 4: Effect of several key hyperparameter

contribute to excessive trivial aspects and low expressiveness of components with low dimension size ($\frac{d}{K} = 4$ of each component).

5.4.2 Effect of Representation Dimension. We vary the number of dimension d from 16 to 80 with step 16. Generally, our model obtains better performance with a larger d. Enlarging d not only makes disentangled representations more informative, but also improves the expressivity of components. We set d as 64 in default since further increasing is costly but provides slight improvement.

5.4.3 Effect of DGCN Layer Number. To investigate the influence of DGCN depth, we vary the number of DGCN layer L in the range of $\{1, 2, 3, 4, 5\}$. We find that DRAN gets much better performance when L reaches 2 and 3. This is because stacking more DGCN layers is able to aggregate useful information from high-order neighbors of target nodes. In particular, Increasing the depth of DGCN is helpful for capturing the complex correlation between relevant POIs. However, The improvement provided by continuing stacking layers is not obvious, which suggests that redundant stacking may introduce noises to the learning process and cause the over-smoothing problem.

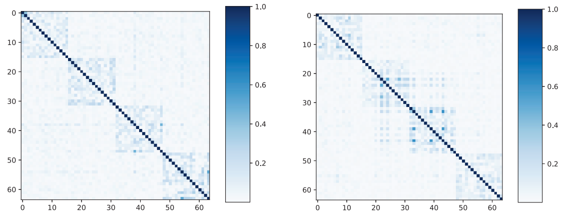
5.5 In-depth Study

As the graph-based disentangled representation is at the core of our model, we further conduct experiments to study the effect of our disentangled representation learning strategy. We first explore how the propagation rule of DGCN affects the model performance, and then investigate whether the diverse components of representation are successfully disentangled through visualization.

Table 3: Performance comparison of different GCN variants

	Foursquare		Gowalla	
	Recall@10	NDCG@10	Recall@10	NDCG@10
GCN+	0.4158	0.3525	0.3128	0.2346
LightGCN+	0.4248	0.3457	0.3177	0.2447
CIGCN+ (4 heads)	0.4518	0.3783	0.3077	0.2366
CIGCN+ (64 heads)	0.2886	0.2128	0.1531	0.1089
DRAN	0.4512	0.3775	0.3291	0.2535

Figure 5: Correlation values of representation dimensions



(a) Foursquare

(b) Gowalla

5.5.1 Effect of Propagation Rule. We maintain other parts of our model and replace DGCN with three graph convolution-based methods including GCN [12], LightGCN [9] and CIGCN [40] respectively. We report the performance of these variants (namely GCN+, LightGCN+ and CIGCN+ respectively) in Table 3. We can find that all of these variants receive varying degrees of performance degradation. GCN adopts a holistic weight matrix for representation propagation, leading to different dimensions strongly entangled and thus reducing the expressiveness of representations. LightGCN directly drops the weight matrix and updates representations by linear transformation, which overlooks the difference between layers. CIGCN proposes to learn disentangled representations by keeping each dimension independent and corresponding a single dimension with an aspect. It is not suitable for the attention mechanism since we need to separate each dimension into an individual attention head (i.e., CIGCN+ (64 heads) in table 3). Such a method suffers from low expressiveness of components since each component only has the dimension size of 1. We also evaluate the CIGCN+ with 4 attention heads and observe slight performance decrease. Our DGCN can be viewed as a generalization of these methods, which can effectively learn disentangled representations for enhancing performance and is more suitable for sequential recommendation.

5.5.2 Visualization. To exhibit that DRAN effectively learns disentangled representations X_{fin} of POIs with DGCN, we calculate the absolute correlation values of representation dimensions on both datasets, and report the result in Figure 5. It can be seen that the correlation result shows four obvious diagonal blocks, which demonstrated that our model has a strong ability in differentiating latent components of disentangled representations.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel Disentangled Representation-enhanced Attention Network (DRAN) for next POI recommendation. To improve the expressivity of representations, DRAN leverage the Disentangled Graph Convolution Network (DGCN) to explicitly models different aspects of POIs when learning POI representations. Extensive experiments on two real-world LBSN datasets demonstrate the superiority of DRAN to state-of-the-art methods and the effectiveness of our proposed disentangled learning strategy. An interesting direction in this field is the interpretability of the disentangled representation. In future work, we will make efforts to couple latent components with POI attributes by introducing POI side information.

ACKNOWLEDGEMENTS

This research is supported in part by the 2030 National Key AI Program of China 2018AAA0100503, National Science Foundation of China (No. 62072304, No. 61772341, No. 61832013), Shanghai Municipal Science and Technology Commission (No. 19510760500, No. 21511104700), the Oceanic Interdisciplinary Program of Shanghai Jiao Tong University (No. SL2020MS032), Scientific Research Fund of Second Institute of Oceanography, the open fund of State Key Laboratory of Satellite Ocean Environment Dynamics, Second Institute of Oceanography, MNR, and Zhejiang Aoxin Co. Ltd.

REFERENCES

[1] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. 2017. Deep Variational Information Bottleneck. In *ICLR*. 11396–11404.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* (2013), 1798–1828.

[3] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*. 2605–2611.

[4] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *WWW*. 1459–1468.

[5] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. 2017. Poi2vec: Geographical latent representation for predicting future visitors. In *AAAI*. 102–108.

[6] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI*. 2069–2075.

[7] Qing Guo, Zhu Sun, Jie Zhang, and Yin-Leng Theng. 2020. An attentional recurrent neural network for personalized next location recommendation. In *AAAI*. 83–90.

[8] Jing He, Xin Li, Lejian Liao, Dandan Song, and William Cheung. 2016. Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In *AAAI*. 137–143.

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.

[10] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2018. Disentangled representation learning for non-parallel text style transfer. *arXiv preprint arXiv:1808.04339* (2018).

[11] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. 197–206.

[12] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[13] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A Hierarchical Spatial-Temporal Long-Short Term Memory Network for Location Prediction.. In *IJCAI*. 2341–2347.

[14] Huayu Li, Yong Ge, Defu Lian, and Hao Liu. 2017. Learning User's Intrinsic and Extrinsic Interests for Point-of-Interest Recommendation: A Unified Approach.. In *IJCAI*. 2117–2123.

[15] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *WSDM*. 322–330.

[16] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*. 3538–3545.

[17] Ranzhen Li, Yanyan Shen, and Yanmin Zhu. 2018. Next point-of-interest recommendation with temporal and multi-level context attention. In *ICDM*. 1110–1115.

[18] Yang Li, Tong Chen, Hongzhi Yin, and Zi Huang. 2021. Discovering collaborative signals for next POI recommendation with iterative Seq2Graph augmentation. *arXiv preprint arXiv:2106.15814* (2021).

[19] Defu Lian, Vincent W Zheng, and Xing Xie. 2013. Collaborative filtering meets next check-in location prediction. In *WWW Companion Volume*. 231–232.

[20] Nicholas Lim, Bryan Hooi, See-Kiong Ng, Xueou Wang, Yong Liang Goh, Renrong Weng, and Jagannadan Varadarajan. 2020. STP-UDGAT: Spatial-Temporal-Preference User Dimensional Graph Attention Network for Next POI Recommendation. In *CIKM*. 845–854.

[21] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*. 194–200.

[22] Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, and Quan Yuan. 2017. An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proceedings of the VLDB Endowment* 10, 10 (2017), 1010–1021.

[23] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. STAN: Spatio-Temporal Attention Network for Next Location Recommendation. In *WWW*. 2177–2185.

[24] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. *arXiv preprint arXiv:1910.14238* (2019).

[25] Andriy Mnih and Russ R Salakhutdinov. 2008. Probabilistic matrix factorization. In *NeurIPS*. 1257–1264.

[26] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*. 811–820.

[27] Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *AAAI*. 214–221.

[28] Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. 2007. Measuring and testing dependence by correlation of distances. *The annals of statistics* 35, 6 (2007), 2769–2794.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*. 5998–6008.

[30] Hao Wang, Huawei Shen, Wentao Ouyang, and Xueqi Cheng. 2018. Exploiting POI-Specific Geographical Influence for Point-of-Interest Recommendation.. In *IJCAI*. 3877–3883.

[31] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge Graph Convolutional Networks for Recommender Systems. In *WWW*. 3307–3313.

[32] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*. 165–174.

[33] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *SIGIR*. 1001–1010.

[34] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *AAAI*. 346–353.

[35] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. 2016. Learning graph-based poi embedding for location-based recommendation. In *CIKM*. 15–24.

[36] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*. 325–334.

[37] Hongzhi Yin, Bin Cui, Xiaofang Zhou, Weiqing Wang, Zi Huang, and Shazia Sadiq. 2016. Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. *ACM Trans. Inf. Syst.* 35, 2 (2016), 1–44.

[38] Kangzhi Zhao, Yong Zhang, Hongzhi Yin, Jin Wang, Kai Zheng, Xiaofang Zhou, and Chunxiao Xing. 2020. Discovering Subsequence Patterns for Next POI Recommendation.. In *IJCAI*. 3216–3222.

[39] Pengpeng Zhao, Anjing Luo, Yanchi Liu, Fuzhen Zhuang, Jiajie Xu, Zhixu Li, Victor S Sheng, and Xiaofang Zhou. 2020. Where to go next: A spatio-temporal gated network for next poi recommendation. *TKDE* (2020).

[40] Tianyu Zhu, Leilei Sun, and Guoqing Chen. 2021. Embedding Disentanglement in Graph Convolutional Networks for Recommendation. *TKDE* (2021).

[41] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2017. What to Do Next: Modeling User Behaviors by Time-LSTM.. In *IJCAI*. 3602–3608.