CLLP: Contrastive Learning Framework Based on Latent Preferences for Next POI Recommendation

Hongli Zhou
220222133@seu.edu.cn
Southeast University
Nanjing, China

Zhihao Jia
zhihaojia@seu.edu.cn
Southeast University
Nanjing, China

Haiyang Zhu
haiyangzhu@seu.edu.cn
Southeast University
Nanjing, China

Zhizheng Zhang*
seu_zzz@seu.edu.cn
Southeast University
Nanjing, China

ABSTRACT

Next Point-Of-Interest (POI) recommendation plays an important role in various location-based services. Its main objective is to predict the users' next interested POI based on their previous check-in information. Most existing studies view the next POI recommendation as a sequence prediction problem but pay little attention to the fine-grained latent preferences of users, neglecting the diversity of user motivations on visiting the POIs. In this paper, we propose a contrastive learning framework based on latent preferences (CLLP) for next POI recommendation, which models the latent preference distributions of users at each POI and then yield disentangled latent preference representations. Specifically, we leverage the cross-local and global spatio-temporal contexts to learn POI representations for dynamically modeling user preferences. And we design a novel distillation strategy to make full use of the collaborative signals from other users for representation optimization. Then, we disentangle multiple latent preferences in POI representations using predefined preference prototypes, while leveraging preference-level contrastive learning to encourage independence of different latent preferences by improving the quality of latent preference representation space. Meanwhile, we employ a multi-task training strategy to jointly optimize all parameters. Experimental results on two real-world datasets show that CLLP achieves the state-of-the-art performance and significantly outperforms all existing solutions. Further investigations demonstrate the robustness of CLLP against sparse and noisy data.

CCS CONCEPTS

• **Information systems → Recommender systems**.

*Corresponding author.

KEYWORDS

Next POI Recommendation; Contrastive Learning; Latent Preference Modeling; Graph Distillation Operator

ACM Reference Format:
Hongli Zhou, Zhihao Jia, Haiyang Zhu, and Zhizheng Zhang. 2024. CLLP: Contrastive Learning Framework Based on Latent Preferences for Next POI Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3626772.3657730

1 INTRODUCTION

With the advances in information and communication technologies, people tend to share daily lives on social media. The location-based services usually employ next Point-Of-Interest (POI) recommendation systems that are developed based on users' check-in behaviors to enhance their experiences. As a specific sub-class of POI recommendation, next POI recommendation aims to predict the next POI that a particular user is most likely to visit based on her/his historical trajectory [26, 34]. This usually requires capturing users' preferences by analyzing their current location, historical check-in sequences and geographical information of POIs [3, 6, 19].

Due to the consensus that users' check-in sequences reflect their behavioral preferences from multiple aspects, the majority of existing methods focus on modeling sequences to extract users' preferences for prediction. Early studies use Markov chains to model sequential transitions patterns [3, 7, 8, 26]. To better model the sequential dependencies of check-in sequences, the sequence models, such as Recurrent Neural Networks (RNN) and its variants, are adopted by several works to leverage spatio-temporal intervals between adjacent POIs [17, 19, 25, 34]. Other attempts turn to extend RNNs with additional gate mechanisms for capturing crucial preference information [41]. And some studies separately model users' short-term and long-term preferences [4, 27, 38], achieving significant results. Besides, attention-based models have been proposed to learn from both successive and non-successive POI transitions due to the great success of Transformer [6, 11, 22, 28]. More recently, some graph-based models build the POI transition graph and user similarity graph to enrich the learned users' preferences and POI representations by explicitly considering collaborative signals of check-in sequences [16, 25, 36].

However, there are several issues in the previous studies. First, they neglect the diversity of user motivations on visiting the POIs. We argue that a user's current preference is composed of multiple latent preferences, and the most crucial latent preferences determine which POI the user will visit next. However, the entanglement of latent preferences makes it challenging to accurately capture the user's main motivations. Figure 1 shows an example. Figure 1(a) represents the two-dimensional projection of a user's preference representation. While in Figure 1(b), it represents the two-dimensional projections of the latent preference representations obtained by decomposing the user preference. It is evident that the user's current preference is mainly influenced by the latent preferences represented by yellow and blue. which implies she is more likely to visit POIs associated with both of these two latent preferences. Second, they fail to dynamically capture users' preference changes at each POI. Most existing methods model the entire check-in sequence to learn user's static preferences for prediction, which ignore the preference variations of users in different spatio-temporal contexts. It may lead the model to exhibit habitual judgments, limiting its ability to discern special circumstances. For example, a user habitually dines at $l1$ (e.g., restaurant) after work, but due to overtime today, she went to $l2$ (e.g., convenience store) for dinner because it is closer. We aim for the model to reduce the probability of user going to $l1$ based on working overtime today, rather than making judgments solely based on user habits. From the perspective of motivational diversity, if her usual post-work preference mainly consists of dining and relaxation, today it might consists of dining and convenience. This shift leads to the different choices of user. Third, the widespread presence of data sparsity problem and cold-start problem results in a lack of high-quality supervised signals to learn users' preference representations with high expressiveness and robustness.

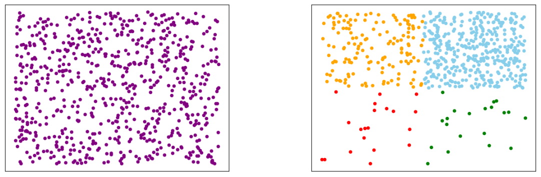

(a) (b)

Figure 1: (a) The two-dimensional projection of the user's preference representation. (b) The two-dimensional projections of the user's latent preference representations.

To address the above problems, inspired by recent developments of Self-Supervised Learning (SSL) [2, 4, 5, 20] and Disentangled Representation Learning [13, 24, 42], we propose a novel Contrastive Learning framework based on latent Preferences for next POI recommendation (CLLP), which focus on learning users' dynamic preferences based on various local contexts, and effectively disentangling users' multiple latent preferences at each POI. Specifically, we propose a enhanced transformer, which both captures users' specific behaviors in various local contexts and their general behavioral habits, to model POI representations. In other words, we aim to capture the general patterns of user preference changes. And we utilize a novel adaptive distillation graph to capture similarity of preference change patterns across all users by introducing cross-sequence collaborative signals. Then, we propose multiple independent preference prototypes, each of which is utilized to represent a aspect of user preference, to disentangle multiple latent preferences of users. To encourage independence of different latent preferences, a robust latent-preference level contrastive learning framework is employed to improve the quality of latent preference representation space, while alleviating data sparsity problem. We employ a multi-task strategy to jointly optimize the CL objective and next POI recommendation objective, due to they share similar object, i.e. modeling spatio-temporal dependencies in sequences. We summarize our contributions as follows:

- We propose a novel sequence encoder, which focuses on dynamically modeling users' preferences and effectively disentangling the multiple latent preferences of users.
- We propose a robust latent-preference level contrastive learning framework, which not only improves the quality of latent preference representation space but also alleviates the data sparsity problem. To the best of our knowledge, this is the first attempt to apply latent-preference level contrastive learning to next POI recommendation task.
- Extensive experiments on two benchmark datasets demonstrate the effectiveness of CLLP. In addition, we conducted a detailed analysis of the robustness of CLLP, and the effectiveness of contrastive learning on representation learning.

2 RELATED WORK

2.1 Next POI Recommendation

Compared with conventional POI recommendation, next POI recommendation focuses more on the spatial and temporal influence of the recent trajectory to predict a user's next moves. Early attempts apply Markov Chain-based methods [3, 7, 8, 26, 39] to estimate the transition probability between POIs. For instance, FPMC [26] applies matrix decomposition to learn a personalized transition matrix for each individual user. Meanwhile, other studies explored the possibility of tailoring the commonly-used matrix factorization or metric embedding technique into the next POI recommendation [7, 8]. Recently, STRNN [19] constructs spatio-temporal matrices by calculating the spatio-temporal distances between consecutive check-ins to extend RNN. DeepMove [6] combines Gated Recurrent Unit (GRU) networks with an attention layer to learn the sequential patterns of users' check-in data. LSTM [27] was also adopted to model users' long-term and short-term preferences. Moreover, STAN [22] explicitly exploits relative spatio-temporal distances of all the check-ins. However, all of these studies neglect the dynamic nature of users' preferences, and they fail to consider the benefits of exploiting fine-grained latent preferences of users.

2.2 Disentangled Representation Learning

Disentangled representation learning aims to explicitly separate and capture influencing factors of variations behind data [18, 30]. MacridVAE [23] is the first attempt to incorporate disentangled representation learning into users' historical data. DSSRec [24] proposes a sequence-to-sequence training strategy to extract extra supervised signals in the disentangled latent space. DGCF [32] devises intent-aware graphs to discover different user intents over items. CIGCN [42] extends GCN to maintain dimensions independent. However, these works excessively entangle in modeling insignificant intents, whereas we believe these factors do not exert a crucial impact on user behavior.

2.3 Contrastive Learning

Contrastive learning has become a major branch of self-supervised learning and has recently achieved remarkable successes in areas ranging from Computer Vision (CV) [2, 37], Natural Language Processing (NLP) [5, 30, 32], and recommender systems. In the CV community, SimCLR [2] proposes a simple contrastive learning framework for visual representations, which studies stochastic image augmentations. Following this, some works [10, 38, 40] propose various self-supervised tasks to create high-quality self-supervised signals. For NLP, the augmentation [37] for text is of a discrete nature. Most of methods [5, 9, 32] construct positive augmentation pairs by deleting, reordering and substituting words in a sentence, which also inspires our work. In recommendation scenarios, contrastive learning is also widely applied [4, 15, 20, 33]. However, it is challenging to devise a contrastive learning framework for next POI recommendation task due to the discreteness and length skewness of the check-in sequences. In this paper, we devise a robust data augmentation strategy to create fine-grained self-supervised signals to optimize representations and alleviate data sparsity problem.

3 PRELIMINARIES

3.1 Problem Formulation

This section presents our formulation of next POI recommendation task and relevant preliminary concepts. Here, we let $U = \{u_1, u_2, ...u_M\}$ be a set of users and $\mathcal{L} = \{l_1, l_2, ...l_N\}$ be a set of locations (POIs), where M and N denote the number of users and POIs, respectively. Each POI l_i owns a geographical coordinates (lat, lon) denoting its latitude and longitude.

Definition 1: (Check-in) A check-in is represented by $c = (u, l, t)$, which denotes that the user u visits the location l at timestamp t.

Definition 2: (User Trajectory) A user trajectory consists of an ordered set of check-in records. Let $T_{u_i} = \{c_1, c_2, ...c_n\}$ represents the historical trajectory of user u_i, where c_k denotes the k^{th} check-in in the trajectory, and n denotes the length of the trajectory.

Definition 3: (Next POI Recommendation) Given the trajectory T_{u_i} of the user u_i, the objective of next POI recommendation is to recommend u_i a list of top-ranked POIs that she may have interests at the next timestamp.

3.2 Data Augmentation Operators

We follow the approach in [20] to employ five augmentation operators, namely Crop (C), Mask (M), Reorder (R), Substitute (S) and Insert (I). These operators are applied to users' original check-in sequences, creating different augmented sequences. For instance, we employ the crop operator to generate augmented sequences for original sequence T_u as follows

Crop (C): Randomly select t different indices $\{idx_1, idx_2, ...idx_t\}$ in the sequence T_u. Then we delete these POIs from T_u, generating augmented suquence T_u^C:

$$T_u^C = C(T_u) = [c_1, ..., c_{idx_i-1}, ..., c_{idx_i+1} ..., c_n] \tag{1}$$

where $t = \lceil \eta * |T_u| \rceil$, T_u^C is the augmented sequence, η is a hyper-parameter that controls the number of deleted POIs, $\lceil \cdot \rceil$ is the ceiling function, and $idx_i \in [1, 2, ..., n]$. The augmentation operations of other operators are similar to the process described above. But when applying substitute operators to the original sequence, we replace each selected POI with the most similar POIs. The similarity between POIs is measured by the ItemCF-IUF [1]. More details regarding data augmentation will be discussed in Section 4.4.

3.3 Latent Preference Definition

Users may have diverse latent preferences (e.g., entertainment, exercise, relaxation, etc.), constituting their subjective motivations to visit the next POI. Hence, in this paper, we aim to project the users' preferences at each POI into K latent categories, namely $D = [d_1, d_2, ...d_k] \in \mathbb{R}^d$. Specifically, the sequence encoder projects each historical POI of T_u into K latent spaces with a certain probability expressed as $D = M_\theta(T_u, d_u)$, where θ is the set containing all the trainable parameters, and then leverage the users' main latent preferences at current POI to predict the next POI.

4 PROPOSED METHOD

In this section, we present the details of our proposed CLLP. The overall architecture of the CLLP is illustrated in Figure 2. It consists of four key components: (1) sequence encoder (part A in Figure 2), which first integrates cross-local contexts, global contexts and spatio-temporal patterns to learn POI representations for dynamically modeling user preferences and employs a new distillation graph to optimize the POI representations, and then disentangle mutiple latent preferences by using predefined preference prototypes; (2) prediction module (part B in Figure 2), which utilizes the disentanged latent preferences to predict the next POI. (3) CL module (part C in Figure 2), which creates latent-preference level self-supervised signals to encourage independence of latent preferences and alleviate data sparsity problem. (4) multi-task training module (part D in Figure 2), which jointly optimizes the CL objective and the next POI recommendation objective.

4.1 Enhanced Transformer Module

In this subsection, we will introduce the reasons for enhancing transformer structure. Transformer [28] is a strong encoder structure, which can effectively model both successive and non-successive POI transitions. It is crucial since the dependencies between POIs are not always sequential. And many of the users' trips are preplanned, implying that the selection of POIs is influenced not only by past behaviors but also by potential future behaviors [31]. Bidirectional Transformer can focus on check-in behaviors at other time points. However, the basic structure is insufficient for modeling the complex relationships among POIs based on the two reasons: First, users' preferences may vary with different local contexts, so it is vital to dynamically capture users' preference changes in various local contexts. Second, the spatio-temporal factor is one of the core foundations for user preference learning, but the transformer fails

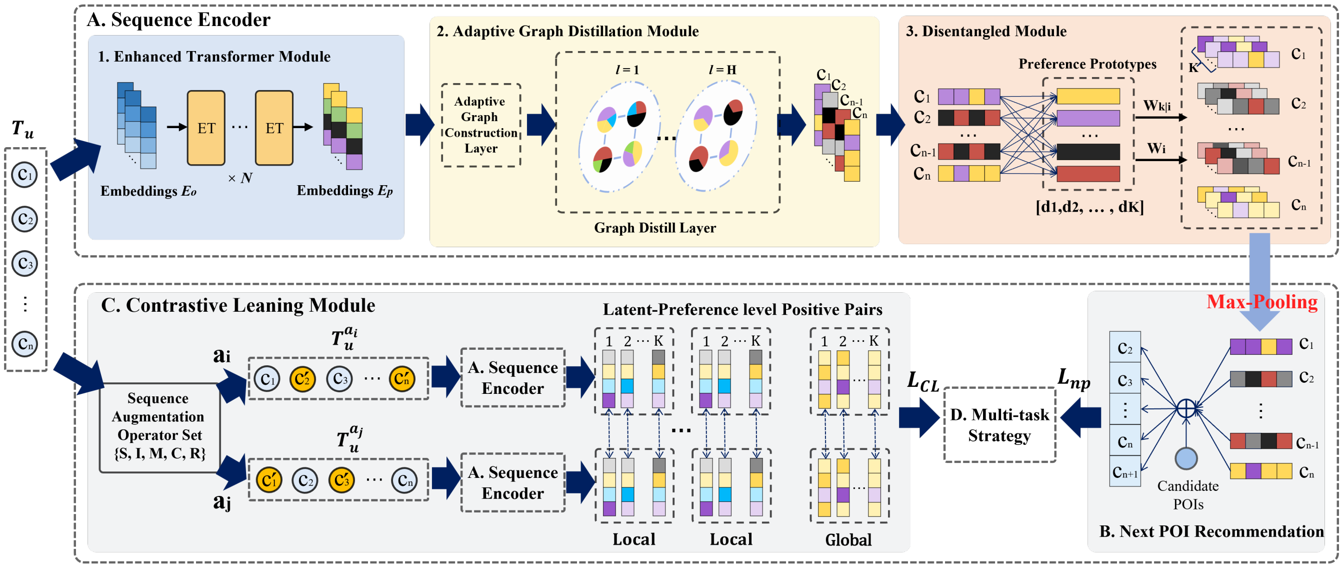


Figure 2: An overview of the proposed CLLP model

to model the spatio-temporal relationships among POIs. Considering these factors, we replace the full attention mechanism in Multi-Head Attention (MHA) with a novel Local-Window Attention (LW-Attention) and append a spatio-temporal layer after the MHA to address the first and second aforementioned limitations.

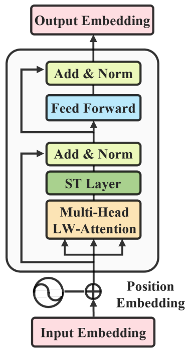


Figure 3: The structure of the Enhanced Transformer

The structure of the Enhanced Transformer (ET) is shown in Figure 3. We first add a learnable position embedding matrix $P \in \mathbb{R}^{n\times d}$ to the embedding matrix $E_0 \in \mathbb{R}^{n\times d}$ of the original sequence T_u, expressed as $E_p \in \mathbb{R}^{n\times d}$, so the original input is represented as $E_P^{(0)}$, the input of the l^{th} transformer is formulated as $E_P^{(l)}$. When the input embedding enters the MHA, we set the local window length to w to partition the entire sequence into multiple local contexts, and then independently calculate attention scores within each window to update the POI representations, capturing users' preferences in different local contexts. The attention pattern of LW-Attention is shown in Figure 4, which is calculated as

$$Attention(E_P^{(l)}) = softmax(\frac{QK^T}{\sqrt{d/h}})V \tag{2}$$

$$LW-Attention(E_P^{(l)}) = softmax(\frac{Q_w K_w^T}{\sqrt{d/h}})V_w \tag{3}$$

where $Q = E_P^{(l)} W_Q, K = E_P^{(l)} W_K, V = E_P^{(l)} W_V$ with $W_Q, W_K, W_V \in \mathbb{R}^{d\times d/h}$ are the projected query, key and value matrices respectively, Q_w, K_w, V_w are the embedding matrices in the windows, $Q_w, K_w, V_w \in \mathbb{R}^{w\times d/h}$, the h is the number of attention heads, the d is the dimension of the POI representations.

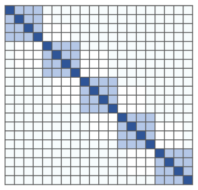


Figure 4: The attention pattern of the LW-Attention

To capture the general patterns of user preference changes, we interact the information learned from different local contexts. For any two POI c_i and c_j, instead of directly computing attention scores between two POIs, which may lead to the loss of learned local information, we use the product of local attention scores between their local representations and self-attention scores between their representations as attention weight. The local representation of a POI is defined as the mean of all POI representations within the same window. Thus, the attention weight is calculated as

$$E_{local_i}^{(l)} = Mean(E_{p_1}^{(l)}, E_{p_2}^{(l)}, ..., E_{p_w}^{(l)}) \tag{4}$$

$$W_{local_{i,j}} = \frac{E^{(l)}local_i \cdot E^{(l)}local_j}{\sum_{k=0}^{a} E^{(l)}local_i \cdot E^{(l)}local_k} \tag{5}$$

$$local-Attention(E_p^{(l)}) = F(W_{local}, Attention(E_P^{(l)})) \cdot E_p^{(l)} \tag{6}$$

$$MHA(E_P^{(l)}) = Attention(Concat(head_1, ..., head_h)) \quad where \quad head_i = local - Attention(E_P^{(l)}) \tag{7}$$

where a is the number of windows, $a = \lceil n/w \rceil$, $W_{local_{i,j}}$ is the local attention score between window i and j, W_{local} is the attention matrix for all windows, W_{global} is the self-attention matrix for all POIs, $F(\cdot)$ is a function calculating the final attention matrix.
Moreover, to enhance spatial and temporal sensitivity, we leverage general behavior patterns of users to update POI representations. In particular, users always visit some POIs (e.g., home, office, etc.) regularly and those POIs with short distance. Inspired by [6], we utilize the global spatio-temporal contexts to design a similarity function $w(\cdot)$, which reflects the correlation between historical POIs c_j and c_i, $j < i$, their similarity is calculated as

$$w_{i,j}(\Delta T_{i,j}, \Delta D_{i,j}) = \mathrm{hvc}(2\pi \Delta T_{i,j}) e^{-\Delta T_{i,j}} e^{-\Delta D_{i,j}} \tag{8}$$

where $hvc(x) = \frac{1+cosx}{2}$ is the Havercosine function modeling the daily periodicity, $\Delta T_{i,j}$ and $\Delta D_{i,j}$ denote the temporal and spatial interval between c_i and c_j, the output of $w(\cdot)$ is limited in $[0, 1]$. Then, the output of the spatio-temporal layer is calculated as

$$E_{p_i}^{(l)} = \frac{\sum_{j=0}^{i} w_{i,j} \cdot E_{p_j}^{(l)}}{\sum_{j=0}^{i} w_{i,j}} \tag{9}$$

where $w_{i,j}$ is the similarity $w_{i,j}(\Delta T_{i,j}, \Delta D_{i,j})$. Loosely speaking, the POI representations encompass information from various contexts, and thus enable the dynamic inference of the user's preferences in the current context based on their behaviors in different contexts. However, when calculating attention weights, the ideal scenario is the similarity relations between local contexts and between POIs always align, either both being similar or dissimilar, but this is not always the case. Thus, some noise information is introduced into POI representations during encoding. Specifically, if two POI representations contain only a small amount of similar information, it is considered as noise.

4.2 Adaptive Graph Distillation Module

In this subsection, we propose a novel adaptive graph distillation strategy (AGDO) to remove noise in POI representations and learn more distinguishable preference information by leveraging collaborative signals from other users. We first transform a user sequence into a weighted graph using the adaptive graph construction layer, and then it will be in turn fed into a graph distillation layer.

4.2.1 Adaptive Graph Construction Layer. To describe the similarity relations, we first transform the current sequence into a fully-connected graph G^*, where nodes in the G^* are expressed as $V = \{c_1, c_2, ..., c_n\}$, and the weight on the edge between nodes c_i and c_j is defined as the cosine similarity between their representations. We remove the edges with weights less than a predefined threshold τ from G^*, and then obtain a new weighted graph G.

4.2.2 Graph Distillation Layer. Inspired by the popular graph convolution operator (GCO) [12, 14, 29], we propose a graph distillation operator (GDO). Different from GCO, which enriches node representations by aggregating messages between neighbors, our GDO focuses on learning crucial and distinguishable information by removing similar information between neighbors. Specifically, we use GDO with H layers, and set a trainable weight matrix $\varphi^{(l)}$ for each layer, which extracts similar information between connected nodes, where $l \in \{1, 2, ..., H\}$. As the feature distributions of node representations at each layer vary, we let all weight matrices to independently learn the capacity for similarity computation. And the weight matrices are shared by all check-in sequences to capture the similarity in preference change patterns across all users. The augmented sequences can also benefit from this, as a large amount of noise will be introduced into augmented sequences due to augmentation operations, which is difficult to distinguish. So we can capture more semantic information from all sequences to better distinguish current noisy items. At each layer $l \geq 0$, the representation of POI c_i is updated as

$$E_{p_i}^{(l+1)} = E_{p_i}^{(l)} - \sum_{p_j \in N_i} \frac{\varphi^{(l)}[E_{p_i}^{(l)}, E_{p_j}^{(l)}]}{|N_i|} + b^{(l)} \tag{10}$$

where N_i is the neighbor set of c_i in graph G, $b^{(l)}$ is the bias, and $\varphi^{(l)} \in \mathbb{R}^{d \times 2d}$ is the neighbor similarity extracting matrix. We output POI representations of the last layer, i,e., $E_p^{(H)} \in \mathbb{R}^d$. It is worth noting that, due to the small proportion of noise information, the τ is usually set to a small value.

4.3 Disentangled Module

The main latent preferences reflect users' subjective motivations for visiting the next POI. To capture these preferences from the POI representations, we employ a disentangled module to disentangle the POI representations into K latent preferences. It is appended after ET and AGDO so as to reuse their expressive power.

4.3.1 Explicit weighting. In practice, we compute the distance between POI representations and latent preference prototypes to derive explicit weight, which is calculated as

$$W_{k|i} = \frac{exp(\frac{1}{\sqrt{d}}(E_{p_i} \cdot d_k))}{\sum_{j=0}^{K} exp(\frac{1}{\sqrt{d}}(E_{p_i} \cdot d_j))} \tag{11}$$

where $W_{k|i}$ is correlation coefficient between POI c_i and k^{th} latent preference, $d_k \in \mathbb{R}^d$ is the k^{th} preference prototype representation. Then we get K latent preference presentations for each position i :

$$E_{p_n}^{(k)} = LN(W_{k|i} \cdot E_{p_i}) \tag{12}$$

where $k \in \{1, 2, ..., K\}$, LN is abbreviate of the LayerNorm layer.

4.3.2 Implicit weighting. To avoid overemphasizing the role of the current POI c_n on predictions and neglecting the importance of historical locations, we calculate implicit weights using the correlation between POIs. Since the earlier POIs, which are similar to the current POI in the vector space, are more likely to be important, the implicit weights is calculated as

$$W_i = \frac{exp(\frac{1}{\sqrt{d}}(E_{p_i} \cdot E_{p_n}))}{\sum_{j=0}^{n} exp(\frac{1}{\sqrt{d}}(E_{p_j} \cdot E_{p_n}))} \tag{13}$$

$$E_{p_n}^{(k)} = \sum_{i=0}^{n} (W_i \cdot E_{p_i}^{(k)}) \tag{14}$$

where W_i is the attention weights between POI c_i and current POI c_n, $E_{p_n}^{(k)}$ is the k^{th} latent preference of POI c_n.

4.3.3 Next POI Recommendation. We predict next POI by using main latent preferences of users at the current POI. Specifically, max-pooling is employed to extract main preference, formulated as

$$E_{p_n}^{(m)} = maxpooling(E_{p_n}^{(1)}, E_{p_n}^{(2)}, ..., E_{p_n}^{(k)}) \tag{15}$$

where $E_{p_n}^{(m)} \in \mathbb{R}^d$ is user's main latent preference at current POI. We feed $E_{p_n}^{(m)}$ into a fully connected layer to generate the probability for all candidate POIs. This process is formulated as

$$y^u = W_f E_{p_n}^{(m)} \tag{16}$$

where $W_f \in \mathbb{R}^{|L| \times d}$ is the learnable weight matrix.

4.4 Contrastive Learning Module

The recommendation performance is positively correlated with the alignment and uniformity of representation space [31], so we employ CL to optimize the vector space of latent preference.

4.4.1 latent-preference level Positive Pairs Construction. Due to the discrete nature and length skewness [21] of check-in sequences, and the relationships among POIs, we generate augmented sequences considering both sequence length and correlations [1] among POIs. We introduce a hyper-parameter M as the length threshold, and apply augmentation operators as

$$T_u^a = \begin{cases} aug(T_u), & aug \in \{S, I\}, & |T_u| \leq M \\ aug(T_u), & aug \in \{S, I, M, C, R\}, & |T_u| > M \end{cases} \tag{17}$$

where $aug(\cdot)$ is an operator selected from the corresponding set, T_u and T_u^a are original sequence and augmented sequence, respectively. To be specific, given a minibatch of sequences $\{T_u\}_{n=1}^{N}$, we sample two operators from the corresponding set for each sequence T_u, which obtain $2N$ augmented sequences as:

$$\{T_{u_1}^{a_i}, T_{u_1}^{a_j}, ..., T_{u_n}^{a_i}, T_{u_n}^{a_j} ... T_{u_N}^{a_i}, T_{u_N}^{a_j}\} \tag{18}$$

where $n \in \{1, 2, ..., N\}$. We input $T_{u_n}^{a_i}$ and $T_{u_n}^{a_j}$ from the same sequence separately into the sequence encoder(part A of Figure 2), and obtain the K latent preference representations for each POI. We compute the average latent preference representations from the same preference prototype across all POIs within window w_i, obtaining the local representations of window w_i.

$$E_{l_i}^{(k)} = Mean(E_{p_1}^{(k)}, E_{p_2}^{(k)}, ..., E_{p_{w_i}}^{(k)}) \tag{19}$$

And we treat the K latent preference representations of current POI c_n as the global representations of the augmented sequence, which incorporate information of all POIs in sequence.

$$E_g^{(k)} = E_{p_n}^{(k)} \tag{20}$$

As both short sequence and long sequence contain users' similar preferences [4, 6], we utilize local and global representations to create positive pairs. Illustrated in part C of Figure 2, we treat representations of the same preference prototype within identical windows as positive pairs, denoted as $((E_{l_i}^{a_i})^{(k)}, (E_{l_i}^{a_j})^{(k)})$ and $((E_g^{a_i})^{(k)}, (E_g^{a_j})^{(k)})$, so as to generate $K \times (a+1)$ latent-preference level positive pairs. And we treat all the latent preference representations from other augmented sequences as negative pairs.

4.4.2 Constrative Loss. Based on the above operations, we can obtain latent-preference level positive pairs of the same user and negative pairs of other users. Referring to general contrastive tasks, we enhance the similarity score between the positive pairs and weaken the similarity score between the negative pairs. Specifically, we adopt the NT-Xent loss [2] as the CL loss, formulated as

$$\mathcal{L}_{CL} = -log \frac{exp(sim((E_{lg}^{a_i})^{(k)}), (E_{lg}^{a_j})^{(k)})}{\sum_{g \in neg} exp(sim((E_{lg}^{a_i})^{(k)}, g))} \tag{21}$$

where $sim(\cdot)$ measures the similarity between two positive pairs, $(E_{lg}^{a_i})^{(k)}$ is the representation of positive pairs, g is the representations of negative pairs, neg is the set of negative pairs.

4.5 Multi-Task Training

In the next POI recommendation task, we employ the cross-entropy function to define loss function as

$$\mathcal{L}_{np} = -\sum_{u=1}^{|U|}\sum_{i=1}^{m}\sum_{k=1}^{|\mathcal{L}|} f_{i,k}^u(l_k) log(y_{i,k}^u) + \beta ||\theta||_2 \tag{22}$$

where m denotes the length of check-in sequence of each user, $f_{i,k}^u$ is an indicator function that is equal to 1 if l_k is the ground truth and 0 otherwise, $||\theta||_2$ represents the $L2$ regularization of all trainable parameters, β is the hyper-parameter to control the influence of regularization, and L is the length of $\mathcal{L}$. We adopt a multi-task strategy to jointly optimize the next POI recommendation task and the CL task. The joint loss is a linear weighted sum calculated as:

$$\mathcal{L}_{joint} = \mathcal{L}_{np} + \lambda \mathcal{L}_{CL} \tag{23}$$

where λ is a hyper-parameter to control the weight of CL task loss.

Table 1: Statistic of Datasets.

	#User	#POI	#Check-in	#Density
Gowalla	52,979	121,851	3,300,986	0.13%
Foursquare	46,065	69,005	9,450,342	1.10%

5 EXPERIMENT

5.1 Experimental Setup

5.1.1 Datasets. We conduct experiments on two real-world datasets: Gowalla and Foursquare. Gowalla contains the check-ins from February 2009 to October 2010. Foursquare [35] is collected from April 2012 to January 2014. Each check-in consists of userID, POIID, latitude, longitude, and timestamp. To demonstrate the effectiveness of the proposed method in alleviating data sparsity problem, we retain all users and all POIs, and sort each user's check-ins in ascending order of timestamp. 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. The statistics of datasets are summarized in Table 1.

Table 2: The POI recommendation performance achieved by different methods on Gowalla and Foursquare datasets. The best results are in boldface, and the second-best results are underlined.

Methods	Gowalla				Foursquare			
	Acc@1	Acc@5	Acc@10	MRR	Acc@1	Acc@5	Acc@10	MRR
FPMC	0.0479	0.1668	0.2411	0.1126	0.0753	0.2384	0.3348	0.1578
PRME	0.0740	0.2146	0.2899	0.1503	0.0982	0.3167	0.4064	0.2040
TribeFlow	0.0256	0.0723	0.1143	0.0583	0.0297	0.0832	0.1239	0.0645
STRNN	0.0900	0.2120	0.2730	0.1508	0.2290	0.4310	0.5050	0.3248
DeepMove	0.0625	0.1304	0.1594	0.0982	0.2400	0.4319	0.4742	0.3270
LBSN2Vec	0.0864	0.1186	0.1390	0.1032	0.2190	0.3955	0.4621	0.2781
STGN	0.0624	0.1586	0.2104	0.1125	0.2094	0.4734	0.5470	0.3283
LightGCN	0.0428	0.1439	0.2115	0.1224	0.0540	0.1790	0.2710	0.1574
Flashback	0.1158	0.2754	0.3479	0.1925	0.2496	0.5399	0.6236	0.3805
STAN	0.0891	0.2096	0.2763	0.1523	0.2265	0.4515	0.5310	0.3420
GETNext	0.1419	0.3270	0.4081	0.2294	0.2646	0.5640	0.6431	0.3988
CLLP	**0.1718**	**0.3950**	**0.4876**	**0.2759**	**0.3115**	**0.6322**	**0.7130**	**0.4613**
Improvement(%)	21.07%	20.81%	19.49%	20.27%	17.72%	12.11%	10.88%	15.69%

5.1.2 Baselines. We consider the following state-of-the-art methods as the baselines in the experiments.

- FPMC [26]: This method estimates a personalized transition matrix via matrix factorization techniques.
- PRME [7]: This method learns user and POI embeddings to capture the personalized POI transition patterns.
- TribeFlow [8]: This method uses a semi-Markov chain model to capture the transition matrix over a latent environment.
- STRNN [19]: This method Extends RNNs by customized spatiotemporal transition matrices.
- DeepMove [6]: This method combines an attention layer with GRU to learn the periodicity and sequential patterns of users' check-ins.
- LBSN2Vec [35]: This is a hyper-graph embedding method. In this method, the POIs are ranked based on their similarities to the user and timestamp embeddings.
- STGN [41]: This method extends the conventional LSTM by adding spatial gates and temporal gates to capture user's preference in space and time dimension.
- LightGCN [6]: This method use lightweight GCN to learns users' preferences on POIs based on a pairwise ranking loss.
- Flashback [34]: This a RNN architecture that leverages spatial and temporal intervals to compute an aggregated hidden state from past hidden states for prediction.
- STAN [22]: This attention-based method explicitly utilizes the spatio-temporal information of check-ins to capture the correlations between POIs.
- GETNext [36]: This is an encoder-decoder framework that exploits the user trajectory graph and the POI transition probability graph for next POI recommendation.

5.1.3 Evaluation Metrics. Following the previous works, we employ two widely used evaluation metrics of ranking evaluation, including Accuracy@k (Acc@k) and Mean Reciprocal Rank (MRR). Acc@k estimates the ratio of true POIs occurring in the top-k recommended POIs. In the experiments, we empirically set k to 1, 5, and 10. As opposed to Acc@k, MRR measures the index of the correctly recommended POI in the ordered result list. Each metric is calculated 10 times and averaged.

5.1.4 Parameter Settings. For our baselines, all parameters are set following the suggestions from the original papers. We adopt $\{S, I, M, C, R\}$ to augment all sequences. For common hyper-parameters, we set the number of encoder layers as 3, the number of attention heads as 4, the embedding dimension as 64, the length of local window as 5. We tune H, K, τ and λ within the ranges of $\{1, 2, 3, 4, 5\}$, $[1, 10]$, $[0.1, 0.6]$, $[0.1, 1]$. We employ the Adam optimizer with default betas, the learning rate of $1e^{-3}$, the dropout rate of 0.2 for Gowalla and 0.4 for Foursquare, the β of $1.5e^{-6}$ for Gowalla and $2e^{-7}$ for Foursquare, the M of 4 and the batch size of 256. All hyper-parameters controlling the augmentation ratios are set to 0.1. We analyze the influence of key parameters in Section 5.4.

5.2 Performance Comparison

Table 2 shows the experiment results on Gowalla and Foursquare datasets. From the results, we have the following observations:

- Our proposed CLLP significantly outperforms all types of baselines in terms of every metric. We credit the improvement to following reasons. (1) The sequence model dynamically captures patterns in user preference changes, thereby enhancing the expressivity of POI representations. And it disentangle more granular users' latent preferences, so to as capture the user's main intention for predicting the next POI. (2) By creating positive and negative samples at the latent-preference level, CLLP adjusts the uniformity and alignment of the latent preference vector space, enabling the sequence model to more easily discern the distribution of latent preferences. (3) The multi-task strategy utilizes similar training objective of two tasks to train parameters of the model, alleviating data sparsity problem.
- Compared to traditional methods, deep learning-based approaches perform better as they can learn better POI representations in an end-to-end manner. Among deep learning-based methods, the best performing RNN variant (Flashback) is less competitive than GETNext, since GETNext leverages the self-attention

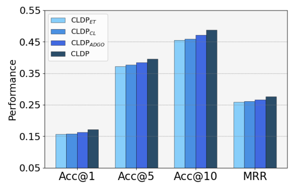
mechanism to capture dependencies among non-successive POIs. And GETNext has better performance than STAN, which is also an attention-based model. It is mainly caused that GETNext considers collaborative signals from other users to alleviate cold-start problem. This further demonstrates the necessity of alleviating data sparsity problem. However, compared with our CLLP, these methods neglect the multiple latent aspects of users' preferences, leading to suboptimal representations. This also just proves the importance of extracting latent preferences.

- As shown in Table 1, compared with Foursquare dataset, Gowalla dataset exhibits lower data density. However, CLLP achieves more improvements on Gowalla dataset. It implies that CLLP has a strong ability to handle sparse data.

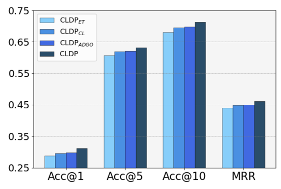
5.3 Ablation Study

To show the effects of main components in our framework, we conduct an ablation study. We denote CLLP as the base model and remove different components to obtain three variants as follows.

- $CLLP_{ET}$: we remove the ET and employ randomly initialized POI representations as input for the AGDO.
- $CLLP_{AGDO}$: we remove the AGDO and employ output of the ET as the input of the disentangled module.
- $CLLP_{CL}$: we remove the CL module and only train our model on raw datasets.



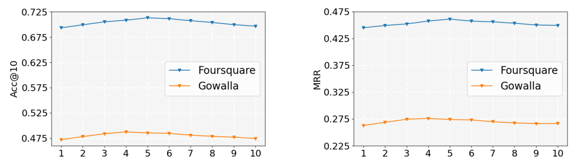
(a) Gowalla



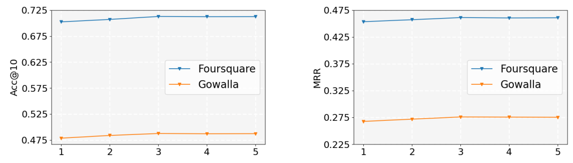
(b) Foursquare

Figure 5: Performance comparison of different variants

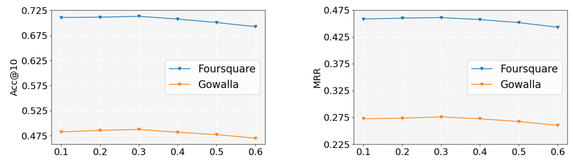
The results are shown in Figure 5. From the results, we have the following findings. First, by dropping ET which is employed to learn POI representations, $CLLP_{ET}$ suffers from a considerable decrease in performance compared with CLLP. This implies that high-quality POI representations, which contain rich users' dynamic preference information, are crucial for disentangling users' latent preferences. Second, after removing the AGDO, the performance of the $CLLP_{AGDO}$ experiences a decline. This indicates that neglecting the impact of collaborative signals from other users may diminish performance, as it aids in removing noise from POI representations, and learning more distinguishable representations. Third, we observe that $CLLP_{CL}$ is less competitive than the base model after removing CL module. One reason is that an incomplete disentanglement of latent preferences after removing CL module, which diminishes the expressivity of latent preference representations, as discussed in Section 5.5.3. And the model will suffer from the data sparsity problem, leading to a lack of sufficient supervised signals and posing challenges in learning high-quality representations. This can also be observed from the greater performance degradation of the $CLLP_{CL}$ on the Gowalla dataset, which exhibits greater sparsity compared to the Foursquare dataset.



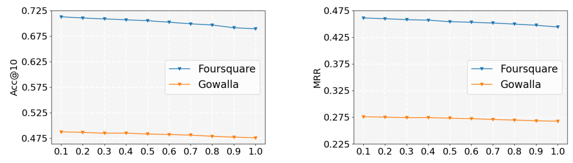
(a) Number K of latent preferences



(b) Number H of GDO layers



(c) Similarity threshold τ



(d) The weight λ of contrastive loss

Figure 6: Effect of several key hyperparameter

5.4 Hyperparameter Study

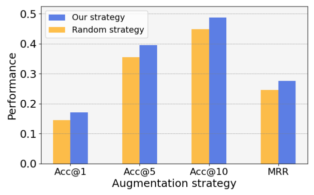
We report the effect of four key hyper-parameters on CLLP. Figure 6 shows the results.

5.4.1 *Impact of the number K of latent preferences.*

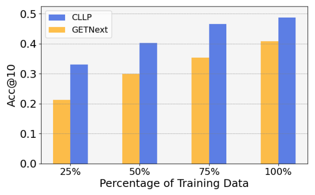
We observe that the performance peaks as K increases, and then begins to deteriorate. The reason is that a single latent preference cannot reflect the real situation of users' preferences when K is 1. Having K too large may contribute to excessive trivial preference aspects and low expressiveness, leading to poor performance. By increasing K from 1 to 5, we can observe a substantial increase of performance on both datasets. This demonstrates the sequence encoder explicitly disentangle users' latent preferences.

5.4.2 *Impact of the number H of GDO layers.*

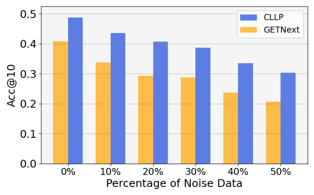
To further investigate the influence of ADGO depth, we vary the number of layer H in the range of {0, 1, ... , 5}. We find that CLLP performs better when H reaches 2 and 3. With the increase of H, ADGO can explicitly remove more noise in representations and learn more distinguishable representations. However, the improvement provided by continuing stacking layers is not obvious, as redundant stacking may no longer significantly enhance the distillation effect.



(a)



(b)



(c)

Figure 7: comparison of different augmentation strategies and the comparison of the robustness of different models.

5.4.3 Impact of the similarity threshold τ. To achieve better distillation effects, setting an appropriate similarity threshold is crucial. we observe a decline in performance with the increase of threshold when τ exceeds 0.3. This implies that with the increase of τ, the connected nodes in the graph may contain more similar information. We aim to selectively remove a small portion of noise information, removing excessive information is evidently undesirable.

5.4.4 Impact of the weight λ of contrastive learning loss. We investigate how the CL loss interacts with the next POI prediction loss. According to the evaluation results, we note that performance starts to deteriorate when λ increases over certain threshold. This implies that the contrastive learning signal complements the learning goal but should not dominate it.

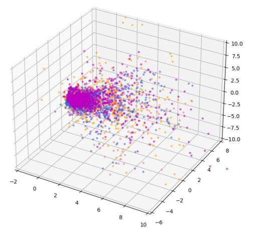
5.5 In-depth Study

Differing from prior contrastive learning researches, we incorporate the domain insight that "user's preference is composed of multiple latent preferences, with the main latent preferences predominantly influencing user's decisions". We conduct a detailed analysis of our CL strategy, and further investigate whether the CL contributes to the disentanglement of latent preferences through visualization.

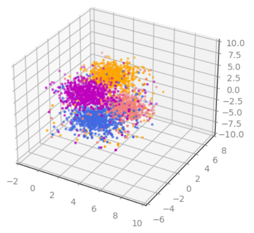
5.5.1 Augmentation Strategy Analysis. To demonstrate the effectiveness of our augmentation strategy, we conduct comparative experiments by using random augmentation strategy from previous works [9, 24], which random selects two operators from the set to replace POIs in sequence for generating augmented sequences, without considering the sequence length and correlations among POIs. The results are shown in Figure 7(a). We observe the performance of random strategy is inferior to our strategy, as it introduces a large amount of noise and exacerbates cold-start problem.

5.5.2 Robustness Analysis of Model. We evaluate the robustness of CLLP from two aspects: sparse data and noisy data. First, we both train CLLP and GETNext with partial training data respectively and keep the test data unchanged. The results are shown in Figure 7(b). We observe that CLLP consistently performs better than GETNext and the performance degradation is slower. The CLLP achieves performance with only 50% of training data close to the GETNext with full training data. This implies that the CL module better alleviate the data sparsity problem by creating more self-supervised signals. Second, we train CLLP and GETNext with full training data, and randomly add a certain proportion of noise POIs to every test sequence for inference. The results are shown in Figure 7(c). We observed the CLLP still outperforms GETNext. This implies that the high quality augmented samples endow CLLP with more robustness against the noisy interactions during the inference stage.

5.5.3 Visualization. To investigate the underlying mechanisms of proposed CL strategy. We compare visualization results of latent preference representations generated by CLLP before and after the addition of the CL. The visualization is based on PCA decomposition, which projects the embedding matrix into 3D. The results are shown in Figure 8. Before incorporating contrastive learning, despite CLLP capturing multiple latent preferences, it is still difficult to explicitly distinguish them. However, as illustrated in Figure 8(b), the CLLP can better distinguish different preferences. Since the CL can make the representations of same latent preferences to be close, and dissimilar latent preferences to be far apart.



(a) Without CL



(b) With CL

Figure 8: Comparison of the latent preference representation space on Gowalla datasets.

6 CONCLUSION

In this paper, we propose a contrastive learning framework based on latent preferences (CLLP) for next POI recommendation, to model the latent preference distributions of users at each POI by dynamically learning their preferences, and then yield disentangled representations, while leveraging contrastive learning to improve the quality of latent preference representation space and alleviate data sparsity problem. Experimental results on two real-world datasets demonstrate the superiority of CLLP to state-of-the-art methods and the robustness of CLLP. For future work, we will involve side information and further interpret disentangled representations better, and explore finer-grained exploration of the relation between augmented sequences and the operators.

ACKNOWLEDGEMENTS

We would like to thank anonymous reviewers for their thorough comments and suggestions.

REFERENCES

[1] John S Breese, David Heckerman, and Carl Kadie. 2013. Empirical analysis of predictive algorithms for collaborative filtering. *arXiv preprint arXiv:1301.7363* (2013).

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[3] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Twenty-Third international joint conference on Artificial Intelligence*.

[4] Chenghua Duan, Wei Fan, Wei Zhou, Hu Liu, and Junhao Wen. 2023. CLSPRec: Contrastive Learning of Long and Short-term Preferences for Next POI Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 473–482.

[5] Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766* (2020).

[6] 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 *Proceedings of the 2018 world wide web conference*. 1459–1468.

[7] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized ranking metric embedding for next new poi recommendation. (2015).

[8] Flavio Figueiredo, Bruno Ribeiro, Jussara M Almeida, and Christos Faloutsos. 2016. Tribeflow: Mining & predicting user trajectories. In *Proceedings of the 25th international conference on world wide web*. 695–706.

[9] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[10] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728* (2018).

[11] Qing Guo, Zhu Sun, Jie Zhang, and Yin-Leng Theng. 2020. An attentional recurrent neural network for personalized next location recommendation. In *Proceedings of the AAAI Conference on artificial intelligence*, Vol. 34. 83–90.

[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[13] 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).

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

[15] Xuewei Li, Aitong Sun, Mankun Zhao, Jian Yu, Kun Zhu, Di Jin, Mei Yu, and Ruiguo Yu. 2023. Multi-Intention Oriented Contrastive Learning for Sequential Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 411–419.

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

[17] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-aware sequential location recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2009–2019.

[18] Zihan Lin, Hui Wang, Jingshu Mao, Wayne Xin Zhao, Cheng Wang, Peng Jiang, and Ji-Rong Wen. 2022. Feature-aware diversified re-ranking with disentangled representations for relevant recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3327–3335.

[19] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[20] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. 2021. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479* (2021).

[21] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S Yu. 2021. Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer. In *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval*. 1608–1612.

[22] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the web conference 2021*. 2177–2185.

[23] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. *Advances in neural information processing systems* 32 (2019).

[24] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 483–491.

[25] Xuan Rao, Lisi Chen, Yong Liu, Shuo Shang, Bin Yao, and Peng Han. 2022. Graph-flashback network for next location recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1463–1471.

[26] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 214–221.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[30] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Longbing Cao. 2019. Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In *International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence.

[31] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*. PMLR, 9929–9939.

[32] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1001–1010.

[33] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.

[34] Dingqi Yang, Benjamin Fankhauser, Paolo Rosso, and Philippe Cudre-Mauroux. 2020. Location prediction over sparse user mobility traces using rnns. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. 2184–2190.

[35] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudre-Mauroux. 2019. Revisiting user mobility and social relationships in lbsns: a hypergraph embedding approach. In *The world wide web conference*. 2147–2157.

[36] Song Yang, Jiamou Liu, and Kaiqi Zhao. 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*. 1144–1153.

[37] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems* 33 (2020), 5812–5823.

[38] Zeping Yu, Jianxun Lian, Ahmad Mahmoody, Gongshen Liu, and Xing Xie. 2019. Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation.. In *IJCAI*. 4213–4219.

[39] Jia-Dong Zhang, Chi-Yin Chow, and Yanhua Li. 2014. Lore: Exploiting sequential influence for location recommendations. In *Proceedings of the 22nd ACM SIGSPATIAL international conference on advances in geographic information systems*. 103–112.

[40] Richard Zhang, Phillip Isola, and Alexei A Efros. 2016. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer, 649–666.

[41] Pengpeng Zhao, Anjing Luo, Yanchi Liu, Jiajie Xu, Zhixu Li, Fuzhen Zhuang, Victor S Sheng, and Xiaofang Zhou. 2020. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 5 (2020), 2512–2524.

[42] Tianyu Zhu, Leilei Sun, and Guoqing Chen. 2021. Embedding disentanglement in graph convolutional networks for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (2021), 431–442.