EEDN: Enhanced Encoder-Decoder Network with Local and Global Context Learning for POI Recommendation

Xinfeng Wang
University of Yamanashi
g22dtsa7@yamanashi.ac.jp

Fumiyo Fukumoto
University of Yamanashi
fukumoto@yamanashi.ac.jp

Jin Cui
University of Yamanashi
g22dtsa5@yamanashi.ac.jp

Yoshimi Suzuki
University of Yamanashi
ysuzuki@yamanashi.ac.jp

Jiyi Li
University of Yamanashi
jyli@yamanashi.ac.jp

Dongjin Yu
Hangzhou Dianzi University
yudj@hdu.edu.cn

ABSTRACT

The point-of-interest (POI) recommendation predicts users' destinations, which might be of interest to users and has attracted considerable attention as one of the major applications in location-based social networks (LBSNs). Recent work on graph-based neural networks (GNN) or matrix factorization-based (MF) approaches has resulted in better representations of users and POIs to forecast users' latent preferences. However, they still suffer from the implicit feedback and cold-start problems of check-in data, as they cannot capture both local and global graph-based relations among users (or POIs) simultaneously, and the cold-start neighbors are not handled properly during graph convolution in GNN. In this paper, we propose an enhanced encoder-decoder network (EEDN) to exploit rich latent features between users, POIs, and interactions between users and POIs for POI recommendation. The encoder of EEDN utilizes a hybrid hypergraph convolution to enhance the aggregation ability of each graph convolution step and learns to derive more robust cold-start-aware user representations. In contrast, the decoder mines local and global interactions by both graph- and sequential-based patterns for modeling implicit feedback, especially to alleviate exposure bias. Extensive experiments in three public real-world datasets demonstrate that EEDN outperforms state-of-the-art methods. Our source codes and data are released at https://github.com/WangXFng/EEDN.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

POI Recommendation, Implicit Feedback, Cold Start Issue, HyperGraph Convolution, Exposure Bias

ACM Reference Format:
Xinfeng Wang, Fumiyo Fukumoto, Jin Cui, Yoshimi Suzuki, Jiyi Li, and Dongjin Yu. 2023. EEDN: Enhanced Encoder-Decoder Network with Local and Global Context Learning for POI Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

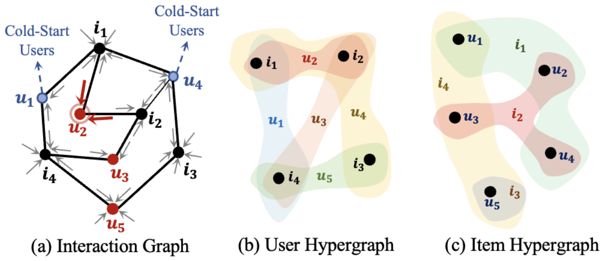
With the breakthrough of mobile internet, location-based social network (LBSN) services have gained major popularity for users to share their experience through checking their routes. Likewise, a large amount of data consisting of check-in event histories are generated in various LBSN services every day. The information obtained by analyzing such rich data is beneficial for many downstream tasks, including POI recommendations. However, as check-ins data are observational, these include various biases, such as selection bias and exposure bias [3]. This causes major obstacles for POI recommendations to mining users' personal preferences. One such obstacle is that the check-in dataset is implicit feedback, where there are no labels such as rating scores indicating the user's preference for POIs. Therefore, a recommendation system is required to estimate users' preferences based on an assumption. One typical assumption is that users with similar preferences in the neighborhood, as reflected in historical behaviors, are likely to have similar interests in the future [20]. Several metrics, such as Euclidean distance and the Jaccard index, are applied to measure local relations, that is, the similarity between a target user and its structural or semantic neighborhood [13]. Some approaches focus on global relations, which exploit users' preferences by leveraging the learned similarity defined on latent feature vectors. However, most of them focused on either local [16] or global relations [32, 35, 39].

Another obstacle is the cold-start issue: inactive users have few visiting histories or some POIs attract little attention from users. A recommendation model fails to collect enough knowledge related to such users or POIs. Ignoring these users or POIs during training leads to poor prediction performance. Several authors have attempted to solve the issue and improve performance by leveraging various implicit contextual features embedded in existing check-in data, such as spatial-temporal information [30, 35], social relations [38], and sequential dependence [19, 35]. However, most of these approaches rely heavily on the availability and quality of auxiliary data, resulting in a lack of robustness of the model.

More recently, POI recommendations based on deep learning (DL) have been intensively studied. These attempts include recurrent neural networks (RNNs) [12], transformer [35], generative adversarial network (DGAN) [43], and graph convolutional networks

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn



(a) Interaction Graph (b) User Hypergraph (c) Item Hypergraph

Figure 1: Illustration of interaction graph and hypergraph.

(GCNs) [31, 33]. These enable the handling of non-linear interactions between users and POIs. On the one hand, several authors have proposed DL methods to solve implicit feedback problems and have attempted to capture both local and global relations to model users' preferences [15, 34]. Despite some successes, it is still limited to modeling users' potential features as their model only utilizes a user-item matrix or a spatial-temporal sequence. On the other hand, the authors attempted to alleviate the cold-start problem by utilizing latent patterns behind the user-item interactions. One major attempt is to use meta-learning [10, 27]. Although they have achieved great success, they do not take high-order collaborative signals into consideration, and thus fail to capture the diversity among various users in different POIs. Graph-based DL methods captured high-order collaborative structure information. However, graph-based approaches, such as GCNs, do not handle cold-start neighbors during graph convolution. For example, (a) in Fig. 1, for the target u_2, the 2-order neighbors u_1, and u_4, are propagated to u_2, although these neighbors are also cold-start users with low-quality embeddings, resulting in representation degeneration.

Motivated by the previous work mentioned above, we propose an enhanced encoder-decoder network (EEDN) to exploit rich latent features between users or POIs, and interactions between users and POIs to alleviate implicit feedback and cold-start problems for POI recommendation. The EEDN is based on the encoder-decoder network, where the encoder uses a hybrid hypergraph convolution. As shown in (b) a user hypergraph and (c) an item hypergraph transformed from (a) an interaction graph of Fig. 1, a hypergraph convolution captures the beyond pairwise relations. For example, u_2 in (b) and i_2 in (c) are enhanced through each graph convolution step and avoid ineffective propagation between cold-start users, u_1 and u_4 in (b) or nodes i_1 and i_3 in (c), learnings to select effective neighbors. The decoder of the EEDN, on the other hand, applies both graph- and sequential-based methods to the same input data to capture rich local and global interactions for modeling implicit feedback. Furthermore, the EEDN can adapt prevalent encoders (e.g., Transformer, Long-Short Transformer (LS-Transformer), and Multi-Layer Perceptron with gating (gMLP)), indicating potential robustness for further improvements. The main contributions of our work can be summarized as follows:

- We propose an enhanced encoder-decoder network (EEDN) to exploit rich latent features between users, POIs, and interactions between users and POIs for POI recommendation.
- We propose a method for mining local and global interactions by leveraging both graph- and sequential-based patterns for modeling implicit feedback, especially to deal with the POIs that users prefer but have not yet visited.
- To handle the cold-start neighbors, we utilize a novel hypergraph convolution encoder to enhance the ability of learning to select effective neighbors. We empirically show its effectiveness by comparing three popular encoders.
- Extensive experiments on three public real-world datasets show that our model achieves state-of-the-art results.

2 RELATED WORK

2.1 Implicit Feedback Problem

Implicit feedback is a common problem in recommendation systems, and many authors have attempted to apply DL methods. Ma et al. proposed a method consisting of a self-attentive encoder and a neighbor-aware decoder (SAE-NAD) to learn complex user-POI relations [19]. Liu et al. proposed a hierarchical Bayesian framework that integrates the stacked denoising autoencoder with probabilistic matrix factorization to leverage both the local and global correlations among the items [15]. Chen et al. proposed efficient neural matrix factorization (ENMF), which is derived from three optimization algorithms: user-, item-, and alternating-based algorithms for efficient recommendations [1]. Han et al. presented a method called contextualized point-of-interest recommendation (CPIR) for ranking POI candidates [9]. Wu et al. proposed a dual attentive neural network for session-based recommendations [34]. Ni et al. presented a recommendation model that encodes heterogeneous similarities in an expanded prediction rule [20]. Wang et al. proposed a strategy called adaptive denoising training, which adaptively prunes noisy interactions during training [29].

Our approach is similar to Ma et al.'s SAE-NAD method; that is, the approach is based on the encoder-decoder model [19]. The difference is that the correlations between POIs and users are learned by utilizing both sequential patterns and structured graphs to mining rich latent features between POIs, users, and the interactions between users and POIs, while they only focused on sequential patterns and applied the attention mechanism to them.

2.2 Cold-Start Problem

Recent attempts related to the cold-start problem can be summarized into two groups: models to utilize auxiliary data and models to mine the underlying patterns from the source data. In the context of the first group, Yin et al. utilized spatial information [37], Zhabiz et al. and Yin et al. incorporated the social trust path [8, 36], and Wang et al. utilized knowledge graphs [28]. However, additional contextual information is not always available, resulting in poor-quality cold-start embeddings [10].

In the context of the second group, one major attempt was to use meta-learning [10, 27]. It learns how to learn new tasks by utilizing prior experience with the related tasks, and is classified into two groups: metric-based [5] and model-based [7, 18, 24] approaches. Although they have achieved promising results, they have not fully considered capturing high-order collaborative signals for recommendations. Graph neural networks have also been well-studied. These attempts include LightGCN [11], self-supervised tri-training (SEPT) [38], self-supervised graph learning (SGL) [32], and simple

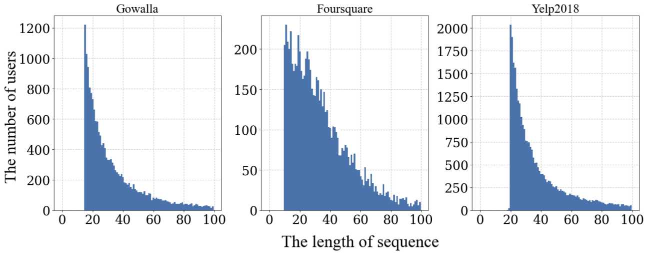


Figure 2: Distribution of users with different activeness

graph contrastive learning (SIMGCL) [39]. However, GNN-based approaches optimize the likelihood of a user adopting an item, which is not a direct improvement of the embedding quality of cold-start users or items. Hao et al. proposed a pre-training GNN to learn high-quality embeddings for cold-start users or items [10]. They utilized meta-learning to select the effective neighbors adaptively while ignoring the rich behavior patterns of users as is often the case in POI recommendation [33].

3 PRELIMINARIES

3.1 Definitions

Definition 1 (Point-of-interest (POI)). A POI $p \in \mathcal{P}$ ($|\mathcal{P}| = N$) is defined as a uniquely identified location that users might be interested in.

Definition 2 (POI-POI graph). The POI-POI graph $\mathcal{G} = (\mathcal{P}, \mathbf{A})$ shows the interaction correlation network between POIs, where $\mathcal{P}$ denotes a set of POIs. $\mathbf{A} \in \mathbb{R}^{N \times N}$ refers to an adjacent matrix. Each element $a_{i,j} = 1$ shows POIs p_i and p_j have been visited by the same user, and otherwise $a_{i,j} = 0$.

Task 1 (POI Recommendation). Given a user $u \in \mathcal{U}$ ($|\mathcal{U}| = M$) visit sequence $\mathcal{S}_u = \{p_k\}_{k=1}^{L}$, where L refers to the number of POIs in the sequence and a POI-POI graph $\mathcal{G}$, the goal is to generate a list of top-k POIs with the highest probabilities of unvisited POI candidates from $\hat{\mathbf{r}}_{u,*} \in \mathbb{R}^N$ for the target user u.

3.2 Hypergraph for Cold-Start Users

The cold-start problem is a difficult challenge, as it leads to a severe lack of information, reflected in the insufficient exposure of users' historical behaviors, which show personalized preferences. As an example of LBSN datasets illustrated in Fig. 2, the number of users decreases exponentially when their activeness increases, falling into the long tail of medium-to-low active users. This indicates that the majority of users are inactive and these users affect overall performance.

To alleviate the issue mentioned above, we focused on hypergraphs for blocking high-order propagation from cold-start users and preserving effectively propagating collaborative signals among nodes (POIs). The hypergraph is a graph to represent bi-directional correlations, where a hyperedge is allowed to connect more than two vertices [40, 41]. We recall that in Fig. 1 (b) and (c), users are vertices, and items are hyperedges in the user hypergraph and vice versa, which are derived from (a) in Fig. 1. Hypergraphs can capture important items (or users) through interactions between users and items. More importantly, hypergraph convolution can restrict information propagation between item nodes, with the hyperedge corresponding to each user during user representation learning. To compensate for the dilemma of missing global signals, global contexts in POI-POI graphs were integrated into hypergraph convolution through iteratively repeating convolution processes for each user.

4 METHODOLOGY

The framework is illustrated in Fig. 3, which consists of a hypergraph convolution encoder (HCE) and a matrix factorization decoder (MFD). The former aims to enhance users' personalized preferences by integrating local and global interaction patterns. The latter explicitly matches the user feature vectors with those of POIs to obtain rating scores. It simultaneously learns implicit features from graph-based and sequential presentations as auxiliary information in recommendation decision-making.

4.1 Hypergraph Convolution Encoder

Standard hypergraph-based functions naturally generate representation from feature maps using transformations of weight matrices and hyperedge matrices. Let $\mathbf{R} \in \mathbb{R}^{M \times N}$ be a rating matrix. The hypergraph Laplacian matrix [42] is defined as follows:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{H}\mathbf{W}_1\mathbf{\Delta}^{-1}\mathbf{H}^{\top}\mathbf{D}^{-\frac{1}{2}}, \tag{1}$$

where the diagonal matrix $\mathbf{D} \in \mathbb{R}^{M \times M}$ denotes degrees of nodes, the diagonal matrix, $\mathbf{W}_1 \in \mathbb{R}^{N \times N}$, indicates the weights of the hyperedges; the diagonal matrix $\mathbf{\Delta} \in \mathbb{R}^{N \times N}$ shows the degrees of the hyperedges; and $\mathbf{H}$=$\mathbf{R} \in \mathbb{R}^{M \times N}$ refers to the hyperedges for the user hypergraph (there are M users and N hyperedges, and each hyperedge contains the user's rating scores for items). We used the user-POI interaction matrix $\hat{\mathbf{H}} \in \mathbb{R}^{M \times N}$ instead of the rating matrix. Here, each row in $\hat{\mathbf{H}}$ is for a hyperedge and $\hat{\mathbf{H}}_{ij} = 1$ if vertex i is connected by hyperedge j and otherwise $\hat{\mathbf{H}}_{ij} = 0$.

Recalling that user hypergraphs are beneficial for extracting not only local relatedness between users' visited POIs [40, 41] but also global contexts across the whole candidate POIs, which can be inferred from the POI-POI graph [11], we incorporated them into our method and proposed a novel hypergraph convolution method. **Hypergraph Convolution Network (HGCN).** For the set of POIs $\mathcal{P}$, we generated the global embedding matrix $\mathbf{E}_p \in \mathbb{R}^{d_m \times N}$, where d_m refers to the dimension of embedding and the i-th column of $\mathbf{E}_p$ indicates the N-dimensional vector for POI p_i. Since the multiplication of $\mathbf{E}_p$ and a hyperedge can be regarded as the slicing operation that selects corresponding POI vectors, the procedure of the slicing operation for user u_k is given as follows:

$$\mathbf{E}_p\hat{\mathbf{H}}_{k,*} = \mathbf{E}_p(\hat{\mathbf{H}}_{k,1}, \hat{\mathbf{H}}_{k,2}, ..., \hat{\mathbf{H}}_{k,N}), \tag{2}$$

where $\hat{\mathbf{H}}_{k,i} = 1$ if user u_k has interacted with POI p_i, $\hat{\mathbf{H}}_{k,i} = 0$ otherwise. Thus, for each POI p_j from the historical visit sequence $\mathcal{S}_u$, we denote its embedding vector to $\mathbf{E}_p\mathbf{y}_j \in \mathbb{R}^{d_m}$, where $\mathbf{y}_j \in \mathbb{R}^N$ is the one-hot vector for p_j and $j \in \{1, 2, ..., L\}$. The check-in POI embedding matrix $\mathbf{E}^{\mathcal{S}_u} \in \mathbb{R}^{d_m \times L}$ of $\mathcal{S}_u$ is given by:

$$\mathbf{E}^{\mathcal{S}_u} = \mathbf{E}_p[\mathcal{S}_u] = \mathbf{E}_p\mathbf{Y}^{\top}, \tag{3}$$

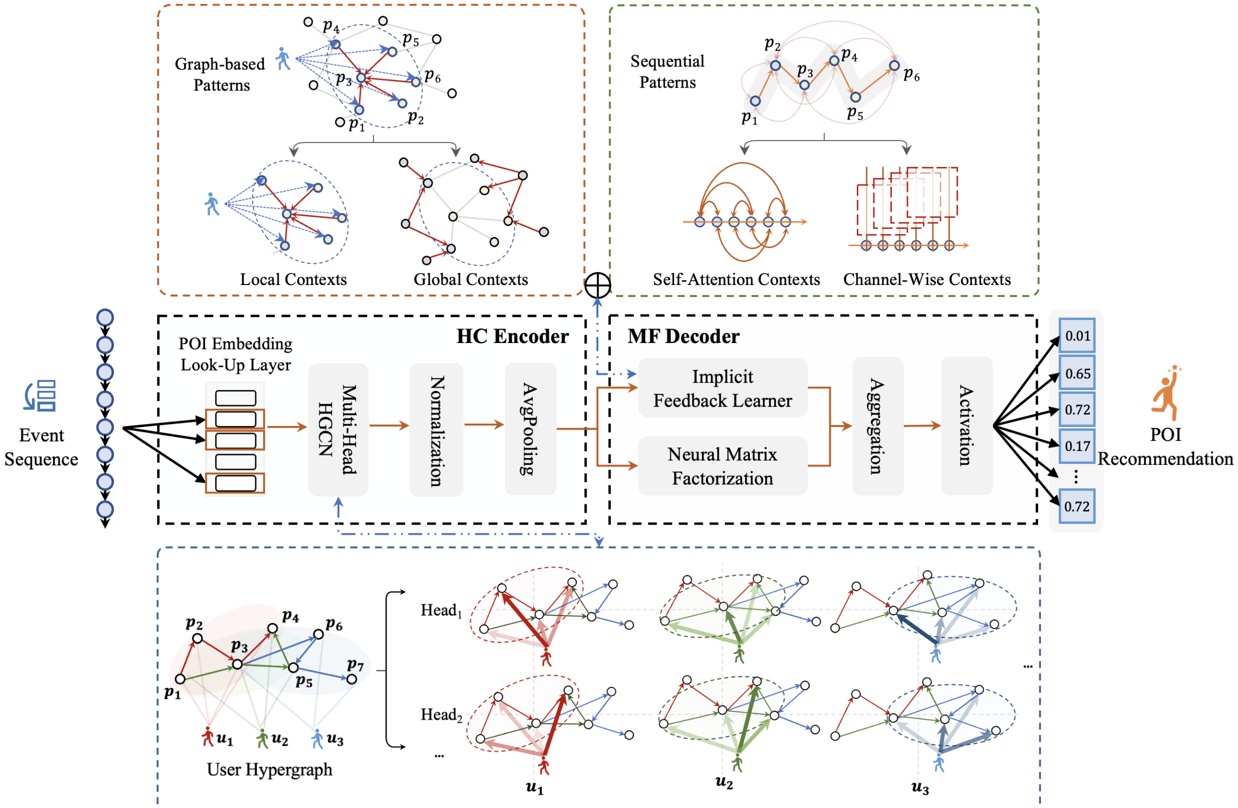


Figure 3: Architecture of proposed framework

where $[\cdot]$ denotes the slicing operation, and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_L] \in \mathbb{R}^{N\times L}$ refers to the POIs in $\mathcal{S}_u$. The k-th column of $\mathbf{E}_p$ shows the embedding of the specific POI $\mathbf{y}_k$ in $\mathcal{S}_u$.

He et al. proposed simplifying the architecture of GCN by eliminating the weight matrix of global nodes ($\mathbf{W}_1 \in \mathbb{R}^{N\times N}$) and nonlinear activation [11]. Following their framework, we removed the global weight matrix to focus on local information integrations among POIs within the hyperedge. As a result, it simplifies computational complexity without sacrificing performance. Specifically, the computational complexity of traditional HGCN blocks is $O(N^2)$ for the item hypergraph, while our model is $O(L^2)$, where $N \gg L$. Here, N denotes the total number of POIs and L indicates the length of the user sequence. Furthermore, we empirically added a local linear feature transformation ($\mathbf{W}_2 \in \mathbb{R}^{d_m\times d_m}$) to enhance its capability for selecting neighbors of user nodes, as the query, key, and value transformations, which are formulated with Transformer [25]. Let $\mathbf{E}^{(0)}$ be the check-in POI embedding matrix $\mathbf{E}^{\mathcal{S}_u}$. We obtained the matrix equivalence form of informative diffusion in HGCN as follows:

$$\hat{\mathbf{E}}^{(l)} = \hat{\mathbf{A}}\alpha_1(\mathbf{E}^{(l)})\mathbf{W}_2, \tag{4}$$

where $\mathbf{E}^{(l)}$ indicates the output of the l-th HGCN layer, α_1 is ELU activation function, and $\hat{\mathbf{A}} \in \mathbb{R}^{L\times L}$ denotes the symmetrically normalized sub adjacent matrix of POIs in $\mathcal{S}_u$, as follows:

$$\hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}[\mathcal{S}_u], \tag{5}$$

where $[\cdot]$ refers to the slicing operation, which is similar to Eq. (3), $\mathbf{A} \in \mathbb{R}^{N\times N}$ denotes the global adjacent matrix for the POI-POI graph, and $a_{ij} = 1$, if POI p_i and POI p_j have been visited by the same user, otherwise 0. $\mathbf{D} \in \mathbb{R}^{N\times N}$ is a diagonal matrix in which each entry d_{ij} denotes the number of nonzero entries in the i-th row vector of the adjacency matrix $\mathbf{A}$. Note that the symmetrical normalization of the adjacent matrix, $\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, is widely used to quantify the global correlations between adjacent nodes and each target node [11, 13, 32], which can be represented as the weights assigned to users' visited POIs when performing hypergraph convolution. Moreover, some POIs have properties with multiple aspects and they mutually affect users' decisions. For example, a restaurant provided delicious food, but the service was disappointing. This indicates that the latent factor also plays an important role to improving performance. The EEDN thus adopts a multi-head (MH) block [25] in the HGCN to strengthen its learning effect for multiple latent factors. The $\mathbf{E}^{(l+1)}$ is given as follows:

$$\mathbf{E}^{(l+1)} = \text{MultiHead}(\hat{\mathbf{E}}_1^{(l)}, \hat{\mathbf{E}}_2^{(l)}, ..., \hat{\mathbf{E}}_{z_h}^{(l)}), \tag{6}$$

where z_h denotes the number of heads in an MH block. Equal-weight summation as the aggregation function works better than a feed-forward network in MH HGCN blocks. Subsequently, let the

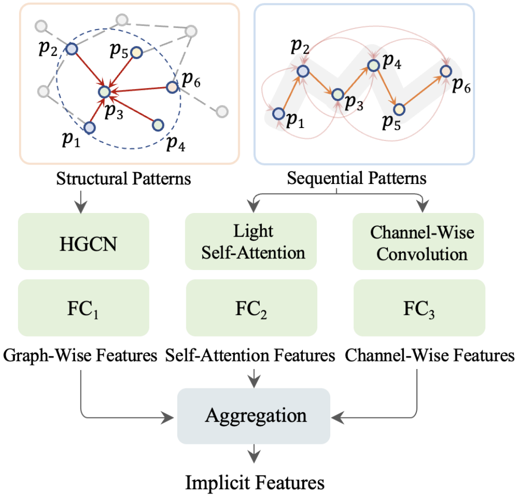


Figure 4: Illustration of Implicit Feedback Learner

output of the last layer be the hidden representation $\mathbf{h} \in \mathbb{R}^{d_m \times L}$, the $\mathbf{h}$ is fed into an average-pooling layer and an L$_2$ normalization layer, which is formulated as follows:

$$\mathbf{e}_u = \mathrm{Norm}(\mathrm{AvgPooling}(\mathbf{h})), \tag{7}$$

where $\mathbf{e}_u$ denotes the representation of the user u.

Recall that user representation is aggregated by the POIs in the user hyperedge. The overall formulation of the hypergraph convolution is rewritten as follows:

$$\mathbf{e}_u = \mathrm{AGG}(\{\mathbf{e}_{p_j} : p_j \in \mathcal{H}_u\}), \tag{8}$$

where $\mathbf{e}_p$ and $\mathbf{e}_u$ refer to the feature vectors of a POI, and a user, respectively. AGG is the aggregation function, and $\mathcal{H}_u$ indicates the hyperedge of the user u.

4.2 Matrix Factorization Decoder

The objective of the decoder is to compute the matching degree of feature vectors of users and POIs to propagate the gradient feedback and distill the implicit features from two aspects: structural and sequential patterns. Thus, the decoder consists of two key modules: implicit feature learner (IFL) and neural matrix factorization (NMF).

Implicit Feature Learner. Indeed, only positive examples are observed in the check-in records, while negative samples always include false-negative samples. In other words, a user would really prefer a POI, but the user has not yet had a chance to visit it. This is called exposure bias, which is one of the vital challenges to address in implicit feedback recommendation approaches. The exposure bias causes defective representations and biased rating scores during training, with no feedback from false-negative samples. To alleviate this issue, the EEDN distills the implicit features from structural and sequential patterns and utilizes these features to assist the MFD with having more generalization. As shown in Fig. 4, graph-wise, self-attention, and channel-wise features were collected from both structural and sequential patterns simultaneously and aggregated as compound implicit features.

Graph-Wise Features. As for structural patterns, after HGCN blocks both local and global aggregate interaction features, a simple but effective linear transformation layer is applied to learn latent graph-wise features:

$$\mathbf{f}_{graph} = \mathbf{e}_u \mathbf{W}_3, \tag{9}$$

where $\mathbf{f}_{graph}$ refers to the graph-wise implicit features, and $\mathbf{W}_3 \in \mathbb{R}^{d_m \times N}$ is the weight matrix shared by all users.

Self-Attention Features. Among users' historical visits, some representative POIs should contribute to reflecting the user's preferences. Motivated by this fact, we applied a self-attention mechanism to integrate the POI representation weighted by the dependencies between users' visited POIs to characterize their latent preferences. Empirically, we found that normal self-attention tends to grab more gradients from NMF, leading to defects in NMF. As a result, this leads to the deterioration of the overall recommendation quality. We thus eliminate the projection transformation of the query, key, and value signals, which we call light self-attention (LSA):

$$\begin{aligned} \mathrm{S}_{att} &= \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}})\mathbf{V}, \\ \mathbf{f}_{seq_1} &= \mathrm{Norm}(\mathrm{AvgPooling}(\mathrm{S}_{att}))\mathbf{W}_4, \end{aligned} \tag{10}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the query, key, and value matrices that are equivalent to the original signal, i.e., the check-in POI embedding matrix $\mathbf{E}^{\mathcal{S}_u}$, S_{att} denotes the output of LSA, $\mathbf{f}_{seq_1}$ refers to the self-attention implicit features, and $\mathbf{W}_4 \in \mathbb{R}^{d_m \times N}$ is the weight matrix. Due to the simplification of the self-attention mechanism, LSA can reduce computational complexity.

Channel-Wise Features. On the one hand, light self-attention features can reveal latent heterogeneity among users' visited POIs by unearthing the global dependencies. On the other hand, it is restricted to mapping features to subspaces and lacks local correlations between POIs. To mitigate this issue, we choose convolution layers [4] to model local relations. Let Conv2D$(\cdot)$ be the stacked two-dimensional convolution layers. The complementary part of sequential features is obtained by:

$$\begin{aligned} \mathrm{S}_{conv} &= \mathrm{Conv2D}(\mathbf{E}^{\mathcal{S}_u}), \\ \mathbf{f}_{seq_2} &= \mathrm{Norm}(\mathrm{AvgPooling}(\mathrm{S}_{conv}))\mathbf{W}_5, \end{aligned} \tag{11}$$

where S_{conv} denotes the output of convolution layers, $\mathbf{f}_{seq_2}$ refers to the channel-wise implicit features, and $\mathbf{W}_5 \in \mathbb{R}^{d_m \times N}$ is the weight matrix.

Neural Matrix Factorization. The rating scores are defined as the inner product of the user and candidate POI representations:

$$\mathbf{r}_{u,p} = \mathbf{e}_u^\top \mathbf{e}_p, \tag{12}$$

where $\mathbf{r}_{u,p}$ denotes the rating score that user u gives to POI p. We used this as a ranking score for the recommendation.

Aggregation. The final rating score with implicit feedback learning is defined by the weighted sum of the explicit rating score and the compound implicit feature:

$$\hat{\mathbf{r}}_{u,*} = \alpha_2(\mathbf{r}_{u,*} + \mathbf{f}_{graph} + \beta_1 \mathbf{f}_{seq_1} + \beta_2 \mathbf{f}_{seq_2}), \tag{13}$$

Table 1: Statistics of Check-in Dataset. "#Avg. Seq." denotes the average number of users' visit sequences.

Dataset	#User	#POI	#Check-ins	#Avg. Seq.
Gowalla	18,737	32,510	1,278,274	67.1
Foursquare	7,642	28,483	512,532	68.2
Yelp2018	31,668	38,048	1,561,406	49.3

where $\hat{\mathbf{r}}_{u,*}$ denotes the final ranking scores of all POI candidates for user u, α_2 is an activation function tanh, and β_1 and β_2 are hyperparameters.

4.3 Loss Function

Many recent works [11, 13, 26, 32, 38, 39] have used Bayesian Personalized Ranking (BPR) loss [22] as their major or a part of their objective function. The idea is to encourage positive samples and punish a certain number of missing values randomly picked from the candidates as negative samples. However, they cannot distinguish between false- and true-negative samples in the missing values with high accuracy. Therefore, instead of penalizing unknown (stochastic) negative samples via the BPR function, the EEDN adopts a cross-entropy objective function that encourages only positive samples. Specifically, the element in $\boldsymbol{\gamma}$ in Eq. (14) is set to 0 regardless of whether it is a true or false negative sample.

It is worth noting that in POI recommendation tasks, the positive examples usually consist of two groups of samples: visited and future-visited POIs. Inspired by [19], we adopted an improved weighting scheme to distinguish not only positive and negative samples but also the confidence levels between visited and future visited POIs. The rating loss is defined as follows:

$$\mathcal{L}_{rating} = -\mathbf{c}_{u,*} \odot \boldsymbol{\gamma}^\top \log\left(\frac{\exp(\hat{\mathbf{r}}_{u,*})}{\sum_j \exp(\hat{\mathbf{r}}_{u,j})}\right) \tag{14}$$

where $\odot$ denotes the element-wise multiplication and $\boldsymbol{\gamma}$ refers to a true label vector, and each dimension of the vector equals 1 when the user has ever visited the POI; otherwise, 0. Specifically, we set the weights of negative (including false negative) samples to the same value, i.e., $\mathbf{c}_{u,p} = 1$ when POI p is a negative sample for user u, and the weights of visited and future visited POIs be λ and δ, respectively. Thus, $\mathbf{c}_{u,p}$ is given as follows:

$$\mathbf{c}_{u,p} = \begin{cases} 1 & \text{user } u \text{ didn't and won't visit POI } p, \\ \lambda & \text{user } u \text{ visited POI } p, \\ \delta & \text{user } u \text{ will visit POI } p, \end{cases} \tag{15}$$

where λ and δ are two hyperparameters to balance the weights between the visited and future visited POIs for users.

5 EXPERIMENT

5.1 Experimental Setup

Datasets and Evaluation Metrics. We performed the experiments on three benchmark datasets, i.e., Gowalla [9, 30], Foursquare, and Yelp2018 [11, 39], to evaluate the EEDN. Gowalla data were collected from February 2009 to October 2010. The Foursquare includes check-in data ranging from April 2012 to September 2013. The Yelp2018 data were collected from the 2018 edition of the Yelp Challenge. The statistical details are shown in Table 1. Following [9, 39], we split each dataset into three folds. For each user, we used the earliest 70% check-ins as training data, the most recent 20% check-ins as test data, and the remaining 10% as validation data. To evaluate the quality of the recommendation, we used two metrics: Recall@K (R@K) and NDCG@K (N@K) with $K \in \{5{,}10{,}20\}$.

Baselines. We compared the EEDN with the following 10 state-of-the-art models which are classified into five groups:

- *Matrix Factorization (MF)-based method*:
- **CPIR** [9] alternately utilizes MF on user and POI similarity to recommend top-k POIs to users.
- **ENMF** [1] applies user-based, item-based, and alternating-based optimization algorithms for efficient recommendations.

- *Self-Attention (SA)-based method*:
- **SAE-NAD** [19] consists of a self-attentive encoder and a neighbor-aware decoder for POI recommendations.
- Spatiotemporal and text representation learning **(STaTRL)** [30] explores multiple contexts to recommend POIs.

- *Graph Convolution Network (GCN)-based method*:
- **LightGCN** [11] simplifies the design of the GCN to make it more concise and appropriate for recommendations.
- **SEPT** [38] builds three graph encoders to capture self-supervisory signals for socially-aware recommendations.

- *Collaborative Filtering (CF)-based method*:
- Directly optimizing alignment and uniformity **(DirectAU)** [26] enhances representations in hypersphere for recommendations.

- *Graph and Contrastive Learning (G&CL)-based method*:
- **SGL** [32] exploits an auxiliary self-supervised task to reinforce node representation for recommendations.
- **NCL** [13] regards users (or items) and their structural neighbors as positive contrastive pairs for recommendations.
- **SIMGCL** [39] adds uniform noise to the embedding space to create contrastive views based on CF for recommendations.

Implementation and Parameter Settings.

The parameters of the EEDN were sampled as follows: β_1 and β_2 were set to 2 and $\frac{1}{2}$, respectively. d_m was 1,024 for Gowalla and Yelp2018, and 768 for Foursquare. γ and δ were 0.4 and 0.5 for Foursquare, 1.5 and 4 for Gowalla, and 1 and 4 for Yelp2018, respectively. The learning rate was 1e-2. The number of HGCN layers z_l was 1, and the number of heads z_h was set to 1, 3, and 2 for Gowalla, Foursquare, and Yelp2018, respectively. The number of convolution layers was 1. The kernel size was **3**×**3**. These hyperparameters were tuned using Optuna[1]. The parameters for the baselines were tuned to attain the best performance or set as proposed by the authors. For a fair evaluation, we conducted each experiment five times and obtained the average result. We implemented our EEDN and experimented with Pytorch on Nvidia GeForce RTX 3090 (24GB memory).

5.2 Experimental Results

Table 2 shows the outstanding competence of the EEDN in comparison to the state-of-the-art models of the three datasets. Overall, the

[1] https://github.com/pfnet/optuna

Table 2: Performance Comparison of Different Recommendation Models. Bold: Best, underline: Second best.

Model	Metric	MF-based		SA-based		GCN-based		CF-based	G&CL-based			Ours	Improv.
		CPIR	ENMF	SAE-NAD	STaTRL	LightGCN	SEPT	DirectAU	SGL	NCL	SIMGCL		
Gowalla	Recall@5	0.0490	0.0452	0.0470	0.0517	0.0508	0.0480	0.0457	0.0524	0.0535	0.0541	**0.0602**	+11.3%
	NDCG@5	0.0785	0.0787	0.0699	0.0840	0.0846	0.0804	0.0733	0.0882	0.0890	0.0885	**0.0996**	+11.9%
	Recall@10	0.0718	0.0686	0.0731	0.0803	0.0781	0.0752	0.0729	0.0824	0.0849	0.0835	**0.0947**	+11.5%
	NDCG@10	0.0745	0.0789	0.0596	0.0876	0.0875	0.0842	0.0781	0.0919	0.0938	0.0924	**0.1040**	+10.9%
	Recall@20	0.1024	0.1037	0.1120	0.1230	0.1196	0.1160	0.1132	0.1255	0.1304	0.1298	**0.1423**	+9.1%
	NDCG@20	0.0907	0.0905	0.0495	0.1018	0.1021	0.0987	0.0928	0.1068	0.1099	0.1090	**0.1203**	+9.5%
Foursquare	Recall@5	0.0380	0.0461	0.0372	0.0472	0.0480	0.0450	0.0482	0.0471	0.0511	0.0464	**0.0554**	+8.4%
	NDCG@5	0.0623	0.0748	0.0603	0.0763	0.0784	0.0722	0.0649	0.0753	0.0834	0.0725	**0.0867**	+4.0%
	Recall@10	0.0675	0.0703	0.0557	0.0735	0.0725	0.0705	0.0714	0.0728	0.0788	0.0732	**0.0843**	+7.0%
	NDCG@10	0.0688	0.0756	0.0496	0.0778	0.0795	0.0751	0.0696	0.0772	0.0854	0.0764	**0.0882**	+3.3%
	Recall@20	0.1024	0.1025	0.0798	0.1106	0.1094	0.1108	0.1091	0.1102	0.1206	0.1146	**0.1268**	+5.1%
	NDCG@20	0.0823	0.0875	0.0396	0.0916	0.0934	0.0920	0.0848	0.0914	0.1012	0.0923	**0.1043**	+3.1%
Yelp2018	Recall@5	0.0192	0.0191	0.0190	0.0215	0.0191	0.0190	0.0238	0.0220	0.0223	0.0242	**0.0267**	+10.3%
	NDCG@5	0.0368	0.0368	0.0341	0.0415	0.0371	0.0360	0.0458	0.0421	0.0428	0.0461	**0.0508**	+10.2%
	Recall@10	0.0342	0.0333	0.0323	0.0372	0.0341	0.0332	0.0413	0.0384	0.0390	0.0426	**0.0457**	+7.3%
	NDCG@10	0.0395	0.0389	0.0311	0.0420	0.0397	0.0385	0.0482	0.0447	0.0456	0.0493	**0.0531**	+7.7%
	Recall@20	0.0548	0.0562	0.0522	0.0662	0.0582	0.0574	0.0705	0.0659	0.0665	0.0721	**0.0760**	+5.4%
	NDCG@20	0.0458	0.0469	0.0277	0.0501	0.0484	0.0476	0.0589	0.0549	0.0558	0.0601	**0.0639**	+6.3%

EEDN consistently achieved a better performance than the other methods, specifically, the improvements compared with the runner-ups NCL and SIMGCL, 9.1% ∼ 11.5% in R@K and 9.5% ∼ 11.9% in N@K on Gowalla, 5.1% ∼ 8.4% in R@K and 3.1% ∼ 4.0% in N@K on Foursquare, and 5.4% ∼ 10.3% in R@K and 6.3% ∼ 10.2% in N@K on Yelp2018 for K=5, 10, and 20. The results demonstrate that the EEDN can exploit rich latent features by learning relations between users, POIs, and interactions between users and POIs. Table 2 also provides the following observations and insights:

- The results obtained by graph- and contrastive learning-based methods, i.e., SGL, NCL, and SIMGCL, are competitive among baselines. NCL performs better than SIMGCL on both Gowalla and Foursquare, while it is worse than SIMGCL on Yelp2018. Note that the Yelp2018 has shorter user sequences than the other two datasets. This indicates that NCL, which leverages nodes (users or items) and their neighbors as positive contrastive pairs, is beneficial for shorter user sequences. In contrast, SIMGCL, which adds uniform noises to the embedding space to create contrastive pairs, works well for longer user sequences.
- Recalling that SGL exploits three operators for upgrading the graph structure based on LightGCN, Table 2 shows that it is slightly inferior to NCL and SIMGCL. This demonstrates that the self-supervised upgrading of the graph is effective in enhancing the representations of users and POIs.
- The InfoNCE loss [21], which promotes more uniform user or POI representations, is widely adopted to mitigate the popularity bias [2], such as SEPT, SGL, NCL, and SIMGCL. In contrast, the EEDN concentrates on exposure bias and obtains better results. This shows that learning more specific representations for each user is more beneficial than learning uniform representations.
- SAE-NAD and STaTRL which are self-attention-based methods had worse results than other methods. The results indicate: (1) the correlations between POIs and users are more suitable for modeling with structured graphs; (2) as shown in Table 1, the average lengths of sequences are 67.1, 68.2, and 49.3 for the Gowalla, Foursquare, and Yelp2018, respectively. This shows that fewer long-term dependencies between visited POIs appeared in the datasets, while they focused on long-term dependencies.
- The results obtained by MF-based CPIR and ENMF showed inferior performance compared to the graph- and context-embedding-based methods. This indicates that encoders capturing complex non-linear features are effective for POI recommendation.

Table 3: Ablation study. "w/o X" denotes the removed parts. "GloIn", "ImFe", and "FeTra" indicate the global contexts, implicit features, and projection layers in HGCN, respectively.

Model	Gowalla		Foursquare		Yelp2018	
	R@20	N@20	R@20	N@20	R@20	N@20
w/o GloIn	0.1361	0.1134	0.1224	0.1008	0.0751	0.0633
w/o ImFe	0.1335	0.1126	0.1223	0.1004	0.0717	0.0597
w/o FeTra	**0.1423**	**0.1203**	0.1261	0.1038	0.0720	0.0600
Full	0.1413	0.1198	**0.1268**	**0.1043**	**0.0760**	**0.0639**

5.3 Ablation Study

We conducted an ablation study to examine the impact of each module on the EEDN. The results are shown in Table 3. The findings are as follows:

- The projection transformation in the HGCN layer helps to improve the performance on the Foursquare and Yelp2018, but does not work well for the Gowalla dataset.
- The model without implicit feature learning performed worse on all datasets. This demonstrates that the EEDN can effectively infer the relevant but non-existing check-in behavior of users by leveraging both structural and sequential patterns.

Table 4: Study of various implicit features. "Gra.", "Seq$_1$", and "Seq$_2$" indicate the graph-, light self-attention- and convolution-based implicit features, respectively.

Category			Gowalla		Foursquare		Yelp2018	
Gra.	Seq$_1$	Seq$_2$	R@20	N@20	R@20	N@20	R@20	N@20
			0.1330	0.1126	0.1215	0.0997	0.0717	0.0597
✓			0.1335	0.1130	0.1221	0.1003	0.0734	0.0614
✓	✓		0.1394	0.1183	0.1232	0.1011	0.0753	0.0633
✓		✓	0.1398	0.1182	0.1252	0.1028	0.0756	0.0636
✓	✓	✓	**0.1423**	**0.1203**	**0.1268**	**0.1043**	**0.0760**	**0.0639**
Improv.			**+6.6%**	**+6.8%**	**+4.4%**	**+4.6%**	**+5.8%**	**+7.0%**

- The global contexts from the POI-POI graph which boosts the interactions among users and POIs contribute to improving the performance of the three datasets. This also demonstrates that incorporating global information into user representation makes users' features more specific.

We further conducted an ablation study to justify the effectiveness of graph and sequential-based implicit features. Table 4 shows the comparison by utilizing three types of implicit features: graph-based, light self-attention, and convolution-based features. Overall, implicit features contribute to the performance, as the improvements were 4.4% ~ 6.6% in Recall@20 and 4.6% ~ 7.0% in NDCG@20 on the three datasets. Specifically, sequential-based implicit features yield more benefits than graph-based implicit features.

5.4 Cold-Start Comparison

Following Yang et al.'s work [35], we ranked the users by the lengths of their visit sequences to evaluate the cold-start recommendation performance. We selected the bottom 15% as inactive users: 2,810, 1,146, and 4,750 users in Gowalla, Foursquare, and Yelp2018, respectively. For simplification, we picked the best three baselines, DirectAU, NCL, and SIMGCL, to compare the EEDN, which are shown in Table 5. We see that the EEDN achieves 4.1% ~ 11.2%, 2.9% ~ 9.3%, and 1.9% ~ 11.3% improvements against the baselines on Gowalla, Foursquare, and Yelp2018, respectively. The results verify that the EEDN is also effective for the cold-start problem. The EEDN provides better performance in scenarios with a smaller K, as the average values of the improvement by R@5 and N@5 are 1.9 ~ 5.1 times those by R@20 and N@20 against all datasets.

5.5 Performance Study Against Encoders

We compared the performance of an HGCN against three encoders that have gained much success in many language processing and vision tasks [6, 14, 17, 25, 44]. These are Transformer [25], MLPs with gating (gMLP) [14], and LS-Transformer [44]. We found that it could achieve better results by adding a residual block to the Transformer and LS-Transformer, but not for gMLP and HGCN. We tuned the hyperparameters to obtain the best performance. The results are shown in Table 6, which prompt the following findings:

- Observing Table 2 and Table 6, the EEDN with any encoders is competitive with NCL and SIMGCL, indicating the potential of the EEDN for further explorations. In particular, the EEDN with any encoders exceeds the baselines of Gowalla and Yelp2018.

Table 5: Cold Start Performance Comparison.

	Model	R@5	N@5	R@10	N@10	R@20	N@20
Gowalla	DirectAU	0.0588	0.0617	0.0912	0.0781	0.1392	0.0969
	NCL	0.0693	0.0729	0.1080	0.0925	0.1641	0.1146
	SIMGCL	0.0698	0.0737	0.1069	0.0923	0.1629	0.1145
	Ours	**0.0763**	**0.0802**	**0.1150**	**0.0994**	**0.1688**	**0.1198**
Foursquare	DirectAU	0.0609	0.0571	0.0935	0.0720	0.1506	0.0922
	NCL	0.0754	0.0726	0.1134	0.0898	0.1714	0.1107
	SIMGCL	0.0708	0.0705	0.1134	0.0902	0.1750	0.1122
	Ours	**0.0829**	**0.0808**	**0.1239**	**0.0971**	**0.1784**	**0.1154**
Yelp2018	DirectAU	0.0276	0.0293	0.0473	0.0398	0.0797	0.0533
	NCL	0.0277	0.0284	0.0462	0.0385	0.0758	0.0509
	SIMGCL	0.0277	0.0297	0.0472	0.0403	0.0812	0.0543
	Ours	**0.0291**	**0.0320**	**0.0526**	**0.0434**	**0.0845**	**0.0565**

Table 6: Comparison of encoders. "LS-Tran." and "Tran." represent Long-Short Transformer and Transformer.

Model	Gowalla		Foursquare		Yelp2018	
	R@20	N@20	R@20	N@20	R@20	N@20
gMLP	0.1381	0.1169	0.1241	0.1021	0.0757	0.0634
LS-Tran.	0.1369	0.1165	0.1238	0.1008	0.0740	0.0622
Tran.	0.1389	0.1183	0.1247	0.1024	0.0744	0.0626
HGCN	**0.1423**	**0.1203**	**0.1268**	**0.1043**	**0.0760**	**0.0639**

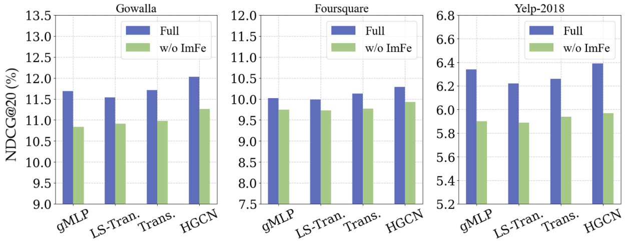
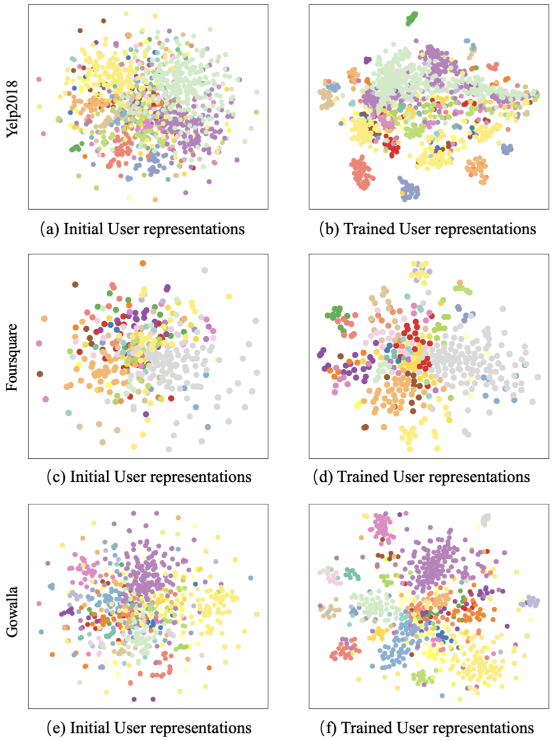


Figure 5: Impact of implicit features to encoders.

- The performance comparison leads to an ordered list of HGCN > Transformer ≈ gMLP > LS-Transformer. This is reasonable for LS-Transformer because it uses local window attention to gain benefits from time complexity but decreases precision.
- The Transformers did not perform the best. There are three possible reasons for this. First, the correlation between visited POIs may not be long-term dependencies, as the average length of the sequence was less than 70 in the three datasets. Second, the standard Transformer does not take global contexts into account although it is effective for POI recommendation. Third, MF-based structures that require a shallow and wide feature extractor, lead to limitations in the performance of the Transformer blocks.

To further explore the adaptability of prevalent encoders and implicit feature learning, we conducted experiments. The results are presented in Fig. 5. Throughout the experiments, we found that the implicit features significantly boosted the recommendation quality in all encoders.



(a) Initial User representations (b) Trained User representations

(c) Initial User representations (d) Trained User representations

(e) Initial User representations (f) Trained User representations

Figure 6: Visualizations of user representation.

5.6 Visualization of User Representation

To examine the effect of user representation learning, we visualize distributions of users on three datasets using t-SNE [23]. In Fig. 6, (a), (c), and (e) plot the distributions of initial users on the Yelp2018, Foursquare, and Gowalla, respectively, while (b), (d), and (f) illustrate the distributions of users after training. Each color denotes users who visited the same POI. Similar users are close to each other. Other findings are as follows:

- We can observe that the same colors are grouped together after training. This indicates that the EEDN is effective in identifying similar behaviors from local and global contexts.
- Overall, the user distributions on Yelp2018 are more ambiguous than those of the other datasets. One reason is that the average length of check-in trajectories of Yelp2018 is short which leads to a lack of knowledge obtained by graph-based and sequential contexts. Table 2 also shows the same observations in that the results on Yelp2018 are worse than those of the other datasets.

5.7 Parameter and Time Complexity Analysis

We examined how key hyperparameters, λ, δ, and the kernel size of the 2D convolution layers affected the performance of the EEDN.

Effect of λ and δ. The hyperparameters of λ and δ were assigned to measure the weight of the visited and future visited POIs. We can see from Fig. 7 that the best values of λ are 1.5 for Gowalla, 0.4 for Foursquare, and 1.0 for Yelp2018, and the best values of δ are 3.6 for Gowalla, 0.7 for Foursquare, and 4.0 for Yelp2018. We can observe that λ is more sensitive than δ, as it has a greater fluctuation range.

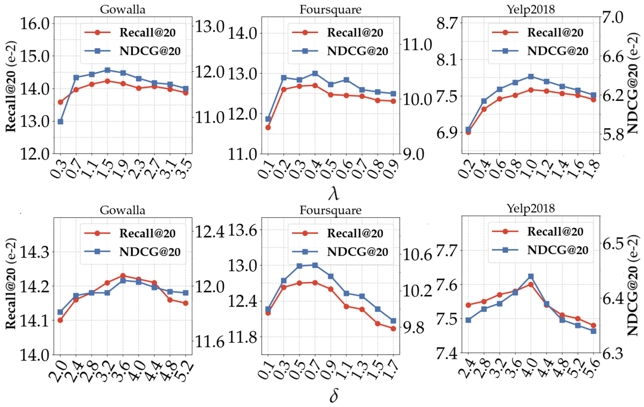


Figure 7: Influence of λ and δ

Table 7: Effect of the kernel size in convolution Layers.

Kernel Size	Gowalla		Foursquare		Yelp2018	
	R@20	N@20	R@20	N@20	R@20	N@20
3×3	0.1423	0.1203	**0.1270**	**0.1047**	**0.0760**	**0.0639**
5×5	0.1410	0.1198	0.1259	0.1040	0.0752	0.0626
7×7	0.1425	0.1206	0.1256	0.1040	0.0751	0.0626
9×9	**0.1429**	**0.1207**	0.1262	0.1038	0.0754	0.0628

Effect of kernel size. Table 7 illustrates the impact of different kernel sizes, with **3×3** beings the best choice in Foursquare and Yelp2018. As for Gowalla, the kernel size of **3×3** performs well as the increase of the size indicates few improvements and more computational costs.

The computational complexity of EEDN depends on HGCN ($O(L^2)$), LSA ($O(L^2)$), and Conv2D ($O(Ld_m)$). Thus, the computational complexity of EEND in a fixed setting is $O(L^2)$, indicating that considering the length of the user sequence is essential in a real deployment.

6 CONCLUSION

In this paper, we proposed an EEDN to alleviate implicit feedback and integrate both local and global interaction patterns for user representation learning. Besides explicitly matching the feature vectors of users and POIs in NMF, the EEDN mines implicit features from graph-based and sequential views to enhance its generalization. Throughout the experiments, we showed the effectiveness of the EEDN. We also verified the robustness of the EEDN by adapting prevalent encoders in recommendation tasks. Future work will include (i) incorporating other contexts for effective user and POI representations, and (ii) extending our framework to a more interpretable solution to address the implicit feedback problem.

ACKNOWLEDGEMENTS

We would like to thank anonymous reviewers for their thorough comments and suggestions. This work is supported by Suzuki Foundation, JKA, JSPS KAKENHI (No.23H03402), and China Scholarship Council (No.202208330093).

REFERENCES

[1] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient neural matrix factorization without sampling for recommendation. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–28.

[2] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[3] Jiawei Chen, Xiang Wang, Fuli Feng, and Xiangnan He. 2021. Bias Issues and Solutions in Recommender System: Tutorial on the RecSys 2021. In *Fifteenth ACM Conference on Recommender Systems*. 825–827.

[4] Qiang Chen, Qiman Wu, Jian Wang, Qinghao Hu, Tao Hu, Errui Ding, Jian Cheng, and Jingdong Wang. 2022. MixFormer: Mixing Features across Windows and Dimensions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5249–5259.

[5] Yudong Chen, Xin Wang, Miao Fan, Jizhou Huang, Shengwen Yang, and Wenwu Zhu. 2021. Curriculum meta-learning for next POI recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2692–2702.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*. PMLR, 1126–1135.

[8] Zhabiz Gharibshah, Xingquan Zhu, Arthur Hainline, and Michael Conway. 2020. Deep learning for user interest and response prediction in online display advertising. *Data Science and Engineering* 5, 1 (2020), 12–26.

[9] Peng Han, Zhongxiao Li, Yong Liu, Peilin Zhao, Jing Li, Hao Wang, and Shuo Shang. 2020. Contextualized point-of-interest recommendation. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*. 2484—-2490.

[10] Bowen Hao, Jing Zhang, Hongzhi Yin, Cuiping Li, and Hong Chen. 2021. Pre-training graph neural networks for cold-start users and items representation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 265–273.

[11] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[12] Chao Li, Zhongtian Bao, Linhao Li, and Ziping Zhao. 2020. Exploring temporal representations by leveraging attention-based bidirectional LSTM-RNNs for multi-modal emotion recognition. *Information Processing & Management* 57, 3 (2020), 102185.

[13] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. In *Proceedings of the ACM Web Conference 2022*. 2320–2329.

[14] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. 2021. Pay attention to mlps. *Advances in Neural Information Processing Systems* 34 (2021), 9204–9215.

[15] Huiting Liu, Huimin Liu, Qiang Ji, Peng Zhao, and Xindong Wu. 2020. Collaborative deep recommendation with global and local item correlations. *Neurocomputing* 385 (2020), 278–291.

[16] Wei Liu, Hanjiang Lai, Jing Wang, Geyang Ke, Weiwei Yang, and Jian Yin. 2020. Mix geographical information into local collaborative ranking for POI recommendation. *World Wide Web* 23, 1 (2020), 131–152.

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10012–10022.

[18] Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on heterogeneous information networks for cold-start recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1563–1573.

[19] Chen Ma, Yingxue Zhang, Qinglong Wang, and Xue Liu. 2018. Point-of-interest recommendation: Exploiting self-attentive autoencoders with neighbor-aware influence. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 697–706.

[20] Yongxin Ni, Xiancong Chen, Weike Pan, Zixiang Chen, and Zhong Ming. 2021. Factored heterogeneous similarity model for recommendation with implicit feedback. *Neurocomputing* 455 (2021), 59–67.

[21] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[22] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[23] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[24] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. A meta-learning perspective on cold-start recommendations for items. *Advances in neural information processing systems* 30 (2017).

[25] 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).

[26] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards Representation Alignment and Uniformity in Collaborative Filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1816–1825.

[27] Chunyang Wang, Yanmin Zhu, Haobing Liu, Tianzi Zang, Jiadi Yu, and Feilong Tang. 2022. Deep Meta-learning in Recommendation Systems: A Survey. *arXiv preprint arXiv:2206.04415* (2022).

[28] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-task feature learning for knowledge graph enhanced recommendation. In *The world wide web conference*. 2000–2010.

[29] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*. 373–381.

[30] Xinfeng Wang, Fumiyo Fukumoto, Jiyi Li, Dongjin Yu, and Xiaoxiao Sun. 2023. STaTRL: Spatial-temporal and text representation learning for POI recommendation. *Applied Intelligence* 53, 7 (2023), 8286–8301.

[31] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive learning for cold-start recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 5382–5390.

[32] 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.

[33] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *Comput. Surveys* 55, 5 (2022), 1–37.

[34] Yuan Wu and Jin Gou. 2021. Leveraging neighborhood session information with dual attentive neural network for session-based recommendation. *Neurocomputing* 439 (2021), 234–242.

[35] 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.

[36] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, Jiali Yang, and Xiaofang Zhou. 2019. Social influence-based group representation learning for group recommendation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 566–577.

[37] Hongzhi Yin, Weiqing Wang, Hao Wang, Ling Chen, and Xiaofang Zhou. 2017. Spatial-aware hierarchical collaborative deep learning for POI recommendation. *IEEE Transactions on Knowledge and Data Engineering* 29, 11 (2017), 2537–2551.

[38] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-aware self-supervised tri-training for recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2084–2092.

[39] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1294–1303.

[40] Wenhui Yu and Zheng Qin. 2019. Spectrum-enhanced pairwise learning to rank. In *The world wide web conference*. 2247–2257.

[41] Wenhui Yu and Zheng Qin. 2020. Graph convolutional network for recommendation with low-pass collaborative filters. In *International Conference on Machine Learning*. PMLR, 10936–10945.

[42] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. 2006. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems* 19 (2006).

[43] Fan Zhou, Ruiyang Yin, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Jin Wu. 2019. Adversarial point-of-interest recommendation. In *The world wide web conference*. 3462–34618.

[44] Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar, and Bryan Catanzaro. 2021. Long-short transformer: Efficient transformers for language and vision. *Advances in Neural Information Processing Systems* 34 (2021), 17723–17736.