# Improving Representation Learning for Session-based Recommendation

Tianwen Chen, Raymond Chi-Wing Wong
*Department of Computer Science and Engineering, The Hong Kong University of Science and Technology*
{*tchenaj,raywong*}*@cse.ust.hk*

***Abstract*—Session-based recommendation aims to predict the next item in an anonymous session. Recent advances have shown the importance of exploiting inter-session dependencies, such as item-item transitions and session-session similarities. However, the existing methods either ignore the relative order of item co-occurrences or assign the same importance to co-occurrence patterns at all distances. Besides, they are prone to extracting wrong signals to learn user preferences from dependencies between sessions. To solve these problems, we propose a model called FOCOL to better exploit the inter-session dependencies by considering Fine-grained item co-Occurrences and applying the COntrastive Learning framework. Specifically, to capture inter-session item-item dependencies, we propose a component called FOGCN (Fine-grained co-Occurrence Graph Convolution Network) to automatically learn the importance of item co-occurrence patterns from a global graph that encodes the detailed information about item co-occurrences such as relative order and distance. To directly capture dependencies between sessions, we view the recommendation task as a clustering problem, and propose a component called CSRL (Contrastive Session Representation Learning) to implicitly group similar sessions (i.e., sessions with the same next item) into the same cluster and push apart sessions at different clusters. Extensive experiments conducted on three public datasets show that the proposed model is superior to the state-of-the-art methods and the proposed two components can learn more informative item and session representations by considering the fine-grained item co-occurrences and directly capturing dependencies between sessions.**

## 1. Introduction

With the explosive growth of information, recommender systems (RSs) become a critical tool to alleviate the information overload problem in many online services such as e-commerce and media sharing websites. Conventional recommendation methods such as collaborative filtering [1, 2] rely on tracking user identities to model each individual user's preferences, which may make them perform poorly in scenarios where user identities cannot be tracked, due to some reasons including anonymous users or privacy issues [3, 4, 5]. Session-based recommendation (SBR) addresses this problem by assuming that users perform actions on a session basis, where a session is a sequence of actions in close temporal proximity. Under this assumption, users' actions in the same session are highly correlated [3, 6, 7], and thus, the sequential and co-occurrence patterns in the active session can be utilized to more accurately model the current user's preferences. Since SBR does not require user information and the "session-based" assumption is a common phenomenon, it is of great practical value and has received much attention from researchers recently.

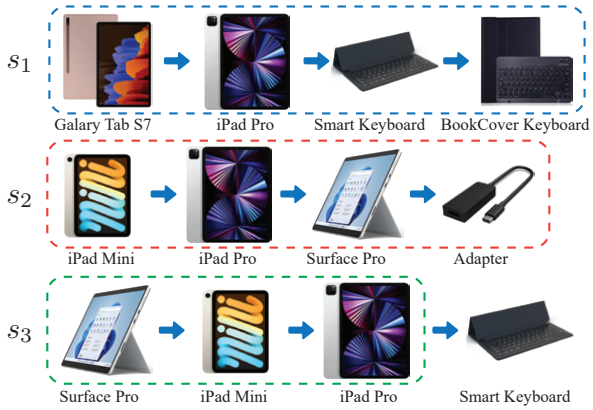


Figure 1. An example of inter-session relationships

The task of SBR is to predict the next action given the historical actions in an anonymous session. Early studies in SBR [3, 7, 8, 9, 10] extract user preferences only from the contextual information *within* the given session. Since sessions are usually very short, it is hard to correctly infer user preferences from the limited number of user actions [9, 10, 11]. To mitigate this *data insufficiency* issue, several recent studies [5, 12, 13, 14] have attempted to utilize the inter-session relationships to more accurately model user preferences. There are two levels of inter-session relationships which are useful to predict users' future intent, namely the item level and the session level.

The *item-level inter-session relationships* refer to the dependencies between item co-occurrences in different sessions. An example is shown in Figure 1. Suppose the current session $s_3$ consists of a sequence of tablets clicked by an anonymous user and the the ground-truth next item we aim to predict is a Smart Keyboard. If only the context information in $s_3$ is used to predict the next item, the RS may think the user wants to see more tablets. However, the user may have already decided to buy iPad Pro and s/he wants to buy a keyboard case next. Thus, the RS fails to correctly infer the user's intent. If the RS is aware of the co-occurrence pattern in another session $s_1$, where a

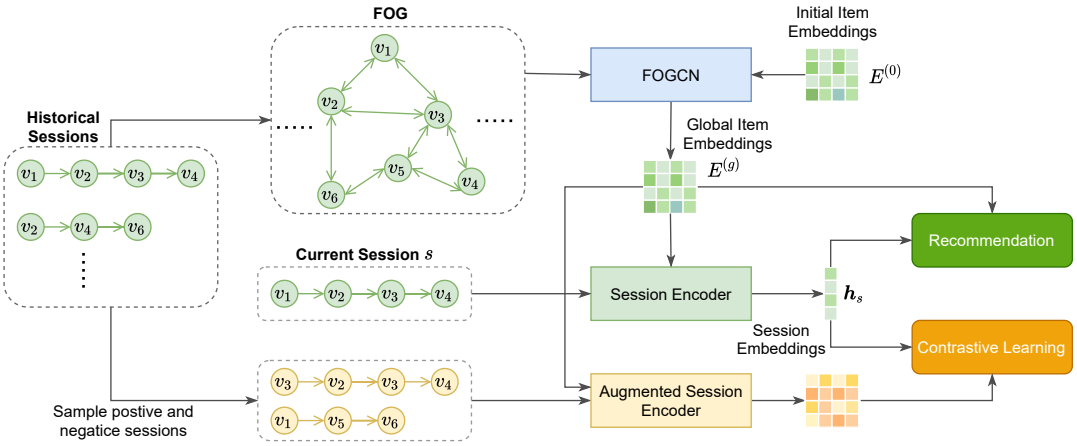


Figure 2. The model architecture of FOCOL

user first clicks the iPad Pro and then clicks the Smart Keyboard, the RS will be able to add Smart Keyboard into the recommendation list. Therefore, it is crucial to utilize the item-level inter-session relationships to correctly capture user preferences. Although methods that predict the next item by considering only the *within-session* information may learn to capture such simple first-order co-occurrences implicitly, they cannot handle more complex higher-order inter-session co-occurrences that involve multiple items and sessions. Recent studies explicitly encode the item-level inter-session relationships by a global item graph. Each edge has a weight to denote the correlation strength between two items, which is empirically defined to be the number of *direction transitions* [14], *co-occurrences within a fixed-sized window* [5], or all *co-occurrences* [13] in the training set. By propagating the item embeddings on the global graph with a multi-layered graph neural network (GNN), the final item representations can effectively capture the complex high-order inter-session item dependencies which can provide rich information for the subsequent session representation learning. However, these methods either do not consider *all* co-occurrences [5, 14] or are unaware of the *relative order* and *distance* of items [5, 13].

The *session-level inter-session relationships* refer to the connections between sessions with similar intents. There are two representative methods along this line of research. CSRM [15] learns a neighbor view of the current session which is a weighted sum of the representations of the top-$k$ most similar sessions in a first-in-first-out memory bank. However, even when there is no similar sessions in the memory bank, a neighbor view is still extracted and contributes to the prediction, which could provide incorrect signals to infer the real user preferences. DHCN [13] captures inter-session relationships by a line graph that contains sessions as nodes, where the edge weight between two sessions is defined to be their Jaccard similarity [16] (i.e., the ratio of intersection over union). The session embeddings (initialized as average item embeddings) are refined by a graph convolution network (GCN) to make adjacent sessions have similar representations. The problem of this approach is that sessions with a large Jaccard similarity does not necessarily reflect similar preferences. For example, in Figure 1, $s_2$ has a larger Jaccard similarity to $s_3$ than $s_1$, but $s_1$ provides a correct hint to the next item of $s_3$ (a keyboard case) while $s_2$ provides an incorrect hint (an adapter).

To tackle the above problems of existing methods in capturing inter-session relationships, we propose a model called FOCOL which considers **F**ine-grained co-**O**ccurrences and applies **CO**ntrastive **L**earning to directly capture inter-session relationships. The model architecture of FOCOL is shown in Figure 2. First, we build a global item graph called FOG (**F**ine-grained co-**O**ccurrence **G**raph) from all historical sessions that encodes the detailed information about item co-occurrence such as the frequency, relative order and distance as weight vectors on directed edges. Then, we propagate the initial item embeddings $\boldsymbol{E}^{(0)}$ on FOG with a graph convolution network called FOGCN which can automatically learn the co-occurrence patterns with different relative order and distance. The final item representations $\boldsymbol{E}^{(g)}$ of FOGCN capture the item-level inter-session relationships. To capture session-level inter-session relationships, given the current session $s$, we sample one positive session that has the same semantics as $s$ and $K_{ns}$ negative sessions that have different semantics. We consider two sessions to have the same semantics if they have the same next item. The current session is encoded by the *session encoder* to obtain the session embedding $\boldsymbol{h}_s$ while the sampled sessions are encoded by the *augmented session encoder* which applies augmentation operations that do not change the semantics before encoding the sampled sessions. The session embeddings are contrasted by the *contrastive session representation learning* (CSRL) component to implicitly group similar sessions and push away different sessions, which can effectively improve session representation learning. Finally, the *recommendation module* generates a probability distribution of the next item from the global item embeddings and the current session embedding.

The main contributions of this study are summarized as

follows. (1) To capture item-level inter-session relationships, we propose FOG to encode the item co-occurrences at a fine-grained level and apply FOGCN to automatically learn the importance of ordered co-occurrences at different distances. (2) To directly capture session-level inter-session relationships, we view the recommendation task as a clustering problem and apply the contrastive learning framework to implicitly groups similar sessions into the same cluster and push away sessions at different cluster, which can effectively improve session representation learning. (3) Extensive experiments are conducted on three public benchmark datasets, which show that the proposed model FOCOL can achieve state-of-the-art performance and is effective in learning more informative item and session representations.

## 2. Related Work

In this section, we review the related work on session-based recommendation (Section 2.1) and constrastive learning (Section 2.2).

### 2.1. Session-based Recommendation

Collaborative filtering (CF) has been widely applied in recommender systems to model users' general preferences by their historical interactions with items. CF methods such as [1, 2] that are based on factorizing the user-item interaction matrix can be adapted for SBR by considering each session as a user. The test sessions are treated as cold-start users and the average of item embeddings in each test session is used as the session representation. The problem of these approaches is that there is a discrepancy between training and testing, and the ordering information in sessions are neglected. Item-based CF has also been applied to SBR by computing an item-item similarity matrix from item co-occurrences in all historical sessions [17]. Similar to matrix factorization-based CF methods, item-based CF methods fail to capture the sequential order of items in sessions.

To model the sequential signals in sessions, Markov chain-based methods can be applied. Each session is modeled as a Markov chain and the user's next action can be inferred from the last state. For example, Rendle et al. [18] proposed FPMC which combines matrix factorization and Markov chain to capture both general user preferences and sequential patterns. Since the number of states grows exponentially when more previous items are considered, most Markov chain-based methods use only the first-order transition matrix, which makes them unable to capture complex high-order sequential patterns.

Recurrent neural networks (RNNs) are inherently designed for modeling sequential data and many effective RNN-based methods have been proposed for SBR. Hidasi et al. [3] proposed the first RNN-based method called GRU4Rec for SBR, which stacks multiple layers of Gated Recurrent Unit (GRU) to model the complete item interaction sequence. Li et al. [8] proposed NARM that incorporates the attention mechanism into GRU to model the user's sequential behavior and main purpose in the active session. ISLF [19] captures the user's main intention with a recurrent variational autoencoder and estimates the user's interests shift by combining both the sequential behavior characteristics and item frequencies in the current session.

Convolutional neural networks (CNNs) are also explored for SBR due to their abilities to extract sequential features. Tuan and Phuong [20] proposed a 3D CNN model for SBR that allows combining different content features such as item descriptions and item categories with character-level encoding for all features. Yuan et al. [6] proposed NextItNet which uses dilated convolution to effectively model long-range dependencies in sessions.

Both RNNs and CNNs are powerful sequential modeling tools, but they mainly focus on modeling consecutive sequential patterns, making them less powerful in capturing non-adjacent item transitions. Recently, graph neural networks (GNNs) have attracted much attention in the field of SBR due to their capabilities to model complex item transition patterns in sessions. Wu et al. [10] proposed SR-GNN which converts a session into an unweighted directed graph and employs a gated GNN (GGNN) to propagate item embeddings along both directions of the edges in the session graph. Following this work, Xu et al. [21] proposed GC-SAN which improves the ability of SR-GNN in capturing long-range dependencies by the self-attention mechanism. Chen and Wong [7] proposed two GNN layers to solve the information loss problems in GNN-based models for SBR. Pan et al. [22] proposed SGNN-HN which contains a star GNN to capture relations between items without direct connections and a highway network with a gating mechanism to alleviate the oversmoothing problem in GNNs.

Recently, it is shown that the global cross-session transition dynamics among items can improve item representation learning and lead to better session recommendation performance. Wang et al. [5] proposed GCE-GNN which constructs a global item graph by connecting items that co-occur within a fixed-sized window. Xia et al. [13] proposed DHCN which performs graph convolution on a hypergraph to obtain global item representations. The hypergraph is equivalent to a global item graph in which items are connected if they are in the same session. Huang et al. [14] proposed MTD which has a directed item graph constructed from the direct transitions among items. However, these methods are unable to distinguish the different importance levels of item co-occurrences with different distances.

### 2.2. Contrastive Learning

Contrastive learning has been widely applied to various deep learning fields as an effective framework for self-supervised learning which can help with a variety of downstream tasks [23, 24, 25, 26, 27, 28, 29]. For the computer vision tasks, the early methods such as CPC [23] and DIM [24] were proposed to contrast the representations of the different scales of the same image as the positive pairs. Some follow-up works such as MoCo [25] and SimCLR [26] considered the augmentations of the same image as positive pairs for contrastive learning. For the natural language

processing problems, Yan et al. [28] proposed ConSERT which applies contrastive learning to solve the collapse issue of BERT-derived sentence representations. Gao et al. [29] proposed SimCSE which uses only the standard dropout augmentation to produce positive pairs.

Contrastive learning has also shown great potential in a variety of recommendation tasks. For the collaborative filtering tasks, SGL [30] generates different views of nodes in the user-item graph with node dropout, edge dropout, and random walk. Yao et al. [31] proposed a data augmentation method that exploits feature correlations and are applicable to heterogeneous categorical features. For the sequential recommendation problems, Zhou et al. [32] proposed to contrast the prediction and ground-truth of attribute-level, item-level, and segment-level representations in the pretraining stage. Xie et al. [33] proposed CL4SRec which is equipped with three data augmentations at the discrete item level and samples two augmentations to construct positive pairs. Although these contrastive learning methods have achieved decent improvements in the target recommendation tasks, they are not suitable for session-based recommendation because their data augmentation strategies either cannot handle sequential data or would lead to unreliable self-supervision signals. DHCN [13] is the most relevant work to ours which constructs a positive pair of session representations by performing graph convolution on an item graph and a session graph. However, the session graph uses Jaccard similarity to measure the similarity between sessions, which can enforce two sessions with different semantics to have similar representations.

## 3. Methodology

In this section, first, we give a formal definition of session-based recommendation in Section 3.1. Then, we introduce our method to capture the item-level and session-level inter-session relationships in Section 3.2 and Section 3.3, respectively.

### 3.1. Problem Definition

Let $\mathcal{I}$ be the set of all unique items in the dataset $\mathcal{D}$ and let $N$ be the number of items. A session $s = [i_{s,1}, i_{s,2}, \cdots, i_{s,|s|}]$ is a sequence of items interacted by an anonymous user and sorted in chronological order, where $i_{s,t} \in \mathcal{I}$ is the $t^{th}$ interacted item in session $s$ and $|s|$ denotes the session length. The objective of a session-based recommender system is to predict the next item $i_{s,|s|+1}$ of session $s$ by generating a probability distribution $\hat{\boldsymbol{y}}^s \in \mathbb{R}^N$ over $\mathcal{I}$, where $\hat{y}^s_i$ is the predicted probability that item $i$ is the next item of $s$. The top-$K$ items with the largest probabilities are recommended as the candidate items.

### 3.2. Capturing Item-level Inter-session Relationships

In this subsection, we introduce how to capture item-level inter-session relationships by FOG and FOGCN.

**3.2.1. Constructing FOG.** First, we build the FOG from all historical sessions. The FOG, denoted by $G = (\mathcal{V}, \mathcal{E})$, is a directed graph containing all items as nodes, i.e., $\mathcal{V} = \mathcal{I}$. There are two edges with opposite directions between item $i$ and item $j$ if they appear in the same session. To encode the fine-grained co-occurrences between items, each edge $(i, j)$ is associated with a weight vector $\boldsymbol{w}_{ij} \in \mathbb{R}^{2L}$, where $L$ is the maximum distance between any two items in any session. The entry of $\boldsymbol{w}_{ij}$ is defined as follows:

$$\boldsymbol{w}_{ij}[k] = \sum_{s \in \mathcal{D}} \sum_{1 \le t, t+k \le |s|} \mathbb{1}_{\{i_{s,t} = i \wedge i_{s,t+k} = j\}} \text{ ,for } 1 \le |k| \le |L| \tag{1}$$

where $\mathbb{1}_A$ is the indicator function of event $A$ which evaluates to 1 if $A$ is true and 0 otherwise. The index of $\boldsymbol{w}_{ij}$ starts from 1 and a negative index $k$ is the Python-style notation that denotes the $(-k)^{th}$ last entry, i.e., $\boldsymbol{w}_{ij}[k] = \boldsymbol{w}_{ij}[2L + k + 1]$, for $-L \le k \le -1$.

Note that $\boldsymbol{w}_{ij}$ encodes the ordered co-occurrences between item $i$ and item $j$ at all distances. Specifically, the first $L$ and the last $L$ entries, count the numbers of forward and backward transitions from item $i$ to item $j$, respectively (order information). The entry $\boldsymbol{w}_{ij}[k]$ denotes the number of times that item $i$ and item $j$ has a relative distance of $k$ (distance information).

The edges are directed because $\boldsymbol{w}_{ij} \neq \boldsymbol{w}_{ji}$. However, the weight vectors of edges with opposite directions are symmetric, i.e., $\boldsymbol{w}_{ij} = \text{reverse}(\boldsymbol{w}_{ji})$.

**3.2.2. Computing Global Item Embeddings with FOGCN.** After constructing the global graph FOG, we propagate the initial item embeddings $\boldsymbol{E}^{(0)} \in \mathbb{R}^{N \times d}$ on FOG with a graph convolution network called FOGCN to capture complex high-order inter-session co-occurrences.

Let $\boldsymbol{E}^{(l)} \in \mathbb{R}^{N \times d}$ be the item embedding matrix at layer $l$, where $d$ is the dimensionality of item embeddings. The refined item embedding of item $i$ at layer $l$, denoted by $\boldsymbol{h}_i^{(l)} = \boldsymbol{E}_i^{(l)} \in \mathbb{R}^d$, is computed as the weighted sum of the neighboring item embeddings in the previous layer:

$$\boldsymbol{h}_i^{(l)} = \sum_{j \in N(i)} \boldsymbol{\alpha}_{ij}^{(l)} \odot \boldsymbol{h}_j^{(l-1)} \tag{2}$$

where $N(i)$ is the set of item $i$'s neighbors, $\odot$ denotes element-wise multiplication. $\boldsymbol{\alpha}_{ij}^{(l)} \in \mathbb{R}^d$ is the importance weight vector of the neighboring item embedding $\boldsymbol{h}_j^{(l-1)}$, where $\alpha_{ijk}^{(l)}$ is the importance weight of the $k^{th}$ feature of $\boldsymbol{h}_j^{(l-1)}$.

The importance weight vector $\boldsymbol{\alpha}_{ij}^{(l)}$ is computed from the edge weight vector $\boldsymbol{w}_{ij}$ as follows:

$$\boldsymbol{\alpha}_{ij}^{(l)} = \frac{\exp\left(\boldsymbol{P}^{(l)} \boldsymbol{w}_{ij}\right)}{\sum_{k \in N(i)} \exp\left(\boldsymbol{P}^{(l)} \boldsymbol{w}_{ik}\right)} \tag{3}$$

where $\boldsymbol{P}^{(l)} \in \mathbb{R}^{d \times 2L}$ is a matrix of learnable parameters. The entry $\boldsymbol{P}_{mn}^{(l)}$ can be interpreted as the contribution of

the co-occurrences at a relative distance of $n$ towards the $m$-th feature in the item embeddings. Therefore, each layer of FOGCN can automatically learn the importance of item co-occurrences at different relative distances.

After propagating the item embeddings for $L_G$ steps, we compute the final global item embeddings as the average of item embeddings at all layers to capture both the low-order and high-order item-level inter-session relationships:

$$\boldsymbol{E}^g = \frac{1}{T+1}\sum_{l=0}^{L_G}\boldsymbol{E}^{(l)} \tag{4}$$

For each item $i$, its global embedding $\boldsymbol{E}_i^g$ can capture the important co-occurrences involving $i$ and up to $L_G$ other items across up to $L_G$ sessions.

### 3.3. Capturing Session-level Inter-session Relationships

In this subsection, we first analyze the procedure of generating recommendations and show that the recommendation task can be formulated as a clustering problem (Section 3.3.1). Then, we propose to facilitate the recommendation task by applying the contrastive learning framework, which can improve session representation learning by directly capturing session-level inter-session relationships.

**3.3.1. Recommendation as a Clustering Problem.** We break the recommendation procedure into two steps, namely *session embedding generation* and *probability generation*.

**Session Embedding Generation**: Given the current session $s = \{i_{s,1}, i_{s,2}, \cdots, i_{s,t}\}$ whose ground-truth next item is $i_{s,t+1}$, where $t$ is the length of $s$, the session embedding generation step generates a fixed-sized vector representation of $s$, denoted by $\boldsymbol{h}_s \in \mathbb{R}^d$.

In this paper, to obtain $\boldsymbol{h}_s$, first, we map each item to its global item embedding to obtain a session embedding matrix $\boldsymbol{E}_s \in \mathbb{R}^{t\times d}$, where $\boldsymbol{E}_{s,t} = \boldsymbol{E}^g_{i_{s,t}}$. This is different from the common approach adopted by the existing studies [5, 8, 10, 13] that maps items to their initial item embeddings $\boldsymbol{E}^{(0)}$ because we found that the global item embeddings contain richer information that are useful for the subsequent sequence modeling.

Then, we refine the global item embeddings by a Transformer $Tr$ [34] to model the sequential dependencies within the session. Since Transformers need additional position embeddings to encode the positional information of the input embeddings, following the previous work [5, 13], we add a learnable reverse position embedding to each item embedding to indicate its position relative to the last item:

$$\boldsymbol{X}^{(0)}_{s,j} = \boldsymbol{E}_{s,j} + \boldsymbol{P}_{t+1-j}, 1 \le j \le t \tag{5}$$

where $\boldsymbol{E}_{s,j} \in \mathbb{R}^d$ is the item embedding at position $j$ and $\boldsymbol{P}_{t+1-j} \in \mathbb{R}^d$ is the corresponding reverse position embedding with reverse position $t+1-j$. For example, the first item embedding $\boldsymbol{E}_{s,1}$ has a reverse position embedding $\boldsymbol{P}_t$. $\boldsymbol{X}^{(0)}_s \in \mathbb{R}^{t\times d}$ is the initial input embedding matrix to the Transformer.

Let $L_T$ be the number of layers in the Transformer encoder, the last item embedding in the last layer is used as the session embedding:

$$\boldsymbol{X}^{(L_T)}_s = Tr\left(\boldsymbol{X}^{(0)}_s\right) \tag{6}$$

$$\boldsymbol{h}_s = \boldsymbol{X}^{(L_T)}_{s,t} \tag{7}$$

**Probability Generation**: The objective of this step is to generate a probability distribution of the next item, denoted by $\hat{\boldsymbol{y}}^s \in \mathbb{R}^N$, over the entire item set. In this paper, we first compute the score of item $i$ being the next item of $s$ as follows:

$$z^s_i = \boldsymbol{h}^T_s \boldsymbol{E}^g_i \tag{8}$$

Then, the predicted probability distribution is computed by a softmax operation:

$$\hat{\boldsymbol{y}}^s = \text{softmax}(\boldsymbol{z}^s) \tag{9}$$

**Analysis**: Suppose we create a cluster in a $d$-dimensional space $\mathcal{M}_d$ for each item and the center of the cluster corresponding to item $i$ is its embedding $\boldsymbol{E}^g_i$, then the recommendation task essentially tries to embed each session into a point (i.e., session embedding) in $\mathcal{M}_d$ which is most probably assigned to the cluster corresponding to the correct next item. Specifically, Equation (8) computes the distance between $\boldsymbol{h}_s$ and the cluster center of item $i$, and Equation (9) normalizes the distances to obtain the probability of $s$ belonging to each cluster.

Let $\boldsymbol{y}^s \in \mathbb{R}^N$ be an one-hot vector denoting ground-truth probability distribution, where $\boldsymbol{y}^s_{i_{s,t+1}} = 1$, the learning objective of the recommendation task is to minimize the expected cross entropy between $\boldsymbol{y}^s$ and $\hat{\boldsymbol{y}}^s$:

$$\mathcal{L}_{rec} = \mathop{\mathbb{E}}_{s\sim\mathcal{D}}\left[-\sum_{i=1}^{N} y^s_i \log \hat{y}^s_i\right] = -\mathop{\mathbb{E}}_{s\sim\mathcal{D}}\left[\log \hat{y}^s_{i_{s,t+1}}\right] \tag{10}$$

Thus, the recommendation task aims to maximize $\hat{y}^s_{i_{s,t+1}}$, which depends on the distances between the session embedding $\boldsymbol{h}_s$ and all the cluster centers, meaning that this learning objective only considers *relationships between sessions and items*. To improve the clustering performance, another effective approach is to directly capture the relationships between sessions by minimizing the distances between session embeddings in the same cluster and maximizing the distances between session embeddings in different clusters, which can be achieved by our contrastive session representation learning (CSRL) component described in Section 3.3.2.

**3.3.2. Contrastive Session Representation Learning.** To directly capture the relationships between sessions with contrastive learning, we consider the session embeddings in the same cluster as different views of the same object (i.e., the cluster). Specifically, given a session $s$, we sample one positive session $s^+$ from the same cluster and $K_{ns}$ negative sessions $\mathcal{N}_s$ from other clusters. Then, we follow InfoNCE [35] to maximize a lower bound of the the mutual

information between the session embeddings in the same cluster:

$$\mathcal{L}_{cl}(s) = -\log \frac{\exp\left(\boldsymbol{h}_s^T \boldsymbol{h}_{s^+}\right)}{\exp\left(\boldsymbol{h}_s^T \boldsymbol{h}_{s^+}\right) + \sum_{s^- \in \mathcal{N}_s} \exp\left(\boldsymbol{h}_s^T \boldsymbol{h}_{s^-}\right)} \tag{11}$$

Due to the popularity bias problem [36], most of clusters only contains a few sessions (i.e., the unpopular items are the next items of only a few sessions), there are not sufficient positive pairs of session embeddings to effectively capture the relationships between sessions in these clusters. Therefore, to address this problem and also to improve the generalization ability of the model, we apply data augmentation on the sampled sessions to generate more positive and negative pairs.

Specifically, let $\boldsymbol{h}_s^A$ be the embedding of session $s$ after applying augmentation operation $A$, we optimize the following contrastive learning objective:

$$\mathcal{L}_{cl}^A(s) = -\log \frac{\exp\left(\boldsymbol{h}_s^T \boldsymbol{h}_{s^+}^A\right)}{\exp\left(\boldsymbol{h}_s^T \boldsymbol{h}_{s^+}^A\right) + \sum_{s^- \in \mathcal{N}_s} \exp\left(\boldsymbol{h}_s^T \boldsymbol{h}_{s^-}^A\right)} \tag{12}$$

where the embeddings of the sampled sessions are replaced by the embeddings of the augmented sessions.

Note that the augmentation operation $A$ should not change the semantics of the session representation, i.e., after applying the data augmentation operation $A$, the embedding of the positive session $\boldsymbol{h}_{s+}^A$ should still be assigned to the same cluster as the current session $s$. Otherwise, $\boldsymbol{h}_s$ and $\boldsymbol{h}_{s^+}^A$ do not form a positive pair and optimizing Equation (12) will produce wrong learning signals.

Existing contrastive learning methods [28, 32] that operate on the discrete data levels (e.g., item masking, sequence cropping, and reordering) do not satisfy this requirement. For item masking, since sessions are usually very short (less than 6 items), there may not be enough information to correctly infer the next item. For sequence cropping and reordering, the augmented session can become a totally different session that have a different next item.

In this paper, given a session $s$, we propose to augment the session at the item embedding level (i.e., the input embeddings $\boldsymbol{X}_s^{(0)}$ to the Transformer) with the following three augmentation operations:

1) *Corrupt*: corrupt $\boldsymbol{X}_s^{(0)}$ by a slight noise sampling from a normal distribution $\mathcal{N}(0, \sigma^2)$.
2) *Dropout*: perform dropout on $\boldsymbol{X}_s^{(0)}$ with drop ratio $r$.
3) *Insert*: insert a learnable mask embedding $\boldsymbol{m}$ into $\boldsymbol{X}^{(0)}$. Suppose we insert $\boldsymbol{m}$ at position $j$, where $1 \leq j \leq t+1$, then the augmented input embedding will be $\{\boldsymbol{X}_{s,1}^{(0)}, \cdots, \boldsymbol{X}_{s,j-1}^{(0)}, \boldsymbol{m}, \boldsymbol{X}_{s,j}^{(0)}, \cdots, \boldsymbol{X}_{s,t}^{(0)}\}$

Let $\boldsymbol{X}_{A,s}^{(0)}$ be the input embeddings altered by augmentation operation $A$. It replaces $\boldsymbol{X}_s^{(0)}$ as the input to the Transformer encoder to obtained the augmented session embedding $\boldsymbol{h}_s^A$.

For *Corrupt* and *Dropout*, they will not change the semantics of the resulted session embedding because we can set $\sigma$ and $r$ to small values (in our experiments, $\sigma = r = 0.1$). For *Insert*, it is easy for the session encoder $Tr$ to learn to ignore $\boldsymbol{m}$. Therefore, the augmented session embedding $\boldsymbol{h}_s^A$ retain the same semantics as the original session embedding $\boldsymbol{h}_s$, which can provide more session pairs for the CSRL component to effectively capture the relationships between sessions.

To allow Equation (12) to capture the relationships between the original sessions, we also create an *Identity* augmentation operation that does nothing on $\boldsymbol{X}_s^{(0)}$. Let $\mathcal{A}$ be the set of all four augmentation operations, the overall learning objective of our model is:

$$\mathcal{L} = \mathcal{L}_{rec} + \beta \cdot \mathcal{L}_{cl} = \mathcal{L}_{rec} + \beta \cdot \mathop{\mathbb{E}}_{s \sim \mathcal{D}, A \sim \mathcal{A}} \left[\mathcal{L}_{cl}^A(s)\right] \tag{13}$$

where $\beta$ is a hyper-parameter controlling the magnitude of the contrastive learning task.

# 4. Experiments

In this section, we first describe the experimental settings and then analyze experimental results.

## 4.1. Datasets

We conducted our experiments on three real-world benchmark datasets: *Tmall*[1], *RetailRocket*[2], and *Diginetica*[3], which are commonly used in the literature of SBR [5, 8, 10, 13, 21]. For fair comparison, we follow the same dataset preprocessing steps in the previous studies [5, 10, 13]. Specifically, we filtered out sessions with length 1 and items with frequency less than 5. Then, the latest data (e.g., the data of the last week in *RetailRocket*) was extracted as the test set and all the previous data was used as the training set. Finally, we applied a data augmentation technique to generate multiple labelled sequences from each session. For example, we would generate a list of sequence-label pairs $([s_1], s_2), ([s_1, s_2], s_3), \cdots, ([s_1, s_2, \cdots, s_{|s|-1}], s_{|s|})$ from session $s = [s_1, s_2, \cdots, s_{|s|}]$. Some statistics of the datasets after preprocessing are shown in Table 1.

TABLE 1. STATISTICS OF DATASETS USED IN THE EXPERIMENTS

| Dataset | #Training Sessions | #Test sessions | #Items | Average Length |
|---|---|---|---|---|
| Tmall | 351,268 | 25,898 | 40,728 | 6.69 |
| RetailRocket | 433,643 | 15,132 | 36,968 | 5.43 |
| Diginetica | 719,470 | 60,858 | 43,097 | 5.12 |

## 4.2. Baseline Methods and Evaluation Metrics

To show the advantages of our method, we used the following representative baselines for SBR in our experiments: (1) **ItemKNN** [17] is an item-based CF methods which recommends items that are the most similar to the

1. https://tianchi.aliyun.com/dataset/dataDetail?dataId=42
2. https://www.kaggle.com/retailrocket/ecommerce-dataset
3. https://competitions.codalab.org/competitions/11161

TABLE 2. PERFORMANCE COMPARISONS ON THREE DATASETS

| Model | Tmall | | | | RetailRocket | | | | Diginetica | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | HR@10 | MRR@10 | HR@20 | MRR@20 | HR@10 | MRR@10 | HR@20 | MRR@20 | HR@10 | MRR@10 | HR@20 | MRR@20 |
| ItemKNN | 7.34 | 3.41 | 9.47 | 3.56 | 23.83 | 11.20 | 29.27 | 11.58 | 25.07 | 10.77 | 35.75 | 11.57 |
| FPMC | 13.10 | 7.12 | 16.06 | 7.32 | 25.99 | 13.38 | 32.37 | 13.82 | 15.43 | 6.20 | 22.14 | 6.66 |
| GRU4Rec | 9.47 | 5.78 | 10.93 | 5.89 | 38.35 | 23.27 | 44.01 | 23.67 | 17.93 | 7.73 | 30.79 | 8.22 |
| NARM | 19.17 | 10.42 | 23.30 | 10.70 | 42.07 | 24.88 | 50.22 | 24.59 | 35.44 | 15.13 | 48.32 | 16.00 |
| STAMP | 22.63 | 13.12 | 26.47 | 13.36 | 42.95 | 24.61 | 50.96 | 25.17 | 33.98 | 14.26 | 46.64 | 15.13 |
| SR-GNN | 23.41 | 13.45 | 27.57 | 13.72 | 43.21 | 26.07 | 50.32 | 26.57 | 38.42 | 16.89 | 51.26 | 17.78 |
| GCE-GNN | 28.01 | 15.08 | 33.42 | 15.42 | 47.36 | 26.97 | 55.57 | 27.54 | 41.16 | 18.15 | **54.22** | 19.04 |
| DHCN | 26.22 | 14.60 | 31.42 | 15.05 | 46.15 | 26.85 | 53.66 | 27.30 | 40.21 | 17.59 | 53.66 | 18.51 |
| FOCOL | **28.51** | **15.73** | **33.71** | **16.04** | **49.80** | **30.18** | **57.83** | **30.96** | **41.23** | **18.20** | 54.08 | **19.11** |

last item, where cosine similarity is adopted to measure the similarity between items. (2) **FPMC** [18] combines MF and Markov-chain for personalized next-basket recommendation. To adapt it for SBR, following [5, 8, 9], we ignored the user latent representations when computing recommendation scores. (3) **GRU4Rec** [3] models user sequences with a multi-layered GRU and trains the model with a session-parallel mini-batch setting with ranking-based loss functions. (4) **NARM** [8] is an RNN-based SBR model that incorporates the attention mechanism into GRU to model users' main purpose and sequential behavior. (5) **STAMP** [9] is an attention-based model for SBR that captures users' short-term interests by the self-attention of the last item. (6) **SR-GNN** [10] employs GGNN to capture complex transition patterns among items in each session. (7) **GCE-GNN** [5] learns session-level and global-level item representations by graph attention networks and employs an attention mechanism with reversed position encoding to extract session representations from two levels of item representations. (8) **DHCN** [13] performs graph convolution on a hypergraph channel and a line graph channel to capture high-order correlations among items and integrate self-supervised learning into SBR.

Following previous studies [5, 6, 7, 8, 9, 10], we adopted the commonly used *hit rate at K* (**HR@K**) and *mean reciprocal rank at K* (**MRR@K**) as the evaluation metrics and reported the results for $K = 10, 20$ in our experiments.

## 4.3. Hyperparameter Setup

Following the existing methods [5, 8, 10, 13], we set the embedding size $d$ to 100, the batch size to 100, and the $L_2$ regularization to $10^{-5}$ for all models. For the baseline models, if their datasets and evaluation settings were the same as ours, we directly report their results in the original papers. Otherwise, we ran their released code with their best parameters setups reported in the original papers to obtain the results. In our model, all parameters were initialized using a Gaussian distribution with a mean of 0 and a standard deviation of 0.1. The mini-batch Adam optimizer with a learning rate of 0.001 was used to optimize our model. We performed grid search to find the optimal hyperparameters on a validation set, which was the last 10% of the training set. The ranges for the hyperparameters were: $\{1, 2, \cdots, 5\}$ for the number of FOGCN layers $L_G$ and the number of Transformer layers $L_T$, $\{0.001, 0.003, 0.01, \cdots, 0.3, 1\}$ for the magnitude of the CSRL task $\beta$.

## 4.4. Performance Comparisons

Table 2 shows the experimental results of all the compared methods and the best results are highlighted in boldface. From the results, we can have the following conclusions:

The conventional methods ItemKNN and FPMC are not competitive because they only use the last item for prediction without considering the complete contextual information of the entire session. The neural network-based methods have much better performance, showing the powerful sequential modeling capability of deep learning models. NARM and STAMP achieve better performance than GRU4Rec, because they use the attention mechanism to dynamically select the important items for learning session representations.

GNN-based methods generally perform much better than the other kinds of neural network models, proving the power of GNNs in capturing complex item transition patterns. GCE-GNN and DHCN obtain better results than SR-GNN, which proves that capturing inter-session relationships can help the model more accurately infer users' preferences in SBR. DHCN is less performant than GCE-GNN. One possible reason is that the global graph in GCE-GNN uses item co-occurrences within a window while the hypergraph in DHCN is equivalent to use all co-occurrences without considering the relative distance, meaning that DHCN is more easily affected by uncorrelated items.

The proposed model FOCOL outperforms all baselines on all the datasets. Particularly, on *Tmall* and *RetailRocket*, FOCOL beats the previous state-of-the-art method GCE-GNN by a large margin, showing the effectiveness of considering the item-level inter-session relationships at a fine-grained level and improving session representation using contrastive learning. Although DHCN also has a contrastive learning module capture the relationships between sessions, it implicitly enforces sessions with large Jaccard similarity to have similar representations, which could introduce wrong signals for session representation learning. Note that FOCOL achieves different performance improvements on different datasets. One possible reason is that the sessions in RetailRocket share the strongest correlation among three

datasets, so our method benefits more from capturing the two levels of inter-session relationships on RetailRocket.

## 4.5. Ablation Study

In the subsection, we conduct ablation study to investigate the contribution of two components, FOGCN and CSRL, to the performance of our model.

TABLE 3. COMPARISON OF DIFFERENT VARIANTS OF FOCOL WITH FOGCN MODIFIED

| Model | Tmall | | RetailRocket | | Diginetica | |
|---|---|---|---|---|---|---|
| | HR@20 | MRR@20 | HR@20 | MRR@20 | HR@20 | MRR@20 |
| FOCOL-NG | 31.32 | 15.01 | 54.67 | 30.48 | 49.76 | 17.93 |
| FOCOL-SG | 32.04 | 15.69 | 55.23 | 30.60 | 52.88 | 18.43 |
| FOCOL | **33.71** | **16.04** | **57.83** | **30.96** | **54.08** | **19.11** |

**4.5.1. Effectiveness of FOGCN.** To study the effectiveness of the proposed FOGCN component, we create two variant of our model: (1) **FOCOL-NG** is a model with the FOGCN removed. We replace the global embedding $\boldsymbol{E}^g$ with the initial embeddings $\boldsymbol{E}^{(0)}$. (2) **FOCOL-SG** is a model which uses a simple graph instead of a FOG. The weights of co-occurrences at different distances are fixed to 1. The results are shown in Table 3. From the results, we can see that FOCOL-NG has the worse performance, which means that capturing item-level inter-session relationships is important to accurately infer user preferences. With a simple graph that captures the item co-occurrences at a coarse level, FOCOL-SG can already significantly outperforms FOCOL-NG. However, FOCOL-SG still has much lower results than our original model, showing the effectiveness of considering item co-occurrences at a fine-grained level.

TABLE 4. COMPARISON OF DIFFERENT VARIANTS OF FOCOL WITH CSRL MODIFIED

| Model | Tmall | | RetailRocket | | Diginetica | |
|---|---|---|---|---|---|---|
| | HR@20 | MRR@20 | HR@20 | MRR@20 | HR@20 | MRR@20 |
| FOCOL-NCL | 32.44 | 15.29 | 55.76 | 28.03 | 53.10 | 18.35 |
| FOCOL-NC | 33.51 | 15.70 | 57.35 | 30.65 | 53.61 | 18.72 |
| FOCOL-ND | 33.57 | 15.79 | 57.46 | 30.72 | 53.74 | 18.81 |
| FOCOL-NI | 33.65 | 15.97 | 57.62 | 30.88 | 53.95 | 19.02 |
| FOCOL-NId | 33.22 | 15.63 | 56.95 | 30.54 | 53.46 | 18.62 |
| FOCOL | **33.71** | **16.04** | **57.83** | **30.96** | **54.08** | **19.11** |

**4.5.2. Effectiveness of CSRL.** To investigate the contribution of CSRL, we create five variants of our model by modifying the CSRL component: (1) **FOCOL-NCL** removes the CSRL component. (2) **FOCOL-NC**, **FOCOL-ND**, **FOCOL-NI**, and **FOCOL-NId** remove one of the augmentation operations from CSRL. For example, FOCOL-NC has no *Corrupt* augmentation, and FOCL-NId has no *Identity* augmentation. The results are shown in Table 4. We can see that the variant FOCOL-NCL with CSRL completely removed has the worse accuracy, proving that capturing the session-level inter-session relationships is essential for improving SBR performance. The variants with one of the augmentation removed achieve better performance than FOCOL-NCL and are just slightly worse than the original model, suggesting than all augmentation operations can help improve session representation learning but none of them plays a pivotal role. Among these four variants, FOCOL-NID has the lowest performance, meaning that the relationships among the original sessions are more informative. FOCOL-NC and FOCOL-ND has similar performance and FOCOL-NI is the best, which may be because the sessions pairs generated by *Corrupt* and *Dropout* are more challenging than those generated by *Insert*.

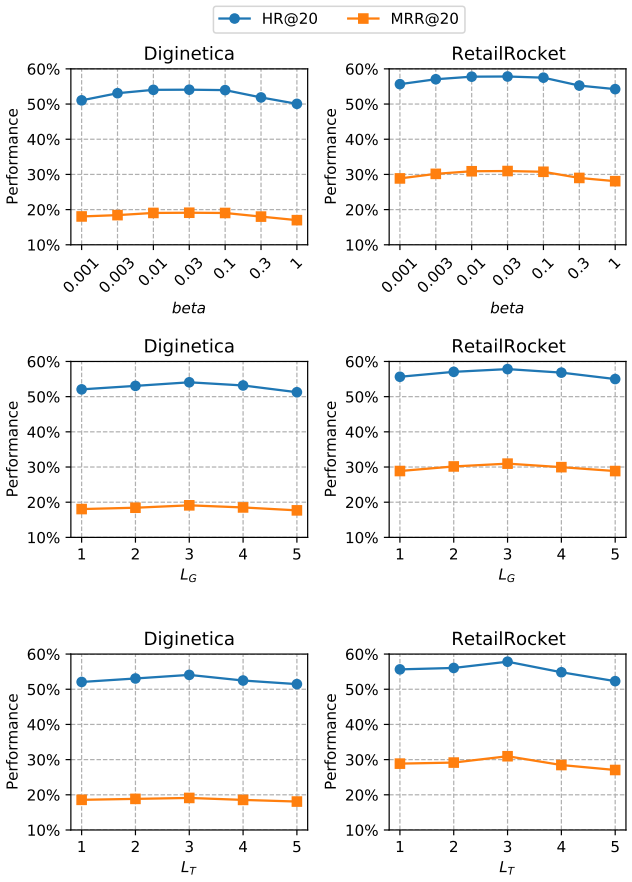


Figure 3. Hyperparameter study

## 4.6. Hyperparameter Study

In this subsection, we study the hyperparameters of our model, including the magnitude of the CSRL loss $\beta$, the number of layers in FOGCN $L_G$, and the number of layers in the Transformer session encoder $L_T$. We fixed the other hyperparameters to their default values ($\beta = 0.03, L_G = 3, L_T = 2$) when varying one of them. We can have the following observations from the results shown in Figure 3.

For the magnitude of the CSRL loss, the best setting is $\beta = 0.03$ for both *Diginetica* and *RetailRocket*. When the value of $\beta$ is in the range of $[0.01, 0.1]$, the performance on both datasets is stable. However, the performance quickly

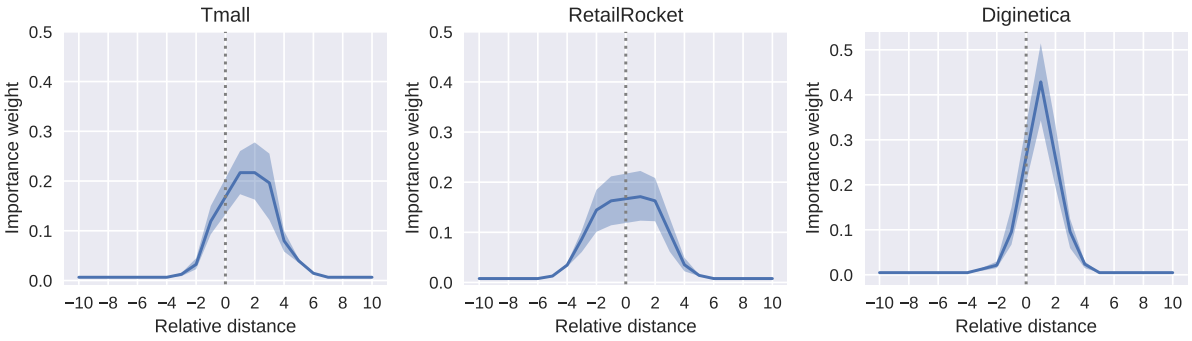


Figure 4. The importance weight w.r.t. the relative distance

decreases when the value becomes smaller or larger. This is because a small value of $\beta$ means the model cannot utilize the learning signals from the CSRL task, while a large value switches the main learning objective to discriminating session representations, while is not fully aligned with the recommendation objective.

For the number of FOGCN layers $L_G$, the optimal value is 3. This is because when $L_G$ is too small, FOGCN cannot capture the useful high-order inter-session item dependencies. When $L_G$ is too large, FOGCN suffers from the over-smoothing problem that is commonly observed in previous studies [5, 7, 13, 37].

For the number of layers in the Transformer encoder $L_T$, we can see its best value is 3 on both datasets. This may be because most of the sessions have a short length, meaning that 3 layers is enough to capture the sequential dependencies in most sessions. When $L_T$ is too large, the model could suffer from the overfitting problem.

### 4.7. Visualizing Learned Importance Weights

In this experiment, we show the importance of considering fine-grained co-occurrences by visualizing the learned importance weights. Specifically, we trained a model with one FOGCN layer, and computed a matrix $\boldsymbol{P} \in \mathbb{R}^{d \times 2L}$ where $\boldsymbol{P}_i = \text{softmax}(\boldsymbol{P}_i^{(0)})$ such that $\sum_{j=1}^{2L} \boldsymbol{P}_{ij} = 1, \forall 1 \leq i \leq d$. Then, $\boldsymbol{P}_{ij}$ can be interpreted as the importance of the co-occurrences at relative distance $j$ to the $i$-th feature. Figure 4 plots the importance weight w.r.t. the relative distance. The blue line is the expectation and the light-blue region denotes the range within one standard deviation, where the expectation and standard deviation are computed over the feature dimension. The importance weights at relative distances larger than 10 or smaller than -10 are not shown because they are negligible. From these plots, we can see that the learned importance weights are different at different relative distances on different datasets. Besides, on *Tmall* and *Diginetica*, the positive relative distances have larger weights than the negative ones, while on *RetailRocket*, the positive and negative relative distances have similar weights. Therefore, it is important to learn the importance weights dynamically to capture the different influences of relative distance and order.

## 5. Conclusion

Recent studies on SBR have shown that capturing inter-session information can help infer user preferences more accurately. However, existing methods do not consider the fine-grained item-level inter-session relationships and are prone to extracting wrong signals to learn user preferences from the session-level inter-session relationships. To fill the gap, we propose FOGCN which can encode the inter-session item co-occurrences in a fine-grained level and automatically learn the importance of ordered co-occurrences at different distances. Furthermore, we view the SBR task as a clustering problem and directly capture the relationships among sessions by contrastive learning. Our experimental results show that the proposed model can outperform the state-of-the-art methods and effectively capture both the item-level and session-level inter-session relationships.

**Acknowledgements**: We are grateful to the anonymous reviewers for their constructive comments on this paper. The research of Tianwen Chen and Raymond Chi-Wing Wong is supported by PRP/026/21FX.

## References

[1] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009, pp. 452–461.

[2] X. He, K. Deng, X. Wang, Y. Li, Y.-D. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *SIGIR*, 2020, pp. 639–648.

[3] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *ICLR*, 2016.

[4] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016, p. 17–22.

[5] Z. Wang, W. Wei, G. Cong, X. Li, X.-L. Mao, and M. Qiu, "Global context enhanced graph neural networks for session-based recommendation," in *SIGIR*, 2020, pp. 169–178.

[6] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, "A simple convolutional generative network for next item recommendation," in *WSDM*, 2019, pp. 582–590.

[7] T. Chen and R. C.-W. Wong, "Handling information loss of graph neural networks for session-based recommendation," in *KDD*, 2020, pp. 1172–1180.

[8] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *CIKM*, 2017, pp. 1419–1428.

[9] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "STAMP: Short-term attention/memory priority model for session-based recommendation," in *KDD*, 2018, pp. 1831–1839.

[10] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural network," in *AAAI*, 2019, pp. 346–353.

[11] P. Ren, Z. Chen, J. Li, Z. Ren, J. Ma, and M. de Rijke, "RepeatNet: A repeat aware neural recommendation machine for session-based recommendation," in *AAAI*, 2019, pp. 4806–4813.

[12] T. Chen and R. C.-W. Wong, "An efficient and effective framework for session-based social recommendation," in *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21)*, 2021, pp. 400–408.

[13] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, and X. Zhang, "Self-supervised hypergraph convolutional networks for session-based recommendation," in *AAAI*, 2021.

[14] C. Huang, J. Chen, L. Xia, Y. Xu, P. Dai, Y. Chen, L. Bo, J. Zhao, and J. X. Huang, "Graph-enhanced multi-task learning of multi-level transition dynamics for session-based recommendation." in *AAAI*, 2021, pp. 4123–4130.

[15] M. Wang, P. Ren, L. Mei, Z. Chen, J. Ma, and M. de Rijke, "A collaborative session-based recommendation approach with parallel memory modules," in *SIGIR*, 2019, pp. 345–354.

[16] P. Jaccard, "The distribution of the flora in the alpine zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[17] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. V. Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath, "The YouTube video recommendation system," in *RecSys*, 2010, pp. 293–296.

[18] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*, 2010, pp. 811–820.

[19] J. Song, H. Shen, Z. Ou, J. Zhang, T. Xiao, and S. Liang, "ISLF: Interest shift and latent factors combination model for session-based recommendation," in *IJCAI*, 2019, pp. 5765–5771.

[20] T. X. Tuan and T. M. Phuong, "3D convolutional networks for session-based recommendation with content features," in *RecSys*, 2017, p. 138–146.

[21] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, "Graph contextualized self-attention network for session-based recommendation," in *IJCAI*, 2019, pp. 3940–3946.

[22] Z. Pan, F. Cai, W. Chen, H. Chen, and M. de Rijke, "Star graph neural networks for session-based recommendation," in *CIKM*, 2020, pp. 1195–1204.

[23] O. J. Hénaff, A. Srinivas, J. D. Fauw, A. Razavi, C. Doersch, S. M. A. Eslami, and A. van den Oord, "Data-efficient image recognition with contrastive predictive coding," *arXiv*, 2019.

[24] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *ICLR*, 2018.

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020, pp. 9729–9738.

[26] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, vol. 1, 2020, pp. 1597–1607.

[27] X. Chen and K. He, "Exploring simple siamese representation learning," in *CVPR*, 2021, pp. 15 750–15 758.

[28] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, and W. Xu, "Consert: A contrastive framework for self-supervised sentence representation transfer," in *IJCNLP*, 2021, pp. 5065–5075.

[29] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *EMNLP*, 2021.

[30] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *SIGIR*, 2021, pp. 726–735.

[31] T. Yao, X. Yi, D. Z. Cheng, F. Yu, T. Chen, A. Menon, L. Hong, E. H. Chi, S. Tjoa, J. Kang, and E. Ettinger, "Self-supervised learning for large-scale item recommendations," *arXiv preprint arXiv:2007.12865*, 2020.

[32] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, "S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *CIKM*, 2020, pp. 1893–1902.

[33] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, B. Ding, and B. Cui, "Contrastive learning for sequential recommendation," *arXiv: Information Retrieval*, 2020.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[35] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv*, 2018.

[36] H. Abdollahpouri, R. Burke, and B. Mobasher, "Controlling popularity bias in learning-to-rank recommendation," in *RecSys*, 2017, pp. 42–46.

[37] R. Qiu, J. Li, Z. Huang, and H. Yin, "Rethinking the item order in session-based recommendation with graph neural networks," in *CIKM*, 2019, pp. 579–588.