An Attribute-Driven Mirror Graph Network for Session-based Recommendation

Siqi Lai
laisiqi@whu.edu.cn
School of Computer Science, Wuhan University
China

Erli Meng
mengerli@xiaomi.com
Xiaomi
China

Fan Zhang†
fan.zhang@whu.edu.cn
School of Information Management, Wuhan University
China

Chenliang Li
cllee@whu.edu.cn
School of Cyber Science and Engineering, Wuhan University
China

Bin Wang
wangbin11@xiaomi.com
Xiaomi
China

Aixin Sun
axsun@ntu.edu.sg
Nanyang Technological University
Singapore

ABSTRACT

Session-based recommendation (SBR) aims to predict a user's next clicked item based on an anonymous yet short interaction sequence. Previous SBR models, which rely only on the limited short-term transition information without utilizing extra valuable knowledge, have suffered a lot from the problem of data sparsity. This paper proposes a novel **m**irror **g**raph enhanced neural model for **s**ession-based recommendation (MGS), to exploit item attribute information over item embedding vectors for more accurate preference estimation.

Specifically, MGS utilizes two kinds of graphs to learn item representations. One is a session graph generated from the user interaction sequence describing users' preference based on transition patterns. Another is a mirror graph built by an attribute-aware module that selects the most attribute-representative information for each session item by integrating items' attribute information. We applied an iterative dual refinement mechanism to propagate information between the session and mirror graphs. To further guide the training process of the attribute-aware module, we also introduce a contrastive learning strategy that compares two mirror graphs generated for the same session by randomly sampling the attribute-same neighbors. Experiments on three real-world datasets exhibit that the performance of MGS surpasses many state-of-the-art models.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

†Fan Zhang is the corresponding author.

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

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531935

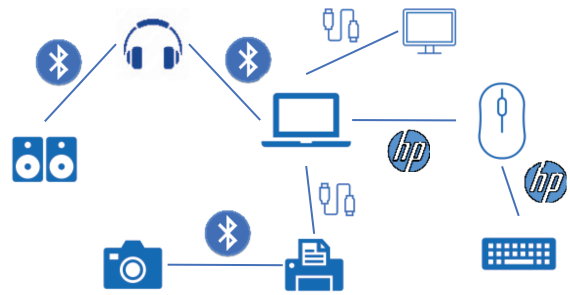


Figure 1: Items are connected by the same attribute values.

KEYWORDS

Recommendation system, Session-based recommendation, Graph neural network, Self-supervised learning

ACM Reference Format:
Siqi Lai, Erli Meng, Fan Zhang, Chenliang Li, Bin Wang, Aixin Sun. 2022. An Attribute-Driven Mirror Graph Network for Session-based Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477495.3531935

1 INTRODUCTION

Recommender systems have been pervasively utilized to serve online shopping and stream platforms due to their great success in recommending useful items for users. Conventionally, most recommendation methods assume that user profiles are included, and long-term behavior records are available. However, such an assumption is challenged by many current real-world circumstances (*e.g.,* online shopping platforms like Tmall, and music websites like last.fm) when user's preferences depend much more on a few recent interactions. For example, a guest who recently needs to buy a laptop and relevant accessories, meanwhile he (or she) is an animal lover, and usually buy many pet supplies on the shopping websites. In that case, some of the user's dated records would be helpless for the recent recommendation. Consequently, session-based recommendation emerges to tackle this challenge, focusing only on the most up-to-date user interactions in the current session to predict the next item. It has received much attention in the past few years, and numerous models have been proposed.

Many deep learning methods have been proposed to tackle the task of session-based recommendation. Recurrent neural networks (RNNs) [6] have become the most prevalent strategy in this realm. Nevertheless, RNNs-based models [5, 8, 10] fail to capture the user preference very well when the sequential relationship is not the only factor to influence user preferences. Recently, graph neural networks (GNNs) [26] have exhibited overwhelming effectiveness in modeling more complex user behaviors and have achieved better performance than RNNs-based models. However, these approaches are also undermined by data sparsity. Due to the absence of long-term user behaviorial information, GNNs-based models [9, 15, 16, 23, 25, 29] have to generate user preferences with the limited interaction data, which is more difficult.

Self-supervised learning (SSL) [11], a refreshing machine learning paradigm that tends to solve the data sparsity problem, has achieved great success in various realms. SSL methods have also been integrated in GNNs for recommendation tasks [24, 27, 28, 31, 32]. Nevertheless, SBR task suffers more deeply from the data sparsity problem than sequence recommendation due to the relatively short interaction sequences.

Furthermore, almost all the previous models generate item representation based only on propagating information over the original interaction sequence while ignoring extra valuable knowledge of each item. Most sequential recommendation studies have comprehensively utilized such information into their models, while session-based counterparts rarely consider them.

To this end, we proposes to take item attribute information, which inherently exists in most real-world datasets (*e.g.,* genre and artist in music broadcast platforms, or category and brand in online shopping scenarios), into our consideration to enhance item representations. By leveraging additional attribute-wise knowledge into item embedding vectors, the user's general preference (*e.g.,* telephone, tool, clothes, etc.) can be captured more accurately. Figure 1 illustrates an example of similar items that are connected through sharing the same attribute (*e.g.,* both the mouse and the keyboard share the same brand attribute value on "HP"). Here, we choose to incorporate attribute information by selecting the most representative item neighbors sharing the same attribute and attribute value (namely *attribute-same*) to generate an attribute-aware graph, namely mirror graph, which enrich the contextual semantics of the session.

Afterwards, we utilize an iterative dual refinement process to update representations of both session and mirror items, which effectively fuses the session-wise and attribute-wise semantics. However, due to the noisy nature of the massive quantity of attribute-same neighbors, it is challenging to efficiently utilize item attribute information. Therefore, we further develop a novel contrastive learning strategy that can further guide the learning of the mirror graph generation. Specifically, this SSL strategy can stimulate the model to recommend attribute-same items with respect to the items in the current session by enhancing the similarity between the attribute-same items.

In summary, we conclude the contributions of this paper as follows:

- We propose to generate an attribute-aware mirror graph, to integrate the item neighbors that share the same attributes as session items to enhance session-based recommendation.
- We apply a novel iterative dual refinement process between the session and mirror graphs to utilize attribute information. Both the session-wise and attribute-wise semantics are fused in an iterative manner for effective representation learning.
- We develop a novel contrastive learning strategy that can supervise the mirror graph generation process by effectively discriminating noisy attribute-same neighbors.

2 RELATED WORK

2.1 Session-based Recommendation

The earliest studies on session-based recommendation mainly focused on discovering short-term item representations in the current session with the Markov chain [17, 18, 30, 33]. Zimdars *et al.*. [33] proposed to use a probability decision tree to model items' transition patterns. Shani *et al.* [18] implemented Markov Decision Process (MDP) to model transition probabilities among items. However, Markov Chain-based methods can only learn information through adjacent interactions, leading to its limitation on capturing interest transfers. Afterwards, collaborative filtering uses a similarity matrix representing the co-occurrence frequency of items to predict the next item. Although collaborative filtering-based models can discover users' general interests, they are incapable of discovering change in user preference.

As deep learning methods became one of the most prevalent modeling techniques, researchers started to utilize recurrent neural networks (RNNs) [6] to model sequential data patterns and succeeded in recommender systems. Hidasi *et al.* [5] proposed a novel method called GRU4REC: the first study of adopting RNNs to model item interactions. Afterwards, Tan *et al.* [19] introduced data augmentation to further develop RNNs-based methods. Li *et al.* [8] proposed to incorporate an attention mechanism into RNNs to model item transitions and user preference in the session. Liu *et al.* [10] developed an attention-based short-term memory networks (STAMP) to capture users' interests with simple multilayer perceptron (MLP). Nevertheless, these RNNs-based methods focus on capturing sequential patterns, and the complex transitions could undermine their effectiveness.

Recently, graph neural networks (GNNs) [26] have been comprehensively used in capturing complex transition relationships and exhibited significant superiority. A session can be well described as a graph. Hence, several efforts have made to adopt GNNs for session recommendation. Specifically, Wu *et al.* [25] proposed to use a gated graph neural network (GGNN) to learn item embedding vectors by propagating information on the session graph, resulting in better session representations. Qiu *et al.* [16] developed a model called FGNN, which can aggregate an item's neighborhood information with multi-head attention. GC-SAN [29] is a derivative of SR-GNN, which applied a self-attention mechanism to model items' co-occurrence. Only recently, works like GCE-GNN[23] and DAT-MDT[1] started to use the information besides the current session. GCE-GNN exploits similar sessions in a more subtle manner for better estimating the user's preference of the current session. DAT-MDT proposes to capture potential mappings across domains to

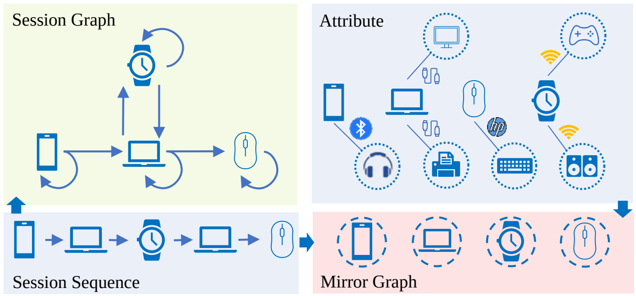


Figure 2: An example of session graph and mirror graph construction.

enrich the contextual information. It is worthwhile to mention that all these methods only consider sparse, short-term user behaviors in a current, which mainly ignore the rich semantics associated with the corresponding attribute information.

2.2 Self-supervised Learning

Self-supervised learning (SSL), a novel technique in the various machine learning realms, has achieved great success in many tasks. Mainly, there are two directions in this line: generation leanring [3, 12, 13] and contrasting learning [2, 7, 14]. The most popular generative method is auto-encoding. It learns to reform the input, which could be filled with noise to enhance the model's robustness. Contrastive models learn to compare through a Noise Contrastive Estimation (NCE) objective. SSL is also applied widely on graph data [4, 22, 24, 27, 28, 31, 32]. DGI [22] proposed to learn node representations based on interactive information among nodes and the local structure. Kaveh *et al.* [4]. developed a contrastive model to learn local and global representation by comparing two different views.

Very recently, SSL has also been validated to be useful for recommendation. S^3-Rec [32] adopted the mutual information maximization principle to learn the potential relationships among items, attributes, and sequence. S^2-DHCN [28] deployed contrastive learning among different hypergraphs by applying self-discrimination. COTREC [27] proposed to compose session data into two views which reflect the internal and external connectivity of sessions. Yu *et al.* [31] proposed a novel structure called tri-training framework to utilize different aspects of social information and generate complementary self-supervision signals. SGL [24] generates multiple views of a node based on three different dropout strategies. Then, the similarity of different views is maximized to eliminate noise and enhance the model's robustness.

3 PRELIMINARIES

In this section, we first present the problem statement, then introduce the session graph construction process and the corresponding graph convolution procedure to derive the representation of each item in a session to encode session-wise semantics.

3.1 Problem Statement

Here, we consider a typical session-based recommendation scenario with an item set $\mathcal{X}$. An anonymous session $s = \{x_1, \ldots, x_n\}$, can be formulated as an interaction sequence by sorting the items clicked by the user in a chronological order, where x_i is the i-th item in this session, and n is the length. The task of session-based recommendation is to identify the next item x_{n+1} that will be interacted by the user for current session s.

3.2 Session Graph

The transition behaviors contained in the session convey lots of user preference. Following the existing work [23], we firstly form a *session graph* $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ where $\mathcal{V}_s \subseteq \mathcal{X}$ is the set of session items in s, and $\mathcal{E}_s$ is the corresponding edge set, which represents the relation between two adjacent items (x_i, x_{i+1}) in s. Here, a *session item* is defined to be a constituent item in the session. To explicitly distinguish complex user behaviors, we formulate $\mathcal{G}_s$ as a directed graph to indicate the user transition behaviors and include a self-loop edge for each session item. An example for the session graph construction is illustrated in the left part of Figure 2.

3.3 Graph Convolution

After modeling the user transition behaviors in $\mathcal{G}_s$, we can perform iterative graph convolution to aggregate a session item's neighborhood for representation learning. Following the setting in [23], four kinds of relations $\mathcal{E} = \{e_{in}, e_{out}, e_{in-out}, e_{self}\}$ can be attached to different neighbors of session item x_i in terms of the associated edges in $\mathcal{G}_s$: 1) the neighbor is an in-degree node for x_i (e_{in}); 2) the neighbor is an out-degree node for x_i (e_{out}); 3) the neighbor holds both bi-directional edges towards x_i (e_{in-out}); and 4) the neighbor is x_i itself (e_{self}). Note that for each neighbor, there could exist only one type from $\mathcal{E}$.

We utilize a graph attention network (GAT) [21] to perform graph convolution over $\mathcal{G}_s$. Specifically, we firstly calculate the neighbor importance for each session item x_i as follows:

$$\alpha_{ij} = \frac{exp(LeakyReLU(\mathbf{e}_{ij}^{\top}(\mathbf{x}_i^{(l-1)} \odot \mathbf{x}_j^{(l-1)})))}{\sum_{x_k \in \mathcal{N}_{x_i}} exp(LeakyReLU(\mathbf{e}_{ik}^{\top}(\mathbf{x}_i^{(l-1)} \odot \mathbf{x}_k^{(l-1)})))} \quad (1)$$

where a_{ij} indicates the importance of x_j to x_i, $\mathbf{x}_i$ is the representation of session item x_i in the $(l-1)$-th layer, $\mathbf{e}_{ij}$ is the embedding of the relation held between x_j and x_i, l indicates the layer number, $\odot$ is the element-wise multiplication and $\mathcal{N}_{x_i}$ is the set of neighboring session items of x_i in $\mathcal{G}_s$. The representation of x_i for l-th layer is then derived as follows:

$$\mathbf{x}_i^{(l)} = \sum_{x_j \in \mathcal{N}_{x_i}} \alpha_{ij} \mathbf{x}_j^{(l-1)} \quad (2)$$

where $\mathbf{x}_i^{(0)}$ is equivalent to the learnable item embedding of item x_i.

4 THE PROPOSED METHOD

With a close eye on recent session-based recommendation models, most of them ignore items' inherent attributes that widely exist in most real-world scenarios. As illustrated in Figure 1, we can follow the associated attributes to reach attribute-same items that

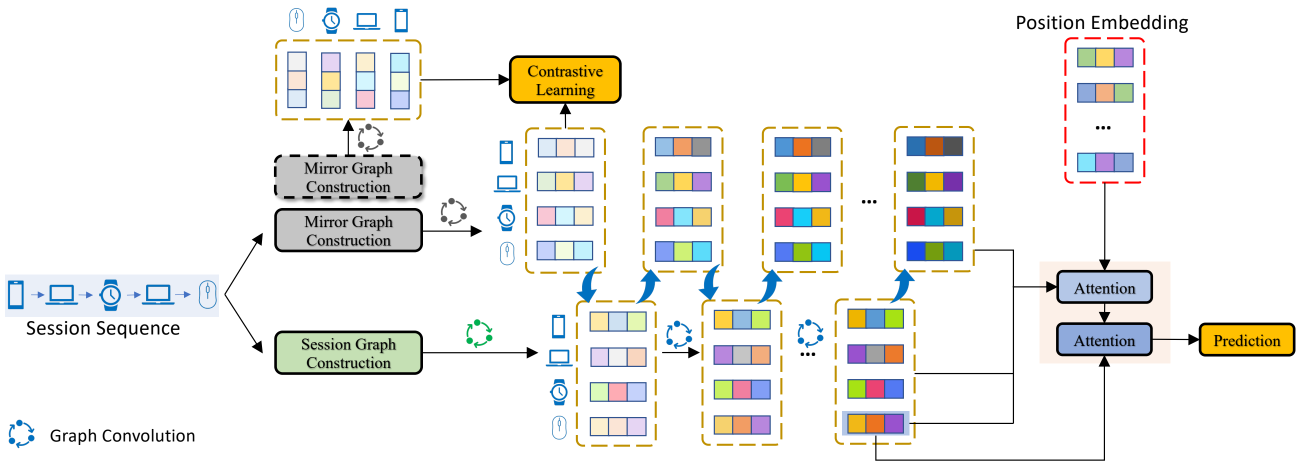


Figure 3: The network architecture of MGS.

share the same attributes. These similar items could provide more semantics to enrich the preference signals not covered in the session graph and guide the preference learning from the latter.

Here, we propose a novel mirror graph enhanced neural model for session-based recommendation (MGS). A mirror graph is like an image of the actual session graph, such that each session item has a corresponding mirror item in the mirror graph. In Section 3.3, we derive the representation of session item in terms of their session-wise transition patterns. On the contrary, we choose to generate the representation of a mirror item from its attribute-same item neighbors. It aims to capture more comprehensive information which can complement the session graph and lead to better user preference understanding.

Figure 3 illustrates the architecture of the proposed MGS. MGS contains four main components: 1) *mirror graph generation*. It builds a mirror graph based on attribute-wise semantics of each session item; 2) *iterative dual refinement*. It guides the session item representation learning via a mutual refinement against each other; 3) *session representation learning*. It aggregates the item representations in both session graph and mirror graph to derive the user preference of the current session by ; 4) *self-supervised learning*. It further enhances the mirror graph generation in an unsupervised manner, where the attention mechanism can be optimized to achieve better discrimination capacity. We next present each of them in detail.

4.1 Mirror Graph Generation

Generally, an item could contain multiple auxiliary attributes. For example, a music song could contain attributes such as artist and genre; a product provided by the online shopping platform could have attributes like category and brand. We denote $\mathcal{R} = \{r_1, \ldots, r_M\}$ as the attribute set (e.g., brand and category), and $\mathcal{A}_{r_p} = \{a_1, \ldots, a_O\}$ as the attribute value set (e.g., Apple and Google), where M and O are the numbers of unique attributes and the attribute values in $\mathcal{R}$ and $\mathcal{A}_{r_p}$, respectively. Given item x contains attribute r_p, we can utilize triplet $< x, r_p, a_{(x,r_p)} >$ to represent this attribute information where $a_{(x,r_p)}$ is the specific value item x holds on this attribute.

Hence, for each mirror item m_i corresponding to session item x_i, we sample b attribute-same items from $\mathcal{X}$. Specifically, a neighbor set $\mathcal{N}_{m_i}^{r_p} = \{x_j | a_{(x_i,r_p)} = a_{(x_j,r_p)}\}$ for a given m_i and attribute r_p pair is sampled where $| \mathcal{N}_{m_i}^{r_p} |= b$.

Then, we perform an attention mechanism to aggregate the neighbors of each attribute r_p for mirror item m_i:

$$\alpha_{ij} = \frac{exp(LeakyReLU(\mathbf{q}_{r_p}^{\top}[\mathbf{x}_i^{(0)} \parallel \mathbf{x}_j^{(0)}]))}{\sum_{x_k \in \mathcal{N}_{m_i}^{r_p}} exp(LeakyReLU(\mathbf{q}_{r_p}^{\top}[\mathbf{x}_i^{(0)} \parallel \mathbf{x}_k^{(0)}]))} \tag{3}$$

$$\mathbf{m}_i^{r_p} = \sum_{x_j \in \mathcal{N}_{m_i}^{r_p}} \alpha_{ij}\mathbf{x}_j^{(0)} \tag{4}$$

where $\mathbf{m}_i^{r_p}$ is the representation of mirror item m_i under attribute r_p, $\mathbf{q}_{r_p}$ is the embedding vector of attribute r_p, $\parallel$ is the concatenation operation. We concatenate $\mathbf{m}_i^{r_p}$ for all the associated attributes of x_i and generate the attribute-wise representation for mirror item m_i as follows:

$$\mathbf{m}_i = \mathbf{W}_1[\mathbf{m}_i^{r_1} \parallel \cdots \parallel \mathbf{m}_i^{r_M}], \tag{5}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times Md}$ is the learnable parameter, and $\mathbf{m}_i$ is the attribute-wise representation for m_i. Normally, an item could only contain a few attributes. If there is only one attribute, we simply take $\mathbf{m}_i^{r_1}$ as $\mathbf{m}_i$.

4.2 Iterative Dual Refinement

As mentioned above, the resultant mirror item representation $\mathbf{m}_i$ is encoded with the attribute-wise semantics. Recall that we iteratively update the session item's representation in session graph $\mathcal{G}_s$ (ref. Section 3.3). Here, we plug in these mirror item representations to guide the session-wise representation learning via an iterative dual refinement process. Both the representations of session items and their corresponding mirror items are used to update each other in an iterative manner, which effectively fuse the session-wise and attribute-wise semantics.

Session Item Refinement. After performing l-th graph convolution for session graph $\mathcal{G}_s$ (ref. Equation 2), we need to fuse the attribute-wise information from the mirror graph to enhance the session items' representations. In detail, for each item x_i in session graph, we choose to refine $\mathbf{x}_i$ based on the element-wise offset between $\mathbf{x}_i$ and $\mathbf{m}_i$ via a weighting scheme:

$$\beta_i = \frac{(\mathbf{W}_1^s \mathbf{x}_i^{(l)})^\top \mathbf{W}_2^s \mathbf{m}_i^{(l-1)}}{\sqrt{d}} \tag{6}$$

$$\mathbf{x}_i^{(l)} = \mathbf{x}_i^{(l)} + \beta_i(\mathbf{m}_i^{(l-1)} - \mathbf{x}_i^{(l)}) \tag{7}$$

where $\mathbf{W}_1^s, \mathbf{W}_2^s \in \mathbb{R}^{d\times d}$ are learnable parameters, $\mathbf{m}_i^{(l-1)}$ is the refined mirror item representation by the previous iteration (and $\mathbf{m}_i^{(0)} = \mathbf{m}_i$).

Mirror Item Refinement. We then utilize the resultant $\mathbf{x}_i^{(l)}$ to further refine the representations of the mirror items. The idea is to fuse the attribute-wise semantics via the guidance of the session-wise semantics. Specifically, we evaluate the importance of each mirror node to every session node by an attention mechanism.

$$a_{ij} = \frac{exp\Big((\mathbf{W}_1^m \mathbf{m}_i^{(l-1)})^\top \mathbf{W}_2^m \mathbf{x}_j^{(l)}\Big)}{\sum_{k=1}^n exp\Big((\mathbf{W}_1^m \mathbf{m}_i^{(l-1)})^\top \mathbf{W}_2^m \mathbf{x}_k^{(l)}\Big)} \tag{8}$$

where $\mathbf{W}_1^m, \mathbf{W}_2^m \in \mathbb{R}^{d\times d}$ are learnable parameters. Then, we derive $\mathbf{m}_i^{(l)}$ for the current layer by a linear combination:

$$\mathbf{m}_i^{(l)} = \sum_{j=1}^n a_{ij} \mathbf{m}_j^{(l-1)} \tag{9}$$

In Equation 8 and 9, we can see that the mirror item identifies the relevant session-wise information and aggregate their corresponding attribute-wise information to update its representation. We have also tried to connect each mirror node m_i only to the adjacent session items of x_i. However, it performs significantly worse instead. We will analysis it in Section 5.6.4.

After the dual refinement through L iterations, we apply a highway network to derive the final representation for each session item as follows:

$$\mathbf{x}_i^{(L)} = \boldsymbol{\pi}_i \mathbf{x}_i^{(0)} + (1 - \boldsymbol{\pi}_i)\mathbf{x}_i^{(L)} \tag{10}$$

$$\boldsymbol{\pi}_i = sigmoid(\mathbf{W}_h[\mathbf{x}_i^{(0)} \parallel \mathbf{x}_i^{(L)}]) \tag{11}$$

where $\mathbf{W}_h \in \mathbb{R}^{d\times 2d}$ is a learnable parameter.

4.3 Session Representation Learning

Now we can obtain session representation by aggregating information from both session item representations and mirror item representations.

Intuitively, session items in different positions contributed differently to the user's preference. Hence, we attach the position embedding to each session item as follows:

$$\mathbf{h}_i = \mathbf{x}_i^{(L)} + \mathbf{p}_i. \tag{12}$$

where $\mathbf{p}_i$ is the embedding of i-th position and $\mathbf{h}_i$ is position fused representation for session item x_i. Afterward, we take $\mathbf{m}_i^{(L)}$ to indicate the user's general intent (*e.g.*, telephone, tools, clothes, etc.). The session representation is initially derived via an attention mechanism as follows:

$$\beta_i = \mathbf{g}^\top sigmoid(\mathbf{W}_2\mathbf{h}_i + \mathbf{W}_3\mathbf{m}_i^{(L)} + \mathbf{W}_4\mathbf{x}_n^{(L)} + \mathbf{b}) \tag{13}$$

$$\mathbf{z}_s = \sum_{i=1}^n \beta_i \mathbf{h}_i, \tag{14}$$

where $\mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_4 \in \mathbb{R}^{d\times d}$ and $\mathbf{g}, \mathbf{b} \in \mathbb{R}^d$ are learnable parameters. Then, a gating mechanism is utilized to fuse the last session item with $\mathbf{z_s}$ as follows:

$$\boldsymbol{\theta} = sigmoid(\mathbf{W}_5[\mathbf{z_s} \parallel \mathbf{x}_n^{(L)}]) \tag{15}$$

$$\mathbf{z}_s = (1 - \mu\boldsymbol{\theta}) \odot \mathbf{z_s} + \mu\boldsymbol{\theta} \odot \mathbf{x}_n^{(L)} \tag{16}$$

where $\mathbf{W}_5 \in \mathbb{R}^{d\times 2d}$ is the learnable parameter, μ is a coefficient to control the impact of the gate mechanism, $\mathbf{z}_s$ is the final session representation.

4.4 Prediction and Model Optimization

After the session representation generation, the recommendation probability $\hat{y}_c$ for a candidate item x_c can be calculated in terms of the inner product followed by a softmax function:

$$\hat{y}_c = \frac{exp(\mathbf{z}_s^\top \mathbf{x}_c^{(0)})}{\sum_{i=1}^N exp(\mathbf{z}_s^\top \mathbf{x}_i^{(0)})}, \tag{17}$$

We use cross-entropy loss to guide the model learning as follows:

$$\mathcal{L}_{main}(\hat{y}) = -\sum_{c=1}^N y_c log(\hat{y}_c) + (1 - y_c) log(1 - \hat{y}_c), \tag{18}$$

where y_c is the ground-truth label for candidate item x_c.

Self-supervised Learning. Due to the massive quantity of attribute-same neighbors available for each session item, the sampled b neighboring items for each associated attribute could certainly contain some noise. The attention mechanism we utilized to perform graph convolution may perform terribly at the beginning. Therefore, we decide to utilize a contrastive learning method to further supervise the mirror graph generation process. Specifically, we generate two mirror graphs for the same session by randomly sampling the attribute-same neighbors. The resultant mirror item representations for same session item x_i are denoted as $\mathbf{m}'_i$ and $\mathbf{m}''_i$. Since they are both learned from attribute-same neighbors, their representations are expected to be similar. That is, the model would more tend to recommend items that share same attributes as items in the current session. Formally, we apply the contrastive loss as follows:

$$\mathcal{L}_{ssl} = -\sum_{i=1}^n log \frac{exp(cos(\mathbf{m}'_i, \mathbf{m}''_i)/\tau)}{\sum_{j=1}^n exp(cos(\mathbf{m}'_i, \mathbf{m}''_j)/\tau)} \tag{19}$$

where $cos(\cdot)$ measures the cosine similarity between two vectors, τ is the hyper-parameter, known as the temperature. Overall, the final loss function of our model is described as follow:

$$\mathcal{L} = \mathcal{L}_{main} + \varphi\mathcal{L}_{ssl} + \lambda \|\Phi\|_2^2, \tag{20}$$

where φ and λ are hyperparameters to control the strength of SSL and L_2 regularization respectively.

Table 1: Statistics of the datasets. "avg. AS num" indicates the average number of each item's attribute-same items.

Dataset	Tmall	Diginetica	30music
# click	818,479	982,961	1,429,251
# train	351,268	719,470	1,153,622
# test	25,898	60,858	122,517
# items	40,728	43,097	132,648
# attribute num	2	1	1
avg. AS num.	848.72	243.58	41.89
avg. len.	6.69	5.12	9.33

5 EXPERIMENTS

In this section we conduct extensive experiments to evaluate the performance of our proposed MGS[1] model by answering the following research questions:

- **RQ1**: Does MGS surpass state-of-the-art SBR baselines in real world datasets?
- **RQ2**: Does incorporating attribute information into item embedding vectors by the mirror graph improve the performance of our model?
- **RQ3**: Does our self-supervised learning module work correctly to guide the mirror graph generation model and cluster similar items?
- **RQ4**: Why do we connect each mirror item to every session item rather than link it only with the adjacent items of its corresponding session item?
- **RQ5**: What is the impact of incorporating the last clicked item into session representation for different datasets?

5.1 Datasets and Preprocessing

We apply three real-world datasets to evaluate our model, known as Tmall[2], Diginetica[3], and 30music[4]. The Tmall dataset is from IJCAI-15 competition, which contains anonymized users' shopping logs on Tmall in the past six months before and on the "Double 11" day. Each item includes two attributes (*i.e.,* category and brand). The Diginetica dataset comes from Personalized E-commerce Search Challenge of CIKM Cup 2016, containing anonymized search and browsing logs, product data, anonymized transactions, and product images. Here, we only consider the transaction data. Diginetica only has a category attribute. Finally, the 30music dataset collects listening and playlists data retrieved from Internet radio stations Last.fm API [20], where the items have only one kind of attribute (*i.e.,* artist).

Following [25, 29], we preprocess three datasets as follows. Sessions with only a single item and items appearing less than 5 times are filtered out for all three datasets. Same as [10], we set the sessions of the most recent ones (*e.g.,* the last week) as the test data, and the remaining for model training. Furthermore, for a session $s = \{x_1, \dots, x_n\}$, we generate sequences and corresponding labels by a sequence splitting process for both training and testing sets across all three datasets (*i.e.,* $([x_1], x_2), ([x_1, x_2], x_3), \cdots, ([x_1, \cdots, x_{n-1}], x_n)$). Note that for every dataset, we utilize all the available attributes. Every item includes an attribute value (*e.g.,* Apple, Google) for every attribute kind (*e.g.,* brand) mentioned above. Table 1 summarizes the statistics of the three datasets after the preprocessing.

[1]Our implementation is available at https://github.com/WHUIR/MGS
[2]https://tianchi.aliyun.com/dataset/dataDetail?dataId=42
[3]https://competitions.codalab.org/competitions/11161#learn_the_details-data2
[4]https://recsys.deib.polimi.it/datasets/

5.2 Evaluation Metrics

We adopt two widely used ranking based metrics: *P@K* and *MRR@K* by following previous work [10, 25].

P@K (Precision): The P@K score measures whether the ground-truth item is included in the top-K list.

$$P@K = \frac{n_{hit}}{|S_{test}|}, \tag{21}$$

where S_{test} is the set of test sequences, and n_{hit} is the number of cases that the ground-truth item is in the top-K list.

MRR@K (Mean Reciprocal Rank): The MRR@K score indicates the position of the ground-truth item in the list of recommendation items. It is set to 0 if the ground-truth item is not in the top-K list, and otherwise is obtained as follows:

$$MRR@K = \frac{1}{|S_{test}|} \sum_{x_{gt} \in S_{test}} \frac{1}{Rank(x_{gt})}, \tag{22}$$

where x_{gt} is the ground-truth item, and $Rank(x_{gt})$ is the position of the ground-truth item in the top-K list.

5.3 Baseline Algorithms

We compare our model with classic methods as well as state-of-the-art models. The following eight baselines are evaluated.

FPMC [17]: It is a next basket recommendation (NBR) model based on Markov Chain, capturing both chronological effects and user preferences. Following the previous works, we ignore the user latent embedding for session-based recommendation.

GRU4REC [5]: It is an RNN-based model that utilizes a session-parallel mini-batch training process and adopts ranking-based loss function to capture user preference.

NARM [8]: It is a variant of RNN-based models, which incorporates attention modules into RNN to enhance its performance.

STAMP [10]: It applies attention mechanism to replace RNN and deploys self-attention to further enhance its performance.

SR-GNN [25]: It is the first work which adopt GNNs to obtain item embedding vectors. The session representation is then derived via a soft-attention mechanism.

GCE-GNN [23]: It is the first work which exploits similar sessions beyond the current session to generate global-level item embedding vectors to enhance its performance.

S^2-DHCN [28]: It constructs two types of hypergraphs to learn inter- and intra-session information and utilizes the self-supervised learning to enhance its performance.

COTREC [27]: It composes session data into two views to reflect the internal and external connectivity of sessions, which are then utilized to supervise each other.

Table 2: Performances(%) of all comparison methods on three datasets. The best and second-best results are highlighted in boldface and underlined, respectively. * denotes MGS significantly surpasses the second-best model using a pair t-test ($p < 0.01$).

Models	Tmall				Diginetica				30music			
	P@10	MRR@10	P@20	MRR@20	P@10	MRR@10	P@20	MRR@20	P@10	MRR@10	P@20	MRR@20
FPMC	13.10	7.12	16.06	7.32	15.43	6.20	26.53	6.95	1.51	0.55	2.40	0.61
GRU4REC	9.47	5.78	10.93	5.89	17.93	7.33	29.45	8.33	15.91	10.46	18.28	10.95
NARM	19.17	10.42	23.30	10.70	35.44	15.13	49.70	16.17	37.81	25.95	39.40	26.55
STAMP	22.63	13.12	26.47	13.36	33.98	14.26	45.64	14.32	36.13	25.97	42.57	26.27
SR-GNN	23.41	13.45	27.57	13.72	36.86	15.52	50.73	17.59	36.49	$\underline{26.71}$	39.93	$\underline{26.94}$
GCE-GNN	29.19	15.55	34.35	15.91	41.54	**18.29**	$\underline{54.64}$	**19.20**	39.93	21.21	44.71	21.55
S^2-DHCN	26.22	14.60	31.42	15.05	41.16	18.15	53.18	18.44	$\underline{40.05}$	17.58	$\underline{45.49}$	17.97
COTREC	$\underline{30.62}$	$\underline{17.65}$	$\underline{36.35}$	$\underline{18.04}$	**41.88**	18.16	54.18	19.07	39.88	17.42	45.15	17.79
MGS	**35.39***	**18.15***	**42.12***	**18.62***	$\underline{41.80}$	$\underline{18.20}$	**55.05***	$\underline{19.13}$	**41.51***	**27.67***	**46.46***	**28.01***

5.4 Parameter Setup

Following previous methods [8, 10, 25], the hidden size of the latent vector is selected in $\{100, 120, \ldots, 180\}$, and the size for mini-batch is set according to each hidden size. To ensure fair comparison, we report the best performance results obtained by deploying the most effective hyper-parameters for each model. The sample number of items used to generate the mirror graph is selected in $\{0, 10, \ldots, 100\}$. The control coefficient μ is chosen from $\{0.0, 0.25, \ldots, 1.25\}$. The temperature coefficient τ for softmax function in the self-supervised module is searched in $\{1.0, 0.5, \ldots, 0.05\}$. The SSL strength φ is selected from $\{1.0, 1.5, 2.0, 2.5\}$. All the parameters are initialized using Gaussian distribution with a mean of 0 and a standard deviation of 0.1. We utilize Adam optimizer with the initial learning rate of 0.001 and decay rate of 0.1 after the first three epochs. The L2 penalty is set to 10^{-5}. The validation dataset is split from the training set with a ratio of 10%.

5.5 Overall Performance (RQ1)

Table 2 reports the performance comparison between these baselines and our proposed MGS on three real-world datasets. Here, we can make the following observations:

RNN-based methods. The RNN-based methods (*e.g.,* GRU4REC, NARM, STAMP) that utilize item transition information tremendously surpass the performance of classical methods (*e.g.,* FPMC) in all the three dataset, indicating the advantages of incorporating sequential information in session-based recommendation. Among them, NRAM and STAMP outperform GRU4REC, which suggests the effectiveness of the attention mechanism.

GNN-based methods. Almost all the GNN-based methods outperform the RNN-based models, indicating that graph neural networks can capture complex transition relationships. On the other hand, it is difficult for RNN-based methods to exploit knowledge beyond the transition patterns. GCE-GNN performs better than SR-GNN in Tmall and Diginetica datasets, suggesting that fusing extra knowledge besides the current session could help learn the users' latent interest better. S^2-DHCN and COTREC model inter- and intra-session information with two graphs, leading to a more robust learning. However, after improving the size of their mini-batch, GCE-GNN outperforms COTREC on the larger dataset, Diginetica, indicating COTREC also has limitations in their self-discrimination SSL, and incorporating extra knowledge is more effective.

Our proposed method. Our proposed MGS almost surpasses all the baselines on these three datasets. Specifically, our model can tremendously outperform on Tmall dataset, which contains two different attributes, indicating MGS has better performance when the dataset includes more attribute information. Additionally, since attribute-same items are clustered by our SSL technique, P@K is significantly higher for all the three datasets. Compared with COTREC and S^2-DHCN that utilize SSL techniques, our proposed MGS is significantly helpful to deliver a more robust mirror graph generation. Specifically, our model can perform much better after using the SSL method, which will describe in Section 5.6.3

Although GCE-GNN and COTREC perform partially better in Diginetica, the GCE-GNN global aggregation component is so complex that it suffers from massive GPU memory use. Meanwhile, COTREC costs too much time for training to reach its optimal performance. Moreover, due to the absence of abundant attribute information, the performance of our model is limited in Diginetica and 30music. Furthermore, GCE-GNN, COTREC, and S^2-DHCN fail to consider the most recent user preference, which is critical on music platforms whose users tend to listen to albums that are similar to their current ones. Hence, their MRR@K scores are much worse than MGS on the 30music dataset.

5.6 Ablation Study

5.6.1 Impact of Attribute Information (RQ2). Here, we discuss the effectiveness of attribute information by comparing MGS with the baseline that excludes the iterative dual refinement and self-supervising module, namely *MGS (No-attri)*. In order to describe its impact more clearly, we conduct two experiments, and develop their evaluation metrics. Firstly, we would like to know whether the performance on sessions, which includes items that share the same attribute value as the ground-truth, is improved. We calculate their MRR scores as follows:

$$MRR_{\alpha} = \frac{1}{|S_{asg}|} \sum_{x_{gt} \in S_{asg}} \frac{1}{Rank(x_{gt})}, \tag{23}$$

where S_{asg} are the sessions including items that have the same attribute value as the ground-truth. Next, we want to know whether session items' attribute-same items are better considered. We calculate their MRR scores as well:

$$MRR_{\beta} = \frac{1}{|V_{ass}|} \sum_{x_{as} \in V_{ass}} \frac{1}{Rank(x_{as})}, \tag{24}$$

where V_{ass} are sessions' attribute-same items, x_{as} is a member of them.

Table 3 exhibits the experiments conducted on Tmall and 30music datasets. Firstly, we can see that the performance on sessions, which contains items that share the same attribute values as their corresponding ground-truth items, is further improved (*i.e., MRR_{α}*). Additionally, thanks to the clustering effect of MGS, although some session's attribute-same items are not the ground-truth, their ranks (*i.e., MRR_{β}*) also moved upwards after considering the attribute information. These two experiments both indicate MGS is more inclined to recommend the attribute-related items.

Table 3: Impact of attribute information.

Method	Tmall		30music	
	MRR_{α}	MRR_{β}	MRR_{α}	MRR_{β}
MGS (No-attri)	26.72	0.37	50.88	5.04
MGS	**28.16**	**0.43**	**51.46**	**5.85**

5.6.2 Impact of Number of Attribute-same Neighbors (RQ2). We next conduct experiments on two datasets to evaluate the impact of different sample numbers of attribute-same items for mirror graph generation. This value is selected from $\{0, 10, 20, 50, 100\}$.

Figure 4a shows the comparison using different numbers on Tmall and 30music datasets. It is clear that our model's performance increases as we sample attribute-same items to build the mirror graph and incorporate this information with item embedding vectors. Additionally, we can also observe that the performance of our model can be improved on MRR@20 by sampling more neighbors when the average number of each item's attribute-same neighbors is relatively few (*e.g.,* 30music). It indicates that the attention based models can capture critical information much more effectively when the sample number is relatively large, and the total quantity is small. For a dataset like Tmall where the amount of each item's neighbors is tremendously massive, a high sample ratio may introduce noise to item embedding vectors and affect the performance negatively. Under this circumstance, the model can select the most relevant ones precisely and cancel the noise by sampling fewer neighbors.

5.6.3 Impact of Self-supervised Learning (RQ3). Here, we investigate the impact of our SSL component. specifically. We remove the self-supervised learning component from MGS (namely MGS-NS). Table 4 shows the results of the two models. It is clear that the full model is improved significantly after incorporating the self-supervised learning, indicating its effectiveness in guiding the training process of the mirror graph generation.

Furthermore, this SSL strategy makes the model would tend to recommend items that share the same attribute values as items in the current session. Table 4 reports the sum of the similarity between attribute-same items (denoted as Sim, the smaller, the more similar) before and after applying the SSL component. The similarity score is obtained as follows:

$$Sim = -\sum_{x_i \in \mathcal{X}} \sum_{x_j \in \mathcal{N}_{x_i}^{as}} log(sigmoid(cos(\mathbf{x}_i, \mathbf{x}_j))) \tag{25}$$

where $\mathcal{N}_{x_i}^{as}$ are attribute-same items of item x_i, $cos(\cdot)$ measures the cosine similarity between two vectors.

Next, we discuss the impact of the temperature coefficient τ in SSL loss function. The τ value is searched from $\{1.0, 0.5, 0.25, 0.1, 0.05\}$. Figure 4b plots performance pattern by varying temperature value on Tmall and Diginetica datasets. We can observe that the performance is limited when the temperature is relatively high. It is reasonable since a much large τ value could introduce much less discriminative signals. Note that we should pay more attention to the cases that are hard to discriminate. Therefore, when we set τ to a small number, the normalized difference between contrast objects is shrunk after applying the softmax function, leading the model's high gradient on hard-negative examples.

Table 5: Impact of different mirror graph refinement methods.

Method	Tmall		Diginetica	
	P@20	MRR@20	P@20	MRR@20
MGS (Adj)	40.46	18.11	54.94	19.01
MGS	**42.12**	**18.62**	**55.05**	**19.13**

5.6.4 Impact of Iterative Dual Refinement (RQ4). To update mirror item representations by identifying the relevant session-wise information and aggregating their corresponding attribute-wise information, we applied a self-attention mechanism to calculate every mirror node's attention weight to each connected session node. Then we obtain the updated mirror node representations by using a linear combination. Here, we investigate another different strategy by only connecting each mirror node to adjacent items of its corresponding session node (namely *MGS (Adj)*).

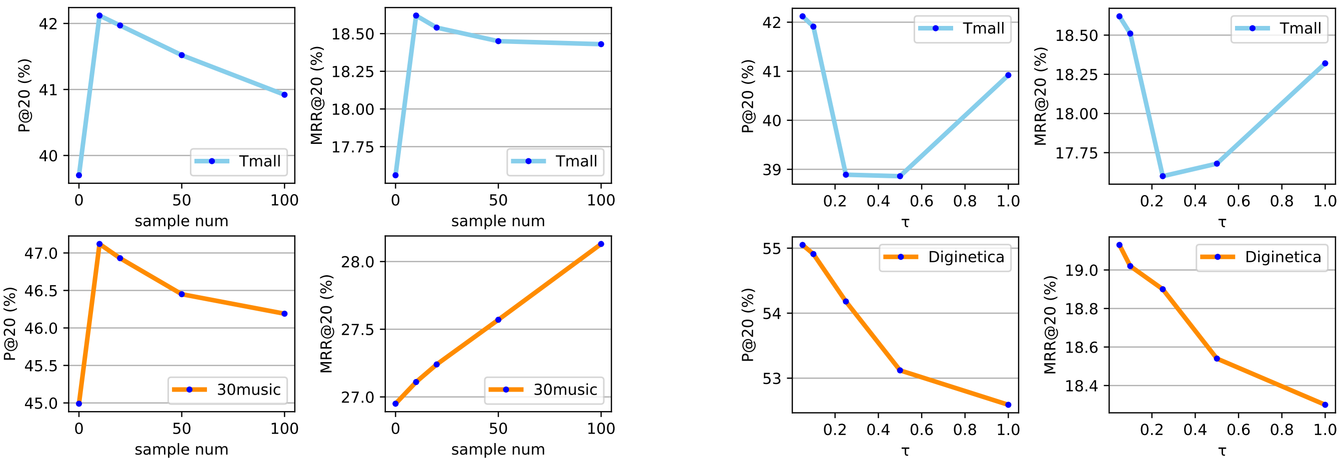
Table 5 reports the results for both MGS and MGS (Adj) on Tmall and Diginetica datasets. We can observe that the connecting each mirror node with every session node outperforms the alternative. This suggests that the relevant session-wise representations can be captured more precisely by exploiting multi-hop global-level information.

5.6.5 Impact of the Last Clicked Item (RQ5). We further examine the impact of incorporating the most recent user's interest towards the resultant session embedding, Specifically, we vary μ value from $\{0.0, 0.25, 0.5, 1.0, 1.25\}$.

Figure 5 plots performance pattern on Tmall and 30music datasets. The Tmall dataset relies less on the last clicked item, indicating the model needs more comprehensive information to better understand a user's preference. On the other hand, the 30music dataset more profoundly relies on the uptodate information, revealing that music software users prefer to listen to similar albums in a short continuous period. Although the performance slightly raises when

Table 4: Impact of self-supervised learning.

Method	Tmall			Diginetica			30music		
	P@20	MRR@20	Sim	P@20	MRR@20	Sim	P@20	MRR@20	Sim
MGS-NS	39.38	17.60	227.36	53.96	18.68	80.65	45.60	23.66	697.79
MGS	**42.12**	**18.62**	**178.86**	**55.05**	**19.13**	**42.18**	**46.46**	**28.01**	**393.94**



(a) (b)

Figure 4: (a) Impact of the sample number of attribute-same neighbor items. (b) Impact of the temperature.

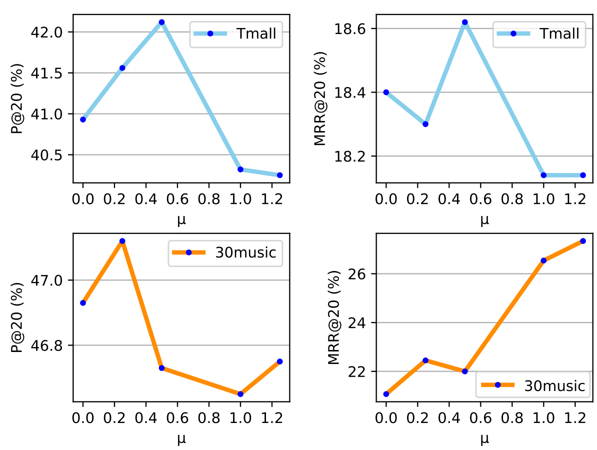


Figure 5: The impact of the last clicked item.

we exclude the last click in terms of P@20, MRR@20 is a more significant indicator that reveals the model's ranking ability, and it increases significantly after incorporating the last clicked item. These results suggest that the gating mechanism with a handcrafted coefficient is practical for different real-world scenarios.

6 CONCLUSION AND FUTURE WORK

The session-based recommendation is a challenging task that suffers from minimal, short, and sparse session data, leading to sub-optimal model performance. This paper proposes a novel session-based recommendation model, known as MGS, to utilize attribute information that inherently existed in most real-world scenarios and is generally ignored by previous studies to profoundly discover user preferences. Additionally, we invent a novel self-supervised learning method to optimize the mirror graph generation process and initial item embedding vectors. In our model, item attribute information and the SSL technic both contribute consistently to learning more comprehensive session representations. Moreover, we technically implement the most recent user preferences into session presentations by a gating network for adapting to scenarios like music broadcast platforms, offering more accurate recommendation performance. Experiments comparing with different baselines and ablation studies exhibit its effectiveness and superiority.

In the future, first and foremost, we plan to challenge datasets that contain more various attribute information, which may also consist of noise that would undermine the model's performance. On the other hand, we would like to take textual reviews that include more fine-grained semantic signals into our consideration.

ACKNOWLEDGMENTS

Fan Zhang is the corresponding author. This work was partially supported by National Natural Science Foundation of China (No. 61872278); and Young Top-notch Talent Cultivation Program of Hubei Province.

REFERENCES

[1] Chen Chen, Jie Guo, and Bin Song. 2021. Dual Attention Transfer in Session-based Recommendation with Multi-dimensional Integration. In *Proceedings of*

the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 869–878.

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

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*. PMLR, 4116–4126.

[5] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[6] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[7] Nikos Komodakis and Spyros Gidaris. 2018. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*.

[8] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[9] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2015. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493* (2015).

[10] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1831–1839.

[11] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[13] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. 2016. Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328* (2016).

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

[15] Zhiqiang Pan, Fei Cai, Wanyu Chen, Honghui Chen, and Maarten de Rijke. 2020. Star graph neural networks for session-based recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1195–1204.

[16] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the item order in session-based recommendation with graph neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 579–588.

[17] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[18] Guy Shani, David Heckerman, Ronen I Brafman, and Craig Boutilier. 2005. An MDP-based recommender system. *Journal of Machine Learning Research* 6, 9 (2005).

[19] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 17–22.

[20] Roberto Turrin, Massimo Quadrana, Andrea Condorelli, Roberto Pagano, and Paolo Cremonesi. 2015. 30Music Listening and Playlists Dataset.. In *RecSys Posters*.

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

[22] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. *ICLR (Poster)* 2, 3 (2019), 4.

[23] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. 2020. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 169–178.

[24] 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.

[25] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 346–353.

[26] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[27] Xin Xia, Hongzhi Yin, Junliang Yu, Yingxia Shao, and Lizhen Cui. 2021. Self-Supervised Graph Co-Training for Session-based Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2180–2190.

[28] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2020. Self-supervised hypergraph convolutional networks for session-based recommendation. *arXiv preprint arXiv:2012.06852* (2020).

[29] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation.. In *IJCAI*, Vol. 19. 3940–3946.

[30] Hongzhi Yin and Bin Cui. 2016. *Spatio-temporal recommendation in social media*. Springer.

[31] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-Aware Self-Supervised Tri-Training for Recommendation. *arXiv preprint arXiv:2106.03569* (2021).

[32] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1893–1902.

[33] Andrew Zimdars, David Maxwell Chickering, and Christopher Meek. 2013. Using temporal data for making recommendations. *arXiv preprint arXiv:1301.2320* (2013).