

# Multi-level Contrastive Learning Framework for Sequential Recommendation

Ziyang Wang\*  
CCIIP Laboratory, Huazhong  
University of Science and Technology  
Alibaba Group  
China  
ziyang1997@hust.edu.cn

Huoyu Liu\*  
Alibaba Group  
China  
huoyu.lhy@alibaba-inc.com

Wei Wei†  
CCIIP Laboratory, Huazhong  
University of Science and Technology  
Joint Laboratory of HUST and Pingan  
Property & Casualty Research (HPL)  
China  
weiw@hust.edu.cn

Yue Hu  
Alibaba Group  
China  
lingshu.hy@alibaba-inc.com

Xian-Ling Mao  
Beijing Institute of Technology  
China  
maoxl@bit.edu.cn

Shaojian He  
Alibaba Group  
China  
shaojian.he@alibaba-inc.com

Rui Fang  
Ping An Property & Casualty  
Insurance company of China, Ltd  
China  
fangrui051@pingan.com.cn

Dangyang Chen  
Ping An Property & Casualty  
Insurance company of China, Ltd  
China  
chendangyang273@pingan.com.cn

## ABSTRACT

Sequential recommendation (SR) aims to predict the subsequent behaviors of users by understanding their successive historical behaviors. Recently, some methods for SR are devoted to alleviating the data sparsity problem (*i.e.*, limited supervised signals for training), which take account of contrastive learning to incorporate self-supervised signals into SR. Despite their achievements, it is far from enough to learn informative user/item embeddings due to the inadequacy modeling of complex collaborative information and co-action information, such as user-item relation, user-user relation, and item-item relation. In this paper, we study the problem of SR and propose a novel multi-level contrastive learning framework for sequential recommendation, named MCLSR. Different from the previous contrastive learning-based methods for SR, MCLSR learns the representations of users and items through a cross-view contrastive learning paradigm from four specific views at two different levels (*i.e.*, interest- and feature-level). Specifically, the interest-level contrastive mechanism jointly learns the collaborative information with the sequential transition patterns, and the feature-level contrastive mechanism re-observes the relation between users and items

via capturing the co-action information (*i.e.*, co-occurrence). Extensive experiments on four real-world datasets show that the proposed MCLSR outperforms the state-of-the-art methods consistently.

## CCS CONCEPTS

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

## KEYWORDS

Sequential Recommendation; Contrastive Learning; Graph Neural Networks; Collaborative Information

## ACM Reference Format:

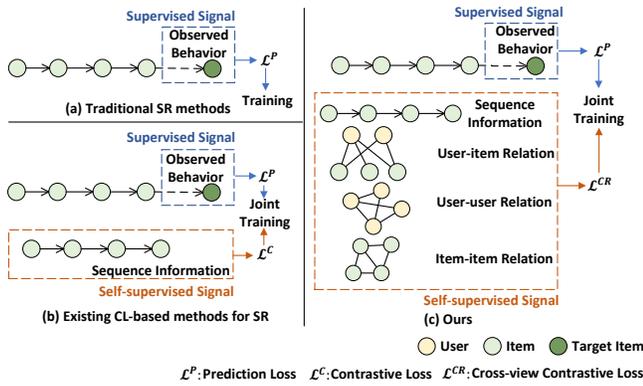
Ziyang Wang, Huoyu Liu, Wei Wei, Yue Hu, Xian-Ling Mao, Shaojian He, Rui Fang, and Dangyang Chen. 2022. Multi-level Contrastive Learning Framework for Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22)*, October 17–21, 2022, Atlanta, GA, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3511808.3557404>

## 1 INTRODUCTION

Recommendation systems play critical roles in many online services such as E-commerce, video streaming, and music platform due to their success in alleviating the information overload problem. Among these applications, sequential recommendation (SR) pays attention to the chronological order of users' behaviors and has become a paradigmatic task in recent years. Given a user behavior history, SR captures the sequential transition patterns among successive items and predicts the next item that the user might be interested in, consistent with many real-world recommendation situations.

The study of the sequential recommendation system is of significant importance and thus has received considerable research interest in recent years. For instance, there exist several works to treat SR as a sequence modeling task, such as GRU4Rec [12], which adopts

\*Both authors contributed equally to this research.  
†Corresponding Author.



**Figure 1: Illustration of training mechanisms of different methods for SR. (a) Traditional methods for SR where the supervised signals are entirely based on the observed user behaviors. (b) Recently contrastive learning-based methods for SR learn the self-supervised signals from the sequence itself. (c) Our proposed method learns rich self-supervised signals by performing cross-view contrastive learning on sequence information, user-item, user-user, and item-item relations.**

recurrent neural networks (RNNs) to model the sequential behaviors of users. Then, SASRec [15] uses the self-attention mechanism to capture high-order dynamics from user behavior sequences. Further, graph-based methods [3, 27, 35] convert each sequence into a graph and model the complex item transitions via graph neural networks (GNNs). However, most of these methods are under a supervised learning paradigm and may suffer from the data sparsity problem since their supervision signal is entirely from the observed user behaviors (shown in Fig 1a), which are highly sparse compared to the entire interaction space [37].

Recently, self-supervised learning (SSL) [19] is proposed to mine the supervised signals from the data itself, which shows promising potential to alleviate the data sparsity problem. As a typical self-supervised learning technique, contrastive learning (CL) has gained increasing attention. By extracting the positive and negative samples from the data, contrastive learning aims to maximize the agreement of positive pairs while minimizing the agreement between negative samples. In this way, it can learn discriminative embeddings without explicit extra labels [34]. Based on the principle of contrastive learning, existing CL-based SR methods apply data-level augmentation ( $S^3$ -Rec [52] and CL4SRec [41]) or model-level augmentation (DuoRec [26]) on user behavior sequence to generate positive and negative pairs, and learn the extra self-supervised signals by contrasting the corresponding pairs.

Despite such achievement, the above methods obtain the self-supervised signals entirely from the sequence itself (shown in Fig 1b), which is insufficient for SR for two reasons. *First*, since each behavior sequence contains a limited number of items, the self-supervised information obtained from the sequence is inadequate. *Second*,  $S^3$ -Rec and CL4SRec generate contrastive pairs by performing simply data augmentation (e.g., item cropping and masking) on behavior sequences, resulting in less information diversity of contrastive pairs, thus the obtained self-supervised signals would

be too weak to learn informative embedding [34]. Due to the insufficiency of directly exploiting contrastive learning on sequential views (i.e. user behavior sequences), it motivates us to explore more views and generate more informative pairs for contrastive learning. However, it is non-trivial to define an appropriate contrastive learning framework with more contrasting views for SR, which requires us to address the following fundamental issues: (1) *How to select proper views for contrastive learning*: As mentioned above, more views are desired for the contrastive learning of SR. An essential requirement is that the selected views should be informative and can reflect user preferences. In fact, collaborative information [33] (e.g., user-item relation) and co-action information [9] (e.g., user-user relation and item-item relation) are two significant factors for user preference learning, which show strong potential to help obtain rich self-supervised signals and should be carefully considered. (2) *How to set a proper contrastive task*: Proper design of contrastive tasks is critical for contrastive learning [34]. In general, similar contrastive views would make the self-supervised signals too weak to learn informative embeddings. Therefore, it is important to ensure a clear diversity of information between the contrasting views.

In light of the aforementioned limitations and challenges, in this paper, we propose a **multi-level contrastive learning framework for sequential recommendation (MCLSR)**. To effectively learn the self-supervised signals, except for the sequential view, we construct three graph views and adopt a multi-level cross-view contrastive mechanism to learn collaborative information, co-action information and sequential transitions (shown in Fig 1c). Specifically, four views of SR are given firstly, i.e., sequential view, user-item view, user-user view, and item-item view. Then MCLSR performs a cross-view contrastive learning paradigm on two levels (i.e., interest- and feature-level). At the interest-level, MCLSR obtains the sequential transition patterns from the sequential view and the collaborative information from the user-item view, where the contrastive mechanism is performed to capture the complementary information between the two views. At the feature-level, MCLSR re-observed the relation between users and items via performing GNNs on the user-user view and item-item view. By applying contrastive learning to learn discriminative information on two views, MCLSR can capture the self-supervised signals from the co-action information between users (items) to further enhance representation learning.

To summarize, this work makes the following main contributions:

- We exploit contrastive learning with collaborative information and co-action information to alleviate the data sparsity problem in studying the sequential recommendation task. Towards this end, we propose a novel recommendation framework that captures sequential transition patterns, collaborative signals and co-action signals from four specific views.
- The proposed MCLSR performs cross-view contrastive learning at interest- and feature-level. The former learns self-supervised signals from collaborative information and sequential transition patterns, and the latter captures the co-action information to learn informative user/item embeddings.
- We conduct extensive experiments on four real-world datasets, and the results demonstrate the superiority of MCLSR and the effectiveness of each key component.

## 2 RELATED WORK

In this section, we will briefly review several lines of work closely related to ours, including sequential recommendation and contrastive learning.

### 2.1 Sequential Recommendation

Compared with session-based recommendation [38], sequential recommendation usually considers user ID and their behavior sequence in a longer time period. Most of the early attempts for SRS are based on Markov Chain, which infers a user’s next action based on the previous one. For example, FPMC [28] captures the sequential patterns by first-order Markov Chain, which is then extend to higher order Markov Chain [10]. To capture long-term and multi-level cascading dependencies, deep learning techniques are introduced into SRS. For instance, RNN-based methods [12, 36, 43] regard SRS as a sequential modeling problem and apply recurrent neural networks to capture the sequential transition patterns. Further, CNN-based methods [30, 48] treat each sequence as an image and adopt convolution networks to model the union-level sequential patterns. Then some advanced techniques are incorporated into SRS, such as self-attention network [15, 17, 20, 29, 32], memory network [1, 5, 13, 50], capsule network [2, 16] and graph neural networks [3, 21, 42, 49]. Typically, SASRec [15] stacks multi-head self-attention blocks to learn dynamic item transition patterns. MIND [16] leverages dynamic routing to obtain multiple interests of users. MA-GNN [21] proposes a memory augmented graph neural network to capture both items’ short-term contextual information and long-range dependencies for sequential recommendation. However, the above methods mainly focus on the modeling of sequential transition in a supervised paradigm, where the supervised signals are entirely based on the observed user behaviors. Due to the limited observed user behaviors, the above methods face the problem of data sparsity. In this paper, we mainly focus on employing a multi-level cross-view contrastive learning paradigm to alleviate the data sparsity problem.

### 2.2 Contrastive Learning

The main idea of contrastive learning is to learn informative representations by contrasting positive pairs against negative pairs, which shows impressive achievement in visual representation learning [4], natural language process [44, 53], and graph neural networks [7, 14, 45].

Recently, some studies are proposed to introduce contrastive learning into recommendation system [22, 24, 25, 39, 40, 46, 47, 51, 54]. For instance, SGL [37] provides an auxiliary signal for existing GCN-based recommendation models by taking node self-discrimination as the self-supervised task. SEPT [46] designs a socially aware self-supervised framework for learning discrimination signals from the user-item graph and social graph. Some efforts also introduce contrastive learning into sequential recommendation [6, 26, 52]. S<sup>3</sup>-Rec [52] devises four auxiliary self-supervised objectives for data representation learning by using the mutual information maximization. CL4SRec [41] applies three data augmentation (*i.e.*, crop, mask and reorder) to generate positive pairs, and contrasts positive pairs to learn robust sequential transition patterns. DuoRec [26] proposes a dropout-based model-level augmentation model with a supervised positive sampling strategy to capture the self-supervised

signal from the sequence. Despite the achievement, the above contrastive learning-based methods for SR mainly focus on learning the self-supervised signals from each sequence. However, due to the limited information within the sequence, the obtained self-supervised signal will be too weak to learn informative embedding.

## 3 PRELIMINARY

In this section, we first formulate the problem of sequential recommendation, then we introduce the construction process of three graph views and the architecture of the graph encoder layer.

### 3.1 Problem Formulation

Assume we have a set of users  $u \in \mathcal{U}$  and a set of items  $v \in \mathcal{V}$ . For each user,  $S^{(u)} = \{v_1^{(u)}, v_2^{(u)}, \dots, v_{|S|}^{(u)}\}$  denotes the sequence of user historical behaviors in chronological order, where  $v_j^{(u)}$  denotes the  $j^{th}$  item interacted by the user. Given an observed sequence  $S^{(u)}$ , the typical task of sequential recommendation is to predict the next items that the user  $u$  is most likely to be interacted with.

### 3.2 Graph Construction

An item can be involved in multiple user behavior sequences, from where we can obtain useful collaborative information [33] and co-action information [9]. Thus extra graph views are constructed here to explore the collaborative signals and co-action signals for SRS. Based on the users’ historical behavior sequences, we first obtain a user-item interaction matrix  $\mathcal{M}^{uv} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{V}|}$ , where  $\mathcal{M}_{ij}^{uv} > 0$  denotes that item  $j$  is appeared in user  $i$ ’s behavior sequence  $S^i$  (*i.e.*,  $v_j \in S^i$ ) and 0, otherwise.

**User-item graph.** The user-item graph is a typical bipartite graph, which is constructed by aggregating cross-user behavior sequences. Let  $\mathcal{G}^{uv} = (\mathcal{V}^{uv}, \mathcal{E}^{uv})$  be the user-item graph, where  $\mathcal{V}^{uv}$  denotes the node set of graph  $\mathcal{G}^{uv}$  that contains all users in  $\mathcal{U}$  and all items in  $\mathcal{V}$ , and  $\mathcal{E}^{uv} = \{e_{ij}^{uv} = \mathcal{M}_{ij}^{uv} | \mathcal{M}_{ij}^{uv} > 0\}$  indicates the edge set of graph  $\mathcal{G}^{uv}$  that contains user-item interactions, where the weight of edge  $e_{ij}^{uv}$  represents the number of user  $i$  interacts with item  $j$ .

**User-user/item-item graph.** The user-user (item-item) graph is constructed to explore the co-action signals between users (items). Based on the interaction matrix  $\mathcal{M}^{uv}$ , we can obtain a user-user matrix<sup>1</sup>  $\mathcal{M}^{uu} = (\mathcal{M}^{uv})(\mathcal{M}^{uv})^T$ . Let  $\mathcal{G}^{uu} = (\mathcal{V}^{uu}, \mathcal{E}^{uu})$  be the user-user graph, where  $\mathcal{V}^{uu}$  denotes the graph node set that contains all users in  $\mathcal{U}$ , and  $\mathcal{E}^{uu} = \{e_{ij}^{uu} = \mathcal{M}_{ij}^{uu} | \mathcal{M}_{ij}^{uu} > 0\}$  indicates graph edge set that contains co-action information, where the weight of each edge denotes the number of co-action behaviors between user  $i$  and user  $j$ .

### 3.3 Graph Encoder Layer

To fully exploit the collaborative information and co-action information from the graphs, a specific graph encoder layer is employed here to extract the node features. Due to the effectiveness and lightweight architecture of LightGCN [11], we employ its message propagation

<sup>1</sup>Here we present how to construct the user-user graph  $\mathcal{G}^{uu}$ , and the item-item graph  $\mathcal{G}^{vv}$  can be constructed similarly.

strategy to encode the node features:

$$\mathbf{X}^{(l)} = \text{GraphEncoder}(\mathbf{X}, \mathbf{A}) = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \mathbf{X}^{(l-1)}, \quad (1)$$

where  $\mathbf{A}$  indicates the adjacency matrix of the graph and  $\mathbf{D}_{ii} = \sum_{j=0} \mathbf{A}_{ij}$  denotes the corresponding diagonal degree matrix. ( $l$ ) indicates the depth of graph encoder layers,  $\mathbf{X}^{(0)}$  indicates the input node features and  $\mathbf{X}^{(l)}$  is the output of the graph encoder layer.

## 4 METHOD

The overview of the proposed multi-level contrastive framework is presented in Figure 2, which comprises four main components: 1) *Graph construction layer*. It constructs user-item, user-user and item-item graphs via aggregating the user behavior sequences; 2) *Interest-level contrastive learning layer*. It first learns the current interest of the user from user behavior sequences and the general interest of the user from the user-item graph, then a cross-view contrastive mechanism is performed. 3) *Feature-level contrastive learning layer*. It obtains the user and item features from the user-item, user-user and item-item graphs, and then it performs cross-view contrastive learning; 4) *Joint training*. It jointly optimizes the prediction loss, interest- and feature-level contrastive loss to update the model parameters. In the following sections, we will present the technical details of MCLSR.

Here, we first construct a user embedding matrix  $\mathbf{H}^u \in \mathcal{R}^{|\mathcal{U}| \times d}$  and an item embedding matrix  $\mathbf{H}^v \in \mathcal{R}^{|\mathcal{V}| \times d}$ , where  $d$  is the dimension of the embedding. The input of our model is the user behavior sequence  $\mathbf{S}^{(u)} = \{v_1^{(u)}, v_2^{(u)}, \dots, v_{|\mathcal{S}|}^{(u)}\}$ , which is fed into a specific embedding layer and transformed to the user embedding  $\mathbf{h}^u \in \mathbb{R}^d$  and the corresponding item embedding matrix  $\mathbf{E}^u = [\mathbf{h}_1^v, \mathbf{h}_2^v, \dots, \mathbf{h}_n^v]$ .

### 4.1 Interest-level Contrastive Learning

Different from previous SR studies [5] that mainly focus on the transition patterns on the current sequence, we aim to introduce collaborative information into SR to capture the general preferences of users. Then by applying contrastive mechanism, the extra self-supervised signals from the complementary information between sequential transition patterns and collaborative information is learned to alleviate the data sparsity problem.

**Current interest learning.** This subsection aims to capture the users' preferences from the user behavior sequences (*i.e.*, sequential view). Since different items have distinct importance for current prediction, a self-attention mechanism [18] is applied to model the user behavior sequences. Given the item embedding matrix  $\mathbf{E}^u$ , we first use a trainable position matrix to incorporate the sequential order information into sequence, *i.e.*,  $\mathbf{E}^{u,p} = [\mathbf{h}_1^v + \mathbf{p}_1, \mathbf{h}_2^v + \mathbf{p}_2, \dots, \mathbf{h}_n^v + \mathbf{p}_n]$ . Then an attention matrix  $\mathbf{A}^s$  is computed as follows:

$$\mathbf{A}^s = \text{softmax} \left( \mathbf{W}_2 \tanh(\mathbf{W}_1 (\mathbf{E}^{u,p})^T) \right), \quad (2)$$

where  $\mathbf{W}_1 \in \mathbb{R}^{4d \times d}$  and  $\mathbf{W}_2 \in \mathbb{R}^{4d}$  are trainable parameters. The dimension of the output matrix  $\mathbf{A}^s$  is  $\mathbb{R}^n$ , where each element  $\mathbf{A}_j^s$  denotes the affinity between the user preference and  $j^{th}$  item in the user behavior sequence. Finally, the user preferences from the sequential view (named current interest) can be obtained by:

$$\mathbf{I}_u^s = \mathbf{A}^s \mathbf{E}^u. \quad (3)$$

**General interest learning.** To fully explore the collaborative information, here we frame the user interests from the cross-user interaction information in the user-item graph  $\mathcal{G}^{uv}$ . To obtain the user features and item features, a graph encoder layer is employed as follows<sup>2</sup>:

$$\mathbf{H}^{all,uv} = \text{GraphEncoder}^{(l)}(\mathbf{H}^{all}, \mathcal{G}^{uv}), \quad (4)$$

where  $\mathbf{H}^{all} \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{V}|) \times d}$  is the initial node feature matrix that contains user and item features (*i.e.*,  $\mathbf{H}^{all} = [\mathbf{H}^u || \mathbf{H}^v]$ , where  $||$  denotes concatenation operation), and GraphEncoder indicates the graph encoder layer that defined in Equation (1).  $\mathbf{H}^{all,uv} \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{V}|) \times d}$  is the learned node feature matrix from user-item graph.

Then for a given user  $u$  and corresponding behavior sequence  $\mathbf{S}^u$ , we can obtain the corresponding user embedding  $\mathbf{h}^{u,uv} \in \mathbb{R}^d$  and item embedding matrix  $\mathbf{E}^{u,uv} \in \mathbb{R}^{n \times d}$  by index selection from the learned node feature matrix  $\mathbf{H}^{all,uv} \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{V}|) \times d}$  according to the indices of user and items. To estimate the importance of each item based on the user preference, an attention matrix is computed based on the user features and item features::

$$\mathbf{A}^c = \text{softmax} \left( \tanh(\mathbf{W}_3 \mathbf{h}^{u,uv}) (\mathbf{E}^{u,uv})^T \right), \quad (5)$$

where  $\mathbf{W}_3 \in \mathbb{R}^{d \times d}$  is a trainable transform weight,  $\mathbf{A}^c \in \mathbb{R}^n$  is the attention matrix between user preference and items. Then the user preferences from the user-item view (named general interest) can be obtained as follows:

$$\mathbf{I}_u^c = \mathbf{A}^c \mathbf{E}^{u,uv}. \quad (6)$$

**Cross-view contrastive learning.** To learn the complementary information from sequential transition patterns and collaborative information, it is meaningful to perform the contrastive learning on the the sequential view (current preference  $\mathbf{I}_u^s$ ) and user-item view (general preference  $\mathbf{I}_u^c$ ). Here, we first feed  $\mathbf{I}_u^s$  and  $\mathbf{I}_u^c$  into a multi-layer perceptron (MLP) to project them into the space where contrastive loss is calculated:

$$\begin{aligned} \mathbf{T}^{l,s} &= \left( \mathbf{W}_2^p \sigma(\mathbf{W}_1^p \mathbf{I}_u^s + \mathbf{b}_1^p) + \mathbf{b}_2^p \right), \\ \mathbf{T}^{l,c} &= \left( \mathbf{W}_2^p \sigma(\mathbf{W}_1^p \mathbf{I}_u^c + \mathbf{b}_1^p) + \mathbf{b}_2^p \right), \end{aligned} \quad (7)$$

where  $\mathbf{W}_*^p \in \mathbb{R}^{d \times d}$ ,  $\mathbf{b}_*^p \in \mathbb{R}^d$  are trainable parameters and  $\sigma$  denotes ELU non-linear activation function.

To learn the self-supervised signal from two views, it is essential to define the positive and negative samples for the interest-level contrastive mechanism. Inspired by the work of contrastive learning in graph neural networks [34], we take the interests of the same user from two views (*i.e.*, sequential view and user-item view) as a pair of positive samples. Moreover, we naturally treat the interests of different users as pairs of negative samples. Then the interest-level contrastive loss can be computed as follows:

$$\mathcal{L}^{ll} = \sum_{i=1} -\log \frac{\Psi(\mathbf{T}_i^{l,s}, \mathbf{T}_i^{l,c})}{\sum_j \Psi(\mathbf{T}_i^{l,s}, \mathbf{T}_j^{l,c}) + \sum_{j \neq i} \Psi(\mathbf{T}_i^{l,s}, \mathbf{T}_j^{l,s})}, \quad (8)$$

where  $\Psi$  denotes  $\exp(\text{sim}(\cdot, \cdot) / \tau)$ ,  $\text{sim}(\cdot, \cdot)$  denotes the cosine similarity function and  $\tau$  is a temperature parameter.

<sup>2</sup>In the equation, we use  $\mathcal{G}^*$  to represent the adjacency matrix of the graph for ease of reading.

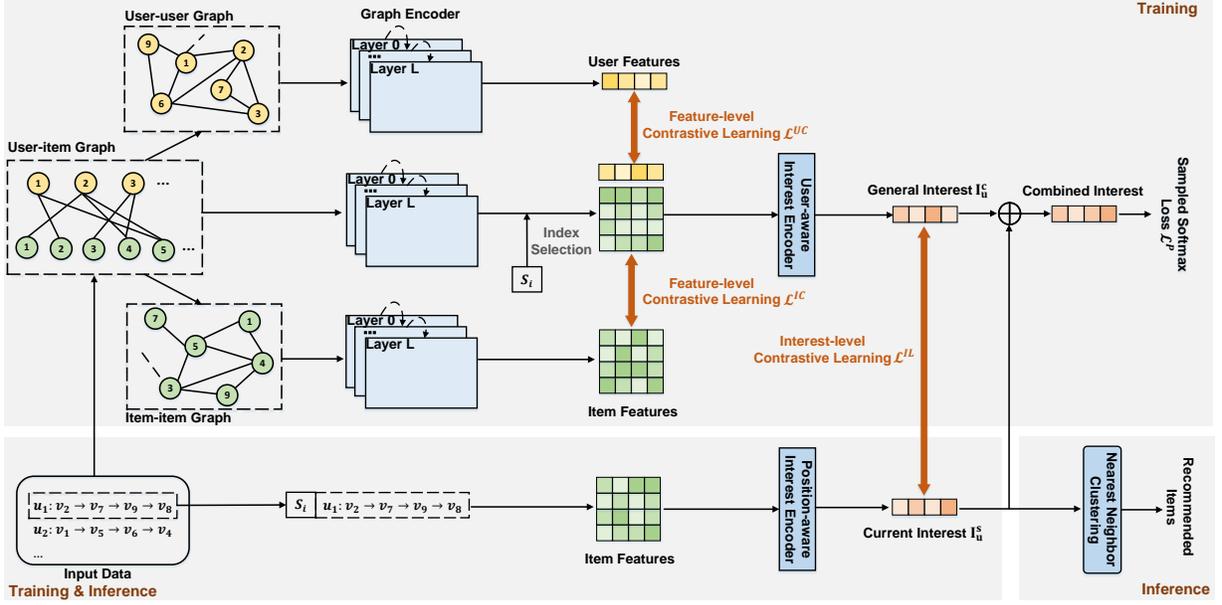


Figure 2: An overview of the proposed framework.  $\oplus$  denotes the element-wise summation.

## 4.2 Feature-level Contrastive Learning

Directly exploring the user-item graph is insufficient to capture the co-action information between users (items). In fact, the co-action information [9] is essential for measuring user-user (item-item) relationships and learning the user preferences. Thus a user-user (item-item) graph is constructed to effectively capture the co-action signals between users (items). For each user<sup>3</sup>, we learn the user features from both the user-item view and user-user view, where the contrastive mechanism is performed to learn self-supervised signals by capturing the discriminative information from two graph views and complement each other.

**Feature learning.** To obtain the collaborative information and co-action information, we first extract the user features from user-item view and user-user view, where a graph encoder layer is applied:

$$\begin{aligned} [\mathbf{H}^{u,uv} || \mathbf{H}^{u,uu}] &= \text{GraphEncoder}^{(l)}([\mathbf{H}^u || \mathbf{H}^v], \mathcal{G}^{uv}), \\ \mathbf{H}^{u,uu} &= \text{GraphEncoder}^{(l)}(\mathbf{H}^u, \mathcal{G}^{uu}), \end{aligned} \quad (9)$$

where  $\mathbf{H}^{u,uv}, \mathbf{H}^{u,uu}$  denotes the user features obtained from user-item graph  $\mathcal{G}^{uv}$  and user-user graph  $\mathcal{G}^{uu}$ , respectively. Note that the weight of edge in  $\mathcal{G}^{uu}$  denotes the number of co-action, which means high co-action pairs show a more critical influence during graph propagation.

**Cross-view contrastive learning.** Then the obtained user features from two graphs are fed into an MLP and projected into the space where contrastive loss is calculated:

$$\begin{aligned} \mathbf{T}^{F,uu} &= \mathbf{W}_4^p \sigma(\mathbf{W}_3^p \mathbf{H}^{u,uu} + \mathbf{b}_3^p) + \mathbf{b}_4^p, \\ \mathbf{T}^{F,uv} &= \mathbf{W}_4^p \sigma(\mathbf{W}_3^p \mathbf{H}^{u,uv} + \mathbf{b}_3^p) + \mathbf{b}_4^p, \end{aligned} \quad (10)$$

where  $\mathbf{W}_*^p \in \mathbb{R}^{d \times d}$ ,  $\mathbf{b}_*^p \in \mathbb{R}^d$  are trainable parameters.

<sup>3</sup>Here we show the *feature-level contrastive learning* for user features, and the process for item features is similarly.

Considering that each user is involved in two graph views, where we can capture the user-item collaborative information and user-user co-action information, respectively. To capture the complementary information between two graph views and obtain discriminative user features, we naturally treat the features of the same user obtained in two graph views as a pair of positive sample while the features of different users as pairs of negative samples:

$$\mathcal{L}^{UC} = \sum_{i=1} -\log \frac{\Psi(\mathbf{T}_i^{F,uv}, \mathbf{T}_i^{F,uu})}{\sum_j \Psi(\mathbf{T}_i^{F,uv}, \mathbf{T}_j^{F,uu}) + \sum_{j \neq i} \Psi(\mathbf{T}_i^{F,uv}, \mathbf{T}_j^{F,uv})}, \quad (11)$$

where  $\mathcal{L}^{UC}$  denotes the contrastive loss for user features and the contrastive loss for item features  $\mathcal{L}^{IC}$  can be calculated in a similar way. The final feature-level contrastive loss  $\mathcal{L}^{FL}$  is computed as follows:

$$\mathcal{L}^{FL} = \mathcal{L}^{UC} + \mathcal{L}^{IC}. \quad (12)$$

## 4.3 Training and Inference

**Training phase.** After computing the user interest representations from sequential view and user-item view, we sum them up to obtain the combined user interest representations:

$$\mathbf{I}_u^{comb} = \alpha \mathbf{I}_u^s + (1 - \alpha) \mathbf{I}_u^c, \quad (13)$$

where  $\alpha$  is a trade-off hyper-parameter. Given a training sample  $(u, o)$  with the user interest embedding  $\mathbf{I}_u^{comb}$  and target embedding  $\mathbf{h}_o^v$ , the likelihood of the user  $u$  interacting with the item  $o$  can be computed by sampled softmax method. Furthermore, the objective function for prediction is to minimize the following negative log-likelihood:

$$\mathcal{L}^p = \sum_{u \in U} -\log \frac{\exp((\mathbf{I}_u^{comb})^T \mathbf{h}_o^v)}{\sum_{k \in \text{Sample}(\mathcal{V})} \exp((\mathbf{I}_u^{comb})^T \mathbf{h}_k^v)}. \quad (14)$$

**Table 1: Statistics of the used datasets.**

Dataset	# user	# item	# interactions	Avg. len.	Sparsity
Books	459,133	313,966	8,898,041	9.7	99.993%
Clothing	39,387	23,034	278,677	6.9	99.969%
Toys	75,258	64,444	1,097,592	9.6	99.977%
Gowalla	65,506	174,606	2,061,264	14.5	99.982%

The overall objective is given as follows:

$$\mathcal{J}(\theta) = \mathcal{L}^p + \beta \mathcal{L}^{IL} + \gamma \mathcal{L}^{FL}, \quad (15)$$

while  $\beta$  and  $\gamma$  are trade-off hyper-parameters. Noted that, we jointly optimize the three throughout the training.

**Inference phase.** For the inference phase, we use the current interest  $\mathbf{I}_u^s$  to perform downstream tasks because: i) To avoid the problem of information leakage, we only use the training data to construct three graphs during training and inference, so the general interest of users cannot be generated during inference. ii) After optimizing  $\mathcal{J}(\theta)$ , the collaborative information and co-action information are learned in the user and item embeddings, thus it is enough to use the current interest  $\mathbf{I}_u^s$  to perform downstream tasks. Then the candidate items are clustered based on the inner product:

$$R(u, N) = \text{Top-}N_{v \in V} \left( (\mathbf{I}_u^s)^T \mathbf{h}_v \right), \quad (16)$$

where  $R(u, N)$  denotes the top- $N$  items to be recommended.

## 5 EXPERIMENT

### 5.1 Experimental Settings

**Datasets.** We conduct experiments on four public datasets.

- **Amazon**<sup>4</sup> consists of product reviews and metadata from Amazon.com [23], and in this study we choose three representative categories: **Books**, **Clothing** and **Toys**.
- **Gowalla**<sup>5</sup> is a widely used check-in dataset which is from a well-known location-based social networking website.

Following [2], we remove the items that appear less than five times, and the max length of each training sample is set to 20. The users of each dataset are split into training, validation, and test sets by the proportion of 8:1:1. The model is trained on the entire click sequences of training users. During the training phase, we incorporate a commonly used set of training sequential recommendation models. In detail, we view each item in the user interaction sequence as a potential target item, where the behaviors happen before the target item is used to generate the users' interest representation. During the inference phase, we choose to generate the users' interest representation from the first 80% of user behaviors and compute the evaluation metric by predicting the remaining 20% of user behaviors by following [2]. The statistics of datasets, after preprocessing, are shown in Table 1.

**Baselines.** To fully evaluate the performance of our method for SR, we compare our method with classic methods as well as state-of-the-art methods.

- **Pop** directly recommends top- $N$  popular items in the training data during inference.
- **GRU4Rec** [12] is the first work that applies recurrent neural network for SR.
- **SASRec** [15] stacks several multi-head self-attention blocks to capture the sequential transition patterns.
- **ComiRec-SA** [2] proposes a multi-interest framework for SR by employing a multi-head self-attention network, where different heads correspond to different interests of users.
- **GCSAN** [42] combines the graph neural network and self-attention mechanism to learn both short- and long-term dependencies between items.
- **S<sup>3</sup>-RecMIP** [52] utilizing the mutual information maximization (MIM) principle to extract the self-supervised signals from the item transitions.
- **CL4SRec** [41] utilizes contrastive mechanism with data augmentation to learn discriminative information.
- **DuoRec** [26] proposes a model-level augmentation method with a positive sampling strategy to capture the self-supervised signal from the user behavior sequences.

**Evaluation metrics.** Following previous work [2, 41], we adopt three widely used ranking-based metrics for sequential recommendation: Recall@N, NDCG@N, and Hit@N. Recall@N denotes the proportion of ground truth items included in the top- $N$  recommended list, NDCG@N measures the positions of recommended items and evaluates the ranking quality of the recommended list, and Hit@N denotes the percentage that the top- $N$  recommended list contains at least one ground truth item.

**Implementation details.** For a fair comparison, all methods are optimized with Adam optimizer with a learning rate of 0.001. The embedding size is set to 64, and the mini-batch size is set to 128. The number of negative samples for sampled softmax loss is set to 1280. For baselines with Transformer blocks (*e.g.*, SASRec, CL4SRec, and DuoRec), we select the number of Transformer blocks in  $\{1, 2, 3\}$  and select the dropout ratio in  $\{0.1, 0.2, \dots, 0.9\}$  in the validation set. For other parameters of baseline methods, we follow the settings given by the original papers if they had provided and otherwise we perform a grid search in the validation set. For our method, the depth of GNN layer is set to 2 selected from  $\{0, 1, 2, 3\}$  in the validation set, and the temperature  $\tau$  is set to 0.5. The trade-off parameters  $\{\alpha, \beta, \gamma\}$  are set to  $\{0.5, 1, 0.05\}$  for all datasets. To decrease the noise and reduce the computational complexity, the neighborhood number of each node on the user-user and item-item graph is set to 50 by filtering edges with small weights.

### 5.2 Performance Comparison

The experimental results of our method with state-of-the-art baselines are listed in Table 2, from where we have the following key findings:

- The performance of POP is the worst since it directly uses rudimentary statistical methods to recommend the most frequently occurring items in the training data, which fails to learn the preference of users. GRU4Rec outperforms POP on four datasets,

<sup>4</sup><http://jmcauley.ucsd.edu/data/amazon/>

<sup>5</sup><https://snap.stanford.edu/data/loc-gowalla.html>

**Table 2: Effectiveness comparison between MCLSR and state-of-the-art approaches. † denotes the performance improvement over the best baseline is statistically significant with p-value < 0.01.**

Datasets	Metric	POPRec	GRU4Rec	SASRec	ComiRec-SA	GCSAN	S <sup>3</sup> -Rec <sub>MIP</sub>	CL4SRec	DuoRec	MCLSR	Improv.
Books	Recall@20	1.368	3.787	6.274	5.489	5.721	6.336	6.544	<u>6.838</u>	<b>7.469</b> <sup>†</sup>	9.2%
	NDCG@20	0.597	1.923	2.825	2.262	2.706	2.964	3.161	<u>3.257</u>	<b>3.479</b> <sup>†</sup>	6.8%
	Hit@20	3.013	8.710	12.765	11.402	11.730	13.052	13.520	<u>14.173</u>	<b>15.542</b> <sup>†</sup>	9.6%
	Recall@50	2.400	6.335	9.349	8.467	8.455	9.684	10.240	<u>10.826</u>	<b>11.583</b> <sup>†</sup>	6.9%
	NDCG@50	0.826	2.600	3.627	3.082	3.434	3.894	4.113	<u>4.308</u>	<b>4.647</b> <sup>†</sup>	7.9%
	Hit@50	5.219	13.597	18.547	17.202	16.865	19.142	20.170	<u>21.366</u>	<b>23.042</b> <sup>†</sup>	7.8%
Clothing	Recall@20	1.200	1.623	2.646	1.678	2.242	2.704	2.863	<u>2.940</u>	<b>3.138</b> <sup>†</sup>	6.7%
	NDCG@20	0.374	0.559	0.854	0.427	0.659	0.873	0.927	<u>1.018</u>	<b>1.081</b> <sup>†</sup>	6.2%
	Hit@20	2.139	2.777	4.188	3.467	3.684	4.343	4.467	<u>4.829</u>	<b>5.138</b> <sup>†</sup>	6.4%
	Recall@50	2.715	2.948	4.505	2.774	3.309	4.522	4.651	<u>4.956</u>	<b>5.352</b> <sup>†</sup>	7.9%
	NDCG@50	0.640	0.778	1.151	0.723	0.829	1.116	1.199	<u>1.356</u>	<b>1.464</b> <sup>†</sup>	8.0%
	Hit@50	4.833	5.085	6.705	5.052	5.812	6.723	7.155	<u>7.785</u>	<b>8.503</b> <sup>†</sup>	9.2%
Toys	Recall@20	0.928	3.214	6.343	5.315	6.593	6.670	6.983	<u>7.841</u>	<b>8.254</b> <sup>†</sup>	10.3%
	NDCG@20	0.510	1.641	2.912	2.114	2.817	3.073	3.072	<u>3.418</u>	<b>3.726</b> <sup>†</sup>	9.0%
	Hit@20	2.496	6.926	12.838	11.075	13.153	13.474	14.079	<u>15.331</u>	<b>16.661</b> <sup>†</sup>	8.7%
	Recall@50	1.844	5.406	10.264	8.962	10.018	10.730	11.300	<u>12.463</u>	<b>13.328</b> <sup>†</sup>	6.9%
	NDCG@50	0.774	2.216	3.899	2.952	3.690	4.072	4.095	<u>4.612</u>	<b>5.081</b> <sup>†</sup>	10.2%
	Hit@50	4.760	11.554	19.837	17.282	19.400	20.363	21.330	<u>23.389</u>	<b>25.462</b> <sup>†</sup>	8.9%
Gowalla	Recall@20	1.206	5.642	8.581	5.559	7.869	7.823	8.804	<u>8.973</u>	<b>9.317</b> <sup>†</sup>	3.8%
	NDCG@20	1.191	5.536	7.546	3.891	6.819	7.351	7.601	<u>7.618</u>	<b>7.759</b> <sup>†</sup>	1.9%
	Hit@20	5.874	22.450	28.931	19.052	26.315	27.676	29.853	<u>30.075</u>	<b>31.832</b> <sup>†</sup>	5.8%
	Recall@50	2.084	9.623	13.838	9.891	12.793	12.710	14.372	<u>15.195</u>	<b>15.972</b> <sup>†</sup>	5.1%
	NDCG@50	1.678	7.784	10.510	5.725	9.107	9.752	10.630	<u>10.735</u>	<b>11.012</b> <sup>†</sup>	2.6%
	Hit@50	9.716	34.321	42.380	32.041	38.613	39.463	43.659	<u>44.618</u>	<b>46.217</b> <sup>†</sup>	3.5%

demonstrating the effectiveness of neural networks for SR. However, GRU4Rec performs poorly compared to other neural network-based methods. It shows that directly using the representation of the last step of RNN is not enough for SR, which is due to the forgetting problem of RNN. As such, dependencies between items cannot be effectively extracted.

- We can observe that SASRec and ComiRec-SA surpass GRU4Rec on four datasets, which demonstrates the strength of the multi-head self-attention mechanism for SR. It may benefit from two aspects: First, the attention mechanism can capture the long-term dependencies within the sequence. Second, it will assign more significant weight to more important items, which filters the noise in the sequence.
- By comparing GCSAN and ComiRec-SA, we can observe the benefits brought by graph neural networks. It is because graph neural networks can capture more complex item transitions by converting transitions within sequences to graphs, resulting in better performance.
- S<sup>3</sup>-Rec<sub>MIP</sub> and CL4SRec exhibit relatively good performance among baseline methods, indicating the significance of contrastive learning for SR. It is because the contrastive mechanism can learn the extra self-supervised signal for SR, which alleviates the data sparsity problem. However, both two methods perform data-level augmentation on each sequence. Due to the less information diversity between the contrastive pairs, the self-supervised signal will be too weak to learn informative embedding.
- DuoRec surpasses S<sup>3</sup>-Rec<sub>MIP</sub> and CL4SRec in most cases. The reason is that DuoRec applies model-level augmentation (*i.e.*,

dropout) to the sequence and performs contrastive learning with a positive sampling strategy, which enhances the diversity of information for contrastive learning. However, the information in each sequence is limited, which means it can obtain limited self-supervised signals without exploring the rich collaborative information and co-action information.

- MCLSR significantly outperforms all baselines overall four datasets consistently. Specifically, the average improvement of MCLSR over the best baseline is 7.0% on four datasets, demonstrating its effectiveness for SR. Different from previous contrastive learning-based methods, the proposed MCLSR leverages a multi-level contrastive mechanism and extracts the complex collaborative information and co-action information for SR, which learns discriminative user and item embeddings.

### 5.3 Ablation Study

As the proposed MCLSR outperforms all kinds of baselines consistently, we investigate the effectiveness of critical components to analyze the proposed method deeply and comprehensively. Specifically, we conduct ablation studies by comparing four variants with the complete model on four datasets. (1) “MCLSR-G” represents removing the graph encoder layer and directly using the initial embedding of user and item. (2) “MCLSR-IF” means removing two levels of contrastive mechanism and only optimizing the loss function for target prediction. (3) “MCLSR-F” directly removes the feature-level contrastive mechanism. (4) “MCLSR-I” represents removing the interest-level contrastive mechanism. From the results in Figure 3, we can make the following observations:

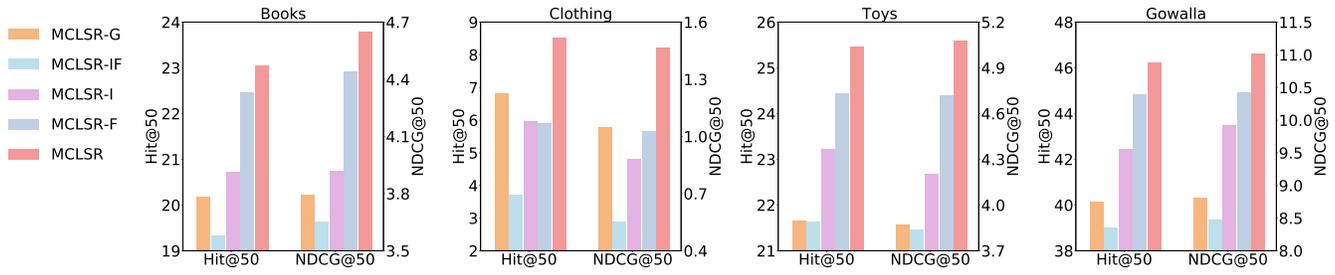


Figure 3: Ablation study on four datasets.

Table 3: The performance of MCLSR with varied depth of GNN layers in terms of Metrics@50.

Depth	Books			Clothing			Toys			Gowalla		
	Recall	NDCG	Hit Rate	Recall	NDCG	Hit Rate	Recall	NDCG	Hit Rate	Recall	NDCG	Hit Rate
$l = 0$	9.881	3.799	20.032	13.025	8.805	40.141	11.052	3.847	21.673	13.265	8.728	40.273
$l = 1$	10.853	4.166	21.782	15.897	10.889	44.543	13.165	4.744	25.069	15.769	10.743	45.620
$l = 2$	<b>11.583</b>	<b>4.647</b>	<b>23.042</b>	<b>15.972</b>	<b>11.012</b>	<b>46.217</b>	13.328	5.081	25.462	<b>15.972</b>	<b>11.012</b>	<b>46.217</b>
$l = 3$	10.085	3.936	20.547	15.021	10.042	43.775	<b>13.720</b>	<b>5.187</b>	<b>26.417</b>	14.948	10.094	43.665

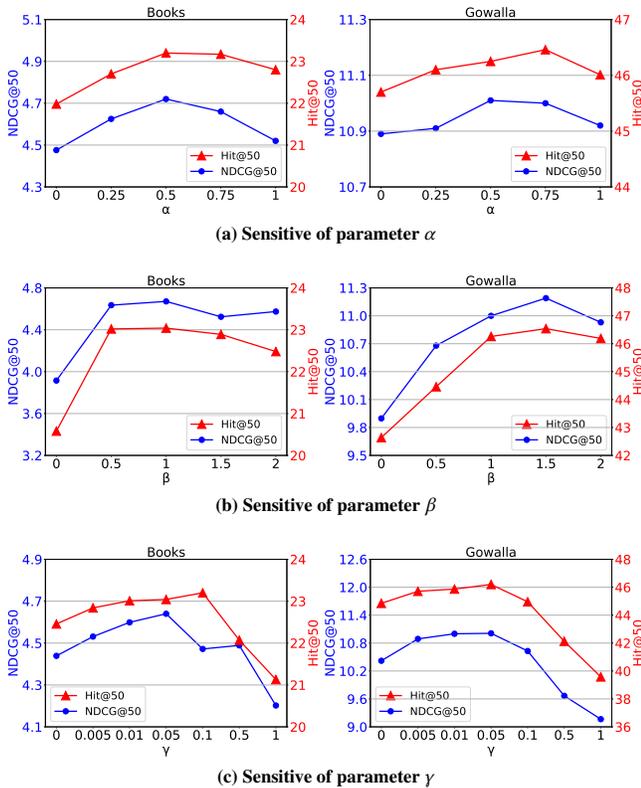


Figure 4: Impact of trade-off parameter.

- There is a significant drop when removing the graph encoder layer (*i.e.*, MCLSR-G). It verifies the significance of the collaborative information hidden in the user-item graph and the co-action

information hidden in the user-user/item-item graph. Besides, it also demonstrates the effectiveness of the graph encoder layer.

- The performance of MCLSR would be remarkably decreased when removing two levels of contrastive mechanism (*i.e.*, MCLSR-IF), indicating the crucial role of two levels of contrastive mechanism in learning the extra self-supervised signals and alleviating the data sparsity problem.
- Compared with MCLSR, MCLSR-I obtains dramatically worse results on four datasets. It is because MCLSR-I loses the self-supervised signal obtained by contrasting the user-item view and sequential view. The results demonstrate the importance of interest-level contrastive mechanisms to learn significant self-supervised signals.
- MCLSR-F is inferior to the complete model MCLSR, especially on Clothing datasets, indicating the importance of co-action information in learning the informative embeddings of users and items from user-user and item-item graphs.

### 5.4 Parameter Sensitive

Here, we investigate the impact of important hyper-parameter settings on the performance of MCLSR, including trade-off parameters  $\alpha$ ,  $\beta$ ,  $\gamma$ , and the number of propagation layers in GNN. The results are shown in Figure 4 and Table 3

**Impact of the parameter  $\alpha$ .** The trade-off parameter  $\alpha$  in Equation 13 controls the proportion of general interest and current interest during train process. It can be seen from Figure 4a that MCLSR performs worst when  $\alpha$  is set to 0, which is due to the inconsistency between the training process and the inference process. Besides, the performance dramatically degrades when  $\alpha$  is set to 1, indicating the importance of general interest. From the results, MCLSR obtains the best performance when  $\alpha = 0.25$  on Books and  $\alpha = 0.75$  on Gowalla.

**Impact of the parameter  $\beta$ .** The trade-off parameter  $\beta$  determines the influence of the interest-level contrastive mechanism during

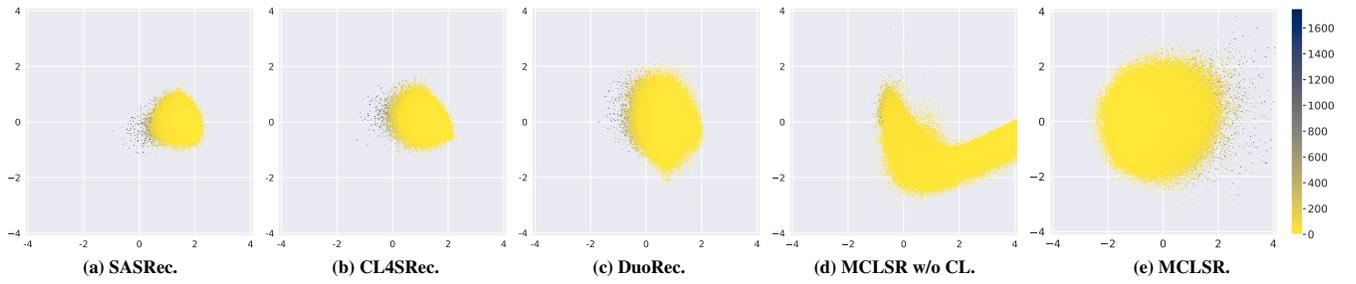


Figure 5: Item embeddings of selected methods on Book dataset.

the training process. From Figure 4b, it can be observed that the performance of MCLSR shows a significant rise when  $\beta$  increases from 0 to 0.5, which demonstrates the crucial role of the interest-level contrastive mechanism. Besides, the results of MCLSR become slightly worse when  $\beta > 1.5$  on Gowalla, which means excessive attention to collaborative information may hurt performance.

**Impact of the parameter  $\gamma$ .** The trade-off parameter  $\gamma$  controls the influence of the feature-level contrastive mechanism. From the results in Figure 4c, MCLSR obtains better performance when  $\gamma$  increases from 0 to 0.05 on two datasets, which demonstrates the effectiveness of feature-level contrastive learning. Besides, the performance of the model drops significantly when  $\gamma$  is set to 1, which shows that focusing too much on the co-action signals of users and items will deteriorate the performance of the model.

**Impact of the depth of GNN layers.** To deeply investigate whether MCLSR benefits from the graph information, we search the number of graph encoder layers  $l$  in the range of  $\{0, 1, 2, 3\}$  and summarize the results in Table 3. It can be observed that:

- The information on the user-item graph, user-user graph, and item-item graph is significant for SR. Specifically, MCLSR with  $l = 0$  obtains dramatically worse results, and there is a significant improvement when setting  $l = 1$  for MCLSR.
- Increasing the depth of GNN is able to enhance the predictive results. More specifically, MCLSR with  $l = 2$  performs better than MCLSR with  $l = 1$  on four datasets and MCLSR achieves the best performance on Toys when  $l = 3$ , which indicates that higher order of propagation obtains more effective collaborative information from three graph views.
- Higher layer of GNN may deteriorate the performance of MCLSR. Specifically, MCLSR with  $l = 3$  performs worse than MCLSR with  $l = 2$  in most cases, which may be due to the overfitting problem of GNNs [33].

## 5.5 Qualitative Analysis

Except for performance scores, we also provide qualitative results to demonstrate the superiority of MCLSR further. Specifically, we project the learned item embedding into two-dimensional space by SVD [26] and show the learned space of selected methods on the Book dataset in Figure 5.

From Figure 5a we can observe that the item embedding learned by SASRec degenerated into a narrow cone. According to [8, 31]

such phenomena deteriorate the model’s capacity as the learned embedding does not have enough capacity to model the diverse features. Comparing Figure 5b, 5c and 5a, the learned embedding spaces of CL4SRec and DuoRec are better than SASRec. It is because CL4SRec and DuoRec devise auxiliary self-supervised objectives for data representation learning based on data-level augmentation and model-level augmentation, respectively. However, directly exploiting the self-supervised signals from the sequence is insufficient for SR. In contrast, Figure 5d and 5e show that the learned embeddings of MCLSR without and with CL. It can be observed that the learned embeddings of MCLSR (Figure 5e) are somewhat uniformly distributed around the origin and not strictly in a narrow cone, which effectively expands the embedding space and has more capacity to model the diverse features of items. We argue that this is because MCLSR learns the representations of users and items through a cross-view contrastive learning paradigm on two levels. Specifically, the interest-level contrastive mechanism jointly learns the collaborative information and the sequential transition patterns, and the feature-level contrastive mechanism captures the co-action signals when learning the user and item features. In this way, MCLSR obtains discriminative item and user representations without extra labels.

## 6 CONCLUSION

This study presents a multi-level contrastive learning framework for sequential recommendation. Different from previous methods, we design four informative views (*i.e.*, sequential view, user-item view, user-user view, and item-item view) and learn the self-supervised signals via cross-view contrastive learning at two different levels. The interest-level contrastive mechanism learns the complementary information from collaborative information and sequential transition patterns and the feature-level contrastive mechanism mines the co-action information between users and items. Comprehensive experiments demonstrate that the proposed method significantly outperforms baselines over four datasets, indicating it has excellent potential to solve real-world recommendation problems.

## ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant No.61602197, Grant No.L1924068, Grant No.61772076, in part by CCF-AFSG Research Fund under Grant No.RF20210005, and in part by the fund of Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL).

## REFERENCES

- [1] Renqin Cai, Jibang Wu, Aidan San, Chong Wang, and Hongning Wang. 2021. Category-aware collaborative sequential recommendation. In *SIGIR*. 388–397.
- [2] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *SIGKDD*. 2942–2951.
- [3] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *SIGIR*. 378–387.
- [4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. 1597–1607.
- [5] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiayi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *WSDM*. 108–116.
- [6] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. In *WWW*. 2172–2182.
- [7] Guanyi Chu, Xiao Wang, Chuan Shi, and Xunqiang Jiang. 2021. CuCo: Graph representation with curriculum contrastive learning. In *IJCAI*. 2300–2306.
- [8] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tiejian Liu. 2018. Representation Degeneration Problem in Training Natural Language Generation Models. In *IJCAI*.
- [9] Ming Gao, Leihui Chen, Xiangnan He, and Aoying Zhou. 2018. Bine: Bipartite network embedding. In *SIGIR*. 715–724.
- [10] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *ICDM*. IEEE, 191–200.
- [11] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.
- [12] Balázs Hidasi, Alexandros Karatzoglou, Lina Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *ICLR*.
- [13] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *SIGIR*. 505–514.
- [14] Xunqiang Jiang, Yuanfu Lu, Yuan Fang, and Chuan Shi. 2021. Contrastive Pre-Training of GNNs on Heterogeneous Graphs. In *CIKM*. 803–812.
- [15] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. IEEE, 197–206.
- [16] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *CIKM*. 2615–2623.
- [17] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *WSDM*. 322–330.
- [18] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *ICLR*.
- [19] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *TKDE* (2021).
- [20] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical gating networks for sequential recommendation. In *SIGKDD*. 825–833.
- [21] Chen Ma, Liheng Ma, Yingxue Zhang, Jianing Sun, Xue Liu, and Mark Coates. 2020. Memory augmented graph neural networks for sequential recommendation. In *AAAI*, Vol. 34. 5045–5052.
- [22] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *SIGKDD*. 483–491.
- [23] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*. 43–52.
- [24] Yuqi Qin, Pengfei Wang, and Chenliang Li. 2021. The world is binary: Contrastive learning for denoising next basket recommendation. In *SIGIR*. 859–868.
- [25] Ruihong Qiu, Zi Huang, and Hongzhi Yin. 2021. Memory Augmented Multi-Instance Contrastive Predictive Coding for Sequential Recommendation. In *ICDM*. IEEE, 519–528.
- [26] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *WSDM*. 813–823.
- [27] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the item order in session-based recommendation with graph neural networks. In *CIKM*. 579–588.
- [28] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*. 811–820.
- [29] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*. 1441–1450.
- [30] Jiayi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*. 565–573.
- [31] Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2019. Improving neural language generation with spectrum control. In *IJCAI*.
- [32] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. 2018. Attention-based transactional context embedding for next-item recommendation. In *AAAI*, Vol. 32.
- [33] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*. 165–174.
- [34] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. 2021. Self-Supervised Heterogeneous Graph Neural Network with Co-Contrastive Learning. In *SIGKDD*. 1726–1736.
- [35] 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 *SIGIR*. 169–178.
- [36] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *WSDM*. 495–503.
- [37] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. 726–735.
- [38] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *AAAI*. 346–353.
- [39] Xin Xia, Hongzhi Yin, Junliang Yu, Yingxia Shao, and Lizhen Cui. 2021. Self-supervised graph co-training for session-based recommendation. In *CIKM*. 2180–2190.
- [40] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-supervised hypergraph convolutional networks for session-based recommendation. In *AAAI*. 4503–4511.
- [41] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Bolin Ding, and Bin Cui. 2021. Contrastive Learning for Sequential Recommendation. *arXiv preprint arXiv:2010.14395* (2021).
- [42] 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*. 3940–3946.
- [43] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Jiajie Xu, Victor S Sheng S. Sheng, Zhiming Cui, Xiaofang Zhou, and Hui Xiong. 2019. Recurrent convolutional neural network for sequential recommendation. In *WWW*. 3398–3404.
- [44] Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing Word Omission Errors in Neural Machine Translation: A Contrastive Learning Approach. In *ACL*. 6191–6196.
- [45] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *NIPS* (2020), 5812–5823.
- [46] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-aware self-supervised tri-training for recommendation. In *SIGKDD*. 2084–2092.
- [47] Junliang Yu, Hongzhi Yin, Xin Xia, Lizhen Cui, and Quoc Viet Hung Nguyen. 2021. Graph Augmentation-Free Contrastive Learning for Recommendation. *arXiv preprint arXiv:2112.08679* (2021).
- [48] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *WSDM*. 582–590.
- [49] Mengqi Zhang, Shu Wu, Xueli Yu, Qiang Liu, and Liang Wang. 2022. Dynamic graph neural networks for sequential recommendation. *TKDE* (2022).
- [50] Shuai Zhang, Huoyu Liu, Aston Zhang, Yue Hu, Ce Zhang, Yumeng Li, Tanchao Zhu, Shaojian He, and Wenwu Ou. 2021. Learning User Representations with Hypercuboids for Recommender Systems. In *WSDM*. 716–724.
- [51] Sen Zhao, Wei Wei, Zou Ding, and Xian-Ling Mao. 2022. Multi-view Intent Disentangle Graph Networks for Bundle Recommendation. In *AAAI*. 1–7.
- [52] 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 *CIKM*. 1893–1902.
- [53] Yutao Zhu, Jian-Yun Nie, Zhicheng Dou, Zhengyi Ma, Xinyu Zhang, Pan Du, Xiaochen Zuo, and Hao Jiang. 2021. Contrastive Learning of User Behavior Sequence for Context-Aware Document Ranking. In *CIKM*. 2780–2791.
- [54] Ding Zou, Wei Wei, Xian-Ling Mao, Ziyang Wang, Minghui Qiu, Feida Zhu, and Xin Cao. 2022. Multi-level Cross-view Contrastive Learning for Knowledge-aware Recommender System. In *SIGIR*. 1358–1368.