ID and Graph View Contrastive Learning with Multi-View Attention Fusion for Sequential Recommendation

Xiaofan Zhou
Worcester Polytechnic Institute
Worcester, USA
xzhou5@wpi.edu

Kyumin Lee
Worcester Polytechnic Institute
Worcester, USA
kmlee@wpi.edu

***Abstract*—Sequential recommendation has become an increasingly prominent subject both in academia and industrial sectors, particularly within the e-commerce domain. Its primary aim is to extract user preference from a user's historical item list and predict the subsequent items that the user might purchase based on that history. Recent trends show a surge in the application of using contrastive learning and graph-based neural network to extract more expressive representation from user's historical item list, where graph contains information of relationship between nodes while ID based representation contains more specific information. However, limited work has explored on multi-view contrastive learning, especially, between the ID and graph to further improve quality of user and item representation learning when only interaction data is available without auxiliary information. To fill the gap, in this study, we propose a novel framework called MultiView Contrastive learning for sequential recommendation (MVCrec). This framework is designed to combine information from both sequential/ID and graph views. It incorporates three facets of contrastive learning: one for sequential view, another one for graph view and the other one for cross-view. To leverage the representations derived from the contrastive learning, we propose a multi-view attention fusion module, which integrates both global and local attentions and measures how likely a target user will purchase a target item. Comprehensive experiments demonstrate the superiority of our model over 11 state-of-the-art baselines, as evidenced by its performance on five real-world benchmark datasets. Our model achieves improvements of up to 14.44% in NDCG@10 and up to 9.22% in HitRatio@10 compared to the best baseline. Our code and datasets are available at https://github.com/sword-Lz/MMCrec.**

***Index Terms*—Sequential recommendation, contrastive learning, graph**

I. Introduction

Sequential recommendation has received increasing attention from both industry and academia, with the primary focus being on recommending items based on users' chronologically ordered purchase histories [1]–[7]. In the early stage, researchers applied recurrent neural network (RNN) and convolutional neural network (CNN) to sequential recommendation [8]–[10]. Additionally, self-supervised methods have been employed in sequential recommendation; for example, BERT4Rec [11] utilizes BERT as an encoder for sequential lists. More recently, contrastive learning and related techniques have been adopted in sequential recommendation to enhance the effectiveness of learned representations [12]–[14].

However, the utilization of contrastive learning (CL) to effectively capture the information of historical sequences remains a challenging research area. Contrastive learning aims to maximize the dissimilarity between different categories of individuals (e.g., users or items) while minimizing the dissimilarity within the same category. The first obstacle often lies in selecting suitable augmentation operations for generating similar instances. To date, three classes of augmentation operations have been established. The first class generates different views of the same sequence through random operations like 'masking', 'cropping', or 'reordering' items [15], [16]. The second class uses variable dropout probabilities at the model level to create different views of the same sequential data [13]. The third class combines 'neural mask', 'layer drop', and 'encoder complement' with data augmentation techniques for constructing positive and negative view pairs [17].

Most of the prior works leverage sequence information to perform contrastive learning on individual sequences. They employ data augmentation or model-level augmentation techniques to augment the historical sequences. Subsequently, the InfoNCE objective function [18] is utilized to compute the contrastive loss. This objective function aims to minimize the distance between augmented sequences generated from the same original sequence, while maximizing the distance between augmented sequences generated from different original sequences.

Although these prior methods have achieved some effectiveness in sequential recommendation, they are suboptimal because of the neglect of structural information which can be obtained/learned from graph-based methods. Graph-based recommendation systems provide a more comprehensive representation of users and items by fully exploiting graph structures, thereby making significant contributions to the field of recommendation systems. In basic recommendation approaches, NGCF [19] and LightGCN [20] integrate graph convolutional networks into the recommendation systems. UltraGCN [21] simplifies GCNs for collaborative filtering by omitting feature transformations and nonlinear activations.

As contrastive learning has developed, VGCL [22] employs variational graph reconstruction to estimate the Gaussian distribution of each node and generates multiple contrastive views through multiple samplings from the estimated distributions. CGCL [23] explore a new way to build contrastive pairs by using similar semantic embeddings. In the realm of sequential recommendation, graph contrastive learning also plays a significant role; MAErec [24] applies graph contrastive learning to adaptively and dynamically distill global item transitional information in self-supervised augmentation scenarios with scarce labels. However, cross-view contrastive learning between graph and sequence information remains a less explored area in sequential recommendation, especially, when given only interaction data without any auxiliary information.

To fill the gap, in this paper, we propose a novel framework based on multi-view contrastive learning, named **M**ulti**V**iew **C**ontrastive learning for sequential **rec**ommendation (MVCrec). Initially, we use contrastive learning to learn each user's historical sequence representation. To make the most of graph structure given the sequence information, we also build an item-based graph and apply contrastive learning to learn the structural representation from the historical sequence. According to common sense, embedding of item IDs provides more item-specific information, whereas utilizing a graph structure to represent items captures more information about their relationships with other items. To further enhance our understanding of structural and sequential representations, we introduce and implement a cross-view contrastive learning strategy. This strategy is designed to pull out more detailed features, generating extra contrastive pairs, which are compared with data-augmented views during the training. Finally, given the two different sequence representations (i.e., item-based sequence representation and graph-based sequence representation) which are created by the contrastive learning, we run our proposed multi-view attention fusion module to combine structural and sequential features. In the experiment, we found that both sequence and graph structures positively contributed to improving the effectiveness, with the graph structure having a greater impact than the sequence view.

In summary, the major contributions of our proposed MVCrec are as follows:

- We propose a novel multi-view contrastive learning approach in the sequential recommendation domain. The proposed model proficiently extracts relevant information from both positive and negative samples by utilizing sequence and graph views derived from users' historical item lists (i.e., prior interaction data).
- A multi-view attention fusion module is proposed to be seamlessly integrated into MVCrec to calculate the recommendation score, utilizing representations from diverse views.
- Through comprehensive experiments across five public benchmark datasets, we demonstrate that MVCrec outperforms 11 state-of-the-art baselines.

II. Related work

A. Sequential recommendation

Sequential recommendation is deployed to forecast user preferences based on their historical purchases. In the initial phase of sequential recommendation development, the Markov chain was utilized to formulate predictions by modeling stochastic transitions and uncovering sequential patterns [25], [26].

With the growth of deep learning in many areas, RNN and Transformer-based methods have been used in sequential recommendation and have achieved good results. They are good at understanding both the long-term and short-term information in users' historical sequences. For example, GRU4rec [8] uses Gated Recurrent Units (GRU) to learn sequential information from the previously consumed items. Caser [9] uses both horizontal and vertical CNNs to understand sequential behaviors. SASRec [10] was the first to use the attention mechanism in sequential recommendation. In terms of Transformer, BERT4Rec [11] uses deep bidirectional self-attention to understand the possible relationships between items and sequences. LinRec [27] introduces a novel method that enhances efficiency while retaining the learning capabilities of traditional dot-product attention through a linear attention module. MELT [28] mutually enhance user and item bilateral branches to deal with long-tailed problem.

While these methodologies have made some advancements in the field of sequential recommendation, most of them have not incorporated structured information, such as graph structures, into their considerations. Unlike the prior works, our approach concurrently uses information derived from both graphs and sequences.

B. Contrastive learning

To enable deep learning models to more accurately differentiate instances pertaining to distinct individuals, contrastive learning was introduced in [29]. The core concept of contrastive learning is to maximize the dissimilarity between varying individuals, and it has witnessed substantial advancements in recent years. The work of [18] introduced the use of mutual information to quantify the similarity between two individuals, considering different views of the same individual as positive pairs. Subsequently, [30] employed a queue to manage the extensive dictionary associated with contrastive learning, while [31] leveraged the remaining pairs in the batch as the negative pairs for the positive pair, introducing a projector to enhance the performance of contrastive learning further. Additionally, [32] explored the execution of contrastive learning tasks without the incorporation of negative samples. In multi-view contrastive learning, MSM4SR [33] proposes fusing text and image views before applying contrastive learning. However, this approach overlooks the interrelationship of cross-view contrastive learning. On the other hand, MMSSL [34] suggests using GCN for cross-view contrastive learning, but it doesn't account for sequential data.

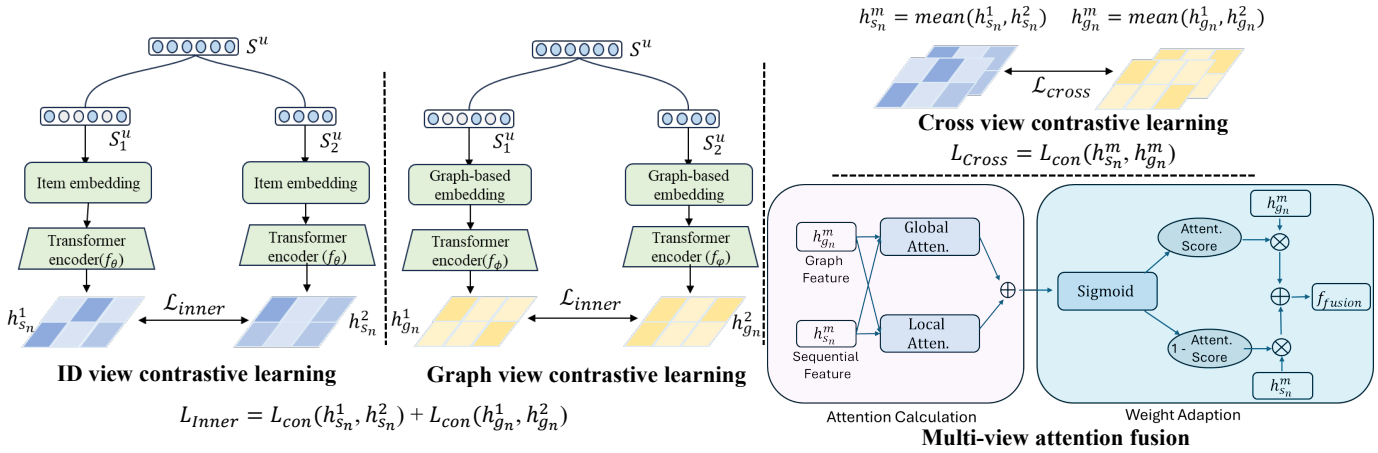


Fig. 1: Our proposed framework, MVCrec, consists of multi-view contrastive learning and multi-view attention fusion module.

Recently, contrastive learning has been used in sequential recommendation to handle issues like not having enough data and having data that's noisy. CL4rec [12] learns about users by comparing different views of the same sequence data. It uses random actions like 'mask', 'crop', or 'reorder' items to create these different views. DuoRec [13] makes pairs to compare by using "dropout" at the model level and suggests using sequences with the same next interaction as matching pairs instead of comparing different data views. MCLrec [14] offers a meta-learning strategy to train contrastive learning with the goal to address the problem of sparse data and create more meaningful representations. EMKD [35] proposes knowledge distillation, which uses contrastive learning to facilitate knowledge transfer between parallel networks, and uses the ensemble of different models as the final prediction. DCrec [36] introduces a new global learning strategy to deal with popularity bias in sequential recommendation. Wu et al. [37] proposed a Multi-behavior Multi-view Contrastive Learning Recommendation (MMCLR), which uses different behaviors of users as positive pair for recommendation, which requires multi-behavior data for implementation. MCLSR [38] proposed multi-level contrastive learning framework for sequential recommendation. Unlike our work utilizing only target behavior with item-based sequence embeddings and graph-based item embeddings, MMCLR employs cross-view contrastive learning by constructing various types of graphs to generate representations for users or items. It also neglects the inner structure in different view, which deserves to apply contrastive learning.

In this paper, the principle of contrastive learning is adapted to extract superior representations of historical interaction sequences, and a new multi-view contrastive learning approach which includes inner view and cross view contrastive leaning is proposed.

C. Graph-based recommendation

User and item interactions in the recommendation task naturally form a graph structure; thus, the incorporation of graph structures is prevalent in recommendation systems. Foundational recommendations like NGCF [19] and LightGCN [20] have advanced the field of recommendation by integrating GCN structures, thus, enhancing the developmental trajectory of recommendation systems. UltraGCN [21] further refines the approach by streamlining GCNs for collaborative filtering and omitting unnecessary feature transformations and nonlinear activations. Additionally, works like CGCL [23] and VGCL [22] have applied graph structures to contrastive learning, utilizing auto-encoders to optimize the process. SRGNN [39] was proposed to use GNN structure to train the sequential recommendation. Within the realm of sequential recommendation, MAErec [24] ingeniously employs graph data in contrastive learning to address issues related to label scarcity. In this paper, we also construct a graph for items to learn their embeddings and user preference representation from the historical sequence via multi-view contrastive learning.

III. Problem Definition

The primary objective of this paper is to predict the next item, c_{n+1}, which a user u is likely to purchase based on the user's historical sequence, denoted as $S^u = [c_1, c_2, \ldots, c_n]$. In this notation, c_i represents the i-th item that the user has purchased, and n is the length of the user's purchasing history.

IV. Proposed Method

A. Overview

As depicted in Figure 1, our proposed MVCrec learns two types of item embedding (typical item embedding and graph-based item embedding), and integrates two contrastive learning approaches: graph-based and sequence-based contrastive learning. Each approach consists of a stochastic data augmentation module, a sequence encoder, and a contrastive loss function [12]. To optimally leverage information from both graph and sequence data, MVCrec employs a cross-view contrastive loss, complementing the two contrastive learning approaches. Additionally, a multi-view attention fusion module

is formulated to amalgamate item-based sequence representation and graph-based sequence representation from both views. In essence, MVCrec consists of five components: (1) stochastic data augmentation module, (2) item embeddings, (3) Transformer-based sequence encoder, (4) multi-view contrastive learning, and (5) multi-view attention fusion module. Detailed information about these modules are described in the following subsections.

B. Stochastic data augmentation module

This module aims to generate two positive views for each historical sequence. Inspired by CL4rec [12], we apply three stochastic data augmentations — 'masking', 'cropping', and 'reordering' — to the historical sequence. The procedure for generating two augmented sequences is as follows:

$$\tilde{S}_1^u = g_1\left(S^u\right), \tilde{S}_2^u = g_2\left(S^u\right) \tag{1}$$

where g_1 and g_2 are a pair of different stochastic data augmentation methods (i.e., randomly select two of 'mask', 'crop' and 'reorder'), and $\tilde{S}_1^u$ and $\tilde{S}_2^u$ are a pair of positive samples.

C. Two types of item embedding

Initially, we project all items into a common embedding space [10]. In this paper, two types of item embedding are used and learned: one is the typical item embedding, and the other one is graph-based embedding. For the typical item embedding, we project all items into $M_s \in \mathbb{R}^{|I| \times d}$ via an embedding layer, where $|I|$ denotes the total number of items, and d represents the dimension of the embedding. For the graph-based item embedding, we use a GCN-based graph encoder to project all items into an embedding space.

D. Graph convolutional encoder

In particular, for the GCN-based graph encoder, we draw upon the concepts presented in [19], [40]. We construct a single graph for the entire dataset to capture node relationships from a global perspective. To build the graph for items, each item within a dataset is viewed as a node. If two items are co-located in less than z distance in a historical sequence, we add an edge between them. Here, z represents a predetermined maximum distance. Initially, we project all items into a common embedding space, $M_g^0 \in \mathbb{R}^{|I| \times d}$, where $|I|$ is the number of items and d is the dimension of embedding, and we treat this as the first layer's item embedding in the graph. Following [19], we discard feature transformation and nonlinear activation for improving efficiency. Then the computation within the GCN-based graph encoder proceeds as follows:

$$\mathbf{m}_i^{l+1} = \mathbf{m}_i^l + \sum_{i' \in \mathcal{N}_i} \mathbf{m}_{i'}^l; \quad \tilde{\mathbf{m}}_i = \sum_{l=0}^{L} \mathbf{m}_i^l \tag{2}$$

where L denotes the total number of layers, and $\mathcal{N}_i$ represents one-hop neighbor nodes of m_i. $\mathbf{m}_i^l, \mathbf{m}_{i'}^l$ represent the embedding of items $i, i' \in |I|$ in the l-th layer. Specifically, we sum up the representations from all layers to obtain the final embedding of an item i, denoted as $\tilde{\mathbf{m}}_i$. We call it graph-based (item) embedding, and all items' graph-based embeddings are represented as a matrix $M_g \in \mathbb{R}^{|I| \times d}$. The graph encoder is designed to convert items into expressive representations based on the structural information in the graph.

E. Transformer-based sequence encoder

Transformer-based sequence encoder is a vital step in the sequential recommendation. It aims to extract the representation from the sequence list. First of all, we describe input to the sequence encoder.

Input to the sequence encoder. Given the input as an interaction history sequence $S^u = [c_1, c_2, ..., c_n]$, the Transformer takes into account the positions of items by initializing the history item list S^u to $e^u \in \mathbb{R}^{n \times d}$ by:

$$\begin{aligned} e_s^u &= [m_{s_1} + p_1, m_{s_2} + p_2, ..., m_{s_n} + p_n]. \\ e_g^u &= [m_{g_1} + p_1, m_{g_2} + p_2, ..., m_{g_n} + p_n]. \end{aligned} \tag{3}$$

where $m_{s_i} \in \mathbb{R}^d$ represents an item's typical item embedding at the i-th position in the sequence, $m_{g_i} \in \mathbb{R}^d$ represents the item's graph-based embedding at i-th position in the sequence, $p_i \in \mathbb{R}^d$ denotes the positional embedding, and n is the sequence length. We note that m_{s_i} and m_{g_i} are extracted from embedding matrices M_s and M_g, respectively, described in the previous subsections.

Sequence encoder. The sequence encoder derives the representation of e^u using a deep neural network (e.g., BERT4Rec) [11]. We use two sequence encoders: one for the sequence of item-based embeddings (e_s^u) and the other one for the same sequence of graph-based embeddings (e_g^u). The sequence encoders are defined as f_θ and f_ϕ, respectively, where θ and ϕ represent each model's parameters. The output representation $H_s^u \in R^{n \times d}$ and $H_g^u \in R^{n \times d}$ are calculated as follows:

$$\begin{aligned} H_s^u &= f_\theta(e_s^u) \\ H_g^u &= f_\phi(e_g^u) \end{aligned} \tag{4}$$

Since our main task is to predict the next item, we employ the final vectors h_{s_n} in $H_s^u = [h_{s_1}, h_{s_2}, ..., h_{s_n}]$ and h_{g_n} in $H_g^u = [h_{g_1}, h_{g_2}, ..., h_{g_n}]$ as the item-based sequence representation and graph-based sequence representation of the historical sequence, respectively. We can interpret them as two types of user representation.

F. Multi-view contrastive learning

Inner view contrastive learning. Inspired by the prior work [12], [14], we utilize InfoNCE as the objective function to optimize features extracted from contrastive learning. We denote the number of historical sequences in each batch by *B*. Given B historical sequences in the batch, each historical sequence goes through the stochastic data augmentation module and returns two augmented sequences, so totally there are *2B* augmented sequences. Since contrastive learning requires positive pairs and negative pairs, given a user's historical sequence (i.e., one of *B* historical sequences in the batch),

we create a positive pair of the sequence via the stochastic data augmentation module. We use the remaining $2(B-1)$ augmented sequences as negative samples for the positive pair.

For each positive pair, contrastive loss is calculated by:

$$\begin{aligned}\mathcal{L}_{\text{con}}\ \left(h_n^1, h_n^2\right) = &- \log \frac{exp^{s\left(h_n^1, h_n^2\right)}}{exp^{s(h_n^1, h_n^2)} + \sum_{h_n \in \text{ neg}}\ exp^{s(h_n^1, h_n)}} \\ &- \log \frac{exp^{s\left(h_n^2, h_n^1\right)}}{exp^{s(h_n^2, h_n^1)} + \sum_{h_n \in \text{ neg}}\ exp^{s(h_n^2, h_n)}}\end{aligned} \tag{5}$$

where h_n^1 and h_n^2 are the positive pair's sequence representations learned from the same Transformer-based sequence encoder (i.e., either f_θ or f_ϕ). $s(,)$ represents the inner product, and neg indicates the set of negative sample embeddings/representations. Since we can create $2(B-1)$ negative pairs for each of h_n^1 and h_n^2, the loss function consists of two terms.

Then, the objective function for optimizing the contrastive learning over the two different views (i.e., item-based sequence representation and graph-based sequence representation via the sequence encoders) is as follows:

$$\mathcal{L}_{\text{Inner}} = \mathcal{L}_{\text{con}}\ \left(h_{s_n}^1, h_{s_n}^2\right) + \mathcal{L}_{\text{con}}\ \left(h_{g_n}^1, h_{g_n}^2\right) \tag{6}$$

Where $h_{s_n}^1$ and $h_{s_n}^2$ are item-based sequence representations of the positive pair, and $h_{g_n}^1$ and $h_{g_n}^2$ are graph-based sequence representations of the positive pair.

Cross-view contrastive learning. In addition to the inner view contrastive learning, we propose a cross view contrastive learning, which learns discriminative features that capture the correspondence between the item-based sequence representation and graph-based sequence representation.

Firstly, the mean of the $h_{s_n}^1$ and $h_{s_n}^2$ obtained from a positive pair is calculated as $h_{s_n}^m$, and the mean of the $h_{g_n}^1$ and $h_{g_n}^2$ obtained from the same positive pair is calculated as $h_{g_n}^m$. Likewise, given *2(B-1)* negative samples in the batch, each two negative samples were originated from the same historical sequence (i.e., *B-1* negative sample pairs). Therefore, we also get each negative sample pair's mean of item-based sequence representations and mean of graph-based sequence representations.

Given the positive pair's mean representations $h_{s_n}^m$ and $h_{g_n}^m$, cross-view contrastive loss is calculated as follows:

$$\mathcal{L}_{\text{cross}} = \mathcal{L}_{\text{con}}\ \left(h_{s_n}^m, h_{g_n}^m\right) \tag{7}$$

$\mathcal{L}_{\text{cross}}$ is designed to maximize the similarity between $h_{g_n}^m$ and $h_{s_n}^m$. This approach compels the model to learn similar item-based and graph-based representations of the same historical sequence (or augmented sequences originated from the same sequence), yielding enhanced representation capability.

Consequently, we combine the aforementioned two contrastive loss functions as follows:

$$\mathcal{L}_{\text{MM}} = \mathcal{L}_{\text{cross}} + \mathcal{L}_{\text{Inner}} \tag{8}$$

G. Multi-view attention fusion module & recommendation prediction

To further utilize the extracted representations, we propose a multi-view attention fusion module designed to amalgamate information from two disparate views – item-based sequences and graph-based sequences – and ultimately predict a target user's preference score for a given item.

The multi-view attention fusion module is executed through an interactive cross-view attention mechanism, which is devised to uncover multi-view global and local dependencies. Given a user's two different view representations, $h_{g_n}^m \in R^{1\times d}$ and $h_{s_n}^m \in R^{1\times d}$, as depicted in Figure 1 (the rightmost figure), we initially calculate the global attention score, $s_{\text{global}}^{\text{attention}}$, and the local attention score, $s_{\text{local}}^{\text{attention}}$:

$$\begin{aligned}s_{global}^{attention} &= Averagepool\left(h_{g_n}^m + h_{s_n}^m\right) \otimes W_g \\ s_{local}^{attention} &= \sigma\left(\left(h_{g_n}^m + h_{s_n}^m\right) \otimes W_l\right)\end{aligned} \tag{9}$$

where $W_g \in R^{1\times 1}$ and $W_l \in R^{d\times d}$ represent global weight and local weight matrix, respectively. *Averagepool* represents an average pool function, which deals the representations from a global view, returns a single numeric value, which is the average of the d values. d is the dimension of a view's sequence representation, and $\otimes$ denotes matrix product. σ represents an activation function. In this paper, we employ ReLU as the activation function.

Given the global and local attention scores, a new fusion task arises as follows:

$$s^{attention} = \text{sigmoid}\left(s_{global}^{attention} \oplus s_{local}^{attention}\right) \tag{10}$$

where $\oplus$ represents the summation between the $s_{global}^{attention}$ and $s_{local}^{attention}$ with broadcasting to handle their different dimensions. We employ the sigmoid function to normalize the scores. These scores are considered as weights for different view representations.

Finally, fused representation $f_{fusion}\left(h_{s_n}^m, h_{g_n}^m\right) \in R^{1\times d}$is calculated as follows:

$$\begin{aligned}&f_{fusion}\left(h_{s_n}^m, h_{g_n}^m\right) \\ &= \left(s^{attention} \circ h_{g_n}^m\right) \oplus \left(\left(1 - s^{attention}\right) \circ h_{s_n}^m\right)\end{aligned} \tag{11}$$

where $\circ$ represents elements-wise product.

Since our final goal is to use this representation for recommendation, we propose a novel strategy to leverage the generated representation:

$$\hat{y} = f_{fusion}\left(h_{s_n}^m, h_{g_n}^m\right) M_s^T + f_{fusion}\left(h_{g_n}^m, h_{s_n}^m\right) M_g^T \tag{12}$$

where $h_{s_n}^m$ and $h_{g_n}^m$ are a target user's item-based sequence representation and graph-based sequence representation, respectively. M_s and M_g are the typical item embedding matrix and graph-based item embedding matrix, respectively, described in Section IV-C.

In our paper, we utilize cross-entropy loss as the objective function, optimizing to improve prediction accuracy.

$$\mathcal{L}_{rec} = H(y, \hat{y}) = -\sum_i y_i \log(\hat{y}_i) \tag{13}$$

where y represents the ground truth label for the user's true preference scores to items.

H. Overall Objective

Finally, the total loss function during the training stage can be represented as:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda \mathcal{L}_{MM} \tag{14}$$

where $\mathcal{L}_{rec}$ is the recommendation objective function in Eq. 13, $\mathcal{L}_{MM}$ represents multi-view contrastive loss, which consists of the inner-view contrastive loss and cross-view contrastive loss, as defined in Eq. 8, and λ is a hyperparameter.

V. Time Complexity

The time complexity of MVCRec is primarily influenced by the training phase, which includes both multi-view contrastive learning and embedding calculations. Training is split into two stages. In the first stage, MVCRec generates augmented views of user sequences and computes item-based and graph-based embeddings, resulting in a complexity of $O(|U|d^2 + |U|d)$, where $|U|$ is the number of users and d is the embedding dimension.

The second stage, which dominates the complexity, involves inner-view and cross-view contrastive learning with a complexity of $O(|U|^2 d)$. This is due to pairwise comparisons across augmented views within each type (item-based and graph-based), enhancing MVCRec's ability to capture user preferences.

However, in the testing phase, the model only requires the multi-view attention fusion module, simplifying the complexity to $O(nd)$, similar to that of an attention-based encoder like SASRec, where n is the sequence length.

VI. Experiment

In this section, we conduct extensive experiments using five real-world datasets to investigate the following research questions (RQs):

- **RQ1**: How is the performance of our MVCrec compared with existing baselines?
- **RQ2**: How effective are the key components of MVCrec in terms of enhancing the model's performance?
- **RQ3**: How do hyperparameters (i.e., the weight of the multi-view contrative loss λ, a batch size, and an embbeding size) affect the performance of MVCrec?

A. Experimental settings

1) ***Dataset****:* To verify the effectiveness of our model, we evaluate its performance using five real-world benchmark datasets: Amazon (Beauty, Sports and Home & Kitchen)[1], Yelp[2] and Reddit dataset[3]. The Amazon datasets contain a series of Amazon product reviews. In our experiments, we use three sub-categories of the Amazon: Beauty, Sports, and Home & Kitchen. The Yelp dataset, containing reviews of businesses listed on Yelp, serves a similar purpose as the Amazon datasets. Reddit dataset contains user interaction on the Reddit platform. In the following experiments, we only use interaction data without any auxiliary data (e.g., text, image). Following the preprocessing steps described in [16], [41], we removed users and items with fewer than five interactions. The statistics of the datasets are summarized in Table I.

TABLE I: The statistics of datasets.

Dataset	#users	#items	#interactions	avg.length	sparsity
Sports	33.6K	18.3K	296.3K	8.3	99.95%
Beauty	22.3K	12.1K	198.5K	8.8	99.93%
Yelp	30.4K	20.0K	316.3K	10.4	99.95%
Home & Kitchen	66.5K	28.2K	551.6K	8.3	99.97%
Reddit	14.5K	15.4K	28.9K	20.95	99.99%

2) ***Baselines****:* We compare our model with 11 state-of-the-art recommendation models, which can be divided into three parts:

Non-sequential models. These baselines are based on collaborative learning and graph convolutional network:

- *BPRMF* [42] uses Bayesian Personalized Ranking (BPR) loss to optimize the matrix factorization model.
- *LightGCN* [19] simplifies the design of GCN to make it more concise and appropriate for recommendation.

General sequential models. These baselines are based on RNN, attention-based neural networks, memory neural networks, GCN-based networks:

- *SRGNN* [39] models the historical item sequence as a graph-structured data to deal with sequential recommendation.
- *GRU4rec* [8] uses Gated Recurrent Unit (GRU) to model for the sequential recommendation.
- *Caser* [9] embeds a sequence of recent items into an "image" in the time and latent spaces, and learns sequential patterns as local features of the image using convolutional filters.
- *SASRec* [10] proposes the first self-attention based sequential model to capture long-term dependencies.

Self-supervised sequential models. These baselines are based on Transformer and collaborative learning:

- *BERT4Rec* [11] trains the bidirectional model using the Cloze task, predicting the masked items in the sequence by jointly conditioning on their left and right context.
- *CL4rec* [12] leverages contrastive learning on the sequential recommendation.
- *MCLSR* [38] 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).
- *MCLrec* [14] innovates the standard contrastive learning framework by contrasting data, and models augmented

[1]https://jmcauley.ucsd.edu/data/amazon/

[2]https://www.yelp.com/dataset

[3]https://www.kaggle.com/datasets/colemaclean/subReddit-interactions/data

views for adaptively capturing the informative features hidden in stochastic data augmentation.

- *DCrec* [36] proposes a global collaborative learning strategy to tackle with the popularity bias for sequential recommendation, considering dependencies between users across sequences.

We note that other existing methods (e.g., UltraGCN, VGCL, CGCL, MAErec, MMSSL [34], MSM4SR [33]) which are not aimed for sequential recommendation or require auxiliary data, are excluded in the baseline list except well-known *BPRMF* and *LightGCN* because their performance would be much lower than sequential recommendation models or sometimes it is hard to run some of their models without auxiliary data (again, in this paper, we only utilize a target behavior's interaction data without any auxiliary data).

*3) **Evaluation metric**:* In accordance with [12], [43]–[46], we employ the leave-one-out strategy to split each dataset into training, validation, and test sets based on the timestamp provided by the dataset. Specifically, we use the last interaction of every user for the test set, and the second-to-last interaction for every user is allocated for the validation set; all remaining interactions are used in the training set. Following the procedure in [47]–[50], we rank the entire item set.

We adopt Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) as evaluation metrics. HR@k measures whether the positive item appears in the top-k recommendation list, and NDCG@k additionally considers its position in the ranking list, where $k \in \{5, 10, 20\}$.

*4) **Implementation Details**:* We implement our method using PyTorch, aligning the implementation of BPRMF, LightGCN, FPMC, GRU4rec, Case, SASRec, CL4rec, and BERT4Rec with the methodologies described in their respective papers. We implemented MCLSR ourselves as the authors did not provide their code in the paper. A graph for the graph encoder is constructed based on the training set. To ensure fairness, we employ BERT as the representation encoder for CL4rec, MCLrec, DCrec, and our MVCrec, setting the number of self-attention blocks and attention heads to 2, and we set the item-to-item distance z as 3 (as mentioned in Section IV-D). All parameters are consistent with those reported in the original papers, and optimal settings are chosen based on model performance on the validation set. We set the embedding size d as 64 and the maximum length of recently consumed items in each user's historical sequence n as 50, selecting a hyperparameter λ from $\{0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$. The learning rate lr is chosen from $\{1e{-}3, 1e{-}4\}$, and weight decay is selected from $\{0, 1e{-}1, 1e{-}2, 1e{-}3, 1e{-}4, 1e{-}5, 1e{-}6\}$. For fairness, we standardize the batch size *B* to 256 for all models and fix temperature of contrastive learing for all contrastive learning method as 1. The models are optimized using the Adam optimizer [22] and are trained with an early stopping strategy based on the performance of the validation set, with the maximum step set to 100. All experiments are conducted on a Tesla T4 GPU.

B. RQ1: overall performance

To clarify the contributions made by MVCrec, we compare its performance with the baselines. The results presented in Table II lead us to several insights:

- Our method outperforms the baselines, attributed to the graph view and the multi-view fusion strategy. For instance, our model surpasses the best baseline by 1.18%$\sim$14.44% on NDCG@10 and 5.03% $\sim$ 9.22% on HR@10 over the five datasets. This superior performance can be explained as follows: (1) The multi-view contrastive learning strategy incorporating both sequence and graph information facilitates the generation of more expressive representations; and (2) The multi-view attention fusion strategy effectively amalgamates item-based sequence representation and graph-based sequence representation. These results confirm the effectiveness of our multi-view contrastive recommendation model, learning more accurate and better representations.
- Self-supervised models (e.g., MCLrec, DCrec) exhibit pronounced efficacy, markedly surpassing classical models such as BPRMF, LightGCN, GRU4rec and Caser. Fundamental Transformer-based methods like SASRec and BERT4Rec excel beyond the classical models, establishing themselves as the secondary tier in sequential recommendation and emphasizing the power of Transformer methods in this realm. In contrast to SASRec and BERT4Rec, models like CL4rec, MCLrec, DCrec, and MVCrec integrate contrastive learning and data augmentation methods for training in the recommendation tasks. This demonstrates contrastive learning's capability to harness more intricate representations from historical sequences by learning features that discern between distinct instances. Interestingly, LightGCN manifests substantial prowess on the Yelp dataset, aligning closely with CL4rec and underscoring the proficiency of graph networks in recommendation systems.
- In comparison to CL4rec, our findings substantiate that a graph structure tailored for sequential recommendation can notably enhance performance. LightGCN also eclipses BPRMF substantially, elevating the graph structure; the graph convolutional network unveils connections between users and items as their interaction is inherently graphical. Concurrently, the results show that our sequence-based graph construction method adeptly discern interactions between varied items by weighing the positioning of items within the sequence.

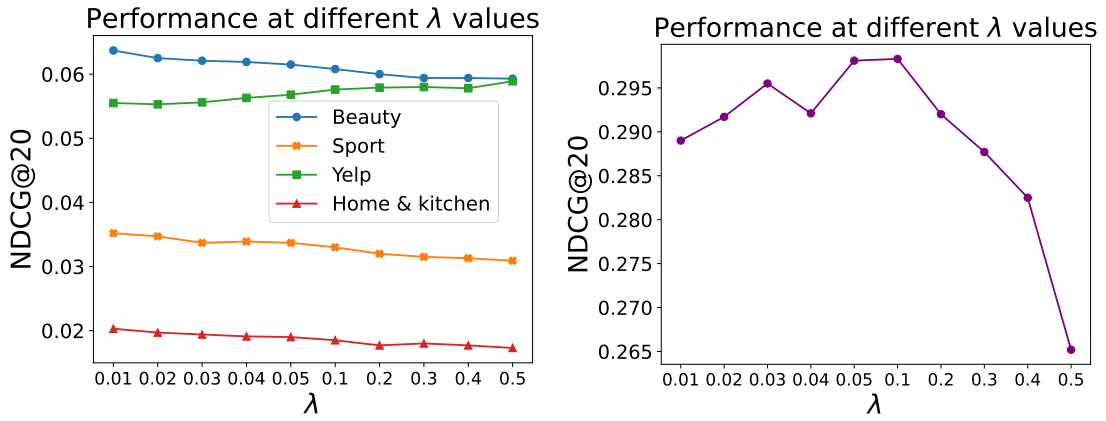
C. RQ2: ablation study

Next, we conduct an ablation study to test whether each proposed component positively contribute to the performance improvement or not.

To further comprehend the efficacy of our proposed model MVCrec, we compare it with three variants of our model: MVCrec(s), MVCrec(g) and MVCrec(mlp). MVCrec(s) employs a single contrastive learning approach based on *only*

TABLE II: Overall coverage where bold means the best performance and underline means the second-best performance. The p-value for the result is less than 0.01.

Dataset	Metric	BPRMF	LightGCN	GRU4rec	Caser	SASRec	BERT4Rec	SRGNN	CL4rec	MCLrec	MCLSR	DCrec	MVCrec	Improv.(%)
Sport	HR@5	0.0144	0.0171	0.0113	0.0060	0.0242	0.0222	0.0214	0.0258	0.0281	0.0186	0.0333	**0.0352**	5.71
	NDCG@5	0.0092	0.0107	0.0073	0.0043	0.0158	0.0147	0.0144	0.0171	0.0191	0.0123	0.0231	**0.0238**	3.03
	HR@10	0.0255	0.0289	0.0182	0.0092	0.0369	0.0351	0.0330	0.0403	0.0428	0.0287	0.0481	**0.0523**	8.73
	NDCG@10	0.0127	0.0146	0.0095	0.0053	0.0199	0.0189	0.0181	0.0218	0.0239	0.0155	0.0278	**0.0293**	5.4
	HR@20	0.0414	0.0471	0.0317	0.0138	0.0550	0.0527	0.0508	0.0607	0.0662	0.0444	0.0683	**0.0760**	11.27
	NDCG@20	0.0168	0.0191	0.0129	0.0065	0.0245	0.0233	0.0226	0.0269	0.0297	0.0195	0.0329	**0.0352**	6.99
Beauty	HR@5	0.0235	0.0262	0.0166	0.0107	0.0466	0.0439	0.0433	0.0516	0.0564	0.0275	0.0614	**0.0647**	5.37
	NDCG@5	0.0143	0.0165	0.0108	0.0068	0.0311	0.0291	0.0304	0.0354	0.0388	0.0189	0.0439	**0.0460**	4.78
	HR@10	0.0397	0.0433	0.0273	0.0174	0.0656	0.0643	0.0620	0.0749	0.0837	0.0410	0.0846	**0.0924**	9.22
	NDCG@10	0.0195	0.0220	0.0142	0.0089	0.0372	0.0356	0.0364	0.0428	0.0476	0.0233	0.0513	**0.0548**	6.82
	HR@20	0.0614	0.0695	0.0446	0.0267	0.0944	0.0935	0.0910	0.1068	0.1166	0.0639	0.1145	**0.1275**	11.35
	NDCG@20	0.0250	0.0286	0.0186	0.0113	0.0444	0.0430	0.0437	0.0509	0.0560	0.0290	0.0588	**0.0637**	8.33
Yelp	HR@5	0.0336	0.0502	0.0134	0.0060	0.0409	0.0419	0.0269	0.0447	0.0531	0.0491	0.0478	**0.0597**	12.43
	NDCG@5	0.0223	0.0357	0.0082	0.0043	0.0331	0.0337	0.0180	0.0328	0.0380	0.0342	0.0374	**0.0447**	17.63
	HR@10	0.0512	0.0730	0.0218	0.0092	0.0551	0.0562	0.0431	0.0642	0.0751	0.0714	0.0654	**0.0811**	7.99
	NDCG@10	0.0280	0.0430	0.0109	0.0053	0.0377	0.0383	0.0232	0.0391	0.0450	0.0414	0.0431	**0.0515**	14.44
	HR@20	0.0812	0.1060	0.0371	0.0138	0.0778	0.0800	0.0673	0.0938	0.1076	0.1025	0.0913	**0.1107**	2.88
	NDCG@20	0.0355	0.0513	0.0147	0.0065	0.0434	0.0443	0.0293	0.0466	0.0532	0.0492	0.0496	**0.0589**	10.71
Home & kitchen	HR@5	0.0054	0.0073	0.0039	0.0042	0.0113	0.0116	0.0066	0.0141	0.0153	0.0046	0.0198	**0.0207**	4.55
	NDCG@5	0.0035	0.0046	0.0024	0.0026	0.0074	0.0076	0.004	0.0096	0.0106	0.0029	0.0146	**0.0147**	0.68
	HR@10	0.0094	0.0122	0.0066	0.0072	0.0180	0.0172	0.0116	0.0211	0.0227	0.0079	0.0269	**0.0288**	7.06
	NDCG@10	0.0048	0.0061	0.0032	0.0036	0.0096	0.0094	0.0057	0.0118	0.0129	0.0039	0.0169	**0.0171**	1.18
	HR@20	0.0158	0.0202	0.0127	0.0129	0.0275	0.0265	0.0196	0.0307	0.0331	0.0144	0.0362	**0.0407**	12.43
	NDCG@20	0.0064	0.0082	0.0048	0.0050	0.0120	0.0117	0.0077	0.0142	0.0156	0.0056	0.0193	**0.0201**	4.15
Reddit	HR@5	0.2805	0.3103	0.1820	0.1662	0.2132	0.2221	0.2458	0.3072	0.3217	0.1767	0.3142	**0.3374**	4.88
	NDCG@5	0.2241	0.2444	0.1525	0.1418	0.1726	0.1756	0.1881	0.2436	0.2557	0.1459	0.2477	**0.2661**	4.07
	HR@10	0.3418	0.3744	0.2166	0.1920	0.2618	0.2750	0.3024	0.3769	0.3859	0.2117	0.3727	**0.4053**	5.03
	NDCG@10	0.2438	0.2650	0.1636	0.1501	0.1883	0.1928	0.2064	0.2561	0.2765	0.1571	0.2667	**0.2880**	4.16
	HR@20	0.4098	0.4526	0.2626	0.2267	0.3280	0.3444	0.3655	0.4483	0.4560	0.2615	0.4417	**0.4773**	4.67
	NDCG@20	0.2610	0.2847	0.1752	0.1588	0.2050	0.2103	0.2223	0.2741	0.2941	0.1696	0.2840	**0.3063**	4.15



(a) Amazon and Yelp

(b) Reddit

Fig. 2: Performance at different λ under NDCG@20.

TABLE III: Ablation study at HR@20 and NDCG@20.

Model		MVCrec	MVCrec(s)	MVCrec(g)	MVCrec(mlp)
Beauty	HR	0.1275	0.1068	0.1183	0.1017
	NDCG	0.0637	0.0509	0.0559	0.0489
Sport	HR	0.0760	0.0607	0.0705	0.0590
	NDCG	0.0352	0.0269	0.0319	0.0271
Yelp	HR	0.1107	0.0938	0.1020	0.0923
	NDCG	0.0589	0.0466	0.0523	0.0449
Home & Kitchen	HR	0.0407	0.0307	0.0362	0.0349
	NDCG	0.0201	0.0142	0.0164	0.0160
Reddit	HR	0.4773	0.4218	0.4348	0.4013
	NDCG	0.3063	0.2697	0.2780	0.2651

item-based sequence information without graph-based sequence information. MVCrec(g) denotes utilization of contrastive learning *solely on the graph-based sequence information* without item-based sequence information. MVCrec(mlp) denotes the use of multilayer perceptron (MLP) by concatenating representations of two views as input and then going through the MLP layers instead of our proposed multi-view attention fusion module. Following [14], we adopt HR@20 and NDCG@20 as evaluation metrics in the ablation study for simplification.

The results are presented in Table III. Analyzing the comparison between our model and three variants yields the following insights:

- A comparison between MVCrec(s) and MVCrec(g) reveals that the graph convolutional layer is more pivotal in terms of representing the history sequence. The information in the graph, constructed by the sequence data, encapsulates extensive user preference.
- Comparing MVCrec(s) and MVCrec shows that our full method significantly outperforms MVCrec(s) – analogous

to CL4rec – attributed to our novel multi-view attention fusion module that harnesses information from both graph and sequence structures to generate more expressive representations.

- Comparing MVCrec(mlp) and MVCrec shows that our proposed multi-view attention fusion module outperforms the MLP significantly, this is because the multi-view attention fusion module utilizes attention to weigh the importance of different views and their features dynamically.

D. RQ3: hyperparameter analysis

1) Hyperparameter Analysis on λ: In this section, we examine the impact of varying λ, a hyperparameter in Eq. 14. We assess the performance of MVCrec across five datasets using different values of λ. Because of the limited space, we report NDCG@20 as the evaluation metric, and the results are illustrated in Figure 2. In the Amazon datasets (i.e., Beauty, Sports, and Home & Kitchen), optimal performance is achieved when λ is set to 0.01. In the Yelp dataset, performance improves with increasing λ, reaching its highest at 0.5. For the Reddit dataset, the best performance is observed with λ set to 0.1. It means both recommendation loss and contrastive loss positively contributed to correctly estimate user-item matching scores and learn better representations. The discrepancy of optimal λ among the datasets can be potentially explained that Amazon dataset have smaller average historical sequence length than Yelp and Reddit datasets. Although we do not report HR@20, We observed similar trends in both HR@20 and NDCG@20.

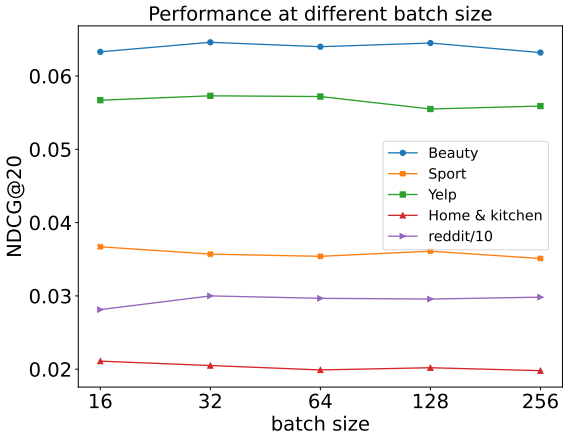


Fig. 3: Performance with different batch sizes under NDCG@20. The results from Reddit have been divided by 10 to ensure its line fits to the same figure with the other four datasets.

2) Hyperparameter Analysis on batch size and embedding size: In this section, we explore how a batch size and an embedding size impact the performance of our MVCrec model. We evaluate MVCrec across five datasets using various batch and embedding sizes. NDCG@20 serves as the main metric, similar to the previous section. The batch size ranges from 16 to 256. Results are illustrated in Figure 3. Optimal batch size varies by each dataset: 32 for Beauty, 16 for Sports, 32 for Yelp, 16 for Home & Kitchen, and 32 for Reddit.

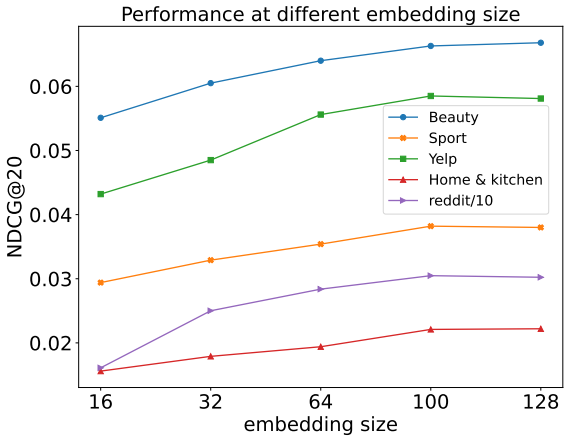


Fig. 4: Performance with different embedding sizes under NDCG@20. The results from Reddit have been divided by 10 to ensure its line fits to the same figure with the other four datasets.

The embedding size ranges from 16 to 128. Results are illustrated in Figure 4. We observe that larger embedding size generally enhances performance across all datasets. Although we do not report HR@20 because of the limited space, we observe that HR@20 has the same trend as NDCG@20 in these experiments.

VII. Conclusion

In this paper, we have proposed a novel contrastive learning framework. Our contrastive learning strategy integrated contrastive learning from two views (i.e., item-based sequence and graph-based sequence), enabling our model to learn better sequence representations. To combine the representations extracted from the two views, we employ the concept of multi-view attention fusion method, to generate/learn more expressive sequence representations. Extensive experiments across five benchmark datasets demonstrated the superiority of our model. In this work, we only used the sequence of consumed items without considering the actual time span between them. In the future, we will explore other potential contrastive learning methods based on the temporal sequence to learn even better user and item representations.

References

[1] W. Liu, X. Zheng, C. Chen, J. Su, X. Liao, M. Hu, and Y. Tan, "Joint internal multi-interest exploration and external domain alignment for cross domain sequential recommendation," in *WebConf*, 2023.

[2] J. Liang, X. Zhao, M. Li, Z. Zhang, W. Wang, H. Liu, and Z. Liu, "Mmmlp: Multi-modal multilayer perceptron for sequential recommendations," in *WebConf*, 2023.

[3] G. Lin, C. Gao, Y. Zheng, J. Chang, Y. Niu, Y. Song, Z. Li, D. Jin, and Y. Li, "Dual-interest factorization-heads attention for sequential recommendation," in *WebConf*, apr 2023.

[4] M. Li, Z. Zhang, X. Zhao, W. Wang, M. Zhao, R. Wu, and R. Guo, "Automlp: Automated mlp for sequential recommendations," in *WebConf*, 2023.

[5] H. Wang, F. Wu, Z. Liu, and X. Xie, "Fine-grained interest matching for neural news recommendation," in *ACL*, 2020.

[6] Y. Yang, C. Huang, L. Xia, Y. Liang, Y. Yu, and C. Li, "Multi-behavior hypergraph-enhanced transformer for sequential recommendation," in *KDD*, 2022.

[7] J. Li, M. Wang, J. Li, J. Fu, X. Shen, J. Shang, and J. McAuley, "Text is all you need: Learning language representations for sequential recommendation," in *KDD*, 2023.

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

[9] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *WSDM*, 2018.

[10] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *ICDM*, 2018.

[11] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *CIKM*, 2019.

[12] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, J. Zhang, B. Ding, and B. Cui, "Contrastive learning for sequential recommendation," in *ICDE*, 2022.

[13] R. Qiu, Z. Huang, H. Yin, and Z. Wang, "Contrastive learning for representation degeneration problem in sequential recommendation," in *WSDM*, 2022.

[14] X. Qin, H. Yuan, P. Zhao, J. Fang, F. Zhuang, G. Liu, Y. Liu, and V. Sheng, "Meta-optimized contrastive learning for sequential recommendation," in *SIGIR*, 2023.

[15] Y. Chen, Z. Liu, J. Li, J. McAuley, and C. Xiong, "Intent contrastive learning for sequential recommendation," in *WebConf*, 2022.

[16] Z. Liu, Y. Chen, J. Li, P. S. Yu, J. McAuley, and C. Xiong, "Contrastive self-supervised sequential recommendation with robust augmentation," *arXiv preprint arXiv:2108.06479*, 2021.

[17] Z. Liu, Y. Chen, J. Li, M. Luo, P. S. Yu, and C. Xiong, "Improving contrastive learning with model augmentation," *arXiv preprint arXiv:2203.15508*, 2022.

[18] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[19] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *SIGIR*, 2020.

[20] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *SIGIR*, 2019.

[21] K. Mao, J. Zhu, X. Xiao, B. Lu, Z. Wang, and X. He, "Ultragcn: Ultra simplification of graph convolutional networks for recommendation," in *CIKM*, 2021.

[22] Y. Yang, Z. Wu, L. Wu, K. Zhang, R. Hong, Z. Zhang, J. Zhou, and M. Wang, "Generative-contrastive graph learning for recommendation," in *SIGIR*, 2023.

[23] W. He, G. Sun, J. Lu, and X. S. Fang, "Candidate-aware graph contrastive learning for recommendation," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 1670–1679.

[24] Y. Ye, L. Xia, and C. Huang, "Graph masked autoencoder for sequential recommendation," in *SIGIR*, 2023.

[25] F. Garcin, C. Dimitrakakis, and B. Faltings, "Personalized news recommendation with context trees," in *Proceedings of the 7th ACM Conference on Recommender Systems*, 2013, pp. 105–112.

[26] S. Feng, X. Li, Y. Zeng, G. Cong, and Y. M. Chee, "Personalized ranking metric embedding for next new poi recommendation," in *IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence*. ACM, 2015, pp. 2069–2075.

[27] L. Liu, L. Cai, C. Zhang, X. Zhao, J. Gao, W. Wang, Y. Lv, W. Fan, Y. Wang, M. He, Z. Liu, and Q. Li, "Linrec: Linear attention mechanism for long-term sequential recommender systems," in *SIGIR*, 2023.

[28] K. Kim, D. Hyun, S. Yun, and C. Park, "Melt: Mutual enhancement of long-tailed user and item for sequential recommendation," in *SIGIR*, ser. SIGIR '23, 2023.

[29] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[30] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[31] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[32] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[33] L. Zhang, X. Zhou, and Z. Shen, "Multimodal pre-training framework for sequential recommendation via contrastive learning," *arXiv preprint arXiv:2303.11879*, 2023.

[34] W. Wei, C. Huang, L. Xia, and C. Zhang, "Multi-modal self-supervised learning for recommendation," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 790–800.

[35] H. Du, H. Yuan, P. Zhao, F. Zhuang, G. Liu, L. Zhao, and V. S. Sheng, "Ensemble modeling with contrastive knowledge distillation for sequential recommendation," 2023.

[36] Y. Yang, C. Huang, L. Xia, C. Huang, D. Luo, and K. Lin, "Debiased contrastive learning for sequential recommendation," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 1063–1073.

[37] Y. Wu, R. Xie, Y. Zhu, X. Ao, X. Chen, X. Zhang, F. Zhuang, L. Lin, and Q. He, "Multi-view multi-behavior contrastive learning in recommendation," in *International conference on database systems for advanced applications*. Springer, 2022, pp. 166–182.

[38] Z. Wang, H. Liu, W. Wei, Y. Hu, X.-L. Mao, S. He, R. Fang, and D. Chen, "Multi-level contrastive learning framework for sequential recommendation," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 2098–2107.

[39] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, p. 346–353, Jul. 2019.

[40] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 27–34.

[41] C. Wang, W. Ma, C. Chen, M. Zhang, Y. Liu, and S. Ma, "Sequential recommendation with multiple contrast signals," *ACM Transactions on Information Systems*, vol. 41, no. 1, pp. 1–27, 2023.

[42] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, ser. UAI '09. Arlington, Virginia, USA: AUAI Press, 2009, p. 452–461.

[43] Z. Fan, Z. Liu, H. Peng, and P. S. Yu, "Mutual wasserstein discrepancy minimization for sequential recommendation," in *WebConf*, 2023.

[44] Y. Lin, C. Wang, Z. Chen, Z. Ren, X. Xin, Q. Yan, M. de Rijke, X. Cheng, and P. Ren, "A self-correcting sequential recommender," in *WebConf*, 2023.

[45] C. Huang, S. Wang, X. Wang, and L. Yao, "Modeling temporal positive and negative excitation for sequential recommendation," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 1252–1263.

[46] Z. He, H. Zhao, Z. Wang, Z. Lin, A. Kale, and J. Mcauley, "Query-aware sequential recommendation," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4019–4023.

[47] Y. Hou, Z. He, J. McAuley, and W. X. Zhao, "Learning vector-quantized item representation for transferable sequential recommenders," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 1162–1171.

[48] Z. Fan, Z. Liu, Y. Wang, A. Wang, Z. Nazari, L. Zheng, H. Peng, and P. S. Yu, "Sequential recommendation via stochastic self-attention," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 2036–2047.

[49] K. Lin, Z. Wang, S. Shen, Z. Wang, B. Chen, and X. Chen, "Sequential recommendation with decomposed item feature routing," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 2288–2297.

[50] Z. Li, X. Wang, C. Yang, L. Yao, J. McAuley, and G. Xu, "Exploiting explicit and implicit item relationships for session-based recommendation," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 2023, pp. 553–561.