Multi-View Contrastive Learning for Sequential Recommendation

by

Xiaofan Zhou

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Data science

by

May 2024

APPROVED:

Professor Kyumin Lee, Advisor

Professor Roee Shraga, Reader

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 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, researchers have not paid attention on multi view contrastive learning between the ID and graph to further improve quality of user and item representation learning. Therefore, in this study, we propose a novel framework called **M**ulti**V**iew **C**ontrastive learning for sequential **rec**ommendation (MVCrec). This framework is designed to combine information from both sequential 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 underscore the superiority of our model, as evidenced by its performance on four real-world benchmark datasets and ablation study. Our code and datasets are available at `https://anonymous.4open.science/r/MMCrec-06BA`.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Kyumin Lee, for his invaluable guidance, patience, and support throughout the course of this research. His insight and expertise have been crucial in the shaping of this thesis, and his encouragement was vital in the face of challenges.

I am also immensely grateful to Prof. Roee Shraga, whose expertise and feedback were indispensable throughout the reviewing process. His contributions have significantly enhanced the quality of my work.

Special thanks go to the faculty and staff of data science program at WPI, whose resources and support have been a cornerstone of my research journey.

I would also like to acknowledge the members of InfoLab at Worcester PolytechnicInstitute, who constantly shared their knowledge.

I owe a debt of gratitude to my family, especially my parents for their understanding, love, and endless support. Their belief in my abilities and their unwavering encouragement have been a source of strength and motivation.

Contents

Chapter 1

Introduction

Sequential recommendation has gotten increasing attention from both industry and academic, with the primary focus being on recommending items based on users' chronologically ordered purchase histories as shown in Figure 1.1 [1]–[7]. In the early stage, researchers applied recurrent neural network (RNN) and convolutional neural network (CNN) to sequential recommendation (e.g., GRU4Rec [8], Caser [9], and SAS4Rec [10]). Additionally, self-supervised methods have been employed in sequential recommendation; for example, BERT4Rec [11] utilizes BERT[12] as an encoder for sequential lists. More recently, contrastive learning and related techniques have been widely adopted in sequential recommendation to enhance the effectiveness of learned representations (e.g., CL4rec [13], Duorec[14] and MCLrec[15]).

However, the utilization of contrastive learning (CL)[16]–[19] 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

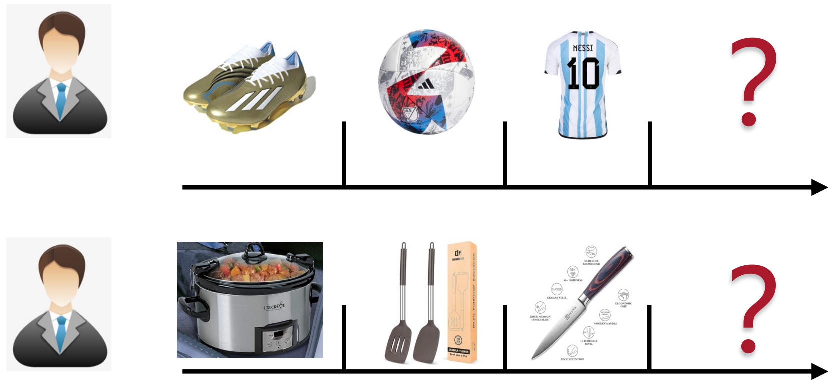

Figure 1.1: A toy example of sequential recommendation. The first user purchased football shoes, football and football t-shirt, and the second user purchased pot, spatula, knife. The goal of sequential recommendation is to predict what item each user will purchase next given the user's historical sequence.

operations have been established. The first class generates different views of the same sequence through random operations like 'masking', 'cropping', or 'reordering' items [20], [21]. The second class uses variable dropout probabilities at the model level to create different views of the same sequential data [14]. The third class combines 'neural mask', 'layer drop', and 'encoder complement' with data augmentation techniques for constructing positive and negative view pairs [22].

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 [17] 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 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 and LightGCN [23], [24] integrate graph convolutional networks into the recommendation systems. UltraGCN [25] simplifies GCNs for collaborative filtering by omitting feature transformations and nonlinear activations. As contrastive learning has developed, VGCL [26] 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 [27] 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 [28] 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[29] between graph and sequence information remains an unexplored area in sequential recommendation, especially, when given only interaction data without any auxiliary information.

To fill the gap, in this thesis, 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 summary, the major contributions of MVCrec are as follows:

To the best our knowledge, we are the first to attempt to propose a novel multi-view contrastive learning approach in the sequential recommendation domain.

- The proposed model proficiently extracts pertinent information from both positive and negative samples, using sequence and graph views extracted from users' historical item lists (i.e., prior interaction data).
- A multi-view attention fusion module is seamlessly integrated within MVCrec to calculate the recommendation score, drawing upon representations from disparate views.
- Via comprehensive experiments across four public benchmark datasets, we substantiate that MVCrec outperforms ten state-of-the-art baselines.

Chapter 2

Related work

2.1 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 [30], [31].

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 [32] introduces a novel method that enhances efficiency while retaining the learning capabilities of traditional dot-product attention through a linear attention module. MELT [33] mutually enhance user and item bilateral branches to deal with long-tailed problem.

Recently, contrastive learning has been used in sequential recommendation to handle issues like not having enough data and having data that's noisy. CL4rec [13] 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 [14] 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 [15] 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 [34] proposes knowledge distillation which use contrastive learning to facilitate knowledge transfer between parallel networks, and use the ensemble of different models as the final prediction. Lastly, DCrec [35] introduces a new global learning strategy to deal with popularity bias in sequential recommendation.

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.

2.2 Contrastive learning

To enable deep learning models to more accurately differentiate instances pertaining to distinct individuals, contrastive learning was introduced in [16]. 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 [17] introduced the use of mutual information to quantify the similarity between two individuals, considering different views of the same individual as positive pairs. Subsequently, [18] employed a queue to manage the extensive dictionary associated with contrastive learning, while [19] 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, [36] explored the execution of contrastive learning tasks without the incorporation of negative samples. In multi-view contrastive learning, MSM4SR [37] proposes the fusion of text and image views prior to contrastive learning. However, this approach overlooks the interrelationship of cross-view contrastive learning. On the other hand, MMSSL [38] suggests using GCN for cross-view contrastive learning, but it doesn't account for sequential data. In this thesis, the principle of contrastive learning is adapted to extract superior representations of historical interaction sequences, and a new multi-view contrastive learning approach is proposed.

2.3 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 and LightGCN [23], [24] have advanced the field of recommendation by integrating GCN structures, thus, enhancing the developmental trajectory of recommendation systems. UltraGCN [25] further refines the approach by streamlining GCNs for collaborative filtering and omitting unnecessary feature transformations and nonlinear activations. Additionally, works like CGCL [27] and VGCL [26] have applied graph structures to contrastive learning, utilizing auto-encoders to optimize the process. At 2019, SRGNN [39] is proposed to use GNN structure to train the sequential recommendation. Within the realm of sequential recommendation, MAErec [28] ingeniously employs graph data in contrastive learning to address issues related to label scarcity. In this thesis, we also construct a graph for items to learn their embeddings and user preference representation from the historical sequence via multi-view contrastive learning.

Chapter 3

Proposed Method

3.1 Problem definition

The primary objective of this thesis 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.

3.2 Overview

As depicted in Figure 3.1, 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 [13]. 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

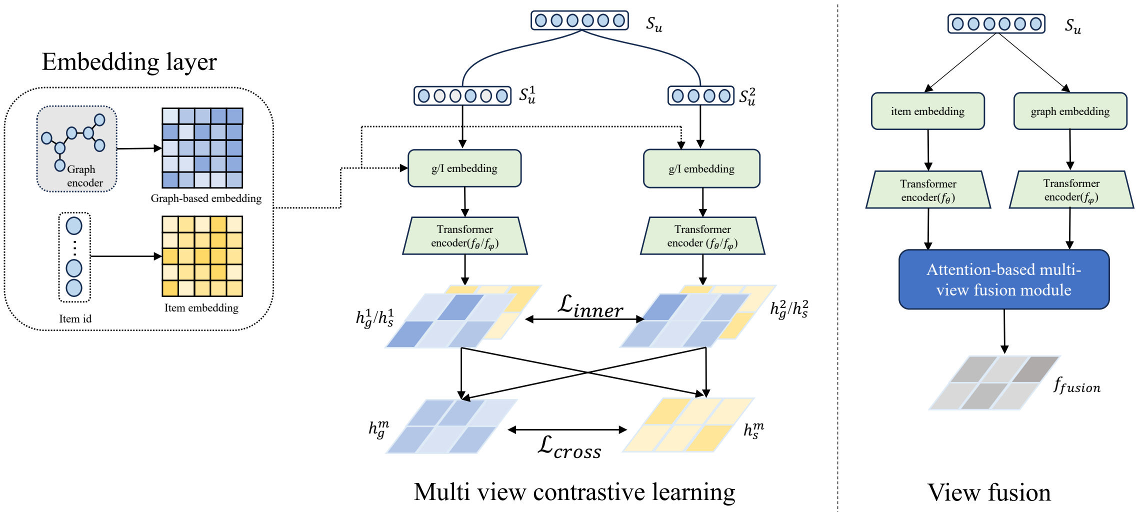

Figure 3.1: Our proposed framework, MVCrec, consists of multi-view contrastive learning and multi-view attention fusion module.

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.

3.3 Stochastic data augmentation

This module aims to generate two positive views for each historical sequence. Inspired by CL4rec [13], we apply three stochastic data augmentations — ‘masking’, ‘cropping’, and ‘reordering’ — to the historical sequence. The procedure for generating two augmented sequences is as follows:

$$\tilde{S}_1^u = g_1\left(S^u\right), \tilde{S}_2^u = g_2\left(S^u\right) \tag{3.1}$$

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.

3.4 Two types of item embedding

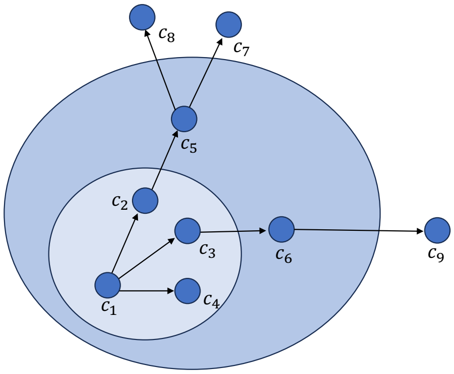

Figure 3.2: Illustration of the GCN-based graph encoder.

Initially, we project all items into a common embedding space [10]. In this thesis, 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.

In particular, for the GCN-based graph encoder, we draw upon the concepts presented in [23], [40]. The structure of the GCN is shown in Figure 3.2. 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. Then the computation within the GCN-based graph encoder proceeds as follows:

$$\mathbf{m}_i^{l+1} = \mathbf{m}_i^l + \sum_{i' \in \mathcal{N}_i} \mathbf{m}_{i'}^l; \quad \tilde{\mathbf{m}}_i = \sum_{l=0}^{L} \mathbf{m}_i^l \tag{3.2}$$

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.

3.5 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:

$$\begin{aligned} e_s^u &= [m_{s_1} + p_1, m_{s_2} + p_2, ..., m_{s_n} + p_n]. \\ e_g^u &= [m_{g_1} + p_1, m_{g_2} + p_2, ..., m_{g_n} + p_n]. \end{aligned} \tag{3.3}$$

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 subsection.

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^u_s) and the other one for the same sequence of graph-based embeddings (e^u_g). The sequence encoders are defined as f_θ and f_ϕ, respectively, where θ and ϕ represent each model's parameters. The output representation $H^u_s \in R^{n\times d}$ and $H^u_g \in R^{n\times d}$ are calculated as follows:

$$\begin{aligned} H^u_s &= f_\theta(e^u_s) \\ H^u_g &= f_\phi(e^u_g) \end{aligned} \tag{3.4}$$

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

3.6 Multi-view contrastive learning

Inner view contrastive learning. Inspired by CL4rec [13] and MCLrec [15], 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:

$$\begin{aligned}\mathcal{L}_{\text{con}}\left(h_n^1, h_n^2\right) = & -\log \frac{exp^{s\left(h_n^1, h_n^2\right)}}{exp^{s(h_n^1, h_n^2)} + \sum_{h_n \in \text{ neg}} exp^{s(h_n^1, h_n)}} \\ & -\log \frac{exp^{s\left(h_n^2, h_n^1\right)}}{exp^{s(h_n^2, h_n^1)} + \sum_{h_n \in \text{ neg}} exp^{s(h_n^2, h_n)}}\end{aligned} \tag{3.5}$$

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:

$$\mathcal{L}_{\text{Inner}} = \mathcal{L}_{\text{con}}\left(h_{s_n}^1, h_{s_n}^2\right) + \mathcal{L}_{\text{con}}\left(h_{g_n}^1, h_{g_n}^2\right) \tag{3.6}$$

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:

$$\begin{aligned}\mathcal{L}_{\text{cross}} = &- \log \frac{exp^{s\left(h_{s_n}^m, h_{g_n}^m\right)}}{exp^{s\left(h_{s_n}^m, h_{g_n}^m\right)} + \sum_{h \in \text{ neg2}} exp^{s\left(h_{s_n}^m, h\right)}} \\ &- \log \frac{exp^{s\left(h_{g_n}^m, h_{s_n}^m\right)}}{exp^{s\left(h_{g_n}^m, h_{s_n}^m\right)} + \sum_{h \in \text{ neg2}} exp^{s\left(h_{g_n}^m, h\right)}}\end{aligned} \tag{3.7}$$

where $neg2$ is the set containing negative sample pair's mean representations. $\mathcal{L}_{\text{cross}}$ 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:

$$\mathcal{L}_{\text{MM}} = \mathcal{L}_{\text{cross}} + \mathcal{L}_{\text{Inner}} \tag{3.8}$$

3.7 Multi-view attention fusion module & recommendation prediction

To further utilize the extracted representations, we propose multi-view attention fusion module, which is aimed at amalgamating information from two disparate view, namely item-based sequence and graph-based sequence.

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, $h_g \in R^{1\times d}$ and $h_s \in R^{1\times d}$, as depicted in Figure 3.1 (the rightmost figure) and Figure 3.3, we initially calculate the global attention score, $s_{\text{global}}^{\text{attention}}$, and the local attention score, $s_{\text{local}}^{\text{attention}}$:

$$\begin{aligned} s_{global}^{attention} &= \sigma\left((h_g + h_s) \otimes W_g\right) \\ s_{local}^{attention} &= \sigma\left((h_g + h_s) \otimes W_l\right) \end{aligned} \tag{3.9}$$

where $W_g \in R^{d\times d}$ and $W_l \in R^{d\times d}$ represent global and local weight matrices, respectively. d is the dimension of a view's sequence representation, and $\otimes$ denotes matrix product. σ represents an activation function. In this thesis, we employ ReLU as the activation function.

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

$$s^{attention} = \text{sigmoid}\left(s_{global}^{attention} \oplus s_{local}^{attention}\right) \tag{3.10}$$

where $\oplus$ represents the summation between the two vectors. We employ the sigmoid function to normalize the scores. These scores are considered as weights for different view representations.

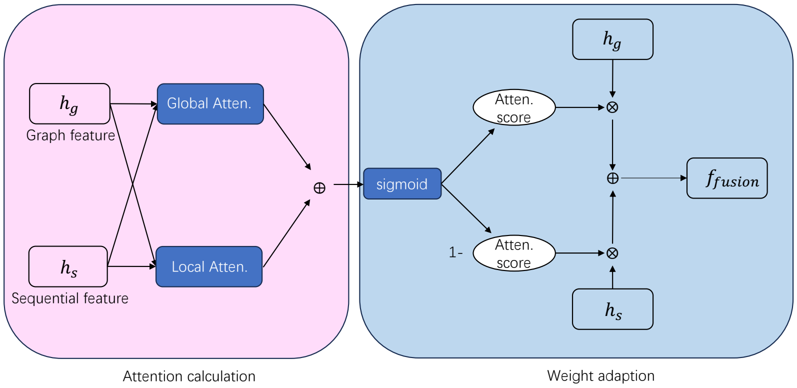

Figure 3.3: Multi-view attention fusion module which consists of two parts: learn (1) attention weights for two representations h_g and h_s, and (2) a new representation based on the attention weights.

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

$$f_{fusion}\left(h_s, h_g\right) = \left(s^{attention} \circ h_g\right) \oplus \left(\left(1 - s^{attention}\right) \circ h_s\right) \tag{3.11}$$

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:

$$\hat{y} = f_{fusion}\left(h_s, h_g\right) M_s^T + f_{fusion}\left(h_g, h_s\right) M_g^T \tag{3.12}$$

where h_s and h_g 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 3.4.

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

$$\mathcal{L}_{rec} = H(y, \hat{y}) = -\sum_{i} y_i \log(\hat{y}_i) \tag{3.13}$$

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

3.8 Overall Objective

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

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda \mathcal{L}_{MM} \tag{3.14}$$

where $\mathcal{L}_{rec}$ is the recommendation objective function in Eq. 3.13, $\mathcal{L}_{MM}$ represents multi-view contrastive loss, which consists of the inner-view contrastive loss and cross-view contrastive loss, as defined in Eq. 3.8, and λ is a hyperparameter.

Chapter 4

Experiment

In this section, we conduct extensive experiments using four 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?

4.1 Experimental settings

Dataset

To verify the effectiveness of our model, we evaluate its performance using four real-world benchmark datasets: Amazon (Beauty, Sports and Home & Kitchen)[1] and Yelp[2]. The Amazon datasets contain a series of Amazon product reviews. In our

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

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

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. In the following experiments, we only use interaction data without any auxiliary data (e.g., text, image). Following the preprocessing steps described in [21], [41], we removed users and items with fewer than five interactions. The statistics of the datasets are summarized in Table 4.1. To be specific, sparsity means the ratio of user and item pair do not appear in data.

Table 4.1: The statistics of datasets.

Dataset	#users	#items	#interactions	avg.length	SD	sparsity
Sports	33.6K	18.3K	296.3K	8.3	6.06	99.95%
Beauty	22.3K	12.1K	198.5K	8.8	8.16	99.93%
Yelp	30.4K	20.0K	316.3K	10.4	10.38	99.95%
Home & Kitchen	66.5K	28.2K	551.6K	8.3	6.81	99.97%

Baselines

We compare our model with ten 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* [23] 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 history 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* [13] leverages contrastive learning on the sequential recommendation.
- *MCLrec* [15] innovates the standard contrastive learning framework by contrasting data, and models augmented views for adaptively capturing the informative features hidden in stochastic data augmentation.
- *DCrec* [35] 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 [38], MSM4SR [37]) 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

Table 4.2: Overall performance where bold means the best performance and underline means the second-best performance.

Dataset	Metric	BPRMF	LightGCN	GRU4rec	Caser	SASRec	BERT4Rec	SRGNN	CL4rec	MCLrec	DCrec	MVCrec	Improv.(%)
Sport	HIT@5	0.0144	0.0171	0.0113	0.0060	0.0242	0.0222	0.0214	0.0258	0.0281	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	0.0231	**0.0238**	3.03
	HIT@10	0.0255	0.0289	0.0182	0.0092	0.0369	0.0351	0.0330	0.0403	0.0428	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	0.0278	**0.0293**	5.4
	HIT@20	0.0414	0.0471	0.0317	0.0138	0.0550	0.0527	0.0508	0.0607	0.0662	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	0.0329	**0.0352**	6.99
Beauty	HIT@5	0.0235	0.0262	0.0166	0.0107	0.0466	0.0439	0.0433	0.0516	0.0564	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	0.0439	**0.0460**	4.78
	HIT@10	0.0397	0.0433	0.0273	0.0174	0.0656	0.0643	0.0620	0.0749	0.0837	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	0.0513	**0.0548**	6.82
	HIT@20	0.0614	0.0695	0.0446	0.0267	0.0944	0.0935	0.0910	0.1068	0.1166	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	0.0588	**0.0637**	8.33
Yelp	HIT@5	0.0336	0.0502	0.0134	0.0060	0.0409	0.0419	0.0269	0.0447	0.0531	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	0.0374	**0.0447**	17.63
	HIT@10	0.0512	0.0730	0.0218	0.0092	0.0551	0.0562	0.0431	0.0642	0.0751	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	0.0431	**0.0515**	14.44
	HIT@20	0.0812	0.1060	0.0371	0.0138	0.0778	0.0800	0.0673	0.0938	0.1076	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	0.0496	**0.0589**	10.71
Home & Kitchen	HIT@5	0.0054	0.0073	0.0039	0.0042	0.0113	0.0116	0.0066	0.0141	0.0153	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	0.0146	**0.0147**	0.68
	HIT@10	0.0094	0.0122	0.0066	0.0072	0.0180	0.0172	0.0116	0.0211	0.0227	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	0.0169	**0.0171**	1.18
	HIT@20	0.0158	0.0202	0.0127	0.0129	0.0275	0.0265	0.0196	0.0307	0.0331	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	0.0193	**0.0201**	4.15

recommendation models or sometimes it is hard to run some of their models without auxiliary data.

Evaluation metric

In accordance with [13], [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 $k \in \{10, 20\}$.

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. A graph for the graph encoder is constructed based on the training set. To maintain 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 distance z as 3. 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. The models are optimized using the Adam optimizer [26] and are trained with an early stopping strategy based on the performance of the validation set, with the maximum step set to 100. Cause our model use some different setting and representation encoder, we have different result with it in [15]. All experiments are conducted on a Tesla T4 GPU.

4.2 RQ1: overall performance

To elucidate the contributions made by MVCrec, we compare its performance with the baselines. The results presented in Table 4.2 lead us to several insights:

- Self-supervised models 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 indicates the capability of contrastive learning 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.
- Compared to graph-based sequential methods or general recommendation methods like SRGNN and LightGCN, our model demonstrates superior performance. Interestingly, SRGNN tends to outperform LightGCN in most datasets. These results underscore the efficacy of graph-based sequential methods. Crucially, the enhanced performance of our model can be attributed to the multi-view strategy, effectively integrating relationship information with ID-specific data.
- 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%∼14.44% on NDCG@10 over the four 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.

4.3 RQ2: ablation study

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

Table 4.3: 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

Quantitative Analysis

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 item-based sequence information. MVCrec(g) denotes utilization of contrastive learning solely on the graph-based sequence information. MVCrec(mlp) denotes the use of multilayer perceptron (MLP) instead of multi-view attention fusion module to fuse representations of two views (concatenate them and go through neural layers). Following [15], we adopt Hit Ratio@20 and NDCG@20 as evaluation metrics in the ablation study for simplification.

The results are presented in Table 4.3. 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. The performance of MVCrec(g) even surpasses or is competitive with MCLrec (the second best baseline) on the Beauty and Sport datasets.
- Comparing MVCrec(s) and MVCrec shows that our proposed 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. This is further confirmed by comparing the performance of MVCrec(g) and MVCrec. While MVCrec(g) attains robust results owing to the graph convolutional encoder, MVCrec surpasses its performance, affirming the efficacy of our proposed framework.
- 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.

Qualitative Analysis

To further elucidate the effectiveness of our model qualitatively, we visualize the learned h_s, h_g, and $f_{\text{fusion}}(h_g, h_s)$ using T-SNE[51]. Within Figure 4.1, “only h_s” represents user preference learned by MVCrec(s), “only h_g” represents user preference

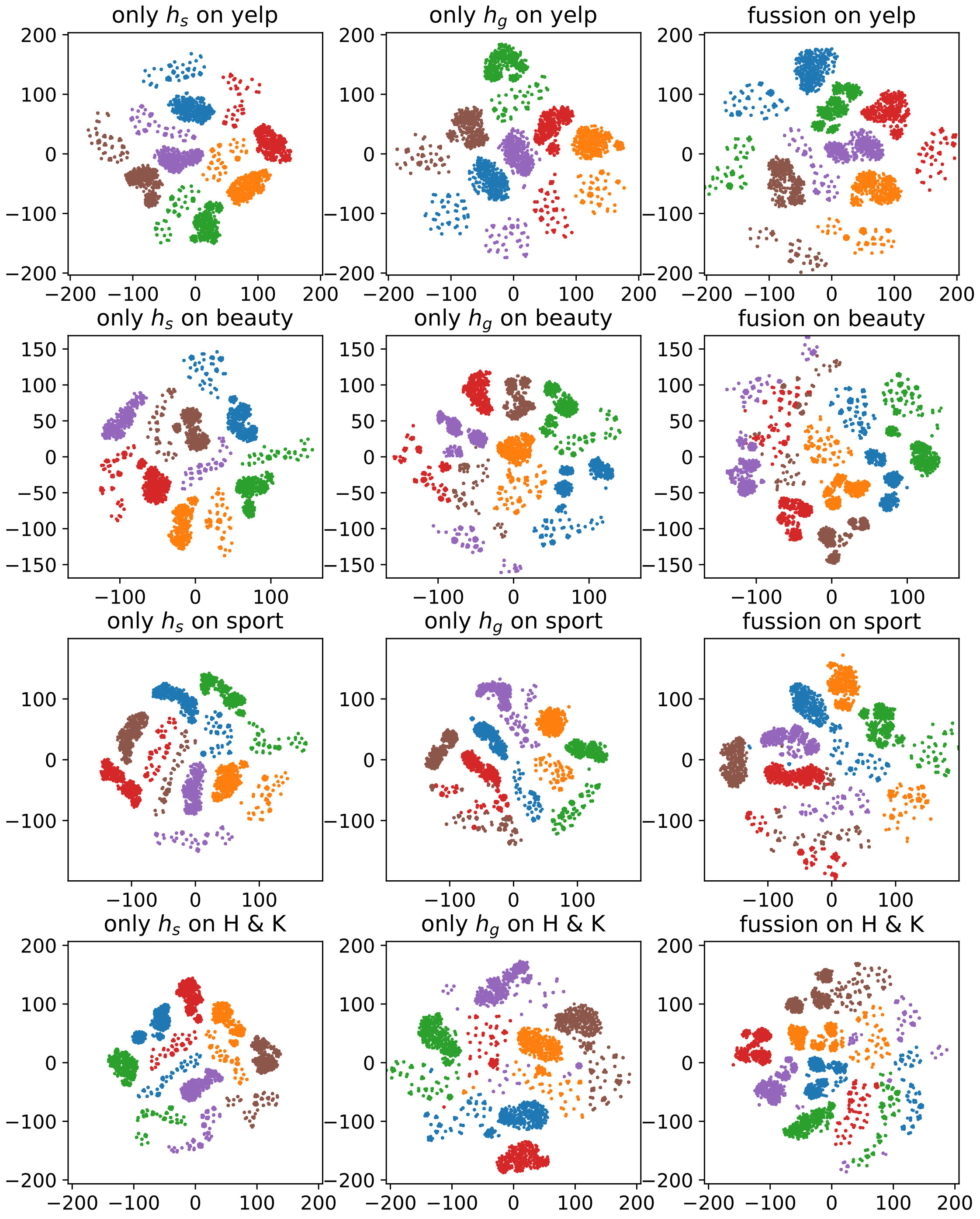

Figure 4.1: T-SNE visualization of sequence representations of the six groups in MVCrec(s), MVCrec(g), and MVCrec(H & K represents the result of Home & Kitchen dataset.).

learned by MVCrec(g), and "fusion" represents $f_{\text{fusion}}(h_g, h_s)$ learned by MVCrec presented in Eq. 3.12. To draw the figure, we first randomly selected 6 historical sequences from each dataset, where each of them consists of recent 50 items so that they can generate enough augmented samples. For each historical sequence, the stochastic data augmentation module at Section 3.3 was used to generate 499 augmented sequences, setting the crop and mask ratios as 0.5 and the max reorder length ratio as 0.8 in order to ensure the augmented sequences are similar to but still different from the original historical sequence. We considered these 499 augmented sequences and 1 original historical sequence as the same group, and the other 2,500 sequences (2,495 augmented samples + 5 original historical sequences) as negative samples. In total, there are six groups and each of which has different color in Figure 4.1. We projected each sequence's representation via T-SNE in the figure. We observe that sequence representations of the different colors, learned by MVCrec(s) and MVCrec(g), are relatively closer than ones learned from MVCrec. In other words, the representations produced by the multi-view fusion module in the different colors are more widely scattered. This indicates that our model is inclined to generate more distinct representations for negative pairs. The figure underscores that the proposed multi-view strategy and the multi-view attention fusion module enhance the efficiency of collaborative learning. This improvement stems from the incorporation of various views in collaborative learning, leading to the generation of more informative representations.

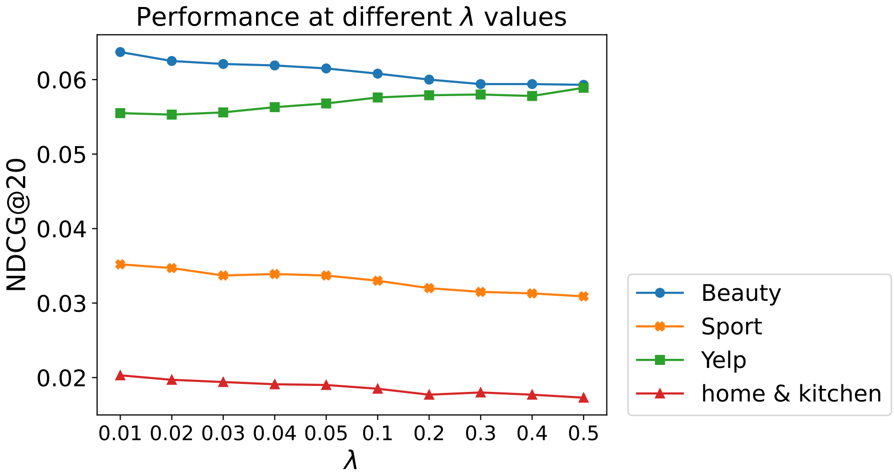

Figure 4.2: Performance at different λ under NDCG@20 on test dataset.

4.4 RQ3: hyperparameter analysis

Hyperparameter Analysis on λ

In this section, we examine the impact of varying λ, a hyperparameter in Eq. 3.14. We assess the performance of MVCrec across four datasets using different values of λ. For simplicity, we employ NDCG@20 as the evaluation metric, and the results are illustrated in Figure 4.2. In the Amazon datasets (i.e., Beauty, Sports datasets, and Home & Kitchen), optimal performance is achieved when λ is set at 0.01, while in the Yelp dataset, performance increases as we increase λ (the highest performance is achieved in 0.5). 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 and Yelp have different user distribution and user behavior patterns.

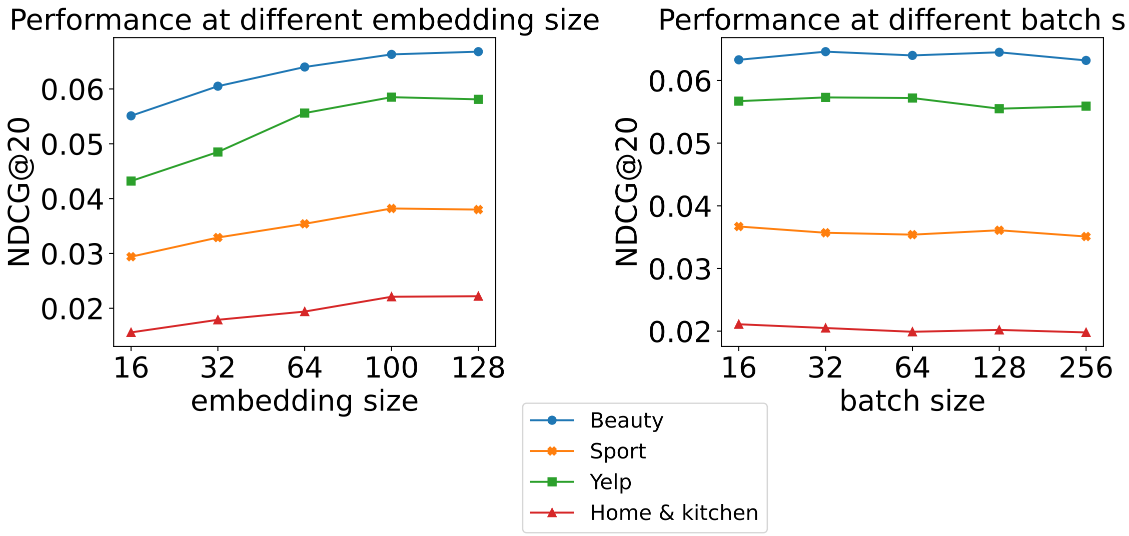

Figure 4.3: Performance at different batch size and embedding size under NDCG@20 on test dataset.

Hyperparameter Analysis on a batch size and an 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 four 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, while the embedding size ranges from 16 to 128. Results are illustrated in Figure 4.3. Optimal batch size varies by each dataset: 32 for Beauty, 16 for Sports, 32 for Yelp, and 16 for Home & Kitchen. We observe that larger embedding size generally enhances performance across all datasets.

Time complexity analysis

In this section, we discuss time complexity of our model in the training and testing phases. In the training phase, we jointly train the model for the contrastive learning loss function and the recommendation loss function, aiming to optimize θ, ϕ and attention-based multi-view fusion module. The time complexity is $O(|U|^2 d + |U| d^2)$, where $|U|$ represents the number of users and d represents an embedding size of the model. In the testing phase, the time complexity of our model is similar to basic sequential recommendation methods, as we only need to use the multi-view attention fusion module, transformer methods and graph convolutional layer. The time complexity is $O(d|I|)$, where $|I|$ represents the number of items. This analysis shows that our MVCrec model attains time complexity on par with state-of-the-art sequential recommendation methods [15], [20].

Chapter 5

Conclusion

In this thesis, 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 four benchmark datasets demonstrated the superiority of our model. In this thesis, we only used the sequence of consumed items without considering actual time span between the consumed items. In the future, we will explore another possible contrastive learning based on the temporal sequence to learn even better user and item representations.

Bibliography

[1] W. Liu, X. Zheng, C. Chen, *et al.*, "Joint internal multi-interest exploration and external domain alignment for cross domain sequential recommendation," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 383–394.

[2] J. Liang, X. Zhao, M. Li, *et al.*, "Mmmlp: Multi-modal multilayer perceptron for sequential recommendations," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 1109–1117.

[3] G. Lin, C. Gao, Y. Zheng, *et al.*, "Dual-interest factorization-heads attention for sequential recommendation," in *Proceedings of the ACM Web Conference 2023*, ACM, Apr. 2023. DOI: 10.1145/3543507.3583278. [Online]. Available: https://doi.org/10.1145%2F3543507.3583278.

[4] M. Li, Z. Zhang, X. Zhao, *et al.*, "Automlp: Automated mlp for sequential recommendations," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 1190–1198.

[5] H. Wang, F. Wu, Z. Liu, and X. Xie, "Fine-grained interest matching for neural news recommendation," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 836–845.

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

ceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, Aug. 2022. DOI: 10.1145/3534678.3539342. [Online]. Available: https://doi.org/10.1145%2F3534678.3539342.

[7] J. Li, M. Wang, J. Li, *et al.*, "Text is all you need: Learning language representations for sequential recommendation," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '23, Long Beach, CA, USA: Association for Computing Machinery, 2023, pp. 1258–1267, ISBN: 9798400701030. DOI: 10.1145/3580305.3599519. [Online]. Available: https://doi.org/10.1145/3580305.3599519.

[8] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1511.06939.

[9] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the eleventh ACM international conference on web search and data mining*, 2018, pp. 565–573.

[10] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE international conference on data mining (ICDM)*, IEEE, 2018, pp. 197–206.

[11] F. Sun, J. Liu, J. Wu, *et al.*, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. [Online]. Available: https://aclanthology.org/N19-1423.

[13] X. Xie, F. Sun, Z. Liu, *et al.*, "Contrastive learning for sequential recommendation," in *2022 IEEE 38th international conference on data engineering (ICDE)*, IEEE, 2022, pp. 1259–1273.

[14] R. Qiu, Z. Huang, H. Yin, and Z. Wang, "Contrastive learning for representation degeneration problem in sequential recommendation," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, ACM, Feb. 2022. DOI: 10.1145/3488560.3498433. [Online]. Available: https://doi.org/10.1145%2F3488560.3498433.

[15] X. Qin, H. Yuan, P. Zhao, *et al.*, "Meta-optimized contrastive learning for sequential recommendation," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Jul. 2023. DOI: 10.1145/3539618.3591727. [Online]. Available: https://doi.org/10.1145%2F3539618.3591727.

[16] 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)*, Jun. 2018.

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

[18] 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.

[19] 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.

[20] Y. Chen, Z. Liu, J. Li, J. McAuley, and C. Xiong, "Intent contrastive learning for sequential recommendation," in *Proceedings of the ACM Web Conference 2022*, ACM, Apr. 2022. DOI: `10.1145/3485447.3512090`. [Online]. Available: `https://doi.org/10.1145%2F3485447.3512090`.

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

[22] 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.

[23] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20, Virtual Event, China: Association for Computing Machinery, 2020, pp. 639–648, ISBN: 9781450380164. DOI: `10.1145/3397271.3401063`. [Online]. Available: `https://doi.org/10.1145/3397271.3401063`.

[24] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, “Neural graph collaborative filtering,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Jul. 2019. DOI: 10.1145/3331184.3331267. [Online]. Available: https://doi.org/10.1145%2F3331184.3331267.

[25] K. Mao, J. Zhu, X. Xiao, B. Lu, Z. Wang, and X. He, “Ultragcn: Ultra simplification of graph convolutional networks for recommendation,” in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, ser. CIKM ’21, Virtual Event, Queensland, Australia: Association for Computing Machinery, 2021, pp. 1253–1262, ISBN: 9781450384469. DOI: 10.1145/3459637.3482291. [Online]. Available: https://doi.org/10.1145/3459637.3482291.

[26] Y. Yang, Z. Wu, L. Wu, *et al.*, “Generative-contrastive graph learning for recommendation,” in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’23, Taipei, Taiwan: Association for Computing Machinery, 2023, pp. 1117–1126, ISBN: 9781450394086. DOI: 10.1145/3539618.3591691. [Online]. Available: https://doi.org/10.1145/3539618.3591691.

[27] 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.

[28] Y. Ye, L. Xia, and C. Huang, “Graph masked autoencoder for sequential recommendation,” in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Jul. 2023.

DOI: 10.1145/3539618.3591692. [Online]. Available: https://doi.org/10.1145%2F3539618.3591692.

[29] D. Zou, W. Wei, X.-L. Mao, *et al.*, "Multi-level cross-view contrastive learning for knowledge-aware recommender system," in *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 2022, pp. 1358–1368.

[30] 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.

[31] 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.

[32] L. Liu, L. Cai, C. Zhang, *et al.*, "Linrec: Linear attention mechanism for long-term sequential recommender systems," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '23, ¡conf-loc¿, ¡city¿Taipei¡/city¿, ¡country¿Taiwan¡/country¿, ¡/conf-loc¿: Association for Computing Machinery, 2023, pp. 289–299, ISBN: 9781450394086. DOI: 10.1145/3539618.3591717. [Online]. Available: https://doi.org/10.1145/3539618.3591717.

[33] K. Kim, D. Hyun, S. Yun, and C. Park, "Melt: Mutual enhancement of long-tailed user and item for sequential recommendation," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '23, ACM, Jul. 2023. DOI: 10.1145/3539618.

3591725. [Online]. Available: http://dx.doi.org/10.1145/3539618.3591725.

[34] H. Du, H. Yuan, P. Zhao, *et al.*, *Ensemble modeling with contrastive knowledge distillation for sequential recommendation*, 2023. arXiv: 2304.14668 [cs.IR].

[35] 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.

[36] J.-B. Grill, F. Strub, F. Altché, *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.

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

[38] 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.

[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, pp. 346–353, Jul. 2019, ISSN: 2159-5399. DOI: 10.1609/aaai.v33i01.3301346. [Online]. Available: http://dx.doi.org/10.1609/aaai.v33i01.3301346.

[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, 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, Montreal, Quebec, Canada: AUAI Press, 2009, pp. 452–461, ISBN: 9780974903958.

[43] Z. Fan, Z. Liu, H. Peng, and P. S. Yu, “Mutual wasserstein discrepancy minimization for sequential recommendation,” in *Proceedings of the ACM Web Conference 2023*, ser. WWW ’23, Austin, TX, USA: Association for Computing Machinery, 2023, pp. 1375–1385, ISBN: 9781450394161. DOI: `10.1145/3543507.3583529`. [Online]. Available: `https://doi.org/10.1145/3543507.3583529`.

[44] Y. Lin, C. Wang, Z. Chen, *et al.*, “A self-correcting sequential recommender,” in *Proceedings of the ACM Web Conference 2023*, ser. WWW ’23, Austin, TX, USA: Association for Computing Machinery, 2023, pp. 1283–1293, ISBN: 9781450394161. DOI: `10.1145/3543507.3583479`. [Online]. Available: `https://doi.org/10.1145/3543507.3583479`.

[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, *et al.*, “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.

[51] L. Van Der Maaten, “Accelerating t-sne using tree-based algorithms,” *The journal of machine learning research*, vol. 15, no. 1, pp. 3221–3245, 2014.