# Efficient Graph Collaborative Filtering with Multi-layer Output-Enhanced Contrastive Learning

Keke Li, Shaoqing Wang(✉), Shun Zheng, Xia Wu, Yao Zhang, and Fuzhen Sun

School of Computer Science and Technology, Shandong University of Technology, Zibo 255091, China
lyscoke@163.com, wsq0533@163.com, wuxia1143@sina.cn, sunfuzhen@sdut.edu.cn

**Abstract.** Recently, Contrastive Learning (CL) is becoming a mainstream approach to reduce the influence of data sparsity in recommendation system. However, existing methods do not fully explore the relationship between the outputs of different Graph Neural Network (GNN) layers and fail to fully utilize the capacity of combining GNN and CL for better recommendation. Within this paper, we introduce a novel approach based on CL, called efficient Graph collaborative filtering with multi-layer output-enhanced Contrastive Learning (GmoCL). It maximizes the benefits derived from the information propagation property of GNN with multi-layer aggregation to obtain better node representations. Specifically, the construction of CL tasks involves considerations from both intra-layer and inter-layer perspectives. The goal of intra-layer CL task is to exploit the semantic similarities of different users (or items) on a certain GNN layer. The inter-layer CL task aims to make the outputs of different GNN layers of the same user (or item) more similar. Additionally, we propose the strategy of negative sampling in the inter-layer CL task to learn the better node representations. The efficacy of the suggested approach is validated through comprehensive experiments conducted on five publicly available datasets.

**Keywords:** Recommendation System · Collaborative Filtering · Contrastive Learning · Graph Neural Network

## 1 Introduction

As Web 2.0 gains widespread popularity, the issue of information overload is progressively intensifying. Recommendation systems, as an effective solution, can alleviate the problem of information overload. Collaborative Filtering (CF) can effectively recommend for users by learning user preference from various feedback, such as clicks, purchases, and adding to cart. Recently, powerful Graph Neural Networks (GNNs) have further enhanced CF by modeling interactive


X. Yang et al. (Eds.): ADMA 2023, LNAI 14176, pp. 755–771, 2023.
https://doi.org/10.1007/978-3-031-46661-8_50

behaviors as graphs. GNNs can learn more effective node representations and make better recommendation performance for users, known as Graph Collaborative Filtering.

Despite the significant success of GNNs, two main problems remain. First, the interactive data of users is usually sparse or noisy, which will leads to inaccurate representations of learned users and items, as graph-based approaches may be more susceptible to data sparsity [24]. Second, existing GNN-based collaborative filtering methods are dependent on explicit interactions to learn node representations, while relationships of outputs of different GNN layers and user or item similarity are not used explicitly to enrich graph information. Contrastive learning methods have been adopted in recent studies to mitigate the scarcity of interaction data [22,24,26,29], however, they are still not fully exploited to mine potential various relationships among users (or items).

Apart from the evident interaction relations between users and items, there are various potential relations, e.g., structural neighbors and semantic neighbors, which are useful for recommendation tasks. NCL [14] takes these aspects into account by constructing contrastive pairs using rich neighbor relations. However, the potential of GNN in facilitating information propagation remains underutilized within NCL. Within the scope of this study, we design a more effective CL approach in order to fully utilize these potential relationships in graph collaborative filtering. Specifically, various potential relationships are utilized after aggregating multiple layers of GNN and further defined in two aspects: (1) intra-layer relationships pertain to the resemblance of output representations among distinct nodes within a given GNN layer, and (2) inter-layer relationships, which refer to the similarity of output representations of the same node at various GNN layers.

To harness the complete potential of output representations from diverse GNN layers, a model-agnostic framework based on contrastive learning for recommendation, called efficient Graph collaborative filtering with multi-layer output-enhanced Contrastive Learning (GmoCL), is proposed. Specifically, the proposed method constructs contrastive targets from two perspectives. From a macro point of view, there are potential relations between some nodes, which may not be explicitly connected on the graph. Inspired by NCL [14], we exploit to divide the similar nodes into same group by means of clustering algorithm. Each node and the cluster center which the node belongs to consist of positive pair, and we consider the other cluster center as the negative samples. In this view, we adopt the outputs of a particular GNN layer to perform cluster operations and construct contrastive targets. So, we denote it as intra-layer perspective and select the outputs of the second layer as representation of nodes in the experiments. From a micro point of view, the outputs of the $k$th GNN layer aggregate the information pertaining to the $k$-hop neighbors. Consequently, the outputs from distinct GNN layers of a given node are employed as positive pairs for contrastive learning. This is called inter-layer CL. The user-item interactive data can construct a bipartite graph. Considering the homogeneity between 2-hop neighbors, we divide the inter-layer CL into two kinds: inter-layer CL on odd layer and inter-layer CL on even layer. Furthermore, in inter-layer CL on even layers, we use a negative sampling strategy, with cluster centers as negative samples.

Within this paper, we fully utilize GNN to mine potential associations between users (or items) and combine these supplementary information and relationships to our CL framework. The outcomes of experiments conducted on five datasets demonstrate that the suggested method contributes to a discernible enhancement in recommendation performance. Within this paper, the contributions can be succinctly summarized as follows:

– We introduce a model-agnostic contrastive learning framework named GmoCL, which leverages the information propagation properties of GNNs and aggregates multiple layers of GNNs to improve graph collaborative filtering.
– We devise an intra-layer contrastive learning task and an inter-layer contrastive learning task, which effectively capture the similarities between outputs from distinct GNN layers, thereby enhancing representation learning. Furthermore, in the inter-layer CL task, we propose inter-layer CL on even-layer with negative sampling and inter-layer CL on odd-layer.
– We conduct experiments on five publicly available datasets, where the outcomes validate the rationality and efficacy of the proposed method. Subsequent ablation experiments demonstrate the individual contributions of each component to the enhancement in performance.

## 2 Preliminary and Definitions

### 2.1 Preliminary

Unlike traditional CF approaches, e.g., matrix decomposition [12,17] based approaches and autoencoder [13,18] based approaches, graph-based collaborative filtering constructs interactions within user-item interaction graphs and derives semantically valuable node representations from the structural information within the graph. Pioneering studies [1,6] extract structural information in the form of random walking on the graph, and later Graph Neural Networks (GNNs) are employed for collaborative filtering [9,18,23,27], where GNNs that introduce convolutional operations into the graph structure are called Graph Convolutional Networks (GCNs). The fundamental concept behind Graph Convolutional Networks (GCNs) is to acquire node representations by diffusing features throughout the graph. This is achieved through iterative graph convolutions, where features are progressively aggregated from neighboring nodes to form the representation of the focal node. For example, NGCF [22] and LightGCN [9] use high-order relations on interactive graphs to improve the performance of recommendation. The effectiveness of NGCF is not significantly influenced by the inclusion of feature transformation and nonlinear activation, two operations inherited from GCN. Therefore, LightGCN with these two operations removed contains only the most essential part of GCN, i.e., neighborhood aggregation for collaborative filtering. This uncomplicated and linear model exhibits improved performance and is more straightforward to train. Therefore, LightGCN is used as the base encoder Within this paper.

Applying a neighborhood aggregation scheme on the graph forms the core of the GCN-based collaborative filtering approach. This scheme involves updating the self-representation by aggregating the representations of neighboring nodes. It can be formulated as two phases.

$$z_u^{(l)} = f_{propagate}\left(\left\{z_v^{l-1} \mid v \in \{N_u \cup u\}\right\}\right), \\ z_u = f_{readout}\left(\left[z_u^{(0)}, z_u^{(1)}, ..., z_u^{(L)}\right]\right). \tag{1}$$

In the interactive graph $G$, where $N_u$ represents the set of neighbors of user $u$, and with $L$ denoting the number of GCN layers, $z_u^{(0)}$ is initialized vector. For user $u$, the propagation function $f_{propagate}$ aggregates its neighbors as well as its own $(l-1)$th layer representation to generate the $l$th layer representation, and there are also some works that aggregate only the neighbors' representations, such as LightGCN. Upon undergoing $l$ iterations of propagation, the $z_u^{(l)}$ representation encapsulates the information derived from $l$-hop neighbors. The readout function $f_{readout}$ is used to receive the ultimate representation of user $u$. Similarly, the representation of item $i$ can be received.

Predicting the probability of user $u$ engaging with item $i$ is the responsibility of the prediction layer. Here, $z_u$ and $z_i$ correspond to the ultimate representations of user $u$ and item $i$, respectively. The prediction score is calculated illustrated as follows:

$$\hat{y}_{ui} = z_u^\top z_i, \tag{2}$$

To capture information directly from the interactions, the Bayesian Personalized Ranking (BPR) [17] loss in pairs, a ranking objective function for recommendations, is used. BPR loss forces unobserved interactions to have lower prediction scores than observed interactions. Outlined below is the objective function:

$$L_{main} = \sum_{(u,i,j)\in O} -log\sigma(\hat{y}_{ui} - \hat{y}_{uj}), \tag{3}$$

where $O = \{(u,i,j) \mid r_{u,i} = 1, r_{u,j} = 0\}$ is the training data in pairs, and $r_{u,j} = 0$ means that item $j$ is not interacted by user $u$. $L_{main}$ is used as the main supervised task for recommendation. By optimizing $L_{main}$, it possesses the capability to forecast interactions between users and items.

## 2.2 Problem Definition

In recommender systems, collaborative filtering aims to provide personalized recommendations to users by suggesting items that align with their potential interests, utilizing observed feedback as a foundation, e.g., click, adding to cart, purchase. To elaborate, considering the sets of users $U$ and items $I$ the observed feedback matrix is represented as $R \in \{0,1\}^{|U|\times|I|}$. When there exist interactions between the user $u$ and the item $i$, then $r_{u,i} = 1$, otherwise 0. The recommender system can predict possible interactions based on the interaction matrix $R$. In addition, the GNN-based collaborative filtering method constructs the interaction matrix $R$ as an interactive graph $G$, thus the objective of the problem involves mapping each node $v$ within the set $V$ into a lower-dimensional spatial representation. This mapping aims to facilitate the recommendation of items that might capture the user's interest.

### 2.3 Notations Definition

The notations used in the paper are shown in Table 1.

**Table 1.** The notations

| Notation | Description |
|---|---|
| $U, I$ | The set of users and items, respectively |
| $V$ | $V = \{U \cup I\}$ the set of all users and items |
| $E$ | $E = \{(u,i) \mid u \in U, i \in I, r_{u,i} = 1\}$ the set of relations |
| $G$ | Interactive graph |
| $R$ | $R \in \{0,1\}^{|U|\times|I|}$ represents the matrix of interactions |
| $N_u$ | Set of neighbor nodes of user $u$ |
| $L$ | Layers of GNN |
| $z_u^{(l)}$ | Representation of user $u$ at layer $l$ |
| $z_i^{(l)}$ | Representation of item $i$ at layer $l$ |
| $z_u$ | The ultimate representation of user $u$ |
| $z_i$ | The ultimate representation of item $i$ |
| $C$ | The set of cluster centers |
| $K$ | The number of clusters |
| $L_{main}$ | The main supervised loss |
| $L_{intra}$ | The intra-layer contrastive loss |
| $L_{inter}$ | The inter-layer contrastive loss |

## 3 Methodology

The model architecture of GmoCL is illustrated in Fig. 1, where the annotation on the diagram takes the user as an example, and the item is similar. Our model has four important parts: 1) Multi-layer aggregation is to learn node representations by smoothing features on the graph, which performs graph convolution iteratively. 2) The intra-layer CL is to exploit the semantic similarity between nodes on a particular layer. 3) The inter-layer CL is to pull the outputs of different layer of same node together. 4) Multi-task learning is the joint training of BPR ranking loss and contrast loss.

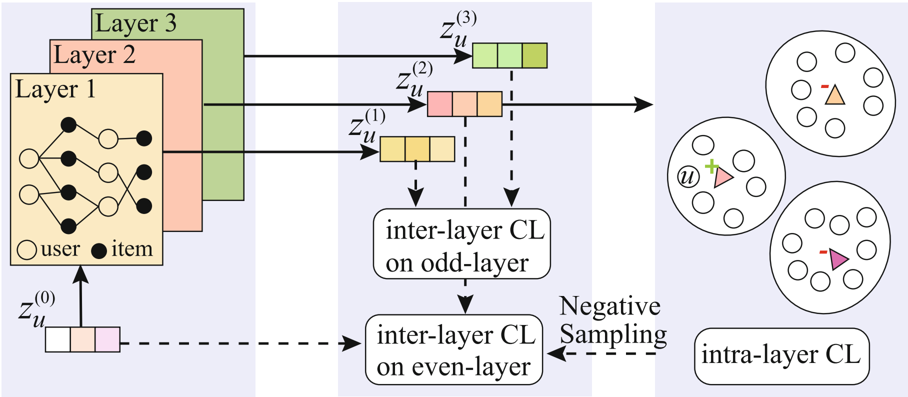



**Fig. 1.** The architecture of GmoCL. The annotation on the diagram takes the user as an example, and the item is similar.

### 3.1 Multi-layer Aggregation

The introduced GmoCL model exhibits model-agnostic properties, enabling its integration into numerous recommendation models based on graph neural networks. But for simplicity, we adopt LightGCN as the basic structure of the encoders. At each layer, we exploit LightGCN model to obtain embeddings of nodes. This whole process is repeated for $L(L = 3)$ times due to the over-smoothing problem. Then, corresponding to $L$ layers, we obtain $L$ output representations of each node.

### 3.2 The Intra-layer Contrastive Learning

The essence of collaborative filtering is to find similar nodes, so it is also essential to mine the relations between similar nodes that are unreachable (i.e., not directly or indirectly connected to each other) on the graph. Given output representations of a particular GNN layer, we construct a contrastive loss function to close the distance of nodes with similar features. It will facilitate nodes to learn a better representation. Specifically, a clustering algorithm is applied to the users and items output representations of a particular GNN layer to obtain the cluster centers, respectively. Our objective is to minimize the distance between the node and the cluster center of the cluster to which the node pertains. In the case of users, the aim of intra-layer contrastive learning is to minimize the subsequent functions:

$$L_{intra}^{U} = \sum_{u \in U} -log \frac{exp\left(cos\left(z_u^{(l)}, c\right)/\tau\right)}{\sum_{c_k \in C} exp\left(cos\left(z_u^{(l)}, c_k\right)/\tau\right)}, \tag{4}$$

where $z_u^{(l)}$ is the output representation of user $u$ at $l$th layer, $cos(\cdot,\cdot)$ is the cosine similarity function, $\tau$ is the temperature hyper-parameter, and $c_k$ is the cluster center obtained by applying the $K$-means algorithm on all user embeddings, while $C$ represents the set of cluster centers, of which there are $K$ in total. The contrasting learning objective for items is similar,

$$L_{intra}^{I} = \sum_{i \in I} -log \frac{exp\left(cos\left(z_i^{(l)}, c\right)/\tau\right)}{\sum_{c_k \in C} exp\left(cos\left(z_i^{(l)}, c_k\right)/\tau\right)}, \tag{5}$$

where $z_i^{(l)}$ is a representation of the output of item $i$ at $l$th layer and $c_k$ is the cluster center of item $i$. The ultimate objective of intra-layer contrastive learning is the summation of weights on both the user and item sides,

$$L_{intra} = L_{intra}^{U} + \alpha L_{intra}^{I}. \tag{6}$$

Here, $\alpha$ serves as the hyper-parameter for weight, maintaining a balance of the two losses.

By applying clustering algorithms, contrastive learning from the intra-layer perspective can alleviate data sparsity and effectively mine similar users or items, thus enabling the model to acquire an improved representation.

### 3.3 The Inter-layer Contrastive Learning

Viewing the interactive graph as a bipartite graph, the aggregation of information through the GNN-based model combines data from both homogeneous and heterogeneous nodes. We propose to exploit the similarity of the output representation of same node on odd or even layers through CL.

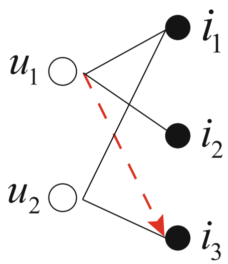



**Fig. 2.** An example of contrast on odd-layer.

**Contrastive Learning on Odd-Layer.** It is well known that multi-layer GCN operation on the graph aggregating information of high-hop neighbors can obtain more accurate node representations. In the recommendation scenario, There is a special bipartite graph of user-item interaction graph. And there are large amount of valuable information between the two-hop neighbors.

Illustrated in Fig. 2, user $u_1$ serves as a representative example. After one layer of GNN propagation, $u_1$ aggregates the information of the interacted items, such as $i_1$. After two layers of GNN propagation, $u_1$ aggregates the information of two-hop neighbors, such as $u_2$. After three layers of GNN propagation, $u_1$ aggregates the information of three-hop neighbors, such as $i_3$. Evident through the dashed line in Fig. 2, since $u_1$ and $u_2$ have interacted with the same item $i_1$, it is inferred that $u_1$ is more likely to interact with $i_3$, so the output representations of $u_1$ at 1st and 3rd layer are used as positive contrast pair, which naturally encodes the items that the user may interact with into the user interest. Specifically, the goal of inter-layer CL on odd-layer is to minimize the distance of outputs of the same node on two consecutive odd layers, such as 1st and 3rd layer. The contrastive loss of user and item are shown in Eq. 7 and Eq. 8.

$$L_{interO}^{U} = \sum_{u \in U} -log \frac{exp\left(cos\left(z_u^{(l)}, z_u^{(1)}\right)/\tau\right)}{\sum_{v \in U} exp\left(cos\left(z_u^{(l)}, z_v^{(1)}\right)/\tau\right)}, \tag{7}$$

$$L_{interO}^{I} = \sum_{i \in I} -log \frac{exp\left(cos\left(z_i^{(l)}, z_i^{(1)}\right)/\tau\right)}{\sum_{j \in I} exp\left(cos\left(z_i^{(l)}, z_j^{(1)}\right)/\tau\right)}, \tag{8}$$

where $l$ is an odd number, and the value of $l$ is 3 in this paper.

Combining the above two losses, contrastive learning objective of inter-layer on odd-layer is constructed as follows:

$$L_{interO} = L_{interO}^{U} + \alpha L_{interO}^{I}. \tag{9}$$

Here, $\alpha$ is the weight hyper-parameter utilized for harmonizing the two losses.

**Contrastive Learning on Even-Layer with Negative Samples.** The motivation of CL on even-layer is similar to that on odd-layer. However, in Sect. 3.2, we obtain the cluster centers based on the output representations of 2nd layer. So we adopt negative sampling strategy. To be specific, the negative samples are the centers of clusters where the target node is not belong to. This is different from the CL on odd-layer, which treats all other nodes in a batch as negative pairs and ignores the rest of the data in other batch, which suffers from data incompleteness. We constructed the contrastive learning objective for user and item as follows,

$$L_{interE}^{U} = \sum_{u \in U} -log \frac{exp\left(cos\left(z_u^{(l)}, z_u^{(0)}\right)/\tau\right)}{\sum_{c_m \in C} exp\left(cos\left(z_u^{(l)}, c_m\right)/\tau\right)}, \tag{10}$$

$$L_{interE}^{I} = \sum_{i \in I} -log \frac{exp\left(cos\left(z_i^{(l)}, z_i^{(0)}\right)/\tau\right)}{\sum_{c_n \in C} exp\left(cos\left(z_i^{(l)}, c_n\right)/\tau\right)}, \tag{11}$$

where $c_m$ and $c_n$ are cluster centers, $l$ is an even number, and the value of $l$ is 2 in this paper.

Merging the losses from both user and item sides, the contrastive learning objective function can be expressed as follows:

$$L_{interE} = L_{interE}^{U} + \alpha L_{interE}^{I}. \tag{12}$$

Here, $\alpha$ functions as the weight hyper-parameter, serving to balance the two categories of contrast losses.

Multi-layer aggregation constructs two contrastive loss objectives for nodes on even as well as odd layers of GNN, explicitly mining the information of homogeneous and heterogeneous nodes, which facilitates representation learning of nodes. Combining the even and odd layer contrastive learning objectives, the final multi-layer aggregation contrastive learning objective is:

$$L_{inter} = L_{interE} + \beta L_{interO}. \tag{13}$$

In this context, $\beta$ represents the weight hyper-parameter that achieves a balance between the two types of contrast losses.

### 3.4 Multi-task Training

Given that the primary objective of collaborative filtering involves predicting potential interactions between users and items, the contrast loss serves as a supplementary component. To simultaneously train the InfoNCE contrast loss and the traditional BPR ranking loss, we utilize a multi-task learning strategy. The overall losses are as follows:

$$L = L_{main} + \lambda_1 L_{intra} + \lambda_2 L_{inter} + \lambda_3 \|\theta\|_2^2. \tag{14}$$

In this equation, $\lambda_1$, $\lambda_2$, and $\lambda_3$ serve as weight hyper-parameters, responsible for achieving a balance between the two proposed contrast losses and normalized terms. Meanwhile, $\theta$ represents the set of parameters within the GNN model.

## 4 Experiments

To validate the efficacy of our proposed model, an extensive array of experiments has been conducted, accompanied by thorough and detailed analyses.

### 4.1 Experimental Setup

In this section, we introduce the dataset used for the experiments, the baseline methods, the evaluation metrics, and some implementation details.

**Datasets.** To evaluate the performance of the proposed model, experiments are conducted using five public datasets: Yelp, MovieLens-1M (ML-1M) [7], Alibaba-iFashion [3], Gowalla [4], and Amazon-Books [16].

**Baseline Methods.** We conduct a comparative analysis of the proposed model in contrast to the subsequent baseline methods:

**BPRMF** [17] uses the matrix factorization (MF) framework to optimize BPR loss to acquire latent representations of users and items with potential.

**NeuMF** [10] uses multilayer perceptron instead of dot product in MF model to learn the matching function of users and items.

**FISM** [11] represents an item-based collaborative filtering model that combines historical interaction representations to capture user interests.

**NGCF** [22] utilizes a bipartite graph structure to integrate higher-order relationships between users and items, while also employing GNN to enhance the collaborative filtering-based model.

**Multi-GCCF** [19] facilitates the propagation of information among users (and items) with higher-order associations, extending beyond the user-item bipartite graph.

**DGCF** [23] disentangles representations of users and items, resulting in enhanced recommendation performance.

**LightGCN** [9] streamlines the GCN architecture for increased simplicity and compatibility within recommendation systems.

**SGL** [24] take advantage of contrastive learning to strengthen recommendation. In this paper, we adopt SGL-ED which is the best instance of SGL.

**NCL** [14] proposes the contrastive learning method involving both structural and semantic neighbors to enhance the recommendation performance. The method proposed in our paper enhances this model and maximizes the advantages of GNN to construct the contrastive learning loss function.

**Table 2.** Datasets

| Datasets | #Users | #Items | #Interactions | Density |
|---|---|---|---|---|
| ML-1M | 6,040 | 3,629 | 836,478 | 0.03816 |
| Yelp | 45,478 | 30,709 | 1,777,765 | 0.00127 |
| Amazon-Books | 58,145 | 58,052 | 2,517,437 | 0.00075 |
| Gowalla | 29,859 | 40,989 | 1,027,464 | 0.00084 |
| Alibaba | 300,000 | 81,614 | 1,607,813 | 0.00007 |

**Evaluation Metrics.** We employ two widely recognized metrics, NDCG@$N$ and Recall@$N$, with $N$ values set at 10, 20, and 50, respectively. These metrics are utilized to assess the top-$N$ performance. Following [24] and [9], we employ a full ranking strategy, i.e., ranking all items with which the user has not engaged.

**Implementation Details.** Our model and all baseline methods are implemented using RecBole [28], a comprehensive open-source framework designed for the development and replication of recommendation algorithms. To ensure just comparisons, the Adam optimizer is employed for optimization across all methods. All parameters are initialized using the Xavier distribution, with a batch size of 4096 and an embedding size of 64. We use the early stop approach to prevent overfitting, the patience value is set to 40 epochs, and set NDCG@10 as the indicator (Table 2).

### 4.2 Overall Performance

The performance of the proposed method, along with other baseline models, is depicted in Table 3 across five datasets. From the results, several observations and conclusions can be drawn:

- Collaborative filtering models that encode representations of historical interaction behaviors as user interests, such as FISM, show better performance on all datasets, which demonstrates the effectiveness of collaborative filtering models. Of all the graph collaborative filtering baseline methods, LightGCN performs best in most datasets, indicating that the simple framework is more effective and robust. Additionally, the decoupled representation learning method DGCF performs less favorably than LightGCN, particularly when dealing with sparse datasets. This may be because, the dimensionality of the decoupled representation is too low to carry enough features.
- In terms of CL methods, SGL and NCL consistently exhibit superior performance over other supervised techniques. SGL obtains data augmented graphs for contrast by randomly perturbing user-item bipartite graphs, and although effective, this approach ignores other potential relationships (e.g., user similarity) in the recommendation system. NCL considers the importance of user (or item) similarity for representation learning from structural and semantic perspectives. While demonstrating effectiveness, it does not fully leverage the high-order information propagation properties inherent in GNNs.
- The superior performance of our model compared to all baseline models showcases the effectiveness of the contrastive learning approach through GNN multi-layer aggregation. Besides, the performance improvement of our method is more obvious on sparse datasets, such as Amazon-Books dataset and Alibaba dataset. This may be due to the fact that sparse datasets have too little interaction information, while our method explores the similarity between the outputs of discontinuous layers of nodes through contrastive learning, making the model to predict more accurate results.

### 4.3 Ablation Experiments

This subsection further analyzes the efficacy of the proposed model through ablation experiments. Figure 3 displays the results for Amazon-Books and ML-1M datasets, as constraints on space prevent the inclusion of additional data.

**Table 3.** Overall performance of different methods

| Dataset | Metric | BPRMF | NeuMF | FISM | NGCF | MultiGCCF | DGCF | LightGCN | SGL | NCL | Ours | Improv. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ML-1M | Recall@10 | 0.1804 | 0.1657 | 0.1887 | 0.1846 | 0.1830 | 0.1881 | 0.1876 | 0.1888 | 0.2048 | 0.2106 | +2.83% |
| | NDCG@10 | 0.2463 | 0.2295 | 0.2494 | 0.2528 | 0.2510 | 0.2520 | 0.2514 | 0.2526 | 0.2727 | 0.2771 | +1.61% |
| | Recall@20 | 0.2714 | 0.2520 | 0.2798 | 0.2741 | 0.2759 | 0.2779 | 0.2796 | 0.2848 | 0.3032 | 0.3108 | +2.51% |
| | NDCG@20 | 0.2569 | 0.2400 | 0.2607 | 0.2614 | 0.2617 | 0.2615 | 0.2620 | 0.2649 | 0.2842 | 0.2897 | +1.94% |
| | Recall@50 | 0.4300 | 0.4122 | 0.4421 | 0.4341 | 0.4364 | 0.4424 | 0.4469 | 0.4487 | 0.4677 | 0.4785 | +2.31% |
| | NDCG@50 | 0.3014 | 0.2851 | 0.3078 | 0.3055 | 0.3056 | 0.3078 | 0.3091 | 0.3111 | 0.3298 | 0.3365 | +2.03% |
| Yelp | Recall@10 | 0.0643 | 0.0531 | 0.0714 | 0.0630 | 0.0646 | 0.0723 | 0.0730 | 0.0833 | 0.0912 | 0.0941 | +3.18% |
| | NDCG@10 | 0.0458 | 0.0377 | 0.0510 | 0.0446 | 0.0450 | 0.0514 | 0.0520 | 0.0601 | 0.0679 | 0.0692 | +1.91% |
| | Recall@20 | 0.1043 | 0.0885 | 0.1119 | 0.1026 | 0.1053 | 0.1135 | 0.1163 | 0.1288 | 0.1358 | 0.1411 | +3.90% |
| | NDCG@20 | 0.0580 | 0.0486 | 0.0636 | 0.0567 | 0.0575 | 0.0641 | 0.0652 | 0.0739 | 0.0815 | 0.0837 | +2.70% |
| | Recall@50 | 0.1862 | 0.1654 | 0.1963 | 0.1864 | 0.1882 | 0.1989 | 0.2016 | 0.2140 | 0.2171 | 0.228 | +5.02% |
| | NDCG@50 | 0.0793 | 0.0685 | 0.0856 | 0.0784 | 0.0790 | 0.0862 | 0.0875 | 0.0964 | 0.103 | 0.1066 | +3.50% |
| Amazon | Recall@10 | 0.0607 | 0.0507 | 0.0721 | 0.0625 | 0.0625 | 0.0737 | 0.0797 | 0.0898 | 0.094 | 0.0986 | +4.89% |
| | NDCG@10 | 0.043 | 0.0351 | 0.0504 | 0.0433 | 0.0433 | 0.0521 | 0.0565 | 0.0645 | 0.0683 | 0.072 | +5.42% |
| | Recall@20 | 0.0956 | 0.0823 | 0.1099 | 0.0991 | 0.0991 | 0.1128 | 0.1206 | 0.1331 | 0.138 | 0.1443 | +4.57% |
| | NDCG@20 | 0.0537 | 0.0447 | 0.0622 | 0.0545 | 0.0545 | 0.064 | 0.0689 | 0.0777 | 0.0817 | 0.086 | +5.26% |
| | Recall@50 | 0.1681 | 0.1447 | 0.1830 | 0.1688 | 0.1688 | 0.1908 | 0.2012 | 0.2267 | 0.2179 | 0.2267 | +4.04% |
| | NDCG@50 | 0.0726 | 0.061 | 0.0815 | 0.0727 | 0.0727 | 0.0843 | 0.0899 | 0.0992 | 0.1028 | 0.1079 | +4.96% |
| Gowalla | Recall@10 | 0.1158 | 0.1039 | 0.1081 | 0.1192 | 0.1108 | 0.1252 | 0.1362 | 0.1465 | 0.1502 | 0.1505 | +0.20% |
| | NDCG@10 | 0.0833 | 0.0731 | 0.0755 | 0.0852 | 0.0791 | 0.0902 | 0.0876 | 0.1048 | 0.1082 | 0.1089 | +0.65% |
| | Recall@20 | 0.1695 | 0.1535 | 0.1620 | 0.1755 | 0.1626 | 0.1829 | 0.1976 | 0.2084 | 0.2129 | 0.215 | +0.99% |
| | NDCG@20 | 0.0988 | 0.0873 | 0.0913 | 0.1013 | 0.0940 | 0.1066 | 0.1152 | 0.1225 | 0.1263 | 0.1274 | +0.87% |
| | Recall@50 | 0.2756 | 0.2510 | 0.2673 | 0.2811 | 0.2631 | 0.2877 | 0.3044 | 0.3197 | 0.3259 | 0.3274 | +0.46% |
| | NDCG@50 | 0.1450 | 0.1110 | 0.1169 | 0.1270 | 0.1184 | 0.1322 | 0.1414 | 0.1497 | 0.1541 | 0.155 | +0.58% |
| Alibaba | Recall@10 | 0.303 | 0.182 | 0.0357 | 0.0382 | 0.0401 | 0.0447 | 0.0457 | 0.0461 | 0.0484 | 0.0498 | +2.89% |
| | NDCG@10 | 0.0161 | 0.0092 | 0.0190 | 0.0198 | 0.0207 | 0.0241 | 0.0246 | 0.0248 | 0.0264 | 0.0272 | +3.03% |
| | Recall@20 | 0.0467 | 0.0302 | 0.0553 | 0.0615 | 0.0634 | 0.0677 | 0.0692 | 0.0692 | 0.0717 | 0.0748 | +4.32% |
| | NDCG@20 | 0.0203 | 0.0123 | 0.0239 | 0.0257 | 0.0266 | 0.0299 | 0.0246 | 0.0307 | 0.0323 | 0.0335 | +3.72% |
| | Recall@50 | 0.0799 | 0.0576 | 0.0943 | 0.1081 | 0.1107 | 0.1120 | 0.1144 | 0.1141 | 0.1155 | 0.1209 | +4.68% |
| | NDCG@50 | 0.0269 | 0.0177 | 0.0317 | 0.0349 | 0.0360 | 0.0387 | 0.0396 | 0.0396 | 0.041 | 0.0427 | +4.15% |

With "w/o inter" and "w/o intra" denote the variables that remove inter-layer CL and intra-layer CL, respectively. As depicted in the figure, removing each aspect leads to a performance decrease, while both variants perform better than the LightGCN. Furthermore, these two relations mutually reinforce each other, contributing to performance improvement through distinct avenues.

### 4.4 Hyper-parameter Analysis

Within this specific section, we analyze the effects of hyper-parameter $\alpha$ and $\beta$. However, due to spatial constraints, we present the outcomes solely for the ML-1M and Amazon-Books datasets in Fig. 4 and 5 respectively.

**Effect of Hyper-parameter $\alpha$.** The coefficient $\alpha$ is used for balancing the user side and item side on the intra-layer contrasts and inter-layer contrasts. To analyze the effect of it, we set its variation range between 0.1 and 2 and report the results in Fig. 4. This suggests that a proper $\alpha$ can be effective to increase the performance of our method. The best results are achieved on the ML-1M dataset with the value of 1.0 and on the Amazon-Books dataset with the value of 0.3, indicating that the homogeneity between the outputs of discontinuous layers is valuable for both users and items.

**Effect of Hyper-parameter** $\beta$**.** Here, we analyze the effects of hyper-parameter $\beta$ for balancing odd- and even-layer contrasts through experiments. To analyze the effect of it, we set its variation range between 0.1 and 2 and report the results in Fig. 5. The results show that the value of $\beta$ achieves the best results differs on different datasets. Specifically, the best results are achieved on the ML-1M dataset with the value of 1.0 and on the Amazon-Books dataset with the value of 0.5.

## 4.5 Distribution of Items Embedding

We show the effects of the proposed model on representation learning in Fig. 6, where our visualization is based on the SVD decomposition, which projects the embedding matrix into a two-dimensional space. From the figure we can see that the distribution of embeddings of low-frequency items is more balanced that they are located around the origin point in our proposed method compared to NCL, and we hypothesize that a more balanced embedding distribution better models different user preferences or item features.

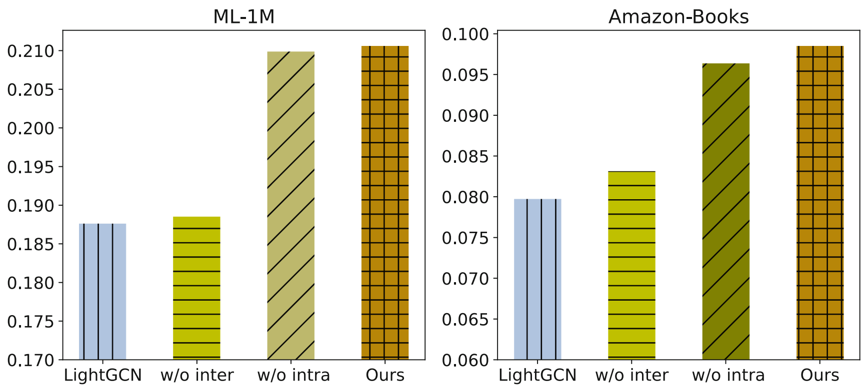



**Fig. 3.** Performance comparison without inter-layer CL and intra-layer CL on two datasets respectively (Recall@10).

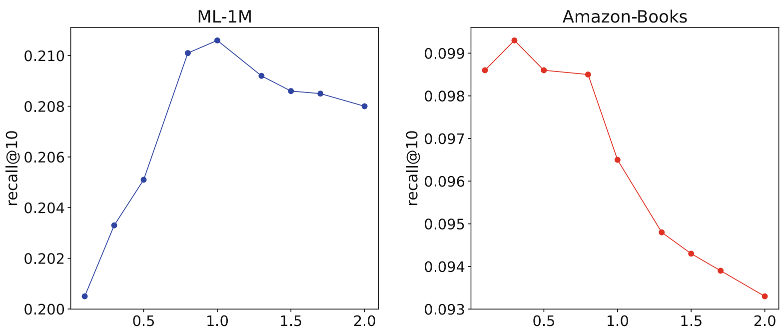



**Fig. 4.** Performance comparison for different $\alpha$

# 5 Related Work

This section provides a concise overview of two pertinent studies: graph-based collaborative filtering and contrastive learning.

**Graph-Based Collaborative Filtering.** The CF-based approach has evolved by now to apply graph neural networks (GNN) to collaborative filtering [9,22,23,27]. For example, utilizing higher-order relations within interaction graphs, NGCF [22] and LightGCN [9] aim to enhance recommendation performance. Furthermore, [19] extends this idea by introducing the construction of multiple interaction graphs to attain more comprehensive association relationships between users and items. Although it is effective, they do not explicitly address the problem of data sparsity [24]. Lately, the integration of self-supervised learning into graph collaborative filtering has emerged as an approach to bolster the efficacy of recommendations. However, most graph-based approaches focus only on interaction history and ignore the potential neighbor relationships between users or items. Some recent self-supervised learning methods are proposed. NCL [14] proposes to consider users (or items) and their homo-

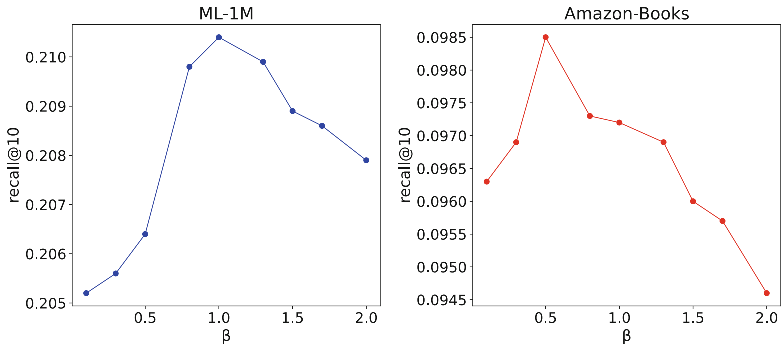
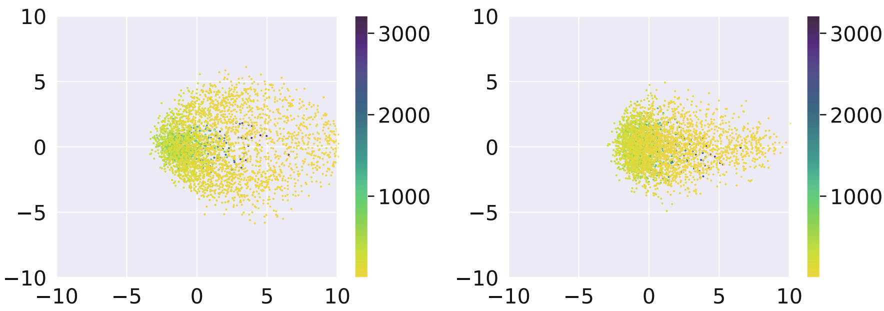



**Fig. 5.** Performance comparison for different $\beta$



(a) NCL (b) Ours

**Fig. 6.** Distribution of item embedding on ML-1M.

geneous neighbors as positive contrastive pairs and construct two self-supervised losses in the structure space and semantic space, respectively. However, the strong GNN is still not well utilized to exploit potential user (or item) relationships.

**Contrastive Learning.** Given the success of contrastive learning in Computer Vision (CV) [2], its application has extended across Natural Language Processing (NLP) [5], recommender systems [20], and graph data mining [15,25]. The primary objective of contrastive learning is to optimize the agreement between positive pairs while minimizing the agreement between negative pairs. DGI [21] regards graph-level representations and node-level representations of the same graph as positive pairs. CMRLG [8] realizes a similar goal by treating adjacency matrix and diffusion matrix as positive pairs. Recently, SGL [24] designs random data augmentation operations and constructed contrastive targets to improve the recommendation performance. More recently, NCL [14] considers the importance of user (or item) similarity for representation learning in structural and semantic aspects, and although effective, It falls short of fully capitalizing on the high-order information propagation characteristics inherent in GNNs. Within this paper, we introduce a contrastive learning framework that operates from both intra-layer and inter-layer perspectives. This framework is devised to comprehensively harness the output representations derived from various GNN layers.

## 6 Conclusion and Future Work

In order to take full advantage of the output representations of different GNN layers, a model-agnostic contrastive learning framework for recommendation, called GmoCL, is proposed. Specifically, contrastive goals are constructed in two respects. For intra-layer CL, we try to find the semantic similar contrastive pair by use of clustering algorithm. For inter-layer CL, we focus on the similar of output representations of two consecutive odd or even layers of the same node. Furthermore, we adopt negative sampling strategy for inter-layer CL on even-layer, which can make the model learn better representations. The effectiveness of the proposed method is underscored by an extensive array of experiments conducted on five publicly accessible datasets.

Going forward, we will place additional emphasis on the matter of positive and negative sampling. In addition, applying our contrast learning framework to different recommendation tasks is also part of our work, and we believe that the potential of GNN still deserves to be explored fully, and combining contrastive learning with heterogeneous graph neural networks will be more beneficial for representation learning and thus can be used better for downstream tasks.

**Acknowledgements.** This work was supported by Shandong Provincial Natural Science Foundation, China (ZR2020MF147, ZR2021MF017).

## References


1. Baluja, S., et al.: Video suggestion and discovery for youtube: taking random walks through the view graph. In: Proceedings of the 17th International Conference on World Wide Web, pp. 895–904 (2008)
2. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607. PMLR (2020)
3. Chen, W., et al.: POG: personalized outfit generation for fashion recommendation at Alibaba iFashion. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2662–2670 (2019)
4. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1082–1090 (2011)
5. Giorgi, J., Nitski, O., Wang, B., Bader, G.: Declutr: deep contrastive learning for unsupervised textual representations. arXiv preprint arXiv:2006.03659 (2020)
6. Gori, M., Pucci, A., Roma, V., Siena, I.: Itemrank: a random-walk based scoring algorithm for recommender engines. In: IJCAI, vol. 7, pp. 2766–2771 (2007)
7. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. ACM Trans. Interact. Intell. Syst. (TIIS) **5**(4), 1–19 (2015)
8. Hassani, K., Khasahmadi, A.H.: Contrastive multi-view representation learning on graphs. In: International Conference on Machine Learning, pp. 4116–4126. PMLR (2020)
9. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: 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, pp. 639–648 (2020)
10. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
11. Kabbur, S., Ning, X., Karypis, G.: FISM: factored item similarity models for top-n recommender systems. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 659–667 (2013)
12. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
13. Liang, D., Krishnan, R.G., Hoffman, M.D., Jebara, T.: Variational autoencoders for collaborative filtering. In: Proceedings of the 2018 World Wide Web Conference, pp. 689–698 (2018)
14. Lin, Z., Tian, C., Hou, Y., Zhao, W.X.: Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In: Proceedings of the ACM Web Conference 2022, pp. 2320–2329 (2022)
15. Liu, Y., et al.: Graph self-supervised learning: a survey. IEEE Trans. Knowl. Data Eng. (2022)
16. McAuley, J., Targett, C., Shi, Q., Van Den Hengel, A.: Image-based recommendations on styles and substitutes. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 43–52 (2015)
17. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)

18. Strub, F., Mary, J., Philippe, P.: Collaborative filtering with stacked denoising autoencoders and sparse inputs. In: NIPS Workshop on Machine Learning for eCommerce (2015)
19. Sun, J., et al.: Multi-graph convolution collaborative filtering. In: 2019 IEEE International Conference on Data Mining (ICDM), pp. 1306–1311. IEEE (2019)
20. Tang, H., Zhao, G., Wu, Y., Qian, X.: Multisample-based contrastive loss for top-k recommendation. IEEE Trans. Multimedia (2021)
21. Veličković, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. arXiv preprint arXiv:1809.10341 (2018)
22. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 165–174 (2019)
23. Wang, X., Jin, H., Zhang, A., He, X., Xu, T., Chua, T.S.: Disentangled graph collaborative filtering. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1001–1010 (2020)
24. Wu, J., et al.: Self-supervised graph learning for recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 726–735 (2021)
25. Wu, L., Lin, H., Tan, C., Gao, Z., Li, S.Z.: Self-supervised learning on graphs: contrastive, generative, or predictive. IEEE Trans. Knowl. Data Eng. (2021)
26. Wu, Y., et al.: Multi-view multi-behavior contrastive learning in recommendation. In: Bhattacharya, A., et al. (eds.) DASFAA 2022. LNCS, vol. 13246, pp. 166–182. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-00126-0_11
27. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 974–983 (2018)
28. Zhao, W.X., et al.: Recbole: towards a unified, comprehensive and efficient framework for recommendation algorithms. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 4653–4664 (2021)
29. Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Graph contrastive learning with adaptive augmentation. In: Proceedings of the Web Conference 2021, pp. 2069–2080 (2021)