Contrastive Co-training for Diversified Recommendation

Xiyao Ma
University of Florida
maxiy@ufl.edu

Qian Hu
Amazon Alexa AI
huqia@amazon.com

Zheng Gao
Amazon Alexa AI
zhenggao@amazon.com

Mohamed AbdelHady
Amazon Alexa AI
mbdeamz@amazon.com

***Abstract*—Beyond accuracy, diversity has become a crucial factor to evaluate a recommendation system as higher diversity helps mitigate echo chamber issue and improve user satisfaction. Recently, great success has been made to improve diversity, but the approaches often sacrifice much lower accuracy. Herein this work, we propose contrastive co-training for diversified recommendation that improves diversity greatly and achieves comparable or even better accuracy. Specifically, we keep two user-item graph views for recommendation and contrastive learning, respectively. Pseudo edges are predicted from current graph view to augment the other graph view by mining the novel items that users might be highly interested in. However, merely leveraging co-training hurts the accuracy since the pseudo labels are sometimes noisy. Therefore, we propose diversified contrastive learning that not only is robust to the noisy pseudo edges but also improves the diversity further by alleviating the popularity and category biases by re-balancing item-level popularity and category-level advantage. The extensive experiments on three public datasets show the superiority of our proposed model in terms of accuracy and diversity compared with strong baselines.**

I. Introduction

The recommender system is a hot topic in data mining and machine learning research community, and accuracy is a widely used evaluation metrics. However, it is not the only golden metrics [1]. Beyond accuracy, diversity has become a crucial factor to evaluate a recommendation system, as it is directly related to user satisfaction [2]. A recommendation system with poor diversity might narrow down the users' horizons and make them frustrated; while recommending relevant and diverse items can improve the exposure of items, broaden the users' horizons, and even help users to discover new content they like [3].

Great efforts have been made to improve recommendation diversity, and they can be roughly categorized into three directions: (1) Rerank-based or post-processing-based methods are proposed to apply on the generated item list. The order of items are determined by balancing the accuracy and diversity with a controllable hyper-parameters. However, these methods highly depend on the quality of the generated items, resulting in suboptimal performance in terms of accuracy [4]–[6]. (2) Methods that take advantages of Determinantal Point Process (DPP) score an entire list of items by taking into account item correlations to diversify the recommended item list. However, these approaches demand that the item embeddings generated in the generation stage to be of high quality, which determines the upper bound of accuracy [7], [8]. (3) Some end-to-end approaches are proposed to directly recommend an item list, formulating it as a ranking problem [9], [10]. However, these approaches achieve high diversity but sacrifice too much accuracy. Therefore, we aim to improve diversity as well as preserve or even improve accuracy.

This work was done during Xiyao Ma's internship in Amazon Alexa. Qian Hu is the corresponding author.

With the huge success of graph-based neural networks (GNNs), GNN-based methods greatly advance the research and have become the mainstream method [11]–[13] for top-k item recommendation due to the high accuracy. Given a user-item bipartite graph where users and items are viewed as nodes, and user-item interactions are viewed as edges, GNN-based models learn user/item embedding by aggregating the embeddings of the connected items/users. High-order proximity between users and items can be effectively captured by stacking several layers of GNNs. However, current GNNs suffer from the popularity and category biases that the learnt user embeddings are very close to the popular items and the items of the advantage categories as they take up the majority of the edges, inherently limiting the model to recommend novel and diversified items [10]. Moreover, GNN learns user and item representations upon the user-item graph that includes fixed number of observed edges during training. However, this ignores the intrinsic nature of implicit feedback recommendation that unobserved edges are mixed with negative feedback and unknown positive feedback [14]. Merely depending on the observed edges and neglecting the unknown positive feedback might lead to inferior diversity.

In this paper, inspired by co-training [15], we propose **C**ontrastive **C**o-**T**raining (CCT) for diversified recommendation. Specifically, we consider two graph views for recommendation and contrastive learning, respectively. The current graph view is utilized to predict pseudo edges that connect users to new items for the other graph view, revealing the novel items that users are highly interested in and improving diversity. However, the predicted pseudo edges are sometimes noisy. Therefore, it is advantageous and natural to perform contrastive learning on the augmented graph views due to its robustness [16], [17]. Moreover, we propose diversified contrastive learning that is carefully designed to alleviate popularity and category biases by re-balancing item-level popularity and category-level advantage. Extensive experiments on

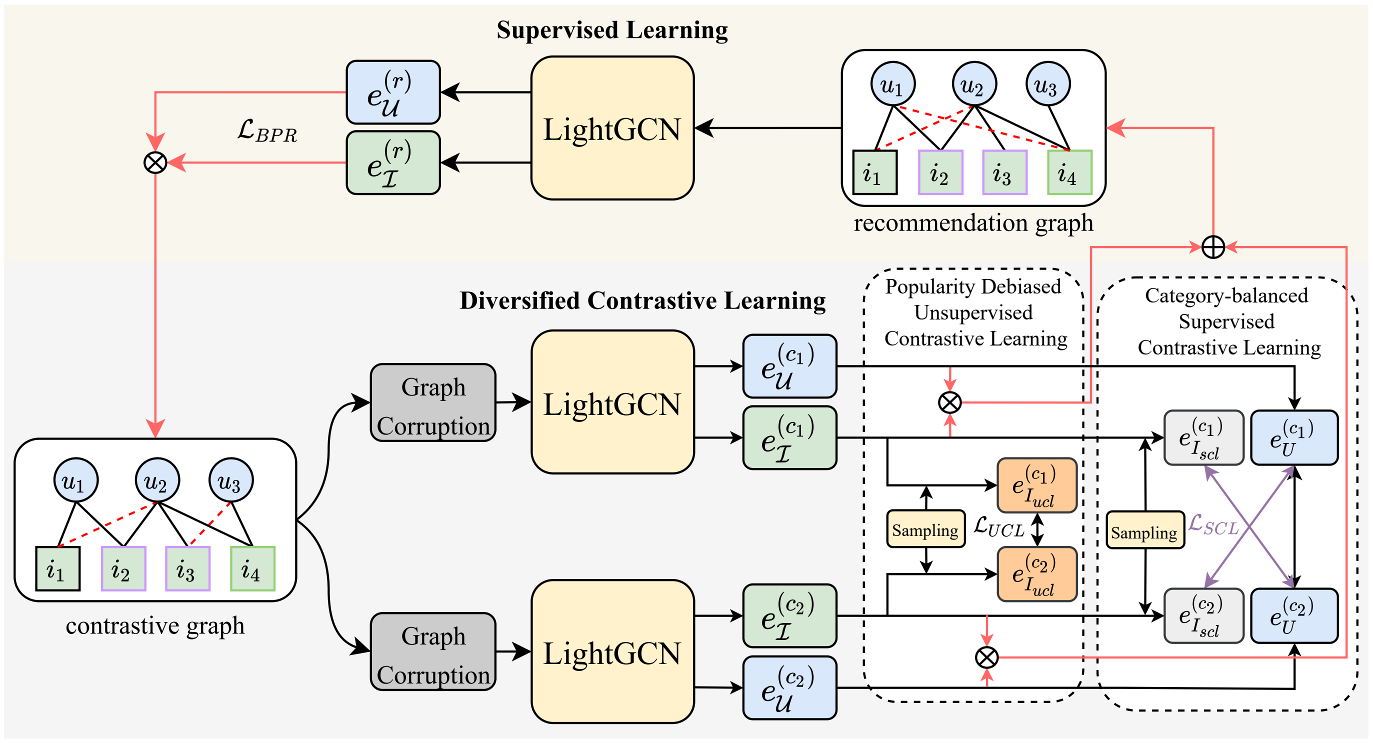


Fig. 1: The overview of our proposed model. The BPR loss and contrastive learning losses are computed for training with the user and item embeddings generated from each graph view, as indicated in the black lines; the red dotted lines in the two graph views represent the pseudo edges, which are predicted by each view, as indicated by the red lines. Item nodes with frames of different colors represents items of different categories.

three public datasets show the effectiveness of our proposed model that significantly outperforms state-of-the-art (SOTA) models. To summarize, the contributions of this paper are three-folds:

- We propose a contrastive co-training framework to improve diversity for recommendation. To the best of our knowledge, this is the first work to effectively integrate co-training and contrastive learning for the purpose of improving diversity.
- We propose diversified contrastive learning to alleviate the popularity and category biases, further improving diversity.
- Extensive experiments show that our proposed model greatly outperforms strong baselines in terms of diversity as well as preserve and even improve accuracy.

II. Problem Formulation

Generally, in a diversified recommendation scenario, we have a set of users $\mathcal{U} = [u_1, u_2, \dots, u_M]$, a set of items $\mathcal{I} = [i_1, i_2, \dots, i_N]$ that each item is associated with a category label $c \in \mathcal{C}$, and observed interactions $\mathcal{Y} = \{y_{ui} = 1 | u \in \mathcal{U}, i \in \mathcal{I}\}$. Following the recent graph-based collaborative filtering approaches [12], [13], we build a user-item bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where node set $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ involves all users and items, and the edge set $\mathcal{E}$ includes all observed interactions $\mathcal{Y}$. In general, we aim to learn a function that predicts the probability $\hat{y}_{ui}$ that a user u would adopt an item i so as to recommend top-k relevant and diverse items for each user.

III. Proposed Method

The overview of the proposed CCT is illustrated in Figure 1, where we consider two graph views $[\mathcal{G}^{(r)} = (\mathcal{V}^{(r)}, \mathcal{E}^{(r)}), \mathcal{G}^{(c)} = (\mathcal{V}^{(c)}, \mathcal{E}^{(c)})]$, namely a recommendation graph view and a contrastive graph view for recommendation and contrastive learning, respectively. The contrastive graph view is further used to generate two corrupted contrastive graph views $\mathcal{G}^{(c_1)} = (\mathcal{V}^{(c_1)}, \mathcal{E}^{(c_1)})$ and $\mathcal{G}^{(c_2)} = (\mathcal{V}^{(c_2)}, \mathcal{E}^{(c_2)})$. We adopt LightGCN [13] as the encoder on each view due to its simplicity and high accuracy, which can be generally formulated as:

$$\boldsymbol{E}_{\mathcal{U}}^{(v)}, \boldsymbol{E}_{\mathcal{I}}^{(v)} = GC(E, \mathcal{G}^{(v)}) \tag{1}$$

where $\mathcal{G}^{(v)}, v \in \{r, c_1, c_2\}$ is any of the views, and GC represents the graph convolution layers in LightGCN. $E \in \mathbb{R}^{(M+N) \times d}$ denotes the initial node embeddings which are shared by the encoder of each view, and $\boldsymbol{E}_{\mathcal{U}}^{(v)} = [e_{u_1}^{(v)}, e_{u_2}^{(v)}, ..., e_{u_M}^{(v)}] \in \mathbb{R}^{M \times d}$ and $\boldsymbol{E}_{\mathcal{I}}^{(v)} = [e_{u_1}^{(v)}, e_{u_2}^{(v)}, ..., e_{u_M}^{(v)}] \in \mathbb{R}^{N \times d}$ are the learnt user and item vectors from the graph view $\mathcal{G}^{(v)}$.

Pseudo edges are predicted to connect user nodes to novel item nodes, augmenting the two graph views iteratively, and the model is trained by jointly minimizing the BPR loss and diversified contrastive learning loss upon the augmented views. We first illustrate the detail of the co-training pipeline in Section III-A, and then we demonstrate the diversified contrastive learning upon the contrastive graph in Section III-B.

A. Contrastive Co-training

The CCT pipeline is illustrated in Algorithm 1, and it consists of the following stages: (1) **Pretraining**: The recommendation and contrastive graph views are firstly initialized by copying the input user-item graph. Then we pretrain LightGCN upon the graph views by minimizing the final loss in Equation 11 (Section III-B). (2) **Predicting Pseudo Edges**: In the co-training paradigm, the two graph views $\mathcal{G}^{(r)}$ and $\mathcal{G}^{(c)}$ are augmented by the pseudo edges predicted by each other. We will demonstrate the detail of this step in Section III-A1. (3) **Training the model on the augmented graphs**: We compute the final loss in Equation 11 with the learnable weights computed by Equation 5 and train the model upon the two augmented graph views as described in Section III-B. Finally, we go back to the second stage and train the model iteratively. We reset the augmented graphs with the input graph for every 50 epochs to avoid error accumulation.

Algorithm 1 Contrastive co-training algorithm

```
Input: Input graph $\mathcal{G}$
Output: Recommended item lists.
1:  Initialize the recommendation graph $\mathcal{G}^{(r)} = (\mathcal{V}^{(r)}, \mathcal{E}^{(r)})$ and the contrastive graph $\mathcal{G}^{(c)} = (\mathcal{V}^{(c)}, \mathcal{E}^{(c)})$ with the input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.
2:  Pretrain the model with diversified contrastive learning in Algorithm 2.
3:  while not converged do
4:      Generate two corrupted contrastive graphs $\mathcal{G}^{(c_1)}$ and $\mathcal{G}^{(c_2)}$ by graph corruption given the contrastive graph $\mathcal{G}^{(c)}$.
5:      Predict pseudo edges $\mathcal{S}^{(r)}$, $\mathcal{S}^{(c_1)}$, and $\mathcal{S}^{(c_2)}$ given the recommendation graph and two corrupted contrastive graphs (Section III-A1).
6:      Augment $\mathcal{G}^{(c)}$: $\mathcal{E}^{(c)} = \mathcal{E}^{(c)} \cup \mathcal{E}_{ps}^{(c)}, \mathcal{E}_{ps}^{(c)} = \boldsymbol{m_1} \odot \mathcal{S}^{(r)}$.
7:      Augment $\mathcal{G}^{(r)}$: $\mathcal{E}^{(r)} = \mathcal{E}^{(r)} \cup \mathcal{E}_{ps}^{(r)}, \mathcal{E}_{ps}^{(r)} = \boldsymbol{m_2} \odot \mathcal{S}^{(c_1)} \cap \mathcal{S}^{(c_2)}$.
8:      Train the model with diversified contrastive learning in Algorithm 2.
9:      Reset the two graph views $\mathcal{G}^{(r)}$ and $\mathcal{G}^{(c)}$ with the input graph $\mathcal{G}$ for every 50 epochs.
10: end while
11: Use the recommendation graph view $\mathcal{G}^{(r)}$ to recommend top-k item list for each user in the testing set.
```

1) Predicting Pseudo Edges: As each view reflects a different aspect of the user-item interactions, it is natural to seek supervisory information from the other view to improve the encoder of the current view. By performing graph convolution over a given graph view $\mathcal{G}^{(v)}, v = \{r, c_1, c_2\}$, the LightGCN learns user and item vectors. For each user u in the dataset, we predict the probability for each unclicked item i by their inner product:

$$\hat{y}_{ui}^{(v)} = e_u^{(v)\top} e_i^{(v)}. \tag{2}$$

where $e_u^{(v)}$ and $e_i^{(v)}$ represent the embedding of user u and item i, respectively. Then we design the following rules to predict pseudo edges:

- We rank the predicted probabilities $\hat{y}_{ui}^{(v)}$ and select the top k unclicked items for each user.
- We then keep the items whose probabilities $\hat{y}_{ui}^{(v)}$ are higher than a threshold (we adopt 0.95 in practice) to filter out the edges with low confidence.
- As we aim to improve diversity, we only keep a subset of items $\mathcal{S}^{(v)}$ that only includes items of the unseen categories that the user has not interacted. Another reason that we prefer the items of unseen categories to the items of seen categories is that adding additional items from seen categories might worsen the aforementioned categories bias, resulting in lower diversity.

After obtaining the pseudo edges candidates set $\mathcal{S}^{(r)}$, $\mathcal{S}^{(c_1)}$, and $\mathcal{S}^{(c_2)}$, we augment the recommendation graph by uniformly sampling T pseudo edges from the consensus of the predictions of the two corrupted contrastive graph, and the contrastive graph is augmented by uniformly sampling T pseudo edges from the predictions of the recommendation graph:

$$\mathcal{E}^{(r)} = \mathcal{E}^{(r)} \cup \mathcal{E}_{ps}^{(r)}, \mathcal{E}_{ps}^{(r)} = \boldsymbol{m_1} \odot \mathcal{S}^{(c_1)} \cap \mathcal{S}^{(c_2)}, \tag{3}$$

$$\mathcal{E}^{(c)} = \mathcal{E}^{(c)} \cup \mathcal{E}_{ps}^{(c)}, \mathcal{E}_{ps}^{(c)} = \boldsymbol{m_2} \odot \mathcal{S}^{(r)}. \tag{4}$$

where $\boldsymbol{m_1} \in \{0,1\}^{|\mathcal{S}^{(c_1)} \cap \mathcal{S}^{(c_2)}|}$ amd $\boldsymbol{m_2} \in \{0,1\}^{|\mathcal{S}^{(r)}|}$ are the mask vectors to sample pseudo edges $\mathcal{E}_{ps}^{(r)}$ and $\mathcal{E}_{ps}^{(c)}$ for the recommendation and contrastive graphs, respectively. Operation $\odot$ is the element-wise product for edge sampling.

a) Noise-robust loss function with learnable weights: The predicted pseudo edges aim to mine the user-interested novel items and improve the diversity. However, the pseudo edges might be noisy even though we filter out the edges with low confidence. Therefore, it might be problematic and result in suboptimal performance if we treat the observed edges and the predicted pseudo edges the same. To tackle the issue, we assign lower weights to the pseudo edges and higher weights to the observed edges, when computing the supervised BPR loss and contrastive learning loss that we will present in Section III-B0c. Instead of using a fixed weight, we train a learnable weight w_{ui} for each pseudo edge under the assumption that closer user embedding and item embedding indicates higher reliability and lower noise:

$$w_{ui} = \begin{cases} 1, & y_{ui} \in \mathcal{Y} \\ \sigma(W(e_u^{(v)} - e_i^{(v)}) + b), & y_{ui} \in \mathcal{E}_{ps}^{(v)} \end{cases} \tag{5}$$

where $e_u^{(v)}$ and $e_i^{(v)}$ are the user embedding and item embedding of each pseudo edge generated upon the graph view $\mathcal{G}^{(v)}$. W and b are trainable weights and bias. Note that observed edges are always associated with weight $w_{ui} = 1$ as the observed edges are ground truth and should be fully utilized.

Algorithm 2 Diversified Contrastive Learning

```
Input: User embeddings, item embeddings, recommendation graph view $\mathcal{G}^{(r)}$, and two corrupted contrastive graph views $\mathcal{G}^{(c_1)}$ and $\mathcal{G}^{(c_2)}$ generated by graph corruption.
1: Initialize the node embeddings of the two graph views with the user embeddings and item embeddings.
2: for all batch$(U, I) \in \mathcal{E}^{(r)}$ do
3:     Compute learnable weights $w_{ui}$ for the pseudo edges in each graph view $\mathcal{G}^{(v)}, v = \{r, c_1, c_2\}$
4:     Compute the supervised BPR loss $\mathcal{L}_{\text{BPR}}$ (Equation 12) on the recommendation graph.
5:     Select item nodes $I_{ucl}$ according to weights $p_i^{ucl}$ (Equation 7) and compute $\mathcal{L}_{UCL}$ (Equation 6).
6:     Select item nodes $I_{scl}$ for the current batch users according to weights $p_{u,l}^{scl}$ (Equation 8) and compute $\mathcal{L}_{SCL}$ (Equation 10).
7:     Update user and item embeddings by minimizing the final loss $\mathcal{L}_{final}$ (Equation 11).
8: end for
```

B. Diversified Contrastive Learning

To further alleviate the issue that the predicted pseudo edges are noisy, we propose to incorporate contrastive learning into the co-training framework due to its effectiveness and robustness by contrasting different corrupted contrastive graph views of the same node against other nodes. However, the vanilla contrastive learning cannot get rid of two types of bias: (1) *Popularity Bias*: Popular items are interacted by a lot of users while unpopular items are interacted by few users. This means that popular items learn better item embeddings than unpopular items as popular items have more edges connected and get involved in training more. (2) *Category bias*: Generally, given historically interacted items of a user, there are advantage categories which the majority of items belong to and disadvantage categories that include few items. The user embeddings learned by LightGCN are more closer to the items of the advantage categories than those of the disadvantage categories, making the model to recommend items from a narrow range of categories. In this section, following our prior work [18], we present graph corruption and the diversified contrastive learning that alleviates the bias issues with popularity debiased unsupervised contrastive learning and category-rebalanced supervised contrastive learning.

a) Graph Corruption: Diversified contrastive learning refines the user and item embeddings by encouraging the consistency between the two corrupted views generated from the input graph. To this end, we adopt the common graph augmentation strategy [16], [17] to corrupt the input graph and generate two different graph views that contain incomplete information of node attributes and node topology by randomly discarding some observed user-item edges and node embedding dropout with a prior probability p.

b) Popularity Debiased Unsupervised Contrastive Learning: To perform unsupervised contrastive learning with node discrimination, the common practice is to adopt the items in the current batch as the anchor nodes and discriminate their different views against other nodes in the current batch [16], formulated as:

$$\mathcal{L}_{UCL}(I_{ucl}^{(c_1)}, I_{ucl}^{(c_2)}) = \sum_{i \in I} -\log \frac{\exp\left(f\left(e_i^{(c_1)}, e_i^{(c_2)}\right)/\tau\right)}{\sum_{z \in I} \exp\left(f\left(e_i^{(c_1)}, e_z^{(c_2)}\right)/\tau\right)} \quad (6)$$

where $f(\cdot,\cdot)$ indicates the cosine similarity function, and τ is the temperature hyper-parameter. (u, i) is the positive pair while (u, z) is the negative pair. However, this exacerbates the popularity bias since the popular items get involved in contrastive learning more than the unpopular items, resulting in that popular items learn better embeddings than unpopular items.

To mitigate the bias, we need to sample more unpopular items and less popular items by assigning higher weights on the unpopular items and lower weights on the popular items during sampling. Accordingly, we compute a weight p_i^{ucl} for each item by:

$$p_i^{ucl} = \frac{1}{D_i^{\alpha}} \quad (7)$$

where D_i is the item degree, which is measured as the quantity of the users that interacted with the item, and α is the coefficient to balance the popularity bias. During batch training, we sample a batch of items I_{ucl} based on the computed weights p_i^{ucl} to compute $\mathcal{L}_{UCL}(I_{ucl}^{(c_1)}, I_{ucl}^{(c_2)})$.

c) Category-balanced Supervised Contrastive Learning: Prior works solely employ the observed user-item interactions $\mathcal{Y}$ to compute the supervised BPR losses, overlooking their potential usefulness when it comes to contrastive learning.

Herein we propose a category-rebalanced supervised contrastive learning module to emphasize the interactions of the items of unpopular categories and the corresponding users. In other words, given a user and her historical interacted items, we intend to sample more items of unpopular categories and less items of popular categories instead of enumerating all the historical interacted items of the user.

To this end, we compute a category-rebalanced weight for each item of every user. Generally, suppose the interacted item set of user u is $I_u = [i_1, i_2, ..., i_l]$ where each item is associated with a category label and the category set interacted by the user is $C_u = [c_1, c_2, ..., c_x]$. Suppose that c_x is the corresponding category of item i_l, we compute a normalized weight for each item i_l as:

$$a_{u,l} = \frac{1}{|c_x|^{\gamma}}, \quad (8)$$

$$p_{u,l}^{scl} = \frac{a_{u,l}}{\sum_{j=1}^{|I_u|} a_{u,j}} \quad (9)$$

where $|c_x|$ and $|I_u|$ indicate the number of items in the category c_x and the total number of items interacted by the user, respectively. γ is used to balance the category bias.

Given the current batch of users U, we sample items according to the above weights to compose the positive pairs, and we sample unclicked items for each user in U to build the negative pairs. In the node embedding space, given an sampled tuple of (a user, a positive item, a negative item), we pull together the user embedding and the positive item embedding while pushing apart the user embedding and the negative item embedding. As more items of disadvantage categories are sampled, the learnt user embedding are pushed to the item embedding of disadvantage categories:

$$\mathcal{L}_{SCL}(U^{(c_1)}, I_{scl}^{(c_2)}) = \sum_{(u,i)\in(U,I_{scl})} -\log \frac{\exp\left(w_{ui} * f\left(e_u^{(c_1)}, e_i^{(c_2)}\right)/\tau\right)}{\sum_{q\in\mathcal{Q}} \exp\left(w_{uq} * f\left(e_u^{(c_1)}, e_q^{(c_2)}\right)/\tau\right)} \tag{10}$$

where $\mathcal{Q}$ indicates the item set that includes one positive item and sampled negative items for user u. w_{ui} indicates the learnable weight given a user and an item.

We trained the model in the multi-task learning fashion with the final loss, as measured by the summation of supervised BPR ranking loss [19], diversified contrastive learning loss, and $L2$ regularization:

$$\mathcal{L}_{final} = \mathcal{L}_{BPR} + \lambda_1 \mathcal{L}_{DCL} + \lambda_2 \|\Theta\|_2^2, \tag{11}$$

$$\mathcal{L}_{BPR} = \sum_{(u,i)\in\mathcal{E}_r, (u,j)\notin\mathcal{E}_r} -w_{ui} * \log\sigma\left(\hat{y}_{ui} - \hat{y}_{uj}\right), \tag{12}$$

$$\begin{aligned}\mathcal{L}_{DCL} =& \mathcal{L}_{UCL}(U^{(c_1)}, U^{(c_2)}) + \mathcal{L}_{UCL}(U^{(c_2)}, U^{(c_1)}) \\ &+ \mathcal{L}_{UCL}(I_{ucl}^{(c_1)}, I_{ucl}^{(c_2)}) + \mathcal{L}_{UCL}(I_{ucl}^{(c_2)}, I_{ucl}^{(c_1)}) \\ &+ \mathcal{L}_{SCL}(U^{(c_1)}, I_{scl}^{(c_2)}) + \mathcal{L}_{SCL}(U^{(c_2)}, I_{scl}^{(c_1)}).\end{aligned} \tag{13}$$

where λ_1 and λ_2 are two hyper-parameters to control the impacts of the diversified contrastive learning loss and $L2$ regularization term on the model parameters θ, respectively. As we fix one graph view and enumerate the users in the other view in Equation 6 and 10, we also need to compute the symmetric contrastive learning loss $\mathcal{L}_{UCL}(I_{ucl}^{(c_2)}, I_{ucl}^{(c_1)})$ and $\mathcal{L}_{SCL}(U^{(c_2)}, I_{scl}^{(c_1)})$, where $I_{ucl}^{(c_1)}$ and $I_{ucl}^{(c_2)}$ are the two different corrupted contrastive views.

IV. Experiments

To validate the superiority of our proposed CCT in terms of accuracy and diversity, we conduct extensive experiments to answer the following research questions:

- **RQ1**: How does CCT perform top-k recommendation task in terms of accuracy and diversity metrics, compared with other state-of-the-art (SOTA) models?
- **RQ2**: How do the components of CCT affect model performance with different settings?
- **RQ3**: How does the proposed model impact eclectic users and long-tail items?

A. Datasets

We adopt three datasets, **Amazon-music**, **Amazon-beauty**, and **Amazon-sport**, from Amazon review dataset collections to evaluate model performances across the experiments [20]. Category information of products is utilized to evaluate diversity. We report the statistics of the three datasets in Table II. For each dataset, we sort all the interactions for each user chronologically. We extract the latest 20% of interactions for each user as the testing set, and we build the validation set by taking the the latest 10% of the rest interactions of each user, using the rest interactions as training data.

TABLE II: Statistics of the datasets.

Datasets	Amazon-music	Amazon-beauty	Amazon-sport
# Users	5,541	22,363	35,598
# Items	3,568	12,101	18,357
# Interactions	64,706	198,502	296,337
# Categories	60	43	149
Density	0.0033	0.0007	0.0004

B. Metrics

We adopt widely-used **Recall@k** and **NDCG@k** [21] as two accuracy metrics. To evaluate model diversity, we employ two common metrics:

- **Intra-list Distance (ILD@k)** [22] measure the average humming distance of the category vectors of any two items in the recommendation item list:

$$ILD@k = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} \frac{1}{k(k-1)} \sum_{(i,j)\in\hat{Y}_u} |\,\mathrm{c}(i) - \mathrm{c}(j)| \tag{14}$$

  where $k = 20$, and $c(\cdot)$ returns the one-hot category vector for an item. $\hat{Y}_u$ is the item list recommended by a model for user u.
- **Category Coverage (CC@k)** [10] reports the average quantity of the unique categories of the recommended items:

$$CC@k = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} |C(u)| \tag{15}$$

  where $|C(u)|$ represents the number of unique categories covered by the recommended item list for user u.

C. Baselines

In the experiment, we mainly adopt different types of models as baselines for performance comparison:

- **MMR** [4] is a re-ranking method for diversified ranking problems by maximizing the marginal relevance.
- **DPP** [8] applies determinantal point process to optimize the trade-off between accuracy and diversity and generates recommendation list through the MAP inference. We use one-hot vector of categories as item features and LightGCN predicted ratings as item scores.
- **NCF** [23] is a popular non-graph-based method that utilizes neural networks with non-linear activation function to model interactions between user and item.
- **LightGCN** [13] is a SOTA graph-based collaborative filtering baseline that presents a light convolution message aggregation function.
- **SGL** [16] proposes unsupervised contrastive learning with a node discrimination task as a supplement to the supervised BPR loss.

TABLE I: Model Performance Comparison in terms of accuracy and diversity.

Datasets	Amazon-music (topk=20)				Amazon-beauty (topk=20)				Amazon-sport (topk=20)			
Methods	Recall	NDCG	ILD	CC	Recall	NDCG	ILD	CC	Recall	NDCG	ILD	CC
MMR	0.1945	0.0738	1.3597	6.6107	0.0994	0.0463	1.5038	7.3531	0.058	0.0213	1.1438	6.0053
DPP	0.1977	0.0765	1.4137	7.2496	0.0969	0.0491	1.5433	7.636	0.0561	0.0223	1.1936	6.3305
NCF	0.1891	0.0879	1.4018	7.1041	0.0812	0.0467	1.5769	8.0937	0.0379	0.0166	1.337	7.4165
LightGCN	0.2501	0.1122	1.3029	6.1782	0.121	0.0709	1.4064	6.5604	0.0639	0.0283	1.1008	5.5769
SGL	0.252	0.1138	1.3109	6.3021	0.125	0.0736	1.4264	6.9801	0.0642	0.0285	1.1569	6.0238
PD-GAN	0.2257	0.1027	1.3507	6.5902	0.092	0.0477	1.4934	7.188	0.0501	0.0207	1.274	5.8987
DGCN	0.1898	0.0843	1.5161	7.9948	0.0868	0.0481	1.5623	8.0356	0.0392	0.0158	1.3974	7.2071
CCT	**0.2536**	**0.113**	**1.5182**	**8.2335**	**0.1266**	**0.075**	**1.5849**	**8.1131**	**0.0643**	**0.0287**	**1.4509**	**8.0337**

- **PD-GAN** [9] proposes an end-to-end Generative Adversarial Network where a DPP-based generator recommend relevant and diversified items while a discriminator aims to distinguish between the recommended items and the ground-truth.
- **DGCN** [10] proposes rebalanced neighbor discovering, category-boosted negative sampling, and adversarial learning for GCN to improve diversity for recommendation.

D. Main Results (RQ1)

We first compare model performances for top-k diversified recommendation in terms of the aforementioned accuracy and diversity metrics and we adopt $k = 20$ across all experiments.

As reported in Table I, using the embeddings or rating scores generated by MMR and DPP achieves high diversity but sacrifice too much accuracy due to its post-processing nature. Compared with NCF that treat each user-item interaction independently, LightGCN shows higher accuracy by capturing the high-order connectivity among users and items on the user-item bipartite graph. However, it achieves lower diversity. High diversity usually comes along with low accuracy. SGL introduces unsupervised contrastive learning to alleviate the label-sparsity issue and learn better embeddings for user and item, improving accuracy and diversity to some extent [24]. SGL achieves the best accuracy among the baselines. Across the three datasets, our proposed model consistently yields the best overall performance in terms of both accuracy and diversity compared with strong baselines. Specifically, our proposed model outperforms SGL in terms of $ILD@20$ by 15.8%, 12.2% and 25.4% on the three datasets with comparable and even better accuracy, respectively. Looking at the comparison with SOTA end-to-end diversified recommendation methods, our proposed model outperforms PD-GAN by a large margin on both accuracy and diversity metrics. Although DGCN achieves comparable diversity to our proposed model, it suffers from much lower accuracy. We finally conduct significance testing (t-test) to show the significance of the improvements of our proposed CCL with fives runs and obtain $p < 10^{-5}$ compared with SGL with the highest accuracy in terms of ILD@20 and CC@20 on the three datasets, respectively.

E. Model Study (RQ2)

a) Ablation Study: We conduct ablation study to quantify the impact of the components in our proposed CCT and report the corresponding degradations in Table III, where DCL, CT, UCL, SCL, and LW indicate the diversified contrastive learning, co-training, unsupervised contrastive learning, supervised contrastive learning, and learnable weights, respectively. Each proposed component contributes in a certain degree to the improvement of model performance, which demonstrates the reasonability of our proposed architecture. Taking a close look at the results, removing contrastive learning and only utilizing self-training delivers higher diversity but lower accuracy, which demonstrates that contrastive learning effectively benefits the model when the pseudo labels are unreliable due to its noise-robust nature. On the other hand, training the model without co-training also hurts diversity, and it shows the effectiveness of co-training that reveals the user-interested novel items. Compared with LightGCN, it still achieves higher accuracy and diversity, demonstrating the validity of the proposed diversified contrastive learning.

b) Effect of the Quantity of Pseudo Edges: The quantity T of the pseudo edges plays a critical role in mining novel items that users like. Therefore, to show the impact of different quantity T, we plot the curves of model performance in terms of $Recall@20$ and $ILD@20$ with regard to different T in Figure 2. Generally, as the quantity T increases, more edges that connect to diverse items are added into the graph, improving the diversity; while more unreliable edges hurt the accuracy at the same time. Thanks to the robustness of contrastive learning, the accuracy degradation is smaller than the gain obtained on the diversity.

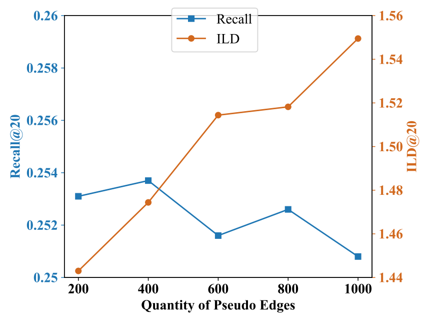
F. Benefits of (RQ3)

a) Effect on Eclectic Users: The problem of diversified recommendation is more critical for users with eclectic interests, and a good diversified recommendation model should have huge benefits to such users. We define eclectic users based on their affinity towards diverse items, as measured by the number of their historical interactions and the number of different categories they interacted with. For example, we extract users that interacted with at least 20 items and at least different 6 categories from each dataset as the eclectic users. We report the models' performance on the eclectic users of the three datasets in Table IV. Our proposed model consistently achieves the almost best accuracy and diversity at the same time.

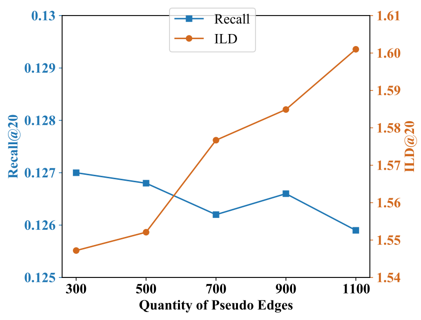
b) Long-tail Item Recommendation: Long-tail items are less interacted by users, and GNN-based models are easily

TABLE III: Ablation study.

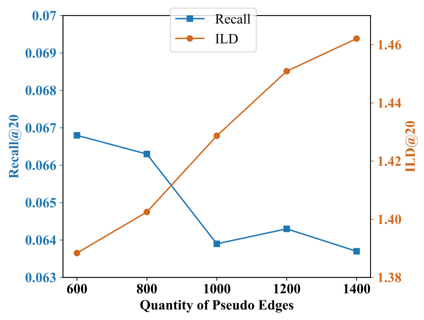
Datasets	Amazon-music (topk=20)				Amazon-beauty (topk=20)				Amazon-sport (topk=20)			
Methods	Recall	NDCG	ILD	CC	Recall	NDCG	ILD	CC	Recall	NDCG	ILD	CC
CCT	0.2536	0.113	1.5182	8.2335	0.1266	0.075	1.5849	8.1131	0.0643	0.0287	1.4509	8.0337
-DCL	0.2379 (-6.2%)	0.1089 (-3.6%)	1.4306 (-5.7%)	7.1037 (-13.7%)	0.1206 (-4.7%)	0.072 (-4%)	1.5147 (-4.4%)	7.4528 (-8.1%)	0.0548 (-14.7%)	0.0211 (-26.5%)	1.4008 (-3.4%)	7.6498 (-4.8%)
-CT	**0.2537 (+0.0%)**	**0.1143 (+1.1%)**	**1.4225 (-6.3%)**	**7.0627 (-14.2%)**	**0.1287 (+1.6%)**	**0.0765 (+2%)**	**1.4874 (-6.1%)**	**7.282 (-10.2%)**	**0.0663 (+3.1%)**	**0.0301 (+4.9%)**	**1.2548 (-13.5%)**	**6.6253 (-17.5%)**
-UCL	0.2477 (-2.3%)	0.1125 (-0.45%)	1.4509 (-4.4%)	7.3439 (-10.8%)	0.1233 (-2.6%)	0.0726 (-3.2%)	1.5243 (-3.8%)	7.5238 (-7.1%)	0.0632 (-1.7%)	0.0275 (-4.1%)	1.3532 (-6.8%)	7.3573 (-8.4%)
-SCL	0.2486 (-2%)	0.1128 (-0.1%)	1.4622 (-3.6%)	7.4903 (-9.0%)	0.1245 (-1.6%)	0.0738 (-1.6%)	1.5365 (-3.0%)	7.5870 (-6.4%)	0.0649 (+0.9%)	0.0289 (-0.7%)	1.3847 (-4.6%)	7.4552 (-7.2%)
-LW	0.2521 (-0.6%)	0.1112 (-1.5%)	1.453 (-4.3%)	7.4811 (-9.1%)	0.1247 (-1.5%)	0.0719 (-4.1%)	1.5339 (-3.2%)	7.5891 (-6.4%)	0.0617 (-4.0%)	0.0268 (-6.6%)	1.3650 (-5.9%)	7.4242 (-7.6%)



(a) Amazon-music



(b) Amazon-beauty



(c) Amazon-sport

Fig. 2: Performance with regard to different quantities of pseudo edges.

TABLE IV: Model performance Comparison on Eclectic Users

Datasets	Eclectic Users of Amazon-music (topk=20)				Eclectic Users of Amazon-beauty (topk=20)				Eclectic Users of Amazon-sport (topk=20)			
Methods	Recall	NDCG	ILD	CC	Recall	NDCG	ILD	CC	Recall	NDCG	ILD	CC
MMR	0.1046	0.0619	1.558	7.7464	0.1264	0.1007	1.6084	7.8801	0.0393	0.0281	1.3599	6.8648
DPP	0.1058	0.0637	1.5778	8.0846	0.1211	0.0983	1.6216	8.0385	0.0408	0.0298	1.4184	7.1428
PD-GAN	0.0923	0.0566	1.5009	7.546	0.1189	0.0937	1.6044	7.774	0.0331	0.0257	1.3374	6.6943
DGCN	0.0749	0.0453	1.6232	8.5768	0.1324	0.1125	1.6692	8.4713	0.027	0.02	1.5003	7.6316
NCF	0.087	0.0539	1.63	8.4818	0.1305	0.1129	1.6548	8.3769	0.0229	0.0157	1.4804	8.1898
LightGCN	0.104	0.0623	1.4317	7.2965	0.1499	0.1282	1.5983	7.6007	0.0413	0.0307	1.3327	6.4831
SGL	0.1081	0.067	1.5201	7.3317	0.1672	0.1496	1.5784	7.5267	0.0417	0.0313	1.3589	6.8319
CCT	**0.1139**	**0.069**	**1.6497**	**8.7966**	**0.1891**	**0.1664**	**1.6801**	**8.5832**	**0.0418**	**0.0318**	**1.5635**	**8.6888**

suffer from long-tail items recommendation as long-tail items have few edges that connected to user nodes, which is hard to learn good embeddings. We further compare model performances in terms of $Recall@20$ over the long-tail items, as reported in Figure 3. We sort all items according to item degrees from low to high, and we select the 10% items that have the least item degrees for each dataset. The proposed CTT consistently yields the best results compared to other methods. This benefits from the diversified contrastive learning as the long-tail items are sampled and updated by minimizing the contrastive learning loss more frequently. On the other hand, co-training connects some long-tail items to users that helps to learn better representations for long-tail items as more edges are involved in graph convolution, further improving the performance over the long-tail items. Therefore, the proposed CCT not only greatly improves individual diversity but also benefit the aggregate diversity.

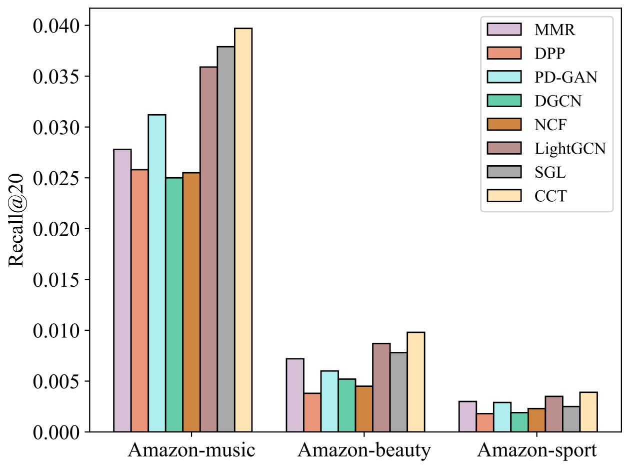


Fig. 3: Performance comparison over long-tail items.

G. Implementation Details

For the diversified contrastive learning, we adopt $\alpha = 0.5$ and $\gamma = 1$ to compute the item weights for item sampling in contrastive learning. Following the baselines, with the same hyper-parameters, we train the models using Adam optimizer [25] with learning rate of $1e-3$. We set embedding dimension as 64 for both user and item embedding and 3 layers of graph neural networks for all GCF methods across the experiments. We empirically adopt $\lambda_1 = 0.1$ and $\tau = 0.2$ for contrastive learning loss and $\lambda_2 = 1e-4$ for $L2$ regularization. We first pre-train the model for $100/60/50$ epochs and then predict pseudo edges on the three datasets, respectively. We expand user-item graph with $T = 750/900/1200$ predicted pseudo edges for every 5 epochs on the three datasets, and we reset the augmented graph with the original input graph for every 50 epochs. We tune the threshold with different values $[0.0, 0.5, 0.9, 0.95]$ according to the results on the validation data.

V. Related Work

a) Diversified Recommendation: Diversity of recommendation can be viewed at two levels: individual diversity and aggregate diversity. Individual diversity aims to measure the dissimilarity of the recommended items for each user as recommending similar items to users would bore users and hurt users satisfaction. Individual-level diversity reflects how many topics the recommendation covers and it can be evaluated by leveraging the category or the genre labels in item recommendation [2]. Aggregate diversity compares the recommendation results of different users and expects them to be dissimilar with each other. Aggregate diversity is sometimes called long-tail recommendation as low aggregate diversity means always recommending popular items to all users while ignoring long-tail items. In this paper, we focus on individual diversity.

Earlier, the rerank-based and post-processing methods (e.g. MMR [4]) are proposed to rerank the generated item candidates by maximizing the marginal relevance that decreases the redundancy of the ranked items. DPP is introduced to enhance the diversity as a parametric model to generate a subset of diversified items from a large pool of item candidates [3]. Several methods are further proposed to improve the efficiency of computing DPP [7], [26]. The post-processing methods are limited by the quality of the item generation stage and decoupled design also results in suboptimal performance. To tackle the issues, some end-to-end approaches are proposed by formulating it as a ranking problem and directly generate a recommended item list at inference time. [22] represent items in a similarity graph and formulate the relevance-diversity trade-off as finding a small set of unrated items that best cover the interacted items. PD-GAN propose an end-to-end GAN model where a DPP-based generator recommends relevant and diversified items while a discriminator aims to distinguish between the recommended items and the ground-truth [9]. DGCN propose rebalanced neighbor discovering, category-boosted negative sampling, and adversarial learning for GCN to improve the diversity for recommendation [10]. EDUA is a bilateral branch network that models both domain-level and user-level diversity for diversified recommendation. BGCN leverages Bayesian GCN to address the uncertainty in the user-item bipartite graph to improve diversity by learning various relationships between users and items [27].

b) Contrastive Learning: A surge of attention on contrastive learning has been dedicated to recommendation tasks to solve the label sparsity issue. SGL [16] generates multiple views of a node, maximizing the agreement between different views of the same node compared to that of other nodes. CCGL [28] and HeCo [24] are its two variants employed on either cascade or heterogeneous graphs. CLCRec [29] conducts a novel contrastive objective function to solve cold-start recommendation problem. GCC [30] is a graph contrastive coding framework to encode and discriminate sampled subgraphs. DHCN [31] and HCGR [32] both propose hypergraph convolutional networks for session-based recommendation with multi-view contrastive learning. NCE-PLRec [33] derives a closed-form of highly efficient linear recommendation algorithm to solve popularity recommendation bias. Different from it, our proposed methods alleviate the popularity bias by sampling anchor, the positive nodes, and the negative nodes for contrastive learning according to the item-level popularity and category-level popularity.

c) Co-training: In the context of recommendation, our work is highly related to EGLN [14] that augments the input user-item graph with a predicted residual graph to discover the user-interested items in the self-training manner. However, the work aims to improve the accuracy of recommendation and a learnable neural network is utilized to recover the adjacency matrix, which brings additional computational complexity. Different from it, our proposed method directly predicts the pseudo edges that connect users with items of novel categories by the inner product of user embedding and item embedding instead of a learnable neural network. [34] also proposes to connect users to diverse items in the input graph iteratively. But it only utilizes pseudo edges without leveraging contrastive learning. On the contrary, we utilize contrastive learning to make the model robust to the unreliable pseudo edges. Moreover, we propose to utilize co-training with learnable weights to improve the quality of the pseudo edges. Another related work is [35] for social recommendation where tri-training is used to distill the social influence from the user-user graph to user-item graph. The work does not add pseudo edges on the graph but adopt a soft way to enforce the consistence between different views with contrastive learning. Distinct from it, our proposed model adds pseudo edges that connect users to user-interested novel items to augment the graph. The pseudo edges are used as supervision signals in both supervised BPR loss and contrastive learning loss. Furthermore, we resort to the specially designed contrastive learning to further improve the diversity.

VI. Conclusion

In this paper, we propose contrastive co-training for diversified recommendation that integrates co-training and contrastive learning which delivers the best diversity and accuracy trade-off at the same time. On the one hand, co-training unearths the novel item that users might be interested in and predicts pseudo user-item edges to augment the graphs. On the other hand, diversified contrastive learning mitigate the damage when the pseudo labels are unreliable and further improve diversity by re-balancing item-level popularity and category-level popularity. In the future, we intend to further improve diversity by exploring methods to learn the uncertainty for the pseudo edges and utilize the edges that are of low confidence but might be true.

References

[1] S. Wu, F. Sun, W. Zhang, and B. Cui, "Graph neural networks in recommender systems: a survey," *arXiv preprint arXiv:2011.02260*, 2020.

[2] C. Gao, Y. Zheng, N. Li, Y. Li, Y. Qin, J. Piao, Y. Quan, J. Chang, D. Jin, X. He, and Y. Li, "Graph neural networks for recommender systems: Challenges, methods, and directions," *ArXiv*, vol. abs/2109.12843, 2021.

[3] P. Cheng, S. Wang, J. Ma, J. Sun, and H. Xiong, "Learning to recommend accurate and diverse items," *Proceedings of the 26th International Conference on World Wide Web*, 2017.

[4] J. G. Carbonell and J. Goldstein-Stewart, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in *SIGIR '98*, 1998.

[5] L. Qin and X. Zhu, "Promoting diversity in recommendation by entropy regularizer," in *IJCAI*, 2013.

[6] C. Sha, X. Wu, and J. Niu, "A framework for recommending relevant and diverse items," in *IJCAI*, 2016.

[7] M. Gartrell, U. Paquet, and N. Koenigstein, "Low-rank factorization of determinantal point processes," in *AAAI*, 2017.

[8] L. Chen, G. Zhang, and E. Zhou, "Fast greedy map inference for determinantal point process to improve recommendation diversity," in *NeurIPS*, 2018.

[9] Q. Wu, Y. Liu, C. Miao, B. Zhao, Y. Zhao, and L. Guan, "Pd-gan: Adversarial learning for personalized diversity-promoting recommendation," in *IJCAI*, 2019.

[10] Y. Zheng, C. Gao, L. Chen, D. Jin, and Y. Li, "Dgcn: Diversified recommendation with graph convolutional networks," *Proceedings of the Web Conference 2021*, 2021.

[11] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

[12] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

[13] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

[14] Y. Yang, L. Wu, R. Hong, K. Zhang, and M. Wang, "Enhanced graph learning for collaborative filtering via mutual information maximization," *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[15] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *COLT' 98*, 1998.

[16] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 726–735.

[17] Z. Liu, Y. Ma, Y. Ouyang, and Z. Xiong, "Contrastive learning for recommender system," *arXiv preprint arXiv:2101.01317*, 2021.

[18] X. Ma, Z. Gao, Q. Hu, and M. AbdelHady, "Contrastive knowledge graph attention network for request-based recipe recommendation."

[19] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009.

[20] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," *Proceedings of the 25th International Conference on World Wide Web*, 2016.

[21] K. Järvelin and J. Kekäläinen, "Ir evaluation methods for retrieving highly relevant documents," *SIGIR Forum*, vol. 51, pp. 243–250, 2017.

[22] S. P. Parambath, N. Usunier, and Y. Grandvalet, "A coverage-based approach to recommendation diversity on similarity graph," *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016.

[23] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," *Proceedings of the 26th International Conference on World Wide Web*, 2017.

[24] X. Wang, N. Liu, H. Han, and C. Shi, "Self-supervised heterogeneous graph neural network with co-contrastive learning," *arXiv preprint arXiv:2105.09111*, 2021.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[26] J. Gillenwater, A. Kulesza, Z. E. Mariet, and S. Vassilvitskii, "A tree-based method for fast repeated sampling of determinantal point processes," in *ICML*, 2019.

[27] J. Sun, W. Guo, D. Zhang, Y. Zhang, F. Regol, Y. Hu, H. Guo, R. Tang, H. Yuan, X. He, and M. J. Coates, "A framework for recommending accurate and diverse items using bayesian graph convolutional neural networks," *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[28] X. Xu, F. Zhou, K. Zhang, and S. Liu, "Ccgl: Contrastive cascade graph learning," *arXiv preprint arXiv:2107.12576*, 2021.

[29] Y. Wei, X. Wang, Q. Li, L. Nie, Y. Li, X. Li, and T.-S. Chua, "Contrastive learning for cold-start recommendation," *arXiv preprint arXiv:2107.05315*, 2021.

[30] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "Gcc: Graph contrastive coding for graph neural network pre-training," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*, 2020, pp. 1150–1160.

[31] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, and X. Zhang, "Self-supervised hypergraph convolutional networks for session-based recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4503–4511.

[32] N. Guo, X. Liu, S. Li, Q. Ma, Y. Zhao, B. Han, L. Zheng, K. Gao, and X. Guo, "Hcgr: Hyperbolic contrastive graph representation learning for session-based recommendation," *arXiv preprint arXiv:2107.05366*, 2021.

[33] G. Wu, M. Volkovs, C. L. Soon, S. Sanner, and H. Rai, "Noise contrastive estimation for one-class collaborative filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 135–144.

[34] R. Ye, Y. Hou, T. Lei, Y. Zhang, Q. Zhang, J. Guo, H. Wu, and H. Luo, "Dynamic graph construction for improving diversity of recommendation," *Fifteenth ACM Conference on Recommender Systems*, 2021.

[35] J. Yu, H. Yin, M. Gao, X. Xia, X. Zhang, and N. Q. V. Hung, "Socially-aware self-supervised tri-training for recommendation," *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021.