Contrastive Collaborative Filtering for Cold-Start Item Recommendation

Zhihui Zhou*
School of Computer Science
Sichuan University, China
zhouzhihui_cs@stu.scu.edu.cn

Lilin Zhang*
School of Computer Science
Sichuan University, China
zhanglilin@stu.scu.edu.cn

Ning Yang†
School of Computer Science
Sichuan University, China
yangning@scu.edu.cn

ABSTRACT

The cold-start problem is a long-standing challenge in recommender systems. As a promising solution, content-based generative models usually project a cold-start item's content onto a warm-start item embedding to capture collaborative signals from item content so that collaborative filtering can be applied. However, since the training of the cold-start recommendation models is conducted on warm datasets, the existent methods face the issue that the collaborative embeddings of items will be blurred, which significantly degenerates the performance of cold-start item recommendation. To address this issue, we propose a novel model called Contrastive Collaborative Filtering for Cold-start item Recommendation (CCFCRec), which capitalizes on the co-occurrence collaborative signals in warm training data to alleviate the issue of blurry collaborative embeddings for cold-start item recommendation. In particular, we devise a contrastive collaborative filtering (CF) framework, consisting of a content CF module and a co-occurrence CF module to generate the content-based collaborative embedding and the co-occurrence collaborative embedding for a training item, respectively. During the joint training of the two CF modules, we apply a contrastive learning between the two collaborative embeddings, by which the knowledge about the co-occurrence signals can be indirectly transferred to the content CF module, so that the blurry collaborative embeddings can be rectified implicitly by the memorized co-occurrence collaborative signals during the applying phase. Together with the sound theoretical analysis, the extensive experiments conducted on real datasets demonstrate the superiority of the proposed model. The codes and datasets are available on https://github.com/zzhin/CCFCRec.

CCS CONCEPTS

• **Information system → Recommender systems**.

KEYWORDS

Recommender Systems, Cold-start Recommendation, Contrastive Learning

ACM Reference Format:
Zhihui Zhou, Lilin Zhang, and Ning Yang. 2023. Contrastive Collaborative Filtering for Cold-Start Item Recommendation. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3543507.3583286

1 INTRODUCTION

In the era of information explosion, recommender systems have been playing a critical role for mitigating information overload in various online applications. As the most successful technique for personalized recommender systems, collaborative filtering (CF) aims at predicting items that are of interest to a specific user, based on the learned user and item embeddings capturing collaborative signals from observed user-item interactions [11]. However, the interactions of new users or new items are often too sparse for reliable user/item representation learning, which leads to the so-called cold-start problem that has stubbornly plagued recommender systems for a long time [25].

In this paper, we focus on cold-start item recommendation, i.e., recommending new items to existing users, which has aroused increasing efforts from the research community. One promising research line falls into the category of content-based methods, which usually train a generative model to project a cold item's content, e.g. attributes, text, and image, etc., onto a warm item embedding space [2, 3, 16, 19, 21, 24, 27, 32, 46]. For example, DropoutNet [32] implicitly transforms a cold-start item's content to a warm embedding with randomly dropping the warm item embeddings that are learned during the training process. MWUF [46] generates warm embeddings for cold-start items based on their features and ID embeddings with two meta networks. Recently, inspired by the impressive success of Generative Adversarial Network (GAN) [9], GAN-based methods have been proposed for cold-start item recommendation. For example, LARA [27] exploits an adversarial neural network with multiple generators to learn a mapping from cold-start items' attributes to user embeddings to generate virtual users for cold-start items.

Essentially, the crux of content-based methods for cold-start item recommendation is to capture collaborative signals from item content so that collaborative filtering can be applied to cold-start items. For this purpose, the existent methods often simulate the cold-start scenario during the training phase by removing the interaction records of warm items from the model input and only preserving the item content, which arguably leads to the issue of **blurry collaborative embeddings**. Figure 1 gives an illustration, where the movies Starsky & Hutch and Rent-A-Cop are a positive sample and a negative sample of the user, respectively. During the training phase, the existent content-based methods will first embed the movie attributes, Genre and Star, and then generate the movie's

[*] Zhihui Zhou and Lilin Zhang share the first authorship.
[†] Ning Yang is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WWW '23, April 30–May 04, 2023, Austin, TX, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583286

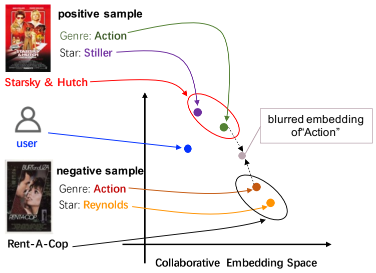


Figure 1: Illustration of blurry collaborative embedding.

Content-Based Collaborative Embedding (CBCE) by aggregating its attribute embeddings. When the positive sample Starsky & Hutch is fed into the existent models, the training algorithm will optimally push the embedding of its Genre value 'Action' to the User Collaborative Embedding (UCE) of the user, and the CBCE of Starsky & Hutch will also be consequently close to it. In contrast, the embedding of the Genre value 'Action' of the negative sample Rent-A-Cop will be pulled away from the UCE so that the CBCE of Rent-A-Cop will be distant from it. Therefore, the final embedding of 'Action' will become blurry with the average of the 'Action' embeddings of the positive and negative samples. If in fact the user likes Action movies and did not watch Rent-A-Cop just because of the dislike of the star Reynolds, then the final blurry embedding of 'Action' will lose the user's actual preference to Action movie, which improperly pulls the positive sample's CBCE away from and pushes the negative one's CBCE close to the UCE, compared with the optimal positions where they should be located.

On the other hand, warm datasets are affluent in co-occurrence collaborative signals of items, i.e., the patterns that some movies are frequently watched together with the same user, which are usually abandoned by the existent methods to mimic the cold-start scenarios on warm data. In sharp contrast with the existent methods, in this paper, we aim at to *capitalize on the co-occurrence collaborative signals in warm data to alleviate the issue of blurry collaborative embeddings for cold-start item recommendation.* Co-occurrence collaborative signals clearly indicate that the embeddings of positive items of a user should be closer to the user's UCE than those of negative items of the same user. Therefore, if we can inject the co-occurrence collaborative signals into the CBCEs, then the blurry CBCEs can be rectified with respect to them. However, it is not trivial to achieve this goal. Unlike warm-start recommendation models which explicitly establish the mapping from historical interactions to embeddings, only item content is available to a cold-start item recommendation model, and CBCEs of cold-start items cannot explicitly and directly encode the co-occurrence collaborative signals because they are unavailable to cold-start items.

To overcome the above challenges, in this paper, we propose a novel model called Contrastive Collaborative Filtering for Cold-start item Recommendation (CCFCRec), of which the main idea is to *teach the CF module to memorize the co-occurrence collaborative signals during the training phase and how to rectify the blurry CBCEs of cold-start items according to the memorized co-occurrence collaborative signals when applying the model.* In particular, we propose a contrastive CF framework consisting of a content CF module and a co-occurrence CF module, by which CCFCRec can generate a CBCE and a **Co-Occurrence Collaborative Embedding (COCE)** for a training item, respectively. As the co-occurrence collaborative signals are unavailable for cold-start items, we present an indirect strategy to inject the co-occurrence collaborative signals into the content CF module during the training process where the co-occurrence collaborative signals are available for training items, instead of directly encoding them in CBCEs. For this purpose, CCFCRec will maximize the mutual information between CBCE and COCE of an item with an elaborate contrastive learning, which enforces the co-occurrence collaborative signals captured by COCE to be memorized by the content CF module in training phase and available in applying phase. At the same time, the two CF modules are jointly trained within a multi-task learning framework. Such multi-task optimization enforces the consistency on the interaction probability predictions that are made respectively by the content CF module and the co-occurrence CF module based on CBCE and COCE of an item, which brings the advantage of the positive transfer of the co-occurrence collaborative signals to the content CF module. At last, we also provide a sound theoretical analysis rooted in information theory to justify our contrastive collaborative filtering from the perspective of reinforcing the blurry CBCEs. The contributions of this paper are summarized as follows:

- We propose a novel model called Contrastive Collaborative Filtering for Cold-start item Recommendation (CCFCRec), which can rectify the blurry collaborative embeddings of cold-start items with the co-occurrence collaborative signals that are memorized during the training phase.
- We devise a contrastive collaborative filtering framework, by which the knowledge about co-occurrence collaborative signals can be transferred to the content CF module for cold-start items.
- We conduct extensive experiments on real datasets to verify the superiority of CCFCRec.

2 PROBLEM FORMULATION

Let $\mathcal{U}$ and $\mathcal{V}$ be a set of users and a set of items, respectively. Let $\mathcal{O} = \{o_{u,v}\}$ where $o_{u,v}$ is the observed interaction between a user $u \in \mathcal{U}$ and an item $v \in \mathcal{V}$. Let $\mathcal{V}_u \subseteq \mathcal{V}$ be the set of items that user $u \in \mathcal{U}$ interacted with, and $\mathcal{U}_v \subseteq \mathcal{U}$ be the set of users who interacted with item $v \in \mathcal{V}$. Each item v is associated with a attribute set $\mathcal{X}_v$ consisting of m attributes $\{x_1^{(v)}, \cdots, x_m^{(v)}\}$, where an attribute is represented as a one-hot vector, e.g., movie director, a multi-hot vector, e.g., movie genre, or a real valued vector, e.g., item image. Let $\mathbf{x}_i^{(v)} \in \mathbb{R}^{d \times 1}$ be the embedding of the ith attribute $x_i^{(v)}$, where d is the dimensionality of embedding and $1 \leq i \leq m$.

Given a warm training dataset $\mathcal{D}$ consisting of $\{\mathcal{U}, \mathcal{V}, \mathcal{O}\}$, we want to learn a cold-start item recommendation model to estimate the probability $\hat{y}_{u,v}$ that a user u will interact with an item v: $\hat{y}_{u,v} = R(u, v, \mathcal{X}_v)$, such that for any pair of $(u^+ \in \mathcal{U}_v, u^- \notin \mathcal{U}_v)$ in the training dataset, $\hat{y}_{u^+,v} > \hat{y}_{u^-,v}$.

3 METHODOLOGY

3.1 Overview

Figure 2 shows the architecture of CCFCRec, where the solid lines represent the paths activated when both training and applying,

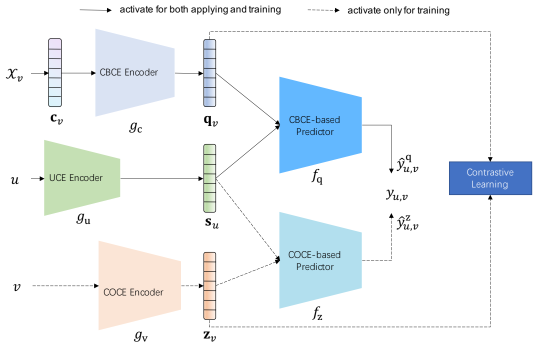


Figure 2: The overview of the proposed model CCFCRec.

while the dashed lines represent the paths activated only when training. In Figure 2, the content CF module includes the CBCE Encoder g_c and the CBCE-based predictor f_q, the co-occurrence CF module includes the COCE Encoder g_v and the COCE-based predictor f_z, and the UCE Encoder g_u is shared between the two CF modules.

For a training sample $(u, v, \mathcal{X}_v, y_{u,v})$, CCFCRec first invokes the encoder g_c to produce the CBCE $\mathbf{q}_v \in \mathbb{R}^{d\times 1}$ based on the content embedding $\mathbf{c}_v$ which is obtained by aggregating the attribute embeddings of $\mathcal{X}_v$. At the same time, CCFCRec also generates the COCE $\mathbf{z}_v \in \mathbb{R}^{d\times 1}$ of item v and the UCE $\mathbf{s}_u \in \mathbb{R}^{d\times 1}$ of user u, via the encoders g_v and g_u, respectively. Then CCFCRec will make the prediction of the interaction probability $\hat{y}^\mathrm{q}_{u,v}$, by invoking the CBCE-based predictor f_q with the CBCE $\mathbf{q}_v$ and the UCE $\mathbf{u}$ as input. Similarly, the COCE-based predictor f_z will be invoked to make the prediction of the interaction probability $\hat{y}^\mathrm{z}_{u,v}$ based on the COCE $\mathbf{z}_v$ and the UCE $\mathbf{s}_u$.

To transfer the knowledge about the co-occurrence collaborative signals into the content CF module, during the training process CCFCRec will conduct a contrastive learning between $\mathbf{q}_v$ and $\mathbf{z}_v$ with optimizing objective to maximize their mutual information, which essentially captures the triple correlation $\mathbf{c}_v \leftrightarrow \mathbf{q}_v \leftrightarrow \mathbf{z}_v$. Although $\mathbf{z}_v$ is unavailable for a cold-start item when applying the model, the content CF module will memorize the triple correlation captured by the contrastive learning during the training phase and rectify a blurry $\mathbf{q}_v$ according to the memorized correlation during the applying phase, as if it were rectified with respect to $\mathbf{z}_v$ (because of the indirect correlation $\mathbf{c}_v \leftrightarrow \mathbf{z}_v$).

At last, it is noteworthy that the two CF modules are jointly trained with the same supervision signal $y_{u,v}$ in a manner of multi-task learning. As mentioned before, the joint training ensures the positive transfer of the co-occurrence collaborative signals from the co-occurrence CF module to the content CF module.

3.2 Collaborative Embedding

3.2.1 CBCE. For a training item v, we first generate the attribute embeddings $\{\mathbf{x}_i^{(v)}\}$, $1 \le i \le m$, which depends on how the attributes are encoded. If an attribute $x_i^{(v)}$ is represented as a one-hot or multi-hot vector, then its attribute embedding can be obtained via a lookup over a learnable embedding matrix $\mathbf{W}_i \in \mathbb{R}^{d\times |x_i^{(v)}|}$, i.e., $\mathbf{x}_i^{(v)} = \mathbf{W}_i x_i^{(v)}$. If $x_i^{(v)}$ is an image, then $\mathbf{x}_i^{(v)}$ can be obtained via a pre-trained convolutional network like VGG [26]. Once the attribute embeddings $\{\mathbf{x}_i^{(v)}\}$ are ready, we generate the content embedding $\mathbf{c}_v \in \mathbb{R}^{md\times 1}$ by a concatenation of $\{\mathbf{x}_i^{(v)}\}$. At last, the CBCE $\mathbf{q}_v$ is produced via g_c with $\mathbf{c}_v$ as input:

$$\mathbf{q}_v = g_\mathrm{c}(\mathbf{c}_v). \tag{1}$$

To capture the nonlinear interplays between attributes, we implement g_c as a two-layer MLP with activation function LeakyReLU.

3.2.2 COCE and UCE. The COCE $\mathbf{z}_v$ and UCE $\mathbf{s}_u$ encode the preference of item v received from users and the preference of user u offered to items, respectively, by capturing the co-occurrence collaborative signals which are hidden in the observed interactions between users and items. Therefore, we represent an item v with a multi-hot vector $\mathbf{v} \in \{0, 1\}^{|\mathcal{U}|\times 1}$, and the u-th component $\mathbf{v}(u) = 1$ if $u \in \mathcal{U}_v$, otherwise $\mathbf{v}(u) = 0$. Similarly, a user u is represented with a multi-hot vector $\mathbf{u} \in \{0, 1\}^{|\mathcal{V}|\times 1}$, and the v-th component $\mathbf{u}(v) = 1$ if $v \in \mathcal{V}_u$, otherwise $\mathbf{u}(v) = 0$. For simplicity, the COCE encoder g_v and the UCE encoder g_u are implemented as a linear combination over the columns of an embedding matrix, i.e.,

$$\mathbf{z}_v = g_\mathrm{v}(\mathbf{v}) = \mathbf{W}_v\mathbf{v}, \tag{2}$$

$$\mathbf{s}_u = g_\mathrm{u}(\mathbf{u}) = \mathbf{W}_u\mathbf{u}, \tag{3}$$

where $\mathbf{W}_v \in \mathbb{R}^{d\times|\mathcal{U}|}$ and $\mathbf{W}_u \in \mathbb{R}^{d\times|\mathcal{V}|}$ are the learnable embedding matrices.

3.3 Contrastive Collaborative Filtering

Basically, CBCE captures the preference of users to attributes of items, while COCE captures the co-occurrence collaborative signals from the observed interactions. As mentioned before, the co-occurrence collaborative signals can help alleviate the issue of blurry CBCE. However, we cannot directly encode them into a CBCE since they are available only for warm-start items during the training phase. Therefore, we take a novel strategy to teach the content CF module to memorize the co-occurrence collaborative signals in its parameters during the training phase, so that the content CF module is able to rectify the blurry CBCEs of cold-start items with respect to the memory during the applying phase. For this purpose, we regard the CBCE $\mathbf{q}_v$ and the COCE $\mathbf{z}_v$ as the content view and behavior view of an item v and conduct a contrastive learning between the two item views. In particular, during the training phase, the parameters of CBCE encoder g_c will be adjusted according to COCE so as to maximize the mutual information between the two item views. To achieve this goal, for a training item v, we build its positive sample set as $\mathcal{N}_v^+ = \{v^+ : \mathcal{U}_v \cap \mathcal{U}_{v^+} \neq \emptyset\}$, i.e., the items with which some user interacted together with v, and its negative sample set $\mathcal{N}_v^- = \mathcal{V} \setminus \mathcal{N}_v^+$. To maximize the mutual information between $\mathbf{q}_v$ and $\mathbf{z}_{v^+}$, and minimize that between $\mathbf{q}_v$ and $\mathbf{z}_{v^-}$, following the idea of InfoNCE [20], we define the contrastive loss as follow:

$$\mathcal{L}_\mathrm{c} = -\mathbb{E}_{v\in\mathcal{D}, v^+\in\mathcal{N}_v^+}\left[\ln \frac{\exp(\frac{\langle \mathbf{q}_v, \mathbf{z}_{v^+}\rangle}{\tau})}{\exp(\frac{\langle \mathbf{q}_v, \mathbf{z}_{v^+}\rangle}{\tau}) + \sum_{v^-\in\mathcal{N}_v^-}\exp(\frac{\langle \mathbf{q}_v, \mathbf{z}_{v^-}\rangle}{\tau})}\right], \tag{4}$$

where $\langle\cdot,\cdot\rangle$ represents inner product.

3.4 Interaction Prediction

During the training phase, CCFCRec will make two predictions of the probability of the interaction between a user u and an item v, invoking the CBCE-based predictor f_{q} and the COCE-based predictor f_{z}, respectively. As the predictors rank the similarity of user and item representations, we implement the predictors as inner product for its simplicity, i.e.,

$$\hat{y}^{\mathrm{q}}_{u,v} = g_{\mathrm{q}}(\mathbf{q}_v, \mathbf{s}_u) = \langle \mathbf{q}_v, \mathbf{s}_u \rangle, \tag{5}$$

$$\hat{y}^{\mathrm{z}}_{u,v} = g_{\mathrm{q}}(\mathbf{z}_v, \mathbf{s}_u) = \langle \mathbf{z}_v, \mathbf{s}_u \rangle. \tag{6}$$

For a training item v, we pair each positive user $u^+ \in \mathcal{V}_u$ with one negative user u^- which is sampled from $\mathcal{V} \setminus \mathcal{V}_u$. Then by applying the popular pair-wise ranking loss BPR [23], we define the following loss functions for the joint training of the two predictors:

$$\mathcal{L}_{\mathrm{q}} = -\sum_{(v,u^+,u^-)\in\mathcal{D}} \ln \sigma(\hat{y}^{\mathrm{q}}_{u^+,v} - \hat{y}^{\mathrm{q}}_{u^-,v}), \tag{7}$$

$$\mathcal{L}_{\mathrm{z}} = -\sum_{(v,u^+,u^-)\in\mathcal{D}} \ln \sigma(\hat{y}^{\mathrm{z}}_{u^+,v} - \hat{y}^{\mathrm{z}}_{u^-,v}), \tag{8}$$

where σ is the sigmoid function.

It is noteworthy that in training phase, the two CF modules share the UCE $\mathbf{s}_u$ and are jointly trained with the same supervision. Such multi-task learning offers a consistent optimizing objective which ensures the positive transfer of the co-occurrence collaborative signals to the content CF module.

3.5 Overall Loss Function

The overall loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{\mathrm{q}} + \mathcal{L}_{\mathrm{z}} + \lambda\mathcal{L}_{\mathrm{c}} + \|\Theta\|, \tag{9}$$

where Θ represents the learnable parameters, and λ is a factor used for regularizing the contribution of the contrastive loss. In this paper, we choose Adam [14] as the optimizer.

3.6 Theoretical Analysis

3.6.1 Justification of Contrastive Collaborative Filtering. Let C, Q, U denote the random variables of item content embedding, item CBCE, and user, respectively. A cold-start item recommendation model can be essentially seen as a projection from item content through CBCE to user, $P(\cdot): C \rightarrow Q \rightarrow \hat{U}$, where Q can be seen as a hidden variable and $\hat{U}$ is the random variable representing a predicted user. Obviously, the projection $P(\cdot)$ can be broken into two steps. The first step is to use an encoder $g(\cdot): C \rightarrow Q$ to generate CBCE based on item content, and then use a predictor $f(\cdot): Q \rightarrow \hat{U}$ to predict the user.

Let U be the clean labels (supervision signals), and then the learning of $P(\cdot)$ can be framed as a Markov chain $U \rightarrow C \xrightarrow{g} Q \xrightarrow{f} \hat{U}$ with objective to maximize the mutual information $I(\hat{U}, U)$, in terms of the theory of information bottleneck [1, 29, 30]. However, the data always come together with noise. For example, in our case, the blurry collaborative embeddings can be viewed as the consequence of noisy interactions. Let U_e be the labels with noise E. For simplicity, we suppose $U_e = U + E$ and $U \perp E$. Then the learning becomes the Markov chain $U_e \rightarrow C \xrightarrow{g} Q \xrightarrow{f} \hat{U}$ which maximizes $I(\hat{U}, U_e)$ instead of $I(\hat{U}, U)$. However, $I(\hat{U}, U_e) = I(\hat{U}, U) + I(\hat{U}, E) \geq I(\hat{U}, U)$. In other words, $I(\hat{U}, U_e)$ is the upper bound of $I(\hat{U}, U)$, and optimizing the upper bound leads to suboptimal result.

When we introduce the contrastive learning, minimizing the contrastive loss defined by Equation (4) is equivalent to minimizing the mutual information between negative samples [20], i.e., $I(\hat{U}, Z^-)$, where Z^- represents the negative sample variable. Now let's take a close look at Figure 1 again. From Figure 1 we can see that the noise is incurred by the negative sample since it confuses the learning of the user's preference to the genre 'Action'. Base on this observation, it is reasonable to regard E as a statistics of Z^-. According to data processing inequality [28], we have $I(\hat{U}, E) \leq I(\hat{U}, Z^-)$. Considering $I(\hat{U}, U) = I(\hat{U}, U_e) - I(\hat{U}, E)$, then we have $I(\hat{U}, U) \geq I(\hat{U}, U_e) - I(\hat{U}, Z^-)$. Therefore, our CCFCRec not only maximizes $I(\hat{U}, U_e)$, but also minimizes $I(\hat{U}, Z^-)$. In other words, CCFCRec maximizes the lower bound of $I(\hat{U}, U)$, which can achieve the solution closer to the optimal one.

3.6.2 Connection to Supervised Contrastive Learning. The projection $P(\cdot): C \rightarrow \hat{U}$ is actually a multi-label classification task, where item content C is the input and user U is the label. Essentially, the contrastive CF selects the positive and negative views for an item with respect to label information, i.e., if $\mathcal{U}_v \cap \mathcal{U}_{v'} \neq \emptyset$, then v' is the positive view of v, otherwise the negative view. As the supervision signals are leveraged for the building of contrastive views, the contrastive CF can be regarded as a Supervised Contrastive Learning (SCL) [13]. In fact, the contrastive loss $\mathcal{L}_{\mathrm{c}}$ defined in Equation (4) can be rewritten as follow:

$$\mathcal{L}_{\mathrm{c}} = -\mathbb{E}_{v\in\mathcal{D}, v^+\in\mathcal{N}^+_v} \sum_{v'\in\mathcal{D}} \mathbb{I}(v \neq v')\mathbb{I}(\mathcal{U}_v \cap \mathcal{U}_{v'} \neq \emptyset) f(v, v'), \tag{10}$$

where $f(v, v') = \ln\left[\frac{\exp(\frac{\langle \mathbf{q}_v, \mathbf{z}_{v'} \rangle}{\tau})}{\sum_{v \neq v'} \exp(\frac{\langle \mathbf{q}_v, \mathbf{z}_{v'} \rangle}{\tau})}\right]$, and $\mathbb{I}(x) = 1$ if x is true, otherwise $\mathbb{I}(x) = 0$. Therefore, the label information ($\mathcal{U}_v$ and $\mathcal{U}_{v'}$) implicitly plays the role of supervision signals for the contrastive CF, which further helps the contrastive CF distinguish the positive sample embeddings from the negative ones.

4 EXPERIMENTS

The experiments aim to answer the following research questions:

- **RQ1** How does CCFCRec perform as compared to the state-of-the-art cold-start item recommendation models?
- **RQ2** How do the contrastive CF and the joint training contribute to the performance of CCFCRec?
- **RQ3** How do the hyper-parameters affect the performance of CCFCRec?
- **RQ4** How can the superiority of CCFCRec be illustrated with intuitive and visualizable case studies?

4.1 Experimental Settings

4.1.1 Datasets. We conduct the experiments on the following two public datasets: MovieLens-20M (ML-20M)[1] and Amazon Video Games (Amazon-VG)[2]. The statistics of the datasets are given in Table 1. On each dataset, we randomly select 70% of the data as training set, 15% as validation set, and 15% as testing set. The raw

[1] https://grouplens.org/datasets/movielens/

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

Dataset	#Interactions	#Users	#Items	Sparsity
ML-20M	19,904,260	138,493	24,003	0.598%
Amazon-VG	475,952	52,965	35,322	0.025%

Table 1: Statistics of the experimental dataset.

embeddings of the images in the datasets are obtained through pre-trained model VGG [26].

4.1.2 Baseline Methods. We compare CCFCRec with the following cold-start item recommendation methods:

- **NFM** [10] is a model for sparse data prediction, which combines the linearity of FM in modeling second-order feature interactions and the non-linearity of neural networks in modeling higher-order feature interactions.
- **LARA** [27] is a cold-start recommendation model based on GAN, which jointly models user profiles from different attributes of the item using an adversarial neural network with multiple generators.
- **MTPR** [6] is a generic learning framework, which optimizes the dual item representations consisting of normal and counterfactual representations so that the training-testing discrepancy can be solved for cold-start items.
- **MvDGAE** [44] is a cold-start recommendation model based on Heterogeneous Information Networks. MvDGAE treats the cold-start problem as a missing data problem and employs a denoise graph auto-encoder framework, which randomly drops out some user-item interaction views in the encoders but forces the decoders to reconstruct the full view information.
- **CVAR** [43] employs latent variables to encode a distribution over item content information and warms up ID embeddings of cold-start items with a conditional decoder.
- **CLCRec** [33] is another cold-start recommendation model based on contrastive learning, which enhances the behavioral features of cold-start items with a contrastive learning between an item's behavior view and content view.

4.1.3 Evaluation Protocols. We adopt the widely used metrics Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) to evaluate the performance of CCFCRec and the baseline methods. In particular, for a testing item v, we generate a ranking user list l_v, and the HR@k of v is $\frac{\sum_{u\in\mathcal{U}_v}\mathbb{I}\left(rank(u,l_v)\le k\right)}{k}$, where $rank(u,l)$ returns the rank of user u in list l, and $\mathbb{I}(x)=1$ if x is true, otherwise $\mathbb{I}(x)=0$. Then the average HR@k is defined as

$$\text{HR@}k=\frac{1}{|\mathcal{D}_t|}\sum_{v\in\mathcal{D}_t}\frac{\sum_{u\in\mathcal{U}_v}\mathbb{I}\left(rank(u,l_v)\le k\right)}{k}, \tag{11}$$

and the NDCG@k is defined as

$$\text{NDCG@}k=\frac{1}{|\mathcal{D}_t|}\sum_{v\in\mathcal{D}_t}\frac{1}{|\mathcal{U}_v|}\sum_{u\in\mathcal{U}_v}\frac{\mathbb{I}\left(rank(u,l_v)\le k\right)}{\log\left(1+rank(u)\right)}. \tag{12}$$

4.1.4 Parameter Setting. During the training of CCFCRec, we set the batch size to 1,024 and the initial learning rates to 5e-6 and 1e-4 on ML-20M and Amazon-VG, respectively. The embedding dimensionality d is set to 128 and 256 on ML-20M and Amazon-VG, respectively. To speed up the contrastive learning process, for each training item in ML-20M, we sample 10 positive samples and 40 negative samples to form 400 positive-negative sample pairs in total, while for each training item in Amazon-VG, we sample 5 positive samples and 40 negative samples to form 200 positive-negative sample pairs in total. The balance factor of contrastive loss λ is set to 0.5 and 0.6 on ML-20M and Amazon-VG, respectively. The parameter τ in Equation (4) is set to 0.1. For fairness, the hyper-parameters of the baseline methods are set to their optimal configuration tuned on validation sets.

4.2 Performances Comparison (RQ1)

The results of the performance comparison between CCFCRec and the baseline methods are shown in Table 2, from which we can make the following observations and analyses:

- On both datasets, CCFCRec consistently outperforms the baseline methods in terms of all the metrics with p-value less than 0.01, which verifies the superiority of CCFCRec in cold-start item recommendation due to its contrastive CF. The contrastive CF enables CCFCRec to rectify the blurry collaborative embeddings by transferring the co-occurrence collaborative signals into the content CF module when training, which enhances the collaborative embeddings of cold-start items when applying.
- The performances of all the methods degrade on Amazon-VG compared with those on ML-20M, which shows that the extreme sparsity of Amazon-VG impacts the learning of high-quality collaborative embeddings.
- Surprisingly, even on the sparser dataset Amazon-VG, CCFCRec achieves more than 25% improvement, which is much greater than on ML-20M. This result demonstrates CCFCRec's better adaptability to sparse training datasets. For a training item, the contrastive CF of CCFCRec takes advantage of both the content views and behavior views of items, which is consequently conducive to alleviate the sparsity problem.
- Basically, the contrastive learning based methods (CLCRec and our CCFCRec) perform better than the other methods on both datasets, since the contrastive learning introduces more information to help learn stronger collaborative embeddings. However, CCFCRec is obviously superior to CLCRec, especially on Amazon-VG. This is because compared with CLCRec which limits the contrastive learning to the views of a training item v itself, CCFCRec extends the contrastive scope to the second-order neighbors (i.e., the items with which some user interacted together with v) of a training item.

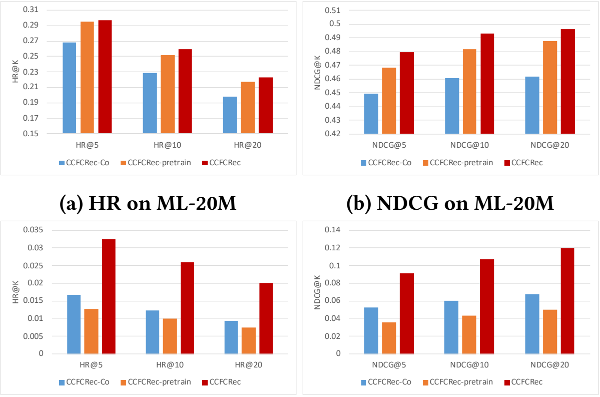
4.3 Ablation Study (RQ2)

Now we investigate the effectiveness of the contrastive CF and the joint training of the two CFs. For this purpose, we compare CCFCRec with its two variants as follows:

- **CCFCRec-Co** is the variant without contrastive CF, where $\mathcal{L}_z$ and $\mathcal{L}_c$ are removed from the overall loss (Equation (9)).
- **CCFCRec-pretrain** is the variant where UCE $\mathbf{s}_u$ and COCE $\mathbf{z}_v$ are pretrained with matrix factorization, and $\mathcal{L}_z$ is removed from the overall loss (Equation (9)). During the training, $\mathbf{z}_v$ is fixed.

Dataset	Baseline	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20
ML-20M	NFM	0.2119	0.1822	0.1531	0.3683	0.3842	0.3921
	LARA	0.2425	0.2165	0.1829	0.4595	0.4541	0.4580
	MTPR	0.2701	0.2393	0.2064	0.4504	0.4588	0.4721
	CVAR	0.2363	0.2188	0.1877	0.4301	0.4359	0.4448
	MvDGAE	0.2789	0.2453	0.2128	0.4586	0.4660	0.4720
	CLCRec	0.2677	0.2371	0.1971	0.4695	0.4820	0.4873
	CCFCRec	**0.2969***	**0.2592***	**0.2230***	**0.4798***	**0.4933***	**0.4962***
	Improv.	6.06%	5.36%	4.57%	2.15%	2.29%	1.79%
Amazon-VG	NFM	0.0162	0.0116	0.0089	0.0462	0.0532	0.0612
	LARA	0.0140	0.0074	0.0044	0.0370	0.0381	0.0400
	MTPR	0.0161	0.0112	0.0083	0.0457	0.0518	0.0587
	CVAR	0.0149	0.0109	0.0078	0.0405	0.0481	0.0547
	MvDGAE	0.0161	0.0126	0.0089	0.0456	0.0539	0.0604
	CLCRec	0.0229	0.0189	0.0150	0.0646	0.0769	0.0879
	CCFCRec	**0.0326***	**0.0260***	**0.0201***	**0.0916***	**0.1074***	**0.1202***
	Improv.	29.75%	27.30%	25.37%	29.48%	28.40%	26.87%

Table 2: Performance comparison. The best runs per metric are marked in boldface. The best runs per metric among baseline methods are underlined. * represents the statistical significance with p-value <0.01. Improv. is the percentage by which CCFCRec improves the performance of the best baseline methods.



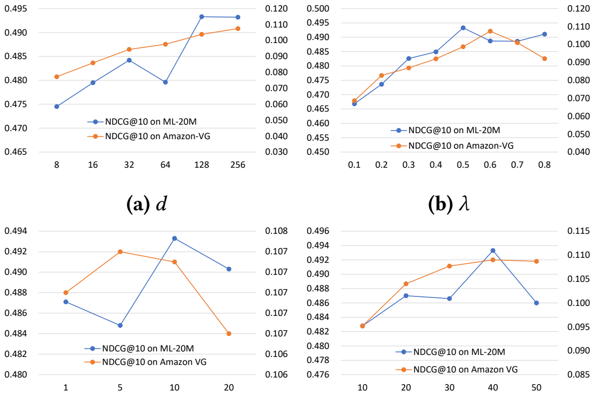
(a) HR on ML-20M

(b) NDCG on ML-20M

(c) HR on Amazon-VG

(d) NDCG on Amazon-VG

Figure 3: Ablation experiments.



(a) d

(b) λ

(c) The number of positive samples

(d) The number of negative samples

Figure 4: Tuning of hyper-parameters

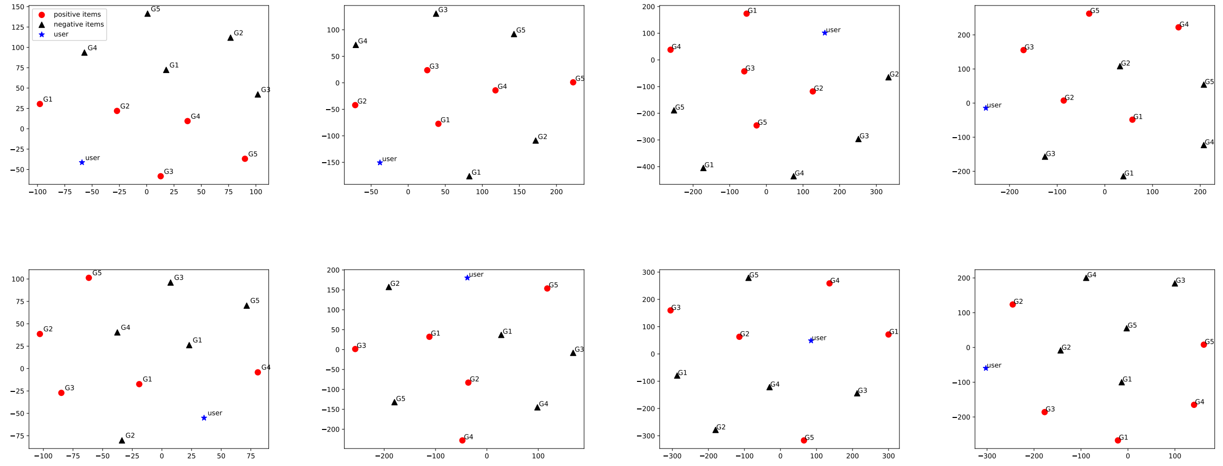
The results are shown in Figure 3. We can see that CCFCRec consistently outperforms the two variants on both datasets. In particular, CCFCRec-Co is significantly inferior to CCFCRec, especially on Amazon-VG, from which we can draw two conclusions: (1) the collaborative embeddings can be enhanced by contrastive CF indeed; (2) the contrastive CF can achieve greater improvement on sparser dataset, due to its extended contrastive scope from which user preference to item contents can be captured more accurately.

Similarly, CCFCRec-pretrain is also inferior to CCFCRec, which shows the advantage of the joint training of the two CF modules. In fact, the joint training brings two advantages: (1) the two CF modules are trained with the same supervision, which leads to the positive transfer of the co-occurrence collaborative signals; (2) the collaborative embeddings will be adjusted according to the ranking supervision signals, which makes them better adaptable to the ranking task. In contrast, the pretrained embeddings will degrade the performance due to the problem of error propagation.

4.4 Tuning of Hyper-parameter (RQ3)

In this section, on the validation sets of both datasets, we investigate the influence of hyper-parameters, the embedding dimensionality d, the balance factor of the contrastive loss λ, the numbers of positive samples and negative samples for a training item, of which the results are shown in Figure 4.

The curves of the embedding dimensionality d are shown in Figure 4(a), from which we observe that NDCG@10 reaches the peak at $d = 128$ on ML-20M and $d = 256$ on Amazon-VG, respectively.



(a) User 10430 in ML-20M (b) User 350 in ML-20M (c) User A1 in Amazon-VG (d) User A16 in Amazon-VG

Figure 5: Visualization of collaborative embeddings. The first row is the results of CCFCRec, while the second row is the results of CCFCRec-Co. One column corresponds to the results of a user.

User 10430	d^+		d^-		$d^- - d^+$		User A1	d^+		d^-		$d^- - d^+$	
	w-Co	w/o-Co	w-Co	w/o-Co	w-Co	w/o-Co		w-Co	w/o-Co	w-Co	w/o-Co	w-Co	w/o-Co
G1	81.53	66.43	137.41	82.28	55.88	15.85	G1	225.42	215.99	605.28	393.04	379.86	177.05
G2	70.97	167.23	204.98	73.66	134.01	-93.57	G2	221.80	199.61	240.48	421.18	18.68	221.57
G3	74.05	123.58	181.43	153.77	107.38	30.19	G3	262.91	405.56	408.33	231.3	145.42	-174.26
G4	109.24	68.18	134.97	120.28	25.73	52.10	G4	425.83	216.19	544.03	205.57	118.20	-10.62
G5	149.3	184.17	192.45	130.46	43.15	-53.71	G5	393.63	366.17	503.68	288.54	110.05	-77.63
User 350	d^+		d^-		$d^- - d^+$		User A16	d^+		d^-		$d^- - d^+$	
	w-Co	w/o-Co	w-Co	w/o-Co	w-Co	w/o-Co		w-Co	w/o-Co	w-Co	w/o-Co	w-Co	w/o-Co
G1	107.61	165.54	123.27	157.78	15.66	-7.76	G1	309.23	349.56	350.92	291.95	41.69	-57.61
G2	113.55	263.17	214.17	154.74	100.62	-108.43	G2	164.77	191.98	306.95	167.13	142.18	-24.85
G3	185.73	282.29	291.10	279.16	105.37	-3.13	G3	187.46	177.79	188.89	470.58	1.43	292.79
G4	207.10	408.44	224.53	352.76	17.43	-55.68	G4	468.36	455.57	469.81	336.45	1.45	-119.12
G5	301.51	157.82	302.54	342.99	1.03	185.17	G5	351.52	468.95	462.30	321.04	110.78	-147.91

Table 3: The distances from item embedding to the user embedding. w-Co represents CCFCRec, while w/o-Co CCFCRec-Co. G1-G5 represent five pairs of positive-negative sample. d^+ represents the positive samples' distances to the user, while d^- the negative samples' distances.

From Figure 4(b) we can see that on both datasets, the performance of CCFCRec increases first with the increasing of λ, which shows that the model performance will improve as more co-occurrence collaborative signals are captured by the contrastive CF of CCFCRec. However, when λ is over 0.5 on ML-20M and 0.6 on Amazon-VG, the model performance begins to drop, because excessive co-occurrence collaborative signals (due to the too small weight of $\mathcal{L}_q$ and $\mathcal{L}_z$) and insufficient content-based collaborative signals (due to the too big weight of $\mathcal{L}_c$) are captured.

Similarly, as shown in Figures 4(c) and 4(d), the model performance curves first rise and then fall with the increasing of the numbers of the positive and negative samples. Obviously, more positive and negative samples for a training item benefits the contrastive learning for the transfer of the co-occurrence collaborative signals, which explains the rising part. However, excessive contrastive samples may incur the risk of overfitting.

4.5 Case Study (RQ4)

Now we visualize the collaborative embeddings learned by CCFCRec and CCFCRec-Co by case studies. We randomly select two users from ML-20M (User 10430 and User 350) and Amazon-VG (User A1 and User A16), respectively. For each user, we pair each of the top-5 positive samples to a randomly chosen negative sample with the same genre or category attribute, and plot the UCEs and the CBCEs with t-SNE algorithm [31] in Figure 5. We also show the distances of item embeddings to corresponding user embeddings in Table 3.

We can observe that for each pair of samples, the positive sample embedding learned by CCFCRec is closer to the user embedding than the negative one as shown in the first row of Figure 5 and the w-Co of $d^- - d^+$ columns in Table 3, while the embeddings learned by CCFCRec-Co are blurred and cannot keep the proper ranking order, as shown in the second row of Figure 5 and the w/o-Co of $d^- - d^+$ columns in Table 3. For example, the bottom figure in Figure 5(a) together with the cell (User 101430, G2, $d^- - d^+$, w/o-Co) in Table 3 shows that in the G2 pair of User 101430, the negative sample embedding is closer to the user embedding than the positive one by 93.57. In contrast, the positive sample embedding learned by CCFCRec is closer than the negative one by 134.01 (as shown in the cell (User 101430, G5, $d^- - d^+$, w-Co) in Table 3). This result demonstrates the superiority of the contrastive CF of CCFCRec in ranking the positive samples ahead of negative ones, due to its ability to leverage the co-occurrence collaborative signals to rectify the blurry collaborative embeddings.

5 RELATED WORKS

5.1 Cold-start Recommendation

The existing works for cold-start recommendation often utilize side information to overcome the problem of sparse data. Content-based works usually capture collaborative signals from item content so that the collaborative filtering can be applicable for cold-start items. For example, NSPR [7] employs DNN to extract content features and use collaborative filtering to predict the user ratings to cold-start items. Different from directly modeling item content, recent content-based methods learn the correlation between side information and item behavior features, and train a generative model to project a cold item's content onto a warm item embedding space [2, 3, 8, 16, 19, 21, 24, 27, 32, 46]. For example, BPR-KNN [8] employs KNN and linear function to constraint item content feature and item behavior feature. LCE [24] combines content information and collaboration information into a common low-dimensional space. LARA [27] exploits an adversarial neural network with multiple generators to learn a mapping from cold-start items' attributes to user embeddings to generate virtual users for cold-start items.

Some efforts tackle the cold-start problem from the perspective of robust learning, which treat cold-start items as warm-start items missing interaction records, and try to infer the warm embeddings of the cold-start items in a robust manner like randomly corrupting the collaborative embedding of warm-start training items. For example, DropoutNet [32] selects a subset of users/items during training, and removes their behavior features to simulate the cold-start scenario. MTPR [6] constructs a counterfactual representation by setting the behavior part of an item's latent representation to zero, and learns pairwise matching between normal representations and counterfactual representations with BPR loss.

Some works use meta-learning to address cold-start recommendation. For example, MeLU [15] uses two sets, namely, the support set and query set. The support set and query set are used for calculating training loss and test loss for each recommendation task. During the local update phase, the model minimizes the training loss based on the support set, while during the global update phase, the model minimizes the test loss based on the query set. MetaHIN [17] extends meta-learning into heterogeneous information networks to address cold-start recommendations. MWUF [46] generates warm embeddings for cold-start items based on their features and ID embeddings with two meta networks.

5.2 Contrastive Learning in Recommendation

Contrastive Learning (CL) is a self-supervised technique, which has achieved an impressive success in solving the problem of data sparsity [4, 12, 37]. Recently, some studies explore contrastive learning for recommender systems [5, 18, 22, 34–36, 38–41, 45]. For example, Zhou *et al.* [45] apply a random masking of attributes and items to build different views and learn the correlations among attributes, items, subsequences, and sequences by maximizing the mutual information between the different views. SGL [34] models user-item interactions as a graph and builds multiple views by conducting node dropout, edge dropout, and random walk on the user-item graph. Similarly, Similarly, Yu *et al.* [40] propose a simple CL method to perturb the graph embedding space with uniform noise to build contrastive views for the training of recommendation model. At the same time, some researchers model user-item interactions with hypergraph, and conduct CL over augmented hypergraph views for the user/item representation learning [35, 39, 42]. Wei *et al.* [33] propose a cold-start recommendation model CLCRec, which fulfills the learning of feature representations for cold-start items by maximizing the mutual information between item content and collaborative signals.

Different from the existing works which often limit the contrastive learning to the views of the same item, our CCFCRec extends the contrastive scope to the views of the second-order neighbors of a training item, which helps capture more collaborative signals and alleviate the sparsity problem as well.

6 CONCLUSION

In this paper, we propose a novel model called Contrastive Collaborative Filtering for Cold-start item recommendation (CCFCRec). CCFCRec can take advantage of the co-occurrence collaborative signals in warm training data to alleviate the issue of blurry collaborative embeddings. In particular, we devise a contrastive collaborative filtering (CF) framework, consisting of a content CF module and a co-occurrence CF module to generate the content-based collaborative embedding and the co-occurrence collaborative embedding for a training item, respectively. During the joint training of the two CF modules, the contrastive learning between the two collaborative embeddings enables the content CF module to memorize the co-occurrence collaborative signals, so that during the applying phase, the blurry collaborative embeddings can be rectified. The extensive experiments conducted on real datasets together with the sound theoretical analysis demonstrate the superiority of CCFCRec.

ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China under grant 61972270.

REFERENCES

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2017. Deep variational information bottleneck. In *ICLR*.

[2] Oren Barkan, Noam Koenigstein, Eylon Yogev, and Ori Katz. 2019. CB2CF: A Neural Multiview Content-to-Collaborative Filtering Model for Completely Cold Item Recommendations. In *RecSys*.

[3] Hao Chen, Zefan Wang, Feiran Huang, Xiao Huang, Yue Xu, Yishi Lin, Peng He, and Zhoujun Li. 2022. Generative Adversarial Framework for Cold-Start Item Recommendation. In *SIGIR*.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*.

[5] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. In *WWW*.

[6] Xiaoyu Du, Xiang Wang, Xiangnan He, Zechao Li, Jinhui Tang, and Tat-Seng Chua. 2020. How to learn item representation for cold-start multimedia recommendation?. In *MM*.

[7] Travis Ebesu and Yi Fang. 2017. Neural semantic personalized ranking for item cold-start recommendation. *Information Retrieval Journal* 20, 2 (2017), 109–131.

[8] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *ICDM*.

[9] J Goodfellow Ian, Pouget-Abadie Jean, Mirza Mehdi, Xu Bing, Warde-Farley David, Ozair Sherjil, and C Courville Aaron. 2014. Generative adversarial nets. In *NeurIPS*.

[10] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*.

[12] Olivier Henaff. 2020. Data-efficient image recognition with contrastive predictive coding. In *ICML*.

[13] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *NeurIPS*.

[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. Melu: Meta-learned user preference estimator for cold-start recommendation. In *KDD*.

[16] Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. 2019. From Zero-Shot Learning to Cold-Start Recommendation. In *AAAI*.

[17] Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on heterogeneous information networks for cold-start recommendation. In *KDD*.

[18] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *KDD*.

[19] Kaixiang Mo, Bo Liu, Lei Xiao, Yong Li, and Jie Jiang. 2015. Image feature learning for cold start problem in display advertising. In *IJCAI*.

[20] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[21] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm Up Cold-Start Advertisements: Improving CTR Predictions via Learning to Learn ID Embeddings. In *SIGIR*.

[22] Ruihong Qiu, Zi Huang, and Hongzhi Yin. 2021. Memory Augmented Multi-Instance Contrastive Predictive Coding for Sequential Recommendation. In *ICDM*.

[23] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2009).

[24] Martin Saveski and Amin Mantrach. 2014. Item cold-start recommendations: learning local collective embeddings. In *RecSys*.

[25] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *SIGIR*.

[26] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[27] Changfeng Sun, Han Liu, Meng Liu, Zhaochun Ren, Tian Gan, and Liqiang Nie. 2020. LARA: Attribute-to-feature adversarial learning for new-item recommendation. In *WSDM*.

[28] MTCAJ Thomas and A Thomas Joy. 2006. *Elements of information theory*. Wiley-Interscience.

[29] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057* (2000).

[30] Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*.

[31] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* (2008).

[32] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. Dropoutnet: Addressing cold start in recommender systems. In *NeurIPS*.

[33] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive learning for cold-start recommendation. In *MM*.

[34] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*.

[35] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-supervised hypergraph convolutional networks for session-based recommendation. In *AAAI*.

[36] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *ICDE*.

[37] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. In *ACL*.

[38] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-aware self-supervised tri-training for recommendation. In *KDD*.

[39] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *WWW*.

[40] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are Graph Augmentations Necessary? Simple Graph Contrastive Learning for Recommendation. In *SIGIR*.

[41] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. 2022. Self-Supervised Learning for Recommender Systems: A Survey. *arXiv preprint arXiv:2203.15876* (2022).

[42] Junwei Zhang, Min Gao, Junliang Yu, Lei Guo, Jundong Li, and Hongzhi Yin. 2021. Double-scale self-supervised hypergraph learning for group recommendation. In *CIKM*.

[43] Xu Zhao, Yi Ren, Ying Du, Shenzheng Zhang, and Nian Wang. 2022. Improving Item Cold-start Recommendation via Model-agnostic Conditional Variational Autoencoder. In *SIGIR*.

[44] Jiawei Zheng, Qianli Ma, Hao Gu, and Zhenjing Zheng. 2021. Multi-view Denoising Graph Auto-Encoders on Heterogeneous Information Networks for Cold-start Recommendation. In *KDD*.

[45] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *CIKM*.

[46] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to Warm Up Cold Item Embeddings for Cold-Start Recommendation with Meta Scaling and Shifting Networks. In *SIGIR*.

A APPENDIX

Table 4: Hyper-parameter setting of baseline methods.

Baseline	**Embedding dimensionality selected from {8, 16, 32, 64, 128, 256}**		**Other hyper-parameters**
	ML-20M	Amazon VG	
NFM	64	64	Number of hidden layers = 2 selected from {1, 2, 3}
LARA	128	64	Number of hidden layers in the Generator and Discriminator = 3 selected from {1, 2, 3, 4}
MTPR	64	128	/
CVAR	64	64	
MvDGAE	128	64	Dropout Rate=0 selected from {0, 0.2, 0.4, 0.6, 0.8, 1.0}
CLCRec	64	256	Number of negative samples k=128 selected from {16, 32, 64, 128, 256}
			The weight of $\mathcal{L}_{RE}$ λ=0.5 selected from {0, 0.1, 0.2, ..., 1.0}