This article has been accepted for publication in IEEE Transactions on Services Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSC.2024.3378925

Enhancing Disentanglement of Popularity Bias for Recommendation with Triplet Contrastive Learning

Jie Liao, Wei Zhou, *Member, IEEE,* Fengji Luo, *Member, IEEE,* and Junhao Wen, *Member, IEEE*

Abstract—Popularity bias is a common phenomenon in the user-item interaction, which means a user interacts with the items just because of the items' popularity, but the user does not actually interest in these items. Neglecting popularity bias in recommendation systems can result in favoring popular items over personal preferences. This paper proposes a new recommendation framework for enhancing the **D**is**E**ntanglement of popularity bias based on **C**ontrastive **L**earning (**DECL**). In DECL, the interest and conformity representation sets of the users and the items are generated through a disentangled representation learning process. A contrastive learning process is then performed to optimize the distributions of the disentangled sets in the representation space. A customized loss function is designed to facilitate the parameter optimization, and the final recommendation is made based on the interest and conformity. Extensive experiments and comparison studies are conducted on three real-world datasets to validate the effectiveness of the proposed DECL framework. The experiment results show that compared with the state-of-the-art methods, DECL can achieve up to 10.69% performance improvement on the Ciao dataset. This indicates the proposed system can effectively disentangle popularity bias in recommendation and has large application potential.

Index Terms—Popularity Bias, Contrastive Learning, Disentangled Representation Learning, Recommender System

1 INTRODUCTION

THE recommendation technology has penetrated many aspects of people's life. A personalized recommender system (PRS) [1] is an expert system that recommends suitable items to a target user through mining the knowledge about the user's preference and interest from observational interaction data between users and items. A variety of PRSs have been developed for addressing recommendation problems in different domains, such as electronic business [2], [3], Web services [4], [5], and energy systems [6], [7].

The popularity bias [8] has become an important issue in recommendation problems. Popularity bias typically means a user interacts with the items just because of the items' popularity, but he/she does not actually interest in these items. For example, a user may choose to buy an item just because of its high sales. The cause of such an interaction is not the user's interest but the conformity that exists in a user population. Neglecting popularity bias in recommendation systems can result in favoring popular items over personal preferences. This will lead to the Matthew effect, reinforcing the popularity of items favored by a large population and decreasing the visibility of other items. Research work has been conducted to incorporate popularity bias in PRS designs [9], [10], [11]. As the most representative work, Ref. [9] develops a popularity bias-aware PRS called DICE, in which the causes of the user-item interaction are disentangled into 2 factors: (1) the users' personal interest or the items' characteristics; and (2) the users' conformity or the items' popularity. For convenience purposes, in this paper, we use the word "interest" to represent both the users' interest and the items' characteristics, and we use the word "conformity" to represent both the users' conformity and the items' popularity. Each of the two causes is represented by a representation set (called the interest representations and the conformity representations), and the two representation sets are determined through a learning process subjected to maximizing a certain distance metric (e.g., the Manhattan distance or Euclidean distance) or minimizing the distance correlation between the two sets.

Despite DICE [9] has shown effectiveness in addressing the popularity bias issue, there are still 2 significant limitations that can be identified in it and are worthy of further investigation:

(1) Firstly, it does not sufficiently preserve the uniformity of the two representation sets in the underlying representation space. In DICE, the two sets are either independent or far from each other in the space, making the representation vectors in the sets far from uniformly distributed in the space. This in turn leads to a significant loss of the information of the users and items. This is because the more uniform the representation vectors distributed in the representation space, the more meaningful information of the users and items can be retained.

(2) Secondly, in DICE, the distance of the interest and conformity representations are always large in the representation space even if the two representations are similar. Such a mechanism cannot truly reflect the semantic structure of the two representations: for a user, when the two representations are very similar, it would indicate the user's personal interest on the item and the item's popularity in a

- J. Liao, W. Zhou, and J. Wen is with School of Big Data and Software Engineering, Chongqing University, Chongqing, China. E-mail: {liaojie, zhouwei, jhwen}@cqu.edu.cn
- F. Luo is with School of Civil Engineering, the University of Sydney, Australia. Email: fengji.luo@sydney.edu.au

Manuscript received April 19, 2005; revised August 26, 2015.

user population are highly consistent; for an item, when the two representations are very similar, it would indicate the popularity of the item is because of its high intrinsic quality. In these situations, maximizing the distance or minimizing the distance correlation between the two representation sets would make the two sets quite different.

Aiming at addressing the above limitations, in this study, we propose a new DisEntangled recommendation framework based on Contrastive Learning (DECL) that can be intelligently aware of popularity in the generation of recommendations. Compared with the state-of-the-art methods [9], [10], [11], the proposed DECL framework integrates with a Contrastive Learning (CL) mechanism, making it capable of better disentangling the users' and items' interest and conformity representations. As a result of our novel design, DECL can effectively address the popularity bias issue, enabling more accurate recommendations that can capture users' real interests. More specifically, the contributions of this paper are 3-fold:

(1) This is the first research that applies contrastive learning to disentangle the users' and items' interest and conformity representations in recommendation problems. In this study, we propose a new CL mechanism that is with two unique features: (i) compared with the existing CL techniques [12], [13], [14] that perform two times data augmentation in the construction of positive samples, the proposed mechanism only requires one time of data augmentation, making it more computationally efficient; and (ii) distinguished from the existing CL techniques that select the negative samples in a random manner, the proposed mechanism utilizes supervised information in the selection of negative samples, and this leads to a higher selection accuracy.

(2) Based on (1), this paper proposes a new recommendation framework (i.e., DECL). In this framework, a disentangled representation learning process is performed to generate two cause representation sets that represent the interest and conformity, respectively. Followed by this, a CL process is performed to optimize the sets' distribution in the representation space. For a user or an item, a triplet is constructed, which consists of an anchor, a positive sample, and a negative sample. A customized loss function is designed based on the triplet to facilitate the parameter optimization, and the final recommendation is made based on the disentangled interest and conformity.

(3) Comprehensive experiments are conducted on three real-world datasets to validate the superiority of the proposed DECL framework. To validate users' genuine interests, we perform an unbiased evaluation [9], [15] on an unbiased test dataset. The dataset comprises interaction records collected from randomly recommending items to users, which hence has eliminated the influence of popularity bias. Additional sensitivity analysis is performed for the critical hyperparameters in DECL, which can provide a guidance to the implementation of DECL.

The rest of the paper is organized as follows. Section 2 briefly reviews the work related to this study; Section 3 presents the design details of DECL; Section 4 reports the experiments conducted to validate the proposed framework. Section 5 concludes the paper and discusses the future research directions.

2 Related Work

This section provides an overview of the representative literature related to this work. The related work is reviewed according to the 3 categories listed below.

2.1 Recommender System

Recommender systems aim at recommending items to target users in a personalized manner through understanding the users' preferences on items. This is usually achieved by analyzing the user-item interaction data. Many of the early recommender system designs are mainly based on the matrix factorization (MF) technique [16], [17]. MF methods map the users' and items' identifications (IDs) into embedding vectors and calculate the inner-product of the embedding vectors to obtain the matching scores of the users on the items; the items with the highest matching scores are then recommended to the target user.

Other techniques are developed to improve the performance of MF methods in recommender systems. The Bayesian Personalized Ranking (BPR) method [17] assumes that a target user should have higher matching scores on the items he/she has interacted with than those on the items he/she has not interacted with. Based on this, BPR refines the loss function in the recommendation systems with conventional MFs to achieve better recommendation performance. Some recent research efforts [18], [19] represent the user-item interaction data as a user-item graph, in which a node in the graph represents a user or an item, and an edge connecting two nodes represents an interaction between a user and an item. Based on this, graph neural networks, which are designed for modeling graph-structured data, are then applied to facilitate the embedding learning process.

Popularity bias has been recognized as an important issue in recommendation problems. It would make a recommender system to recommend popular items to a target user without sufficiently considering their actual interest and preference. A number of techniques have been developed to alleviate this issue in recommender systems, which are reviewed below.

2.2 Popularity Bias in Recommendation

Generally, the methods that address the popularity bias issue in recommendation problems can be categorized into two types: Inverse Propensity Scoring (IPS)-based methods [10], [11], [20] and causal embedding-based [9], [15] methods. IPS-based methods set up weights (also known as propensity scores) for different training samples based on the reverse order of the samples' popularity values. The samples containing popular items would be assigned low weights, and those containing unpopular items get high weights. In this way, the impact of popular items on the training process is reduced, representing a correction of the popularity bias. The major limitations of IPS-based methods are two-fold: firstly, they usually fail to estimate the propensity scores accurately. Secondly, the estimated propensity scores are usually with a high variance. Several methods are developed to reduce the variance of the estimated propensity scores in IPS-based methods. IPS-C [20] sets up an upper limit for the propensity score. The work in [10] applies both the upper limit and normalization operation to the propensity score; based on this, the work further develops a recommendation method, which combines a power-law transformation and a normalization operation to it to smooth out the propensity scores.

Representative work in causal embedding-based methods includes CausE [15] and DICE [9]. In CausE, the training process is performed on 2 datasets: a large-biased dataset and a small unbiased dataset. For each user and each item, two embeddings are trained based on the two datasets, respectively. L1 and L2 regularizations are then used to make the two embeddings as similar as possible. In this way, the unbiased dataset is utilized to reduce the popularity bias in the biased dataset. In DICE, two cause-specific datasets are constructed. For each user and each item, two embeddings are trained based on the datasets: one is called "interest embedding" and the other is called "conformity embedding". The distance correlation between the two embedding sets (one set contains the interest embeddings and one contains the conformity embeddings) is then minimized.

2.3 Contrastive Learning and Its Application in Recommendation

Contrastive learning is an emerging machine learning technique. The basic idea of contrastive learning is to learn a representation space in which similar samples are close to each other, and unsimilar samples are far apart from each other. Contrastive learning can help to improve a recommender system's performance by improving the quality of the users' and items' representation distribution. In recommendation problems, a target user's matching score on an item is usually calculated by applying a dot product operation on the representation vectors of the user and the item, and this can be regarded as a measurement of the similarity between the target user's and the item's representations. By applying contrastive learning, positive and negative sample pairs are constructed (a positive pair contains an anchor and a sample that is similar in semantics of the anchor, and a negative sample pair contains an anchor and a sample that is not similar in semantics of the anchor). Then, contrastive learning makes the positive sample pairs' representations close and the negative sample pairs' representations far apart.

The construction process of positive and negative sample pairs can be supervised or unsupervised. In the supervised construction, the loss functions used in contrastive learning are based on distance metric learning, and the most widely used loss functions are the contrastive loss function [21] and the triplet loss function [22]. The contrastive loss function is based on multiple positive sample pairs and negative sample pairs, and the triplet loss function calculates the loss based on triplets, where each triplet consists of an anchor, a positive sample, and a negative sample. The positive sample is with the same class of the anchor, and the negative sample is with a different class. In unsupervised construction, the loss functions are usually Noise Contrastive Estimation (NCE) loss functions [23], [24] and N-pair loss functions [25]. These loss functions use one positive sample and multiple negative samples of each anchor to calculate the loss. The positive sample is usually constructed using data augmentation [13] or co-occurrence [26] methods, and the negative samples are randomly chosen (e.g., they can be the other samples in the same training batch of the anchor).

Recent work [14], [27] has applied contrastive learning to the design and development of recommender systems. Ref. [14] develops a recommendation framework called SGL. By applying contrastive learning principles, the framework firstly uses data augmentation techniques on the user-item interaction graph twice to generate two different graphs, in which each node represents a user or an item. An encoder is then applied to both graphs to generate the two representations of a node, which form a positive sample pair. Meanwhile, a negative sample pair consists of the anchor and a representation of a node that is in one training batch. A system called DuoRec is developed in [27], which performs a supervised sampling process to construct positive sample pairs and performs an unsupervised data argumentation process to construct negative sample pairs. Based on the positive and negative sample pairs, NCE-based loss functions are used in both systems [14], [27] to optimize the representation distribution.

3 Design of the DECL framework

A recommendation problem with incorporation of popularity bias can be generally expressed as follows: Let $\mathcal{U}, \mathcal{I}$ be the sets of N users and M items, respectively, i.e., $|\mathcal{U}| = N$, $|\mathcal{I}| = M$. Let $\mathcal{O} = \{(u, i) | u \in \mathcal{U}, i \in \mathcal{I}\}$ represents the observed user-item interaction records, in which i is a positive item (i.e., an item with which the user u has interacted). Based on $\mathcal{O}$, a user-item interaction graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed, which consists of a set of nodes $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ and a set of edges $\mathcal{E} = \mathcal{O}$. The weights of all the edges are fixed at 1. Let $\mathcal{O}_{train}$ denotes the training dataset and $\mathcal{O}_{test}$ denotes the testing dataset, respectively. Let p_i denotes the popularity of an item i, which is measured by the number of interactions between the users and the item i.

To consider popularity bias, we construct the training and testing datasets not to be independent and identically distributed (IID) in terms of the popularity distribution of the items. The recommendation target then is to predict the matching scores of a target user on the M items. Based on the above principle, we present the design details of the DECL framework as follows.

3.1 Overview of the Framework

The architecture of DECL is shown in Fig. 1. The framework performs popularity bias-aware recommendations based on the interaction of 4 critical modules:

(1) *Module 1* - Generate the training sample sets. In this process, each record in $\mathcal{O}_{train}$ is expanded to include a negative sample.

(2) *Module 2* - Disentangle the causes of the user-item interaction. Based on the training datasets generated in *Module 1*, DECL performs a disentangled representation learning process to generate 2 cause representation sets, which represent the interest and conformity, respectively.

(3) *Module 3* - Distribution optimization of the cause representation sets. This is achieved by performing a contrastive learning process on the 2 sets generated from *Module 2*.

This article has been accepted for publication in IEEE Transactions on Services Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSC.2024.3378925

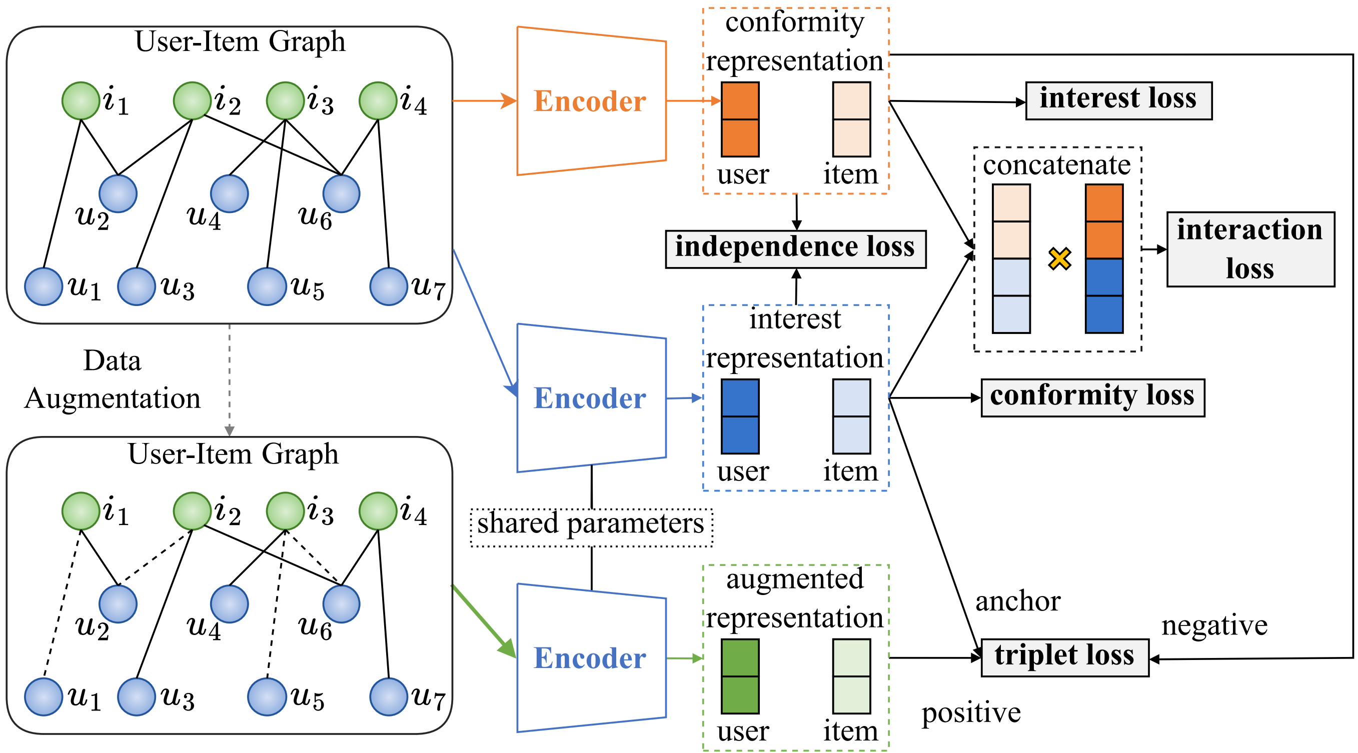


Fig. 1. Architecture design of DECL.

TABLE 1
Nomenclature

Notations	Descriptions
$\mathcal{U}$	User set
$\mathcal{I}$	Item set
N	The number of users
M	The number of items
$\mathcal{O}$	The user-item interaction records, i.e., $\mathcal{O} = \{(u,i) \mid u \in \mathcal{U}, i \in \mathcal{I}\}$
$\mathcal{V}$	The vertex set containing users and items, i.e., $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$
$\mathcal{E}$	The edge set containing user-item interaction records, i.e., $\mathcal{E} = \mathcal{O}$
$\mathcal{G}$	The user-item interaction graph, i.e., $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$
$\mathbf{A}$	The adjacency matrix of $\mathcal{G}$
$\mathbf{D}$	The degree matrix of $\mathcal{G}$
d	The embedding dimension
$\mathbf{E}^I \in \mathbb{R}^{(N+M)\times d}$	The interest representation matrix
$\mathbf{E}^C \in \mathbb{R}^{(N+M)\times d}$	The conformity representation matrix
$\mathbf{E} \in \mathbb{R}^{(N+M)\times d}$	The interaction representation matrix: $\mathbf{E} = concatenate(\mathbf{E}^I, \mathbf{E}^C)$
$\mathbf{E}^A \in \mathbb{R}^{(N+M)\times d}$	The augmented representation matrix
$y_{ui}^{interaction}$	The interaction matching score of user u on item i
$y_{ui}^{interest}$	The interest matching score of user u on item i
$y_{ui}^{conformity}$	The conformity matching score of user u on item i
p_i	The popularity of item i

(4) *Module 4* - Train the recommendation model. DECL performs a multi-task and curriculum learning process to train the recommendation model.

3.2 Module 1: Generation of Training Sample Set

The aim of this module is to construct two mutually exclusive training sample sets O_1 and O_2, which will be used to train the disentangled representation sets in *Module 2*. For each sample (u, i) in the training dataset $\mathcal{O}_{train}$, we sample a negative item j (i.e., an item with which the user u has not interacted) to construct a new 3-turple training data sample (u, i, j). We generate O_1 such that it contains the samples so that for each sample (u, i, j), it satisfies $p_i > p_j$. Similarly, we generate O_2 so that for each sample (u, i, j) in O_2, there is $p_i < p_j$.

Based on the realization that the interaction between a user and an item could be caused by both interest and conformity, the matching score of a user u on an item i (denoted as $y_{ui}^{interaction}$), which implicitly represents the possibility the user will interact with the item, can be expressed as:

$$y_{ui}^{interaction} = y_{ui}^{interest} + y_{ui}^{conformity}, \tag{1}$$

where $y_{ui}^{interest}$ and $y_{ui}^{conformity}$ depict the matching scores of the user u on the item i under the cause of interest and under the cause of conformity, respectively.

For each sample (u, i, j) in O_1, since it needs to satisfy the condition of $p_i > p_j$, the following two inequities can be inferred:

$$y_{ui}^{conformity} > y_{uj}^{conformity}, \tag{2}$$

$$y_{ui}^{interest} + y_{ui}^{conformity} > y_{uj}^{interest} + y_{uj}^{conformity}. \tag{3}$$

This article has been accepted for publication in IEEE Transactions on Services Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSC.2024.3378925

Similarly, for each sample (u, i, j) in O_2, the following inequities can be inferred based on the condition of $p_i < p_j$:

$$y_{ui}^{conformity} < y_{uj}^{conformity}, \tag{4}$$
$$y_{ui}^{interest} + y_{ui}^{conformity} > y_{uj}^{interest} + y_{uj}^{conformity} \tag{5}$$
$$y_{ui}^{interest} > y_{uj}^{interest}. \tag{6}$$

For the above inequalities (2-6), the larger the confidence interval, the higher the confidence. In other words, the larger the popularity difference between the items i and j, the higher the probability of the inequalities to hold. Given an interaction record (u, i), a negative item j is sampled from the items that the user u has not interacted and are with the popularity values larger than $p_i + m_{up}$ or lower than $p_i - m_{down}$, where m_{up} and m_{down} are pre-specified parameters.

3.3 Module 2: Disentangle the Causes of the User-Iterm Interaction

Based on the training sample sets O_1 and O_2 generated in *Module 1*, disentangled representation learning is performed to generate 2 cause representation sets.

At first, 2 matrices are initialized, which represent the interest embedding matrix and the conformity embedding matrix, respectively. Then, a graph-based recommendation model LightGCN [18] is applied. It takes the inputs of: (i) either the initialized interest embedding matrix or the initialized conformity embedding matrix, and (ii) the user-item interaction graph ($\mathcal{G}$). Then, it propagates the inputted matrix in a graph convolution network. For either the interest embedding or the conformity embedding, the following propagation process is applied:

$$\mathbf{E}^{I,(k+1)} = (\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{E}^{I,(k)}, \tag{7}$$
$$\mathbf{E}^{C,(k+1)} = (\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{E}^{C,(k)}, \tag{8}$$

where $\mathbf{E}^{I,(k)}, \mathbf{E}^{C,(k)} \in \mathbb{R}^{(N+M)\times d}$ denote the interest and conformity embedding matrix outputted by the kth layer propagation and $\mathbf{E}^{I,(0)}, \mathbf{E}^{C,(0)}$ denote the initial interest and conformity embedding matrix. $\mathbf{A} \in \mathbb{R}^{(N+M)\times(N+M)}$ represents the adjacency matrix, and $\mathbf{D} \in \mathbb{R}^{(N+M)\times(N+M)}$ represent the degree matrix of the user-item interaction graph. After K layers of propagation, the final interest/conformity representation matrix (denoted as $\mathbf{E}^I$ or $\mathbf{E}^C$) of the users and the items is obtained by summing up the output of each propagation layer:

$$\mathbf{E}^{I} = \sum_{k=0}^{K} \frac{1}{K+1}\mathbf{E}^{I,(k)}, \tag{9}$$

$$\mathbf{E}^{C} = \sum_{k=0}^{K} \frac{1}{K+1}\mathbf{E}^{C,(k)}. \tag{10}$$

Denote $\mathbf{e}_u^I$ and $\mathbf{e}_i^I$ as two different rows in $\mathbf{E}^I$, which represent the interest representations of the user u and the item i, respectively. Based on the representations, the interest matching score between u and i is:

$$y_{ui}^{interest} = (\mathbf{e}_u^I)^T \cdot \mathbf{e}_i^I. \tag{11}$$

Similarly, denote the conformity representations of u and i as $\mathbf{e}_u^C$ and $\mathbf{e}_i^C$, respectively, which are from $\mathbf{E}^C$. The conformity matching score $y_{ui}^{conformity}$ between u and i is obtained as:

$$y_{ui}^{conformity} = (\mathbf{e}_u^C)^T \cdot \mathbf{e}_i^C. \tag{12}$$

The interaction matching score between u and i is calculated as:

$$y_{ui}^{interaction} = (\mathbf{e}_u)^T \cdot \mathbf{e}_i, \tag{13}$$
$$\mathbf{e}_u = \mathbf{e}_u^I || \mathbf{e}_u^C, \tag{14}$$
$$\mathbf{e}_i = \mathbf{e}_i^I || \mathbf{e}_i^C, \tag{15}$$

where $||$ denotes the concatenation operation.

In this study, the Bayesian Personalized Ranking (BPR) loss function [17] is used to update the interest and conformity matrices. The loss function enables the positive items to have higher matching scores than the negative items:

$$\mathcal{L}_{BPR}(y_{ui}, y_{uj}) = -\ln\sigma(y_{ui} - y_{uj}), \tag{16}$$

where $\sigma(\cdot)$ is an activation function; y_{ui} denotes the matching score between the user u and the positive item i; y_{uj} denotes the matching score between u and the negative item j.

Based on inequality (2), in the dataset $\mathcal{O}_1$, the matching score between a user and a positive item is higher than that between the user and a negative item. Similarly, according to inequality (4), the matching score between the user and a positive item in $\mathcal{O}_2$ is lower than that between the user and a negative item. Hence, the loss function models the conformity in $\mathcal{O}_1$ and in $\mathcal{O}_2$ (denoted as $\mathcal{L}_{conformity}^{\mathcal{O}_1}, \mathcal{L}_{conformity}^{\mathcal{O}_2}$, respectively) as Eqs. (17) and (18), respectively:

$$\mathcal{L}_{conformity}^{\mathcal{O}_1} = \sum_{(u,i,j)\in\mathcal{O}_1} \mathcal{L}_{BPR}(y_{ui}^{conformity}, y_{uj}^{conformity}), \tag{17}$$

$$\begin{aligned}\mathcal{L}_{conformity}^{\mathcal{O}_2} &= \sum_{(u,i,j)\in\mathcal{O}_2} \mathcal{L}_{BPR}(-y_{ui}^{conformity}, -y_{uj}^{conformity}) \\ &= -\sum_{(u,i,j)\in\mathcal{O}_2} \mathcal{L}_{BPR}(y_{ui}^{conformity}, y_{uj}^{conformity}),\end{aligned} \tag{18}$$

The loss function models the conformity in the whole training dataset is:

$$\mathcal{L}_{conformity}^{\mathcal{O}_1+\mathcal{O}_2} = \mathcal{L}_{conformity}^{\mathcal{O}_1} + \mathcal{L}_{conformity}^{\mathcal{O}_2}. \tag{19}$$

Based on inequality (6), the loss function models the interest in $\mathcal{O}_2$ (denoted as $\mathcal{L}_{interest}^{\mathcal{O}_2}$) as:

$$\mathcal{L}_{interest}^{\mathcal{O}_2} = \sum_{(u,i,j)\in\mathcal{O}_2} \mathcal{L}_{BPR}(y_{ui}^{interest}, y_{uj}^{interest}). \tag{20}$$

Based on inequalities (3) and (5), the loss function models the interaction as:

$$\mathcal{L}_{interaction}^{\mathcal{O}_1+\mathcal{O}_2} = \sum_{(u,i,j)\in\mathcal{O}_1+\mathcal{O}_2} \mathcal{L}_{BPR}(y_{ui}^{interaction}, y_{ui}^{interaction}). \tag{21}$$

The distance correlation (dCor) [28] metric is used to minimize the distance correlation between $\mathbf{E}^I$ and $\mathbf{E}^C$:

$$\mathcal{L}_{independence} = dCor(\mathbf{E}^I, \mathbf{E}^C). \tag{22}$$

3.4 Module 3: Contrastive Learning-Based Representation Distribution Optimization

In this study, we use the contrastive learning technique to optimize the representation distribution of the users and the items. We use the triplet loss function [22] for such an optimization. The triplet loss function aims at minimizing the distance between an anchor $\mathbf{e}$ and a positive sample $\mathbf{e}^+$ (i.e., a sample that is similar in semantics to $\mathbf{e}$) and maximizing the distance between $\mathbf{e}$ and a negative sample $\mathbf{e}^-$ (i.e., a sample that is not similar in semantics to $\mathbf{e}$). In the proposed DECL, the anchor $\mathbf{e}$ is set as the interest representation of a user or an item; the positive sample is generated by performing data augmentation to the anchor, and the negative sample is set to be the conformity representation.

To generate a positive sample, we randomly drop some edges in the user-item interaction graph $\mathcal{G}$:

$$\mathcal{G}' = (\mathcal{V}, \mathbf{M} \odot \mathcal{E}), \tag{23}$$

where $\mathcal{V}$ and $\mathcal{E}$ are the vertex set and the edge set of the user-item interaction graph, respectively; $\odot$ denotes the element-wise product operation; $\mathbf{M} \in \{0,1\}^{|\mathcal{E}|}$ is a masking vector on $\mathcal{E}$. For each element m in $\mathbf{M}$, it is generated following the probabilities $P(m=1)=\rho$ and $P(m=0)=1-\rho$. The augmented representation matrix $\mathbf{E}^A$ is obtained by the LightGCN method, which takes the interest embedding matrix and the augmented graph $\mathcal{G}'$ as input, i.e., $\mathbf{E}^A = f(\mathbf{E}^{I,(0)}, \mathcal{G}')$. For a user u or an item i, its corresponding positive sample (denoted as $\mathbf{e}_u^A$ or $\mathbf{e}_i^A$) is a row of $\mathbf{E}^A$.

Based on the triplet $(\mathbf{e}_u^I, \mathbf{e}_u^A, \mathbf{e}_u^C)$, the triplet loss on the users is:

$$\mathcal{L}_{triplet}^{user} = \sum_{u \in \mathcal{U}} max(0, s(\mathbf{e}_u^I, \mathbf{e}_u^A) - s(\mathbf{e}_u^I, \mathbf{e}_u^C) + \epsilon), \tag{24}$$

where ϵ is a margin that is enforced between the positive pair (i.e., $\mathbf{e}_u^I$ and $\mathbf{e}_u^A$) and the negative pair (i.e., $\mathbf{e}_i^I$ and $\mathbf{e}_i^C$). $s(\cdot)$ measures the similarly between two vectors:

$$s(\mathbf{e}', \mathbf{e}'') = \frac{exp(cos(\mathbf{e}', \mathbf{e}''))}{\tau}, \tag{25}$$

where τ is a hyperparameter. Similarly, the triplet loss on the items $\mathcal{L}_{triplet}^{item}$ can be obtained.

$$\mathcal{L}_{triplet}^{item} = \sum_{u \in \mathcal{U}} max(0, s(\mathbf{e}_i^I, \mathbf{e}_i^A) - s(\mathbf{e}_i^I, \mathbf{e}_i^C) + \epsilon), \tag{26}$$

The overall triplet loss is then calculated as:

$$\mathcal{L}_{triplet} = \mathcal{L}_{triplet}^{user} + \mathcal{L}_{triplet}^{item} \tag{27}$$

3.5 Module 4: Model Training Process Based on Multi-task Curriculum Learning

In DECL, a multi-task training process is used to jointly optimize the embeddings $(\mathbf{E}^{I,(0)}, \mathbf{E}^{C,(0)})$:

$$\begin{aligned}\mathcal{L} =& \mathcal{L}_{interaction}^{\mathcal{O}_1+\mathcal{O}_2} + \alpha(\mathcal{L}_{interest}^{\mathcal{O}_2} + \mathcal{L}_{conformity}^{\mathcal{O}_1+\mathcal{O}_2}) + \\ & \beta\mathcal{L}_{independence} + \gamma\mathcal{L}_{triplet},\end{aligned} \tag{28}$$

where α,β, and γ are hyperparameters that control the effect of each loss term. We use the curriculum learning principle in the multi-task training process to train the framework parameters from easy to hard. That is, when the margin values m_{up} and m_{down} are large, there is a high confidence degree on inequalities (2-6) for interest, conformity, and interaction modeling, indicating the loss terms $\mathcal{L}_{interest}^{\mathcal{O}_2}$, $\mathcal{L}_{conformity}^{\mathcal{O}_1+\mathcal{O}_2}$, and $\mathcal{L}_{interaction}^{\mathcal{O}_1+\mathcal{O}_2}$ are easy to train. The framework then increases the training difficulty by multiplying the loss weight of interest and conformity modelling (α) and the margin values m_{up} and m_{down} with an attenuation coefficient in each epoch. This multiplication operation is also applied to the loss weight of contrastive learning (γ) to decrease the impact of the loss term ($\mathcal{L}_{triplet}$) on the training process.

TABLE 2
Statistics of three datasets

Dataset	User	Item	Interaction	Density
Ciao	2,577	3,796	77,683	0.79%
LastFM	1,779	1,402	61,147	2.45%
Movielens-10M	69,878	10,677	10,000,054	1.34%

4 Experiments

This section reports the experiment conducted to validate DECL. The framework is implemented by PyTorch[1], and all programs are executed on a computer with dual Intel Xeon E5-2678 v3 processors and an RTX 3090 GPU with 24 GB of memory.

4.1 Experiment Set Up

The main settings of the experiments in this study are given below.

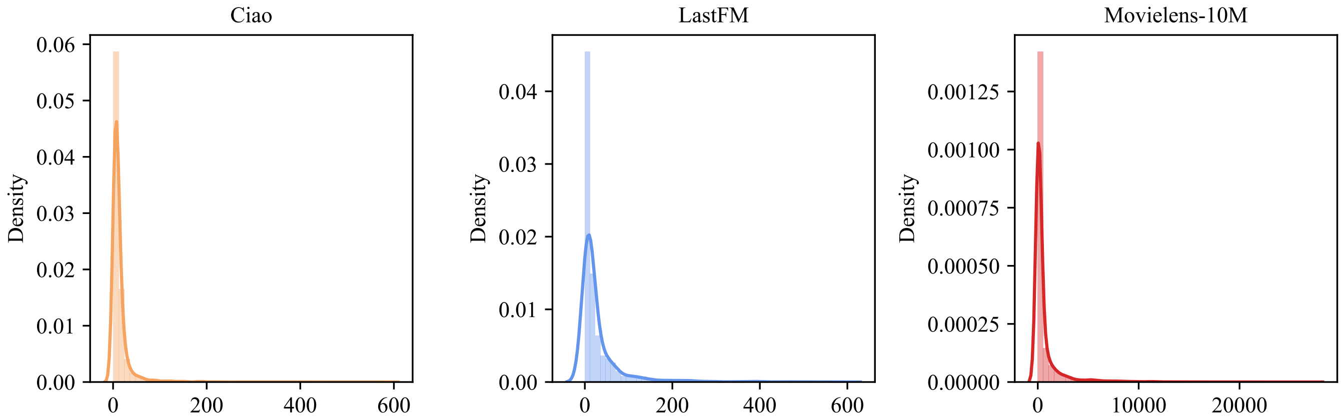
4.1.1 Datasets

The experiment is conducted on three real-world datasets: (1) the Ciao dataset[2], which consists of the interaction records of 2,577 users and 1,402 items; (2) the LastFM dataset[3], which consists of the interaction records of 1,779 users and 1,402 items; and (3) the Movielens-10M dataset[4], which consists of the rating records of 69,878 users on 10,677 items. The statistics of the datasets are shown in Table 2.

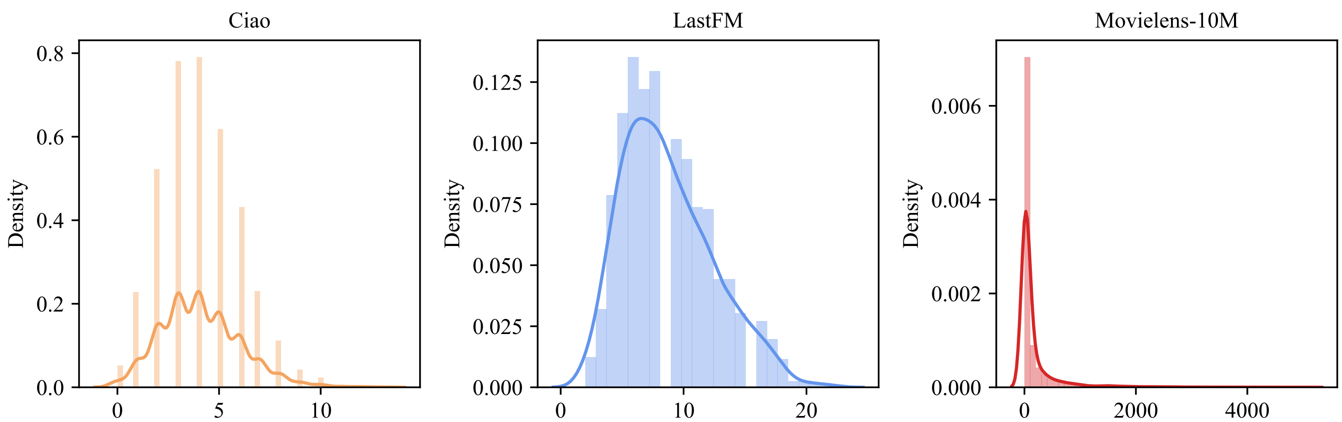
4.1.2 Data Preprocessing

We perform preprocessing for the raw data contained in the dataset. Since the proposed system makes recommendations based on the implicit feedback from users, a binarization operation [29], [30] is applied to the interaction data in the dataset. In the LastFM dataset, all the interaction data items are set to be 1. In the Movielens-10M dataset and Ciao dataset, if a user has rated an item, the interaction data value is set to be 1. After the binarization, we follow Ref. [9], [15] to further preprocess the three datasets: for each dataset, we construct a skewed dataset that contains 40% records of the dataset and keep the remaining 60% records in the dataset. Based on the item's popularity, the probability of the item i to be selected into the skewed dataset is $\frac{p_{min}}{p_i}$, where p_i is the popularity of the item i and p_{min} is the minimum

1. https://github.com/leo0481/DECL
2. https://www.cse.msu.edu/~tangjili/datasetcode/truststudy.htm
3. https://grouplens.org/datasets/hetrec-2011/
4. https://grouplens.org/datasets/movielens/



(a) Training dataset



(b) Testing dataset

Fig. 2. Gaussian kernel density estimation of popularity on the training and testing datasets.

popularity among the items. The selection probability is capped at 0.9 to limit the number of unpopular items that can be selected into the skewed dataset. In this way, the item's popularity is eliminated in the skewed dataset - in other words, all the interactions in the skewed dataset are caused by the users' interests. This step of preprocessing is to validate the performance of the recommendation models in capturing the user's real interest. Then, for each of the 3 original datasets, we divide it into a training dataset, a validation dataset, and a testing dataset with the data amount proportion of 70/10/20. The training dataset is composed of 25% samples in the skewed dataset and the rest 60% samples of the original dataset (i.e., all the samples except for the samples in the skewed dataset). 75% samples in the skewed dataset (except for those used in the training dataset) are used as both of the validation and testing datasets. In this way, the distributions of the items' popularity in the training and testing datasets are not IID, as shown in Fig. 2. The figure shows that the popularity distribution in the training dataset is more uneven, which shows that most users only interact with a few items. The popularity distribution in the testing dataset is more uniform, which better represents the users' real interests.

4.1.3 Benchmark Methods

We compare DECL with the following six state-of-the-art causal recommendation methods:

- **IPS** [11]: it assigns an inverse propensity score to each training sample to eliminate popularity bias. Specifically, it assigns low weights to popular items and high weights to unpopular items.
- **IPS-C** [20]: this method adds an upper limitation to the IPS values based on the IPS method.
- **IPS-CN** [10]: an improved IPS-C method. It performs normalization to the IPS values based on the IPS-C method.
- **IPS-CNSR** [10]: this method is based on IPS-CN; it applies power low transformation and normalization to the IPS values.
- **CausE** [15]: this method trains two embeddings for each user and each item on a large biased dataset and a small unbiased dataset; it then forces the two embeddings to be similar to each other.
- **DICE** [9]: this method trains the interest embedding and conformity embedding of each user/item on a cause-specific dataset; it then forces the two embeddings to be different from each other.

We test each of the above six benchmark methods under two recommendation models respectively: matrix fac-

This article has been accepted for publication in IEEE Transactions on Services Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSC.2024.3378925

torization (MF) [16] and LightGCN [18]. For DECL, only LightGCN is used as the recommendation model as MF is not applicable to graph-based user-item interaction data. To more comprehensively perform the validation, we also compare DECL with two additional benchmark cases in which MF and LightGCN are directly used to generate recommendations, respectively (i.e., without combining MF or LightGCN and a causal approach).

4.1.4 Evaluation Metrics

Four metrics are used in this experiment: Recall, Hit Ratio (HR), Normalized Discounted Cumulative Gain (NDCG), and Mean Average Precision (MAP). For each of the 4 metrics, a higher value indicates a better recommendation performance. The recall and HR metrics are expressed as Eqs. (29) and (30), respectively:

$$Recall = \frac{\#tp}{\#tp + \#fn}, \tag{29}$$

$$HR = \frac{\#tp}{\#tp + \#fp}, \tag{30}$$

where $\#tp$, $\#fp$, $\#fn$ denote the numbers of true-positive, false-positive, and false-negative recommended items, respectively. The true-positive and false-positive recommended items are the ones a target user has interacted with, while the target user has not interacted with the false-negative recommended items. The NDCG metric is expressed as:

$$NDCG@N = \frac{r(1) + \sum_{i=2}^{N} \frac{r(i)}{log_2^i}}{\sum_{i=1}^{|REL|} \frac{r(i)}{log_2^{(i+1)}}}, \tag{31}$$

where $r(i)$ is the relevance score: $r(i) = 1$ indicates that the recommended item has interacted with the user; $r(i) = 0$ indicates that there is no interaction. The notation $|REL|$ represents the sum of the top-N recommended items that are sorted in descending order of the relevance scores.

Let Y_{k_i} denote the number of items with which a target user has interacted among the top k_i recommended items; let r denote the number of items with which a target user has interacted among the top-N recommended items. Then, the MAP metric is defined as:

$$MAP = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{\sum_{i=1}^{r} \frac{Y_{k_i}}{k_i}}{r}. \tag{32}$$

4.1.5 Parameter Setting

For the IPS, IPS-C, IPS-CN, and IPS-CNSR methods, the embedding dimensionality is fixed as 128. For CausE, DICE, and DECL, since each of them contains two sets of embeddings, the embedding dimensionality for these methods is fixed as 64 to ensure fair comparisons. The values of α and β are set to be 0.1 and 0.01, respectively. The values of γ, ρ, τ, and ϵ are set to be $1e^{-5}$, 0.5, 0.5, and $1e^{-8}$, respectively. For each of the methods, we use BPR [17] as the loss function and Adam [31] as the model parameter optimizer. The grid search method is used to tune the hyperparameters.

4.2 Recommendation Performance Evaluation

Table 3 reports the recommendation performance of the different methods on the 3 datasets in terms of the four evaluation metrics. It can be seen that DECL achieves the best performance in all the cases. For example, compared with DICE, DECL achieves 10.69% and 8.65% higher values in terms of NDCG@5 and MAP@5 on the Ciao dataset and 3.97% and 3.88% higher values in terms of HR@5 and MAP@5 on the LastFM dataset. We further perform a paired t-test between the results produced by DECL and by the benchmark method that performs best on the 3 datasets (i.e., DICE), and the p-value is smaller than 0.01. This indicates the statistical test result is significant. Overall, the results in Table 3 proves that by introducing contrastive learning to optimize the distributions of the users' and items' representations, the recommendation efficiency can be enhanced.

It also can be seen from Table 3 that by using LightGCN as the recommendation model, higher recommendation performance can be achieved compared with the cases with MF as the recommendation model. This can be interpreted as LightGCN can better capture the complex nonlinear relationship between the users and items. Moreover, the results in Table 3 show that when the training dataset and testing dataset are non-IID, most causal approaches outperform the cases of solely using MF or LightGCN to perform recommendations; this validates the intuitive realization that by properly addressing the popularity bias (either eliminating it or leveraging it), the recommendation performance can be improved.

4.3 Validation of Contrastive Learning in Recommendation Performance Enhancement

We organize experiments to validate the effectiveness of the contrastive learning mechanism in DECL in improving the recommendation performance. For this purpose, in this section, we particularly compare DECL and DICE, as the latter can be essentially regarded as a variant of DECL without contrastive learning. In the comparison, we evaluate the uniformity of the representation set and the tolerance of the model on the semantically similar samples in the 3 datasets.

4.3.1 Uniformity of the Representation Set

We use a uniformity metric proposed by [32] to measure the uniformity of a representation set:

$$U = -\log \mathop{\mathbb{E}}_{\mathbf{e}', \mathbf{e}'' \sim p_{rep}} [exp(-2||\mathbf{e}' - \mathbf{e}''||_2^2)], \tag{33}$$

where p_{rep} denotes the representation set. The higher the uniformity, the more meaningful information of the users and items can be retained. Based on Eq. (33), we calculate the uniformity of 3 representation sets (i.e., the interest set, the conformity set, and the interaction representation set) in DECL and DICE, respectively. The results are shown in Fig. 3. Notably, on the Ciao dataset, the uniformity values of the three representation sets generated by DECL surpass those produced by DICE. This higher uniformity indicates that DECL achieves a better recommendation performance than DICE.

Fig. 3 also shows on the LastFM dataset, the uniformity of the interest representation set produced by DECL is

TABLE 3
Comparison of DECL and benchmark cases on the Ciao, LastFM, and Movielens-10M datasets

Dataset		Ciao				LastFM				Movielens-10M			
Model	Method	Recall	HR	NDCG	MAP	Recall	HR	NDCG	MAP	Recall	HR	NDCG	MAP
MF	None	0.0097	0.0547	0.0132	0.0260	0.0565	0.3153	0.0826	0.1574	0.0513	0.5328	0.1761	0.2458
	IPS	0.0094	0.0524	0.0129	0.0256	0.0603	0.3339	0.0900	0.1706	0.0584	0.5637	0.1914	0.2703
	IPS-C	0.0102	0.0578	0.0144	0.0272	0.0611	0.3384	0.0902	0.1665	0.0572	0.5541	0.1898	0.264
	IPS-CN	0.0095	0.0567	0.0140	0.0266	0.0617	0.3423	0.0914	0.1686	0.0575	0.5549	0.1910	0.2653
	IPS-CNSR	0.0091	0.0466	0.0119	0.0221	0.0558	0.3064	0.0829	0.1540	0.0563	0.5506	0.1897	0.2619
	CausE	0.0070	0.0373	0.0084	0.0151	0.0482	0.2754	0.0735	0.1469	0.0479	0.5304	0.1714	0.2375
	DICE	0.0114	0.0702	0.0169	0.0347	0.0765	0.4008	0.1156	0.2090	0.0611	0.5562	0.1873	0.2750
GCN	None	0.0134	0.0780	0.0203	0.0404	0.0798	0.4132	0.1208	0.2128	0.0625	0.5613	0.1909	0.2767
	IPS	0.0158	0.0861	0.0225	0.0435	0.0841	0.4255	0.1260	0.2142	0.0629	0.5664	0.1927	0.2806
	IPS-C	0.0140	0.0830	0.0201	0.0388	0.0862	0.4334	0.1297	0.2259	0.0633	0.5671	0.1932	0.2811
	IPS-CN	0.0142	0.0842	0.0220	0.0431	0.0863	0.4384	0.1304	0.2281	0.0631	0.5668	0.1925	0.2799
	IPS-CNSR	0.0140	0.0768	0.0202	0.0394	0.0803	0.4058	0.1177	0.2024	0.0638	0.5673	0.193	0.2813
	CausE	0.0067	0.0431	0.0093	0.0175	0.0600	0.3395	0.0925	0.1792	0.0612	0.5514	0.1843	0.2695
	DICE	<u>0.0183</u>	<u>0.1024</u>	<u>0.0262</u>	<u>0.0497</u>	<u>0.0963</u>	<u>0.4682</u>	<u>0.1474</u>	<u>0.2474</u>	<u>0.0799</u>	<u>0.6406</u>	<u>0.2388</u>	<u>0.3184</u>
	Our method	**0.0193**	**0.1067**	**0.0290**	**0.0540**	**0.0987**	**0.4868**	**0.1502**	**0.2570**	**0.0812**	**0.6488**	**0.2424**	**0.3232**
	Improv.	5.46%	4.20%	10.69%	8.65%	2.49%	3.97%	1.90%	3.88%	1.63%	1.28%	1.51%	1.51%

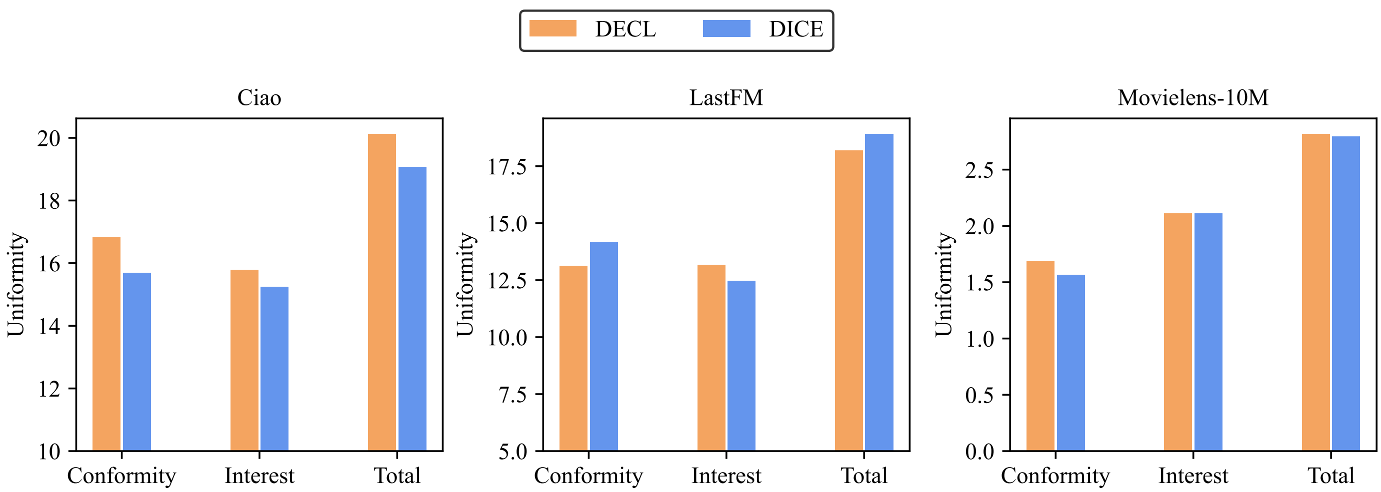


Fig. 3. Uniformity of the representation sets trained by DECL and DICE on the three datasets.

TABLE 4
Tolerance values generated by DECL and DICE on the three datasets

Dataset	Ciao	LastFM	Movielens-10M
DICE	5.027	2.533	3.099
DECL	10.389	5.310	3.123

TABLE 5
Distance correlation values of the interest and comformity representation sets trained by DECL and DICE on the three datasets

Dataset	Ciao	LastFM	Movielens-10M
DICE	34.08	16.85	457.59
DECL	34.83	16.56	450.05

higher than that produced by DICE, but the uniformity values of the conformity representation set and the interaction representation set are lower. Despite this, DECL still performs better than DICE on this dataset in terms of recommendation accuracy. This indicates that on unbiased testing datasets, DECL is more capable of capturing the real interests of the users and items. On the MovieLens-10M dataset, the uniformity values of the three representation sets obtained by DECL and DICE are similar; as a result, the recommendation performance enhancement of DECL on this dataset is marginal.

4.3.2 *Tolerance of the Recommendation Model*

By applying the data augmentation method in Section 3.4 to the anchor $\mathbf{e}$, the sematic of the generated positive sample $\mathbf{e}^+$ is similar with the semantic of the anchor. This implies the two samples should be clustered in the representation space; otherwise, the model would mistakenly lean the semantic structure, degrade the representation ability, and will subsequentially affect the recommendation performance. In the experiment, we use a metric proposed by [33] to measure the model's tolerance:

$$T = \mathop{\mathbb{E}}_{(\mathbf{e},\mathbf{e}^+)\sim p_{pos}} [\mathbf{e}^T \mathbf{e}^+], \tag{34}$$

where p_{pos} denotes the distribution of the sample pair of the anchor and the corresponding positive sample. Using this metric, the tolerance values of DECL and DICE on the 3 datasets are reported in Table 4. It can be seen that the tolerance of DECL is significantly larger than that of DICE on the Ciao and LastFM datasets; DECL is with a slightly larger tolerance than DICE on the MovieLens-10M dataset. The result of the models' tolerance values indicates that by integrating contrastive learning principles, DECL can effectively make similar samples get closer to each other in the representation space. This makes the proposed model not only capable of capturing the user's interest when it is different with the item popularity, but also appliable to the situation where the user's interest aligns with the broader population. In the latter situation, the model will make the user's interest representation similar to the conformity representation. Such an enhanced local density in the representative space will affect the recommendation performance - this is reflected in Tables 3 and 4, where we can find that the difference of tolerance values between DECL and DICE is smallest on the MovieLens-10M dataset; as a result, DECL achieves the least recommendation performance improvement on this dataset.

Besides evaluating the uniformity of the representation set and the model's tolerance as above, we compare the distance correlation values of the representation distributions trained by DECL and DICE, respectively. The result is given in Table 5. On the Ciao dataset, the distance correlation between the 2 representation sets trained by DECL is slightly smaller than the distance correlation between the representation sets trained by DICE; on the other 2 datasets, such a distance correlation in DECL is slightly larger than that in DICE. From all these results, we can see that compared with DICE, DECL does not significantly reduce the distance correlation between the two representation sets; this indicates that the main reason for the superior performance of DECL is that it can effectively optimize the distributions of the representation sets.

4.4 Sensitivity Analysis for Hyperparameters

We further conduct sensitivity analysis for investigating the impact of three important hyperparameters on DECL's recommendation performance: the contrastive learning loss weight (γ), the dropout ratio (ρ), and the temperature (τ). In the sensitivity analysis, the variation ranges of the hyperparameters are set as follows: γ varies within $\{0, 1e^{-7}, 1e^{-6}, \ldots, 1e^{-3}\}$; ρ varies within $\{0.1, 0.2, \ldots, 0.9\}$; and τ varies within $\{0.1, 0.2, \ldots, 1\}$. The sensitivity analysis results are shown in Figs. 4-6.

Fig. 4 shows that the recommendation performance of DECL increases rapidly on the Ciao dataset when the value of γ increases from 0 to $1e^{-7}$. At $\gamma = 1e^{-7}$, DECL achieves the best performance. The recommendation performance then drops when γ is larger than $1e^{-6}$. Similar trends are also observed on the LastFM and MovieLens-10M datasets, where DECL achieves the best performance at $\gamma = 1e^{-5}$ and then steadily declines. These observations indicate that the contrastive learning mechanism can improve the recommendation performance even if the value of γ is very small, while an overly large value of γ would mislead the training process of the model and consequentially degrade the performance. The value of ρ determines the similarity between the positive sample and the anchor in a training sample. The larger the value of ρ is, the less similar the augmented representation and the interest representation is. Fig. 5 shows that when the value of ρ is small, only marginal recommendation performance improvement can be achieved by increasing the value of ρ. On the Ciao dataset, DECL achieves the best performance at $\rho = 0.8$; on the other two datasets, DECL performs the best at $\rho = 0.5$. The recommendation performance then slightly decreases with a continuously increase of ρ. The result indicates that when the two representations are moderately unsimilar, the contrastive learning mechanism can enhance DECL's performance; however, if such an unsimilar degree is very large, the framework's performance will be negatively affected. As for the hyperparameter τ, Fig. 6 shows that on the Ciao and MovieLens-10M datasets, DECL's recommendation performance rises sharply when the value of τ varies from 0.1 to 0.2. Further increasing the value of τ will not lead to significant performance enhancement for DECL. On the LastFM dataset, DECL's performance increases rapidly when the value of τ increases from 0.1 to 0.5. When $\tau = 0.5$, DECL achieves the best performance. When τ is larger than 0.6, no obvious recommendation performance variation can be observed.

5 Conclusion and Future Work

This paper proposes a new recommendation framework that enhances DisEntanglement of popularity bias based on Contrastive Learning called DECL. The framework integrates a new contrastive learning mechanism to effectively disentangle the users' and the items' interest and conformity. This makes the proposed framework can well address the popularity bias issue, which is an important consideration in recommendation problems. The experiment results demonstrate that by integrating the contrastive learning mechanism, DECL can optimize the distributions of the three representation sets (i.e., the interest, the conformity, and the interaction representation sets) in the representation space while enhancing the tolerance of the recommendation model on the semantically similar samples. As a result, DECL can achieve superior recommendation performance on three real-world datasets compared with the state-of-the-art methods. Overall, the methodological design and experiments reported in this paper have demonstrated that DECL can effectively mitigate the negative impact caused by popularity bias, implying its high applicability in real-world recommendation scenarios.

Future work can be conducted in different ways. As the authors' current work in this direction, methodologies are being developed for further disentangling the two macrolevel causes (i.e., interest and conformity) into finer causes. It is also worthy of studying how to separate the representation sets of different causes while sufficiently preserving the distributions of the representation sets. Apart from these theoretical studies, applications of the recommender system proposed in this paper in the domains where recommender systems have been widely applied (e.g., agri-

This article has been accepted for publication in IEEE Transactions on Services Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSC.2024.3378925

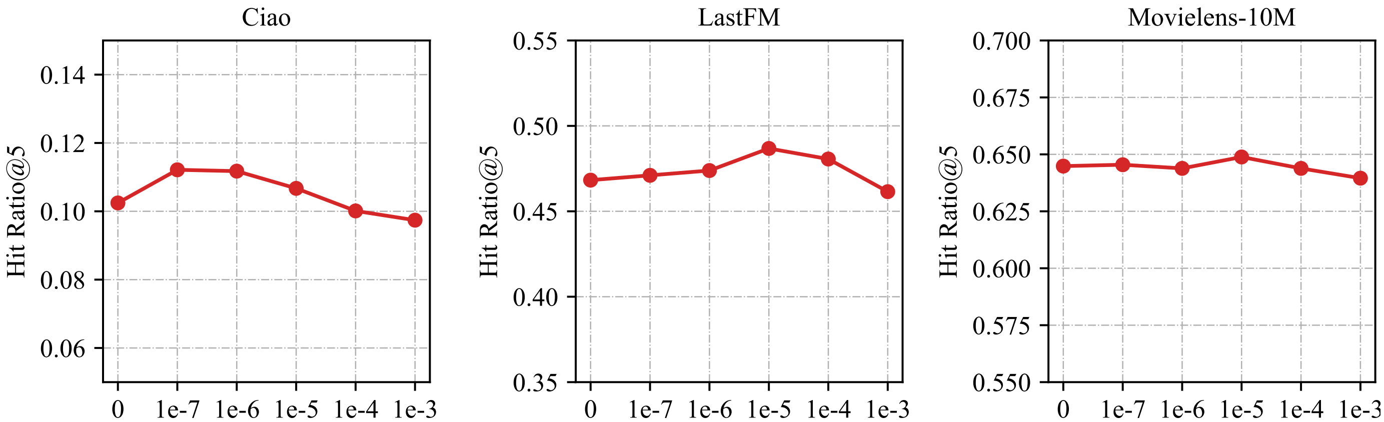


Fig. 4. Impact of different values of γ on the HR metric (in this experiment, the values of ρ and τ are fixed at 0.5 and 0.5, respectively).

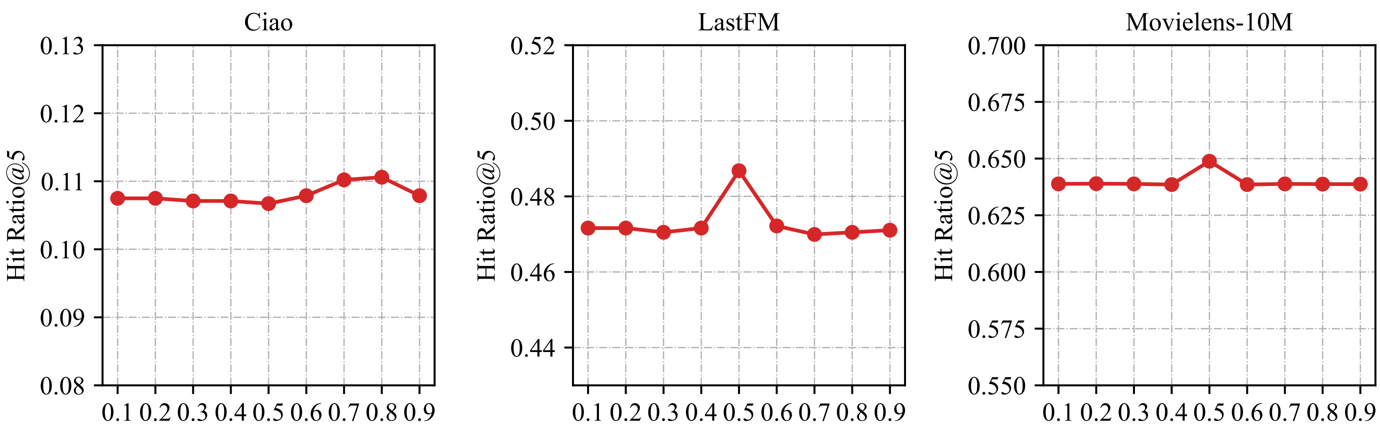


Fig. 5. Impact of different values of ρ on the HR metric (in this experiment, the values of γ and τ are fixed at $1e^{-5}$ and 0.5, respectively).

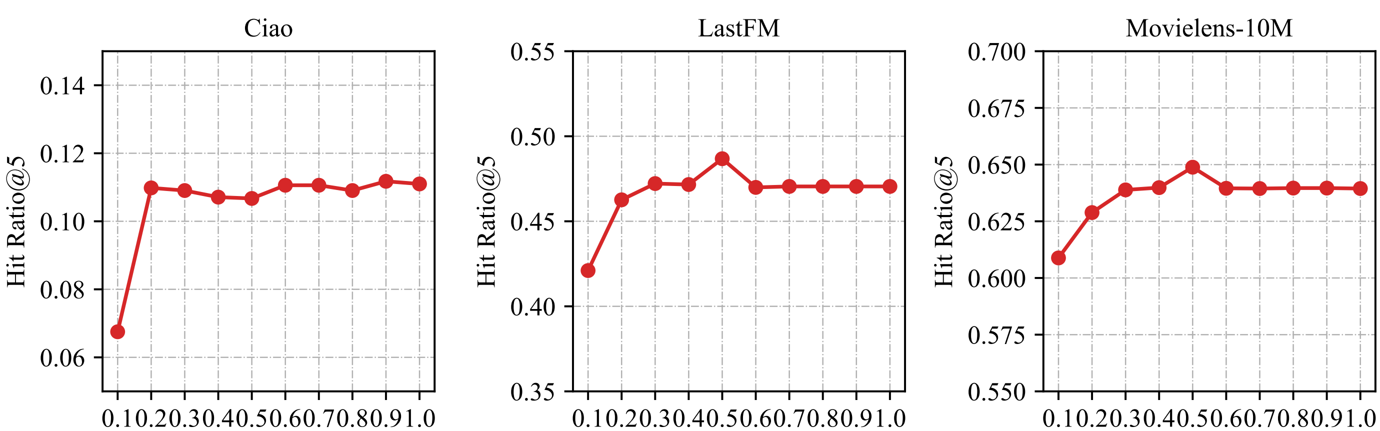


Fig. 6. Impact of different values of τ on the HR metric (in this experiment, the values of γ and ρ are fixed at $1e^{-5}$ and 0.5, respectively).

culture [34], Web services [35] and movies [36]), deserve specific investigations.

6 ACKNOWLEDGMENTS

This work is in part supported by the National Natural Science Foundation of China (Grant No. 62072060, 72074036). The 3rd author's work is supported by the Australian Research Council through a Discovery Project (DP220103881).

REFERENCES

[1] J. Liao, W. Zhou, F. Luo, J. Wen, M. Gao, X. Li, and J. Zeng, "Sociallgn: Light graph convolution network for social recommendation," *Information Sciences*, vol. 589, pp. 595–607, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025522000019

[2] X. Zheng, M. Wang, R. Xu, J. Li, and Y. Wang, "Modeling dynamic missingness of implicit feedback for sequential recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 405–418, 2022.

[3] B. Wu, X. He, Y. Chen, L. Nie, K. Zheng, and Y. Ye, "Modeling product's visual and functional characteristics for recommender systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 3, pp. 1330–1343, 2022.

[4] L. Ren and W. Wang, "A granular svm-based method for top-n web services recommendation," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 457–469, 2022.

[5] S. Badsha, X. Yi, I. Khalil, D. Liu, S. Nepal, E. Bertino, and K. Y. Lam, "Privacy preserving location-aware personalized web service recommendations," *IEEE Transactions on Services Computing*, vol. 14, no. 3, pp. 791–804, 2021.

[6] F. Luo, G. Ranzi, W. Kong, G. Liang, and Z. Y. Dong, "Personalized residential energy usage recommendation system based on load monitoring and collaborative filtering," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1253–1262, 2021.

[7] F. Luo, G. Ranzi, X. Wang, and Z. Y. Dong, "Social information filtering-based electricity retail plan recommender system for smart grid end users," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 95–104, 2019.

[8] R. Cañamares and P. Castells, "Should i follow the crowd? a probabilistic analysis of the effectiveness of popularity in recommender systems," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, ser. SIGIR '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 415–424. [Online]. Available: https://doi.org/10.1145/3209978.3210014

[9] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, and D. Jin, "Disentangling user interest and conformity for recommendation with causal embedding," *The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021*, pp. 2980–2991, 2021.

[10] A. Gruson, P. Chandar, C. Charbuillet, J. McInerney, S. Hansen, D. Tardieu, and B. Carterette, "Offline evaluation to make decisions about playlistrecommendation algorithms," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, ser. WSDM '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 420–428. [Online]. Available: https://doi.org/10.1145/3289600.3291027

[11] T. Joachims, A. Swaminathan, and T. Schnabel, "Unbiased learning-to-rank with biased feedback," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, ser. WSDM '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 781–789. [Online]. Available: https://doi.org/10.1145/3018661.3018699

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

[13] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 5812–5823. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/3fe230348e9a12c13120749e3f9fa4cd-Paper.pdf

[14] 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*, ser. SIGIR '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 726–735. [Online]. Available: https://doi.org/10.1145/3404835.3462862

[15] S. Bonner and F. Vasile, "Causal embeddings for recommendation," in *Proceedings of the 12th ACM Conference on Recommender Systems*, ser. RecSys '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 104–112. [Online]. Available: https://doi.org/10.1145/3240323.3240360

[16] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

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

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

[19] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, and R. Burke, "A graph-based approach for mitigating multi-sided exposure bias in recommender systems," *ACM Trans. Inf. Syst.*, vol. 40, no. 2, nov 2021. [Online]. Available: https://doi.org/10.1145/3470948

[20] L. Bottou, J. Peters, J. Quiñonero Candela, D. X. Charles, D. M. Chickering, E. Portugaly, D. Ray, P. Simard, and E. Snelson, "Counterfactual reasoning and learning systems: The example of computational advertising," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 3207–3260, Jan. 2013.

[21] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 539–546 vol. 1.

[22] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[23] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 297–304. [Online]. Available: https://proceedings.mlr.press/v9/gutmann10a.html

[24] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv e-prints*, pp. arXiv–1807, 2018.

[25] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf

[26] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=Bklr3j0cKX

[27] R. Qiu, Z. Huang, H. Yin, and Z. Wang, "Contrastive learning for representation degeneration problem in sequential recommendation," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, ser. WSDM '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 813–823. [Online]. Available: https://doi.org/10.1145/3488560.3498433

[28] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and

testing dependence by correlation of distances," *The annals of statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[29] H. Wu, C. W. Wong, J. Zhang, Y. Yan, D. Yu, J. Long, and M. K. Ng, "Cold-start next-item recommendation by user-item matching and auto-encoders," *IEEE Transactions on Services Computing*, vol. 16, no. 4, pp. 2477–2489, 2023.

[30] X. Wang, Q. Li, D. Yu, P. Cui, Z. Wang, and G. Xu, "Causal disentanglement for semantic-aware intent learning in recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 9836–9849, 2023.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[32] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 9929–9939. [Online]. Available: https://proceedings.mlr.press/v119/wang20k.html

[33] F. Wang and H. Liu, "Understanding the behaviour of contrastive loss," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 2495–2504.

[34] M. Garanayak, G. Sahu, S. Mohanty, and A. Jagadev, "Agricultural recommendation system for crops using different machine learning regression methods," *International Journal of Agricultural and Environmental Information Systems*, vol. 12, pp. 1–20, 01 2021.

[35] S. N. Mohanty, J. R. Parvin, K. V. Kumar, K. C. Ramya, S. S. Rani, and S. K. Lakshmanaprabu, "Optimal rough fuzzy clustering for user profile ontology based web page recommendation analysis," *J. Intell. Fuzzy Syst.*, vol. 37, no. 1, pp. 205–216, 2019. [Online]. Available: https://doi.org/10.3233/JIFS-179078

[36] S. S. Choudhury, S. N. Mohanty, and A. K. Jagadev, "Multimodal trust based recommender system with machine learning approaches for movie recommendation," vol. 13, no. 2, pp. 475–482. [Online]. Available: https://doi.org/10.1007/s41870-020-00553-2

Jie Liao received his bachelor's degree in economics from Chongqing University, in 2020. He is currently working toward the Ph.D. degree at the School of Big Data and Software Engineering, Chongqing University, China. His research interests include recommendation systems, social computing, self-supervised learning and graph neural network.

Wei Zhou received PhD degree in computer science from Chongqing University in 2015. He is an associate professor at the School of Big Data and Software Engineering, Chongqing University. He was a researcher at the University of Auckland, the Hong Kong Polytechnic University and Shenzhen Institutes of Advanced Technology of CAS respectively. His research interests include recommendation systems, social computing and data mining.

Fengji Luo (S'10-M'13) received his bachelor and master degrees in Software Engineering from Chongqing University, China in 2006 and 2009. He received his Ph.D degree in Electrical Engineering from The University of Newcastle, Australia, 2014. Currently, he is a Lecturer and Academic Fellow in the School of Civil Engineering, The University of Sydney, Australia. His research interests include energy demand side management, smart grid, energy informatics, and service-oriented computing and its aqpplications in energy systems. He has 150+ technical publications in these areas that have attracted 6,000+ citations from peers (Google Scholar). He received the Pro-Vice Chancellor's Research and Innovation Excellence Award of The University of Newcastle in 2015 and ATSE's Australia-Japan Emerging Research Leader Program in 2016. In 2020, he is named as one of the Top 50 engineering and computer science Researchers in Australia by The Australian. He is an Associate Editor of Frontiers in Energy Research, IET Renewable Power Generation, and Energy Conversion and Economics.

Junhao Wen received PhD degree in computer science from Chongqing University in 2005. He is a professor and dean at the School of Big Data and Software Engineering, Chongqing University. His research interests include recommendation systems, social computing and service computing.