Socially-aware Dual Contrastive Learning for Cold-Start Recommendation

Jing Du*
The University of New South Wales
Sydney, Australia
jing.du2@unsw.edu.au

Zesheng Ye*
The University of New South Wales
Sydney, Australia
zesheng.ye@unsw.edu.au

Lina Yao*
The University of New South Wales
Sydney, Australia
lina.yao@unsw.edu.au

Bin Guo‡
Northwestern Polytechnical University
Xi'an, Shaanxi, China
guobin.keio@gmail.com

Zhiwen Yu‡
Northwestern Polytechnical University, China
Xi'an, Shaanxi, China
zhiweny@gmail.com

ABSTRACT

Social recommendation with Graph Neural Networks (GNNs) learns to represent cold users by fusing user-user social relations with user-item interactions, thereby alleviating the cold-start problem associated with recommender systems. Despite being well adapted to social relations and user-item interactions, these supervised models are still susceptible to *popularity bias*. Contrastive learning helps resolve this dilemma by identifying the properties that distinguish positive from negative samples. In its previous combinations with recommender systems, social relationships and cold-start cases in this context are not considered. Also, they primarily focus on collaborative features between users and items, leaving the similarity between items under-utilized. In this work, we propose socially-aware dual contrastive learning for cold-start recommendation, where cold users can be modeled in the same way as warm users. To take full advantage of social relations, we create dynamic node embeddings for each user by aggregating information from different neighbors according to each different query item, in the form of user-item pairs. We further design a dual-branch self-supervised contrastive objective to account for user-item collaborative features and item-item mutual information, respectively. On one hand, our framework eliminates *popularity bias* with proper negative sampling in contrastive learning, without extra ground-truth supervision. On the other hand, we extend previous contrastive learning methods to provide a solution to cold-start problem with social relations included. Extensive experiments on two real-world social recommendation datasets demonstrate its effectiveness.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; • **Information systems → Social recommendation**.

KEYWORDS

social recommendation, cold-start, contrastive learning

ACM Reference Format:
Jing Du, Zesheng Ye, Lina Yao, Bin Guo, and Zhiwen Yu. 2022. Socially-aware Dual Contrastive Learning for Cold-Start Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531780

1 INTRODUCTION

The cold-start problem has plagued recommender systems for a long time. Though attempted have been made to model cold users using auxiliary information such as user profile and item content, these static features do not adapt dynamically as the system evolves. An alternative overcomes this issue, where social networks are integrated with user preferences on items, known as social recommendations [6]. Those with whom users have social connections can influence their potential preferences, which might be effective in modeling cold users who rarely have interactions with items.

Graph Neural Networks (GNNs) are proven to be capable of modeling complex patterns of user behaviors and learning effective representations for users and items. GCMC [2] present differentiable message-passing to derive latent features of user-item interactions. ICMC [24] exten GCMC to inductive representation learning for generalizing to unseen users and items. DiffNet [20] simulate social diffusion with hierarchical influence propagation, while DiffNetLG [17] bake implicit social influence into the embeddings of users and items. GraphRec [5] incorporate different relations into a heterogeneous graph and fuse user-user social relations with user-item interactions. ConsisRec [21] tackle social inconsistency by sampling consistent neighbors and generating dynamic node embeddings for different query user-item pairs. In a nutshell, GNNs enable efficient embeddings of users and items by associating user-user relations with user-item interactions.

Nonetheless, *popularity bias* [1] might still occur with these supervised methods that rely on sufficient records of user-item interactions, as items with higher popularity tend to be interacted with more frequently. In this situation, items that rarely receive attention may contain valuable information that is not fully investigated. Contrastive learning [8, 15] can eliminate such impact of item popularity[10, 11, 16] by pairing up under-investigated (negative)

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531780

items with popular (positive) items, and spotting their semantic differences in a self-supervised manner. Such paradigm suggests a workaround for generalizing popular items to under-investigated counterparts without additional user-item interaction records. However, neither are they designed for the cold-start problem nor assembled with mutual information between items considered [19], but instead focus on collaborative features between users and items. Moreover, they do not explicitly take user-user social relationships into consideration, thus failing to facilitate social recommendation.

To this end, targeting the cold-start problem, we propose a contrastive representation learning framework for social recommendation, with three key designs included: a) **dynamic node embedding**, b) **dual contrastive organization** and c) **debiased contrastive objectives**. Our framework operates over a heterogeneous graph that fuses user-user social relations and user-item interactions, and encodes both user and item nodes into a unified embedding space, making it reasonable to compare the pairwise similarity between any two nodes. Each user is paired with every item, forming a user-item duo. In particular, **dynamic node embedding** allows a user to consult different socially-connected neighbors to determine an adaptive embedding for different paired query item, and vice versa. We further develop a **dual-branch contrastive** strategy upon the embedding pairs to enforce similar users/items to be attracted but dissimilar ones are repellent. We optimize the similarity between user and favorite item in one branch, while focusing on the mutual information between items in another. Furthermore, considering *sampling bias* resulting from common negative sampling strategies could cause inaccurate estimation of user preferences, a down-scaling multiplier is applied to the related terms in **debiased contrastive objectives**. In this way, even if there is no interaction data, cold users can be dynamically encoded through the integration of user-user social connections and user-item collaborative features, with item-item correlations used to address the *popularity bias* therein.

We summarize the contribution of this work as

- We present a **S**ocially-aware **D**ual **C**ontrastive learning framework for cold-start **Rec**ommendation (SDCRec), in which user-user relations, user-item interactions and item-item similarity are combined to modulate the representations under a semi-supervised setting. Cold users are incorporated using social relations and modeled as warm users seaminglessly, without additional user-item interaction records.
- We feature dynamic node embedding with contrastive learning to diminish *popularity bias.* The dual-branch sampling strategy constructs user-item and item-item contrastive tuples, permitting user-item collaborative features and item-item mutual information to be jointly explored. A simple yet effective coefficient is further deployed to addresses *sampling bias* inherent in negative sampling.
- We evaluate the proposed method with two real world datasets of social recommendation. Empirical results on warm and cold scenarios demonstrate its effectiveness.

2 PRELIMINARIES

Say we have a heterogeneous graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{R} \rangle$, where the node set $\mathcal{V} = \mathcal{U} \cup \mathcal{T}$ consists of a user set $\mathcal{U} = \{u_i\}_{i=1:m}$ with m users and an item set $\mathcal{T} = \{t_j\}_{j=1:n}$ including n items. The edge set $\mathcal{R} = \mathcal{S} \cup \mathcal{I}$ contains two types of relations $\mathcal{S} = \{S_{i,j}\}$ and $\mathcal{I} = \{I_{i,k}\}$. Let $S_{i,j} = \langle u_i, u_j \rangle$ indicate the user-user social connection from u_i to u_j, $S_{i,j} = 1$ if there exist edge between u_i and u_j, $S_{i,j} = 0$ otherwise. Let $I_{i,k} = \langle u_i, t_k \rangle$ denote observed user-item rating score from u_i to t_k upon previous interactions, where $I_{i,j}$ is proportional to u_i's preference on t_k. With $\mathcal{I}$, the user set is split into two disjoint subsets $\mathcal{U} = \mathcal{U}^w \cup \mathcal{U}^c$, where $\mathcal{U}^w$ contains q warm users u^w, each are associated with at least one user-item interaction, while $\mathcal{U}^c$ gathers $m - q$ cold users u^c without interactions with any item. Given user u and query item t that does not appear in the user's interaction records, the task is to predict user rating score $\hat{I}_{u,t}$.

3 METHODOLOGY

3.1 Overview

SDCRec deliberates on node representation learning under a semi-supervised setting, where interactions $\mathcal{I}$ provide input to the model, whereas they do not provide direct supervision. In other words, SDCRec is optimized with a self-supervised objective that is independent of $\mathcal{I}$. Concretely, we aim to train a GNN-based embedding model f that encodes $\mathcal{G}$'s nodes into a d-dimensional unified embedding space making no distinction to user and item node embedding $v \mapsto \mathbf{e}_v \in \mathbb{R}^d$. Furthermore, different relationships are fused into a hybrid relation, giving cold users $u^c \in \mathcal{U}^c$ multi-hop connections to items $t \in \mathcal{T}$ through social links with warm users $u^w \in \mathcal{U}^w$. The same emebdding process can be used for cold and warm users in this case. Hence, we use u to represent both warm and cold users, and r for both social and interaction relations. Relations are also mapped to vectorized embeddings $r \mapsto \mathbf{e}_r \in \mathbb{R}^d$.

Fig 1 (b) overviews the workflow of SDCRec. One goes through four steps to predict the rating score from a user $u \in \mathcal{U}$ to a query item $t \in \mathcal{T}$, including a) *Dynamic Node Embedding*, b) *Dual-branch Organization*, c) *Contrastive Optimization*, and d) *Rating Prediction*.

Fig 1 (b) omits *Rating Prediction* since it is not part of the training process, but is used to evaluate the model afterwards. The dot-product similarity $\hat{I}(u,t) = \mathbf{e}_u \cdot \mathbf{e}_t$ between embeddings of user u and query item t is used as predicted user rating. We elaborate *Dynamic Node Embedding*, *Dual-branch Organization* and *Contrastive Optimization* in Sec 3.2, Sec 3.3 and Sec 3.4, respectively.

3.2 Dynamic Node Embedding

For user u and a query item t, embedding model f takes $\mathcal{G}$ as input and generates user embedding $\mathbf{e}_u$ and item embedding $\mathbf{e}_t$. Considering that t might be more relevant to those users who have interacted with it, user embedding $\mathbf{e}_u$ should not be fixed but adapt to different query items by consulting different neighbors, and vice versa. Thus, the strategy of dynamic sampling and aggregation guides all nodes $\forall\, v \in \mathcal{V}$.

3.2.1 Neighborhood Sampling. First, we define the neighborhood function of node v given $\mathcal{G}$,

$$\mathcal{N}(v) = \begin{cases} \{u_i \mid \forall u_i \in \mathcal{S}_{v,i}\} \cup \{t_j \mid \forall t_j \in \mathcal{I}_{v,t_j}\} & \text{if } v \in \mathcal{U} \\ \{u_i \mid \forall u_i \in \mathcal{I}_{i,v}\} \cup \{t_j \mid J(v,j) \geq \epsilon, \forall t_j \in \mathcal{T}\} & \text{if } v \in \mathcal{T} \end{cases} \tag{1}$$

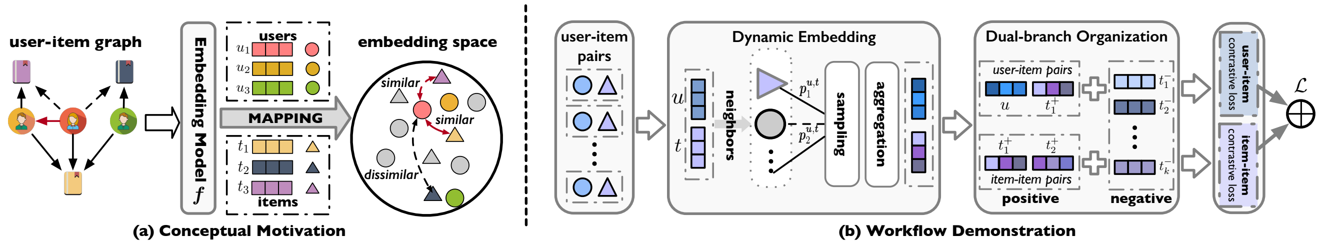

Figure 1: Overview of the proposed framework. The overview of SDCRec, consisting of Dynamic Embedding, Dual-branch organization, and Debiased Contrastive Optimization.

where $J(v, t)$ is Jaccard distance calculating the overlap of interacted users between two item nodes. ϵ is the threshold.

Next, a sampling probability $p_i^{(u,t)}$ is assigned to each neighbor node $i \in \mathcal{N}(v)$ of v, determined by the distance between its embedding $\mathbf{e}_i$ and the query user-item pair $(\mathbf{e}_u, \mathbf{e}_t)$, enabling dynamic neighbor selection when dealing with different query items.

$$p_i^{(u,t)} = \frac{\phi\left(\mathbf{e}_u \parallel \mathbf{e}_t\right) \cdot \mathbf{e}_i}{\sum_{k \in \mathcal{N}(v)} \phi\left(\mathbf{e}_u \parallel \mathbf{e}_t\right) \cdot \mathbf{e}_k}, \quad \forall i \in \mathcal{N}(v) \tag{2}$$

where $\phi(\cdot)$ implies non-linear transformation, $\parallel$ denotes vector concatenation and $\cdot$ is dot-production between two vectors.

3.2.2 Neighborhood Aggregation. Last, one aggregates the sampled neighbor nodes through a GNN with K-layers [7].

$$\mathbf{h}_v^k \leftarrow \Phi^k\Big(\sum_{i \in \mathcal{N}'(v)} \mathbf{h}_v^{k-1} \parallel \alpha_i^k\, \mathbf{h}_i^{k-1}\Big), \quad \text{for } k = 1 \ldots K \tag{3}$$

where $\Phi^k(\cdot)$ defines a non-linear transformation at k-th layer, $\parallel$ is vector concatenation, $\mathcal{N}'(v)$ refers to sampled neighbor nodes for current query user-item pair. Due to difference in the value of relation $r(v, i)$, the contribution provided by each neighbor i differs. It is measured with coefficient α_i^k for i at the k-th layer, and calculated by self-attention on the concatenation of the neighbor node's hidden state $\mathbf{h}_i^{k-1}$ at previous layer and edge embedding $\mathbf{e}_{r(v,i)}$. Let the final hidden state of v be its embedding $\mathbf{e}_v \leftarrow \mathbf{h}_v^K$.

3.3 Dual-branch Organization

The embedding model f outputs a series of user-item pairs for each user u. These paired embeddings intuitively lead to further optimization of similarity between nodes in a contrastive manner. We propose a dual-branch structure namely *User-Item Sampling* and *Item-Item Sampling*, where mutual information between user and the preferred item, as well as density ratio of similar items are maximized therein, respectively.

3.3.1 User-Item Sampling. Recall that the user and item embeddings are not distinguished in the unified space, suggesting that they may be more collaboratively associated if their embeddings are more similar. Therefore, item ranking the highest in similarity to user u is considered positive, while other k negative samples are randomly drawn from non-interacted items with scores below than a certain threshold, too. As such, we can dot product the embeddings to calculate similarity scores for u and each query item t. Note that here we do not use any additional user-item interaction data for cold users by purely looking up social relationships. The user-item contrastive tuples can then be defined

$$\begin{cases} y_{u,t}^{+} = (u, t_1^{+}) & u \cdot t_1^{+} \geq u \cdot t, \quad \forall t \in \mathcal{T} \\ y_{u,t}^{-} = \left\{(u, t_i^{-})\right\}_{i=1:k} & (u \cdot t_i^{-} < \eta) \wedge (t_i^{-} \notin \mathcal{I}_u) \end{cases} \tag{4}$$

where $y_{u,t}^{+}$ and $y_{u,t}^{-}$ collects the positive tuple and negative tuples for user u. $\cdot$ calculates dot-product similarity, t_1^{+} is the positive item to user u. t_i^{-} represents one of k negative items. η is the threshold to filter out negative samples, we use the mean of pairwise similarities between user u and all items in practice.

3.3.2 Item-Item Sampling. We additionally set up Item-Item branch that utilizes the similarity between items when they come to user u to determine the embedding quality [18]. Let two items that have the highest similarity to u be the positive pair. A random collection of k non-interacted items constitute negative pairs with the top item t_1^{+}. The Item-Item sampling is therefore formulated as

$$\begin{cases} y_{t,t}^{+} = (t_1^{+}, t_2^{+}) & u \cdot t_1^{+} > u \cdot t_2^{+} \geq u \cdot t, \quad \forall t \in \mathcal{T} \\ y_{t,t}^{-} = \left\{(t_1^{+}, t_i^{-})\right\}_{i=1:k} & (u \cdot t_i^{-} < u \cdot t_2^{+}) \wedge (t_i^{-} \notin \mathcal{I}_u) \end{cases} \tag{5}$$

The User-Item branch examines if a query item matches the user's preference, whereas the Item-Item branch identifies items that are not in line with the user's preferred item. In both branches, negative sampling ensures user's embedding $\mathbf{e}_u$ takes all the items into account, including those under-explored, thus overcoming *popularity bias*. Having two contrastive branches combined enables to create a representation space that highlights the items that satisfy users, while identifying dis-similar items more precisely.

3.4 Debiased Contrastive Optimization

The sampled contrastive tuples are handled with a contrastive objective in each branch for all users $u \in \mathcal{U}$. We adopt InfoNCE [15] to maximize the density ratio of positive samples among candidate pairs. Particularly, we employ a simple yet effective strategy to mitigate *sample bias*, arising from the fact that items in the negative samples can still be of interest to the user. To reduce their influence, negative terms are multiplied by a down-scaling factor $\gamma (\leq 1)$. We define the overall objective that calculates the expectation across

Table 1: Overall Comparisons of Model Performances. Numerical results are reported as the mean ± standard deviation over 5 different random seeds.

	Methods	Ciao				Epinions			
		warm		cold		warm		cold	
		n@10(10^{-2})	**m@10**(10^{-2})	**n@10**(10^{-3})	**m@10**(10^{-3})	**n@10**(10^{-2})	**m@10**(10^{-2})	**n@10**(10^{-3})	**m@10**(10^{-3})
Vanilla (for RQ1)	SoReg	5.593(±0.0002)	6.314(±0.0008)	3.492(±0.0001)	3.502(±0.0001)	9.578(±0.0003)	10.63(±0.0040)	4.099(±0.0001)	4.134(±0.0001)
	SoRec	6.346(±0.0004)	7.195(±0.0003)	3.560(±0.0001)	3.573(±0.0001)	8.731(±0.0006)	10.37(±0.0019)	4.390(±0.0001)	4.403(±0.0001)
	PMF	6.248(±0.0028)	6.922(±0.0112)	**9.859(±0.0001)**	**9.732(±0.0021)**	5.622(±0.0005)	6.329(±0.0029)	**13.51(±0.0001)**	**13.40(±0.0073)**
	SoMF	**8.827(±0.0001)**	**9.219(±0.0046)**	4.660(±0.0001)	3.653(±0.0001)	9.427(±0.0004)	10.97(±0.0001)	5.490(±0.0017)	5.188(±0.0012)
	CUNE	6.280(±0.0006)	7.077(±0.0006)	3.558(±0.0001)	3.597(±0.0001)	6.410(±0.0001)	6.998(±0.0001)	4.210(±0.0001)	4.208(±0.0002)
	GraphRec	7.424(±0.0001)	7.001(±0.0002)	4.898(±0.0001)	4.286(±0.0001)	11.17(±0.0003)	10.50(±0.0004)	11.77(±0.0015)	13.32(±0.0001)
	ConsisRec	6.560(±0.0012)	6.943(±0.0009)	5.948(±0.0020)	5.826(±0.0034)	**11.70(±0.0001)**	**12.42(±0.0013)**	12.04(±0.0017)	11.98(±0.0008)
w/ DC (for RQ2)	SoReg++	6.712(±0.0003)	6.993(±0.0008)	3.936(±0.0003)	3.876(±0.0005)	9.246(±0.0003)	10.96(±0.0004)	6.710(±0.0001)	7.279(±0.0002)
	SoRec++	6.346(±0.0001)	7.147(±0.0002)	5.285(±0.0006)	5.515(±0.0009)	9.338(±0.0032)	10.24(±0.0036)	8.590(±0.0003)	9.620(±0.0005)
	PMF++	6.959(±0.0033)	8.171(±0.0047)	11.14(±0.0010)	13.27(±0.0020)	6.09(±0.0049)	7.199(±0.0059)	14.49(±0.0004)	17.52(±0.0009)
	SoMF++	9.125(±0.0029)	10.79(±0.0038)	4.465(±0.0003)	45.23(±0.0005)	8.337(±0.0026)	6.824(±0.0025)	8.550(±0.0036)	9.601(±0.0060)
	CUNE++	6.155(±0.0006)	6.928(±0.0006)	5.449(±0.0008)	5.328(±0.0015)	6.059(±0.0026)	6.824±0.0025)	7.575(±0.0023)	8.290(±0.0040)
	GraphRec++	8.183(±0.0002)	13.63(±0.0003)	7.261(±0.0080)	7.920(±0.0134)	12.78(±0.0083)	14.73(±0.0035)	24.51(±0.0001)	24.88(±0.0001)
	ConsisRec++	11.33(±0.0002)	13.63(±0.0002)	11.34(±0.0077)	13.60(±0.0134)	19.75(±0.0002)	23.83(±0.0001)	**24.54(±0.0001)**	24.25(±0.0001)
	SDCRec	**11.79(±0.0062)**	**14.36(±0.0137)**	**11.50(±0.0012)**	**14.02(±0.0023)**	**20.58(±0.0017)**	**24.01(±0.0028)**	24.43(±0.0033)	**26.13(±0.0060)**

all users $\forall\, u \in \mathcal{U}$ that optimize both branches jointly

$$\mathcal{L} = \lambda\, \underbrace{\mathbb{E}_{u\in\mathcal{U}} \left[-\log \frac{\rho(y^+_{u,t})}{\rho(y^+_{u,t}) + \gamma \sum_{c_{u,t}\in y^-_{u,t}} \rho(c_{u,t})} \right]}_{\text{User–Item Contrastive InfoNCE Loss}} + (1-\lambda)\, \underbrace{\mathbb{E}_{u\in\mathcal{U}} \left[-\log \frac{\rho(y^+_{t,t})}{\rho(y^+_{u,t}) + \gamma \sum_{c_{t,t}\in y^-_{t,t}} \rho(c_{t,t})} \right]}_{\text{Item–Item Contrastive InfoNCE Loss}} + \|\Theta\|_2 \tag{6}$$

with

$$\rho(\mathbf{a},\mathbf{b}) = \frac{1}{\tau} \exp \frac{\psi(\mathbf{a}) \cdot \psi(\mathbf{b})}{\|\psi(\mathbf{a})\| \cdot \|\psi(\mathbf{b})\|} \tag{7}$$

where $\psi(\cdot)$ denotes non-linear projection head, τ is the temparature coefficient, λ balances the tradeoff between two branches, $\|\Theta\|_2$ is L2-regularization of model f's parameters.

4 EXPERIMENTS

We present empirical studies that examine the following research questions:

RQ 1) Does the proposed SDCRec outperform other social recommendation models under both warm and cold-start settings?

RQ 2) Can the introduced dual-branch contrastive objective benefit a broad spectrum of social recommendation models?

RQ 3) Is there any relationship between various hyper-parameters and the model's performance?

4.1 Experimental Setup

We conduct evaluations on two datasets, Ciao[1] and Epinions[2]. We perform experiments under both warm and cold-start settings. In warm scenarios, the train/test split is on the whole dataset in a 4:1 ratio regarding user-item interactions. A cold-start scenario is simulated by randomly partitioning 20% of users as cold users with their interaction records shielded but only access to their social relations, while others remain same as in warm settings.

SDCRec is compared with seven baselines for evaluation, including MF-based methods (PMF [14], SoRec [12], SoReg [13], SoMF [9]), non-GNN graph embedding method (CUNE [22]), and GNN-based methods (GraphRec [5], ConsisRec [21]).

4.1.1 Implementation Details. All the baseline models as well as the proposed approach are implemented with PyTorch[3]. The implementation of MF-based methods are based on [23] and are adapted to the design of our experiment scenarios. Adam optimizer is used for training. The embedding size of user and item is 64, batch size is 32, learning rate is 0.0003. For hyperparameters, The number of negative samples is searched in 4, 8, 16, 32, 64. The size of projection head is searched in 16, 32, 64, 128. The ratio of item-item contrastive loss and user-item contrastive loss vary from 0 to 1. Unbiased negative sampling percentage range from 0.1 to 0. To prevent overfitting, early-stopping was applied to SDCRec and all comparison methods.

4.2 Experimental Results

4.2.1 Overall comparison. We summarize the best top-k results in terms of Normalized Discounted Cumulative Gain@10 (N@10) and Mean Reciprocal Rank (M@10), shown by Tab 1. In both warm and cold-start scenarios, the proposed SDCRec outperforms all other baselines with a significant performance gain for two datasets (best results are marked in **bold**). When compared with other vanilla models, SDCRec outperforms its first runner-up counterpart (indicated in **underlined bold**) by 78% in warm settings and 64% on average in cold settings. The second-best results across all experiments are marked by underline. The performances of all models surge after contrastive learning are introduced (e.g., GraphRec++).

4.2.2 Ablation study. We perform ablation studies to evaluate each component of SDCRec, which consists of four variants: SDCRec(w/o

[1]https://www.cse.msu.edu/~tangjili/datasetcode/epinions.zip
[2]https://www.cse.msu.edu/~tangjili/datasetcode/ciao.zip
[3]https://www.pytorch.org

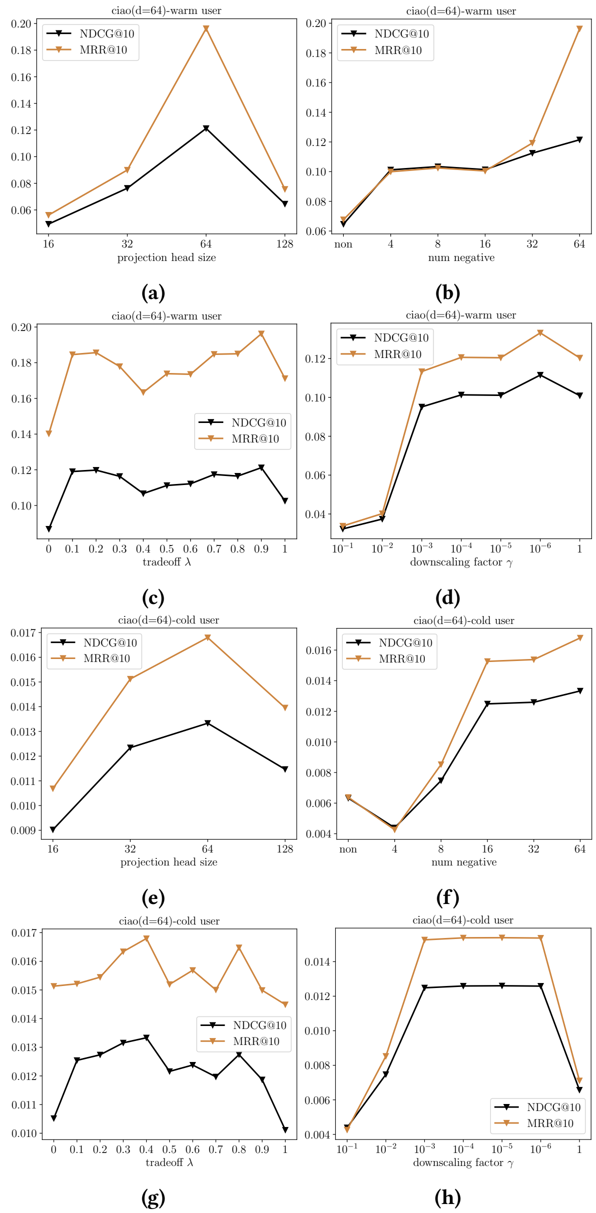

Figure 2: Parameter sensitivity.

Table 2: Ablation study

		SDCRec (w/o DC)	SDCRec (w/o tt)	SDCRec (w/o ut)	SDCRec (w/o γ)	SDCRec
warm	N@10	0.0646	<u>0.1025</u>	0.0868	0.1009	**0.1215**
	M@10	0.0676	<u>0.1712</u>	0.1402	0.1203	**0.1962**
cold	N@10	0.0063	0.0101	<u>0.0105</u>	0.0066	**0.0126**
	M@10	0.0064	0.0145	<u>0.0151</u>	0.0071	**0.0154**

DC) that implements dynamic embedding with no contrastive learning; SDCRec(w/o tt) that removes the Item-Item contrastive branch; SDCRec(w/o ut) that removes the User-Item contrastive branch; and SDCRec(w/o γ) that disregards γ in the negative sampling terms from both User-Item branch and Item-Item branch.

We can only depict the experimental results on the Ciao dataset in Table 2 due to page limitation. Among comparisons, SDCRec(w/o DC) yields the lowest results, owing to the absence of contrastive learning. Adding either contrastive branch substantially enhances the performance. Both User-Item branch and Item-Item branch serve critical roles in learning user/item representations, however, the absence of the Item-Item branch has a greater impact in cold-start situations. Incorporating the debiasing factor γ further improves contrastive learning. In particular, excluding γ may hamper cold-start scenarios. As a result, each component of SDCRec contributes positively to determining the user/item representations. SDCRec with all components assembled reports the highest results.

4.2.3 Parameter sensitivity. Further tests are conducted, including down-scaling multiplier γ, the trade-off coefficient λ, and projection head size d. Fig 2 shows the results of the sensitivity test. Using $\lambda = 0.9$ in warm recommendations leads to the highest results, whereas setting it to 0.4 brings about the best result in cold-start recommendations. Besides, γ assumes the proportion of false-negative samples that would appear during negative sampling. Observations indicate that the peak performance occurs when γ is 10^{-6} in warm recommendation and 10^{-5} in cold recommendation, respectively. In addition, the quality of representation in contrastive learning is sensible to different hidden size of the projection head [3, 4]. Studies about contrastive learning often compare the effects of different projection heads. Our experiments suggest that 64 is the ideal representation dimension for both warm and cold-start recommendations.

5 CONCLUSION

In this work, we present a contrastive representation learning framework for cold-start recommendation with social relations considered. By casting item similarity as part of the self-supervised optimization objective, we combat *popularity bias* encountered by supervised social recommendation models without requirement of extra data. We also extend previous contrastive recommendation models to include social relations when coping with cold-start problem. Experiments verify that such contrastive structure could give rise to any social recommendations. We plan on generalizing this framework to sequential recommendation in future works.

6 ACKNOWLEDGMENTS

We would like to thank the reviewers for their comments. Jing Du is supported by Chinese Scholarship Council(CSC) under grant (No. 202006290008).

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing popularity bias in recommender systems with personalized re-ranking. In *The thirty-second international flairs conference*.
[2] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).
[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.
[4] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297* (2020).
[5] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 417–426. https://doi.org/10.1145/3308558.3313488

[6] Ido Guy. 2015. Social recommender systems. In *Recommender systems handbook*. Springer, 511–543.
[7] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 1025–1035.
[8] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670* (2018).
[9] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*. 135–142.
[10] Fangquan Lin, Wei Jiang, Jihai Zhang, and Cheng Yang. 2021. Dynamic Popularity-Aware Contrastive Learning for Recommendation. In *Asian Conference on Machine Learning*. PMLR, 964–968.
[11] Zhuang Liu, Yunpu Ma, Yuanxin Ouyang, and Zhang Xiong. 2021. Contrastive learning for recommender system. *arXiv preprint arXiv:2101.01317* (2021).
[12] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. 2008. SoRec: Social Recommendation Using Probabilistic Matrix Factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (Napa Valley, California, USA) *(CIKM '08)*. Association for Computing Machinery, New York, NY, USA, 931–940. https://doi.org/10.1145/1458082.1458205
[13] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 287–296.
[14] Andriy Mnih and Russ R Salakhutdinov. 2008. Probabilistic Matrix Factorization. In *Advances in Neural Information Processing Systems*, J. Platt, D. Koller, Y. Singer, and S. Roweis (Eds.), Vol. 20. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2007/file/d7322ed717dedf1eb4e6e52a37ea7bcd-Paper.pdf
[15] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).
[16] Yuqi Qin, Pengfei Wang, and Chenliang Li. 2021. *The World is Binary: Contrastive Learning for Denoising Next Basket Recommendation*. Association for Computing Machinery, New York, NY, USA, 859–868. https://doi.org/10.1145/3404835.3462836
[17] Changhao Song, Bo Wang, Qinxue Jiang, Yehua Zhang, Ruifang He, and Yuexian Hou. 2021. Social Recommendation with Implicit Social Influence. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1788–1792.
[18] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. 2006. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 501–508.
[19] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive Learning for Cold-Start Recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia* (Virtual Event, China) *(MM '21)*. Association for Computing Machinery, New York, NY, USA, 5382–5390. https://doi.org/10.1145/3474085.3475665
[20] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 235–244.
[21] Liangwei Yang, Zhiwei Liu, Yingtong Dou, Jing Ma, and Philip S. Yu. 2021. ConsisRec: Enhancing GNN for Social Recommendation via Consistent Neighbor Aggregation. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021).
[22] Chuxu Zhang, Lu Yu, Yan Wang, Chirag Shah, and Xiangliang Zhang. 2017. Collaborative user network embedding for social recommender systems. In *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 381–389.
[23] Honglei Zhang, Gangdu Liu, and Jun Wu. 2018. Social Collaborative Filtering Ensemble. In *PRICAI 2018: Trends in Artificial Intelligence*. 1005–1017.
[24] Muhan Zhang and Yixin Chen. 2020. Inductive Matrix Completion Based on Graph Neural Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=ByxxgCEYDS