SGCCL: Siamese Graph Contrastive Consensus Learning for Personalized Recommendation

Boyu Li
University of Technology Sydney
Sydney, NSW, Australia
boyu.li@student.uts.edu.au

Ting Guo*
University of Technology Sydney
Sydney, NSW, Australia
ting.guo@uts.edu.au

Xingquan Zhu
Florida Atlantic University
Boca Raton, Florida, United States
xzhu3@fau.edu

Qian Li
Curtin University
Perth, WA, Australia
qli@curtin.edu.au

Yang Wang
University of Technology Sydney
Sydney, NSW, Australia
yang.Wang@uts.edu.au

Fang Chen
University of Technology Sydney
Sydney, NSW, Australia
fang.chen@uts.edu.au

ABSTRACT

Contrastive-learning-based neural networks have recently been introduced to recommender systems, due to their unique advantage of injecting collaborative signals to model deep representations, and the self-supervision nature in the learning process. Existing contrastive learning methods for recommendations are mainly proposed through introducing augmentations to the user-item (U-I) bipartite graphs. Such a contrastive learning process, however, is susceptible to bias towards popular items and users, because higher-degree users/items are subject to more augmentations and their correlations are more captured. In this paper, we advocate a **S**iamese **G**raph **C**ontrastive **C**onsensus **L**earning (SGCCL) framework, to explore intrinsic correlations and alleviate the bias effects for personalized recommendation. Instead of augmenting original U-I networks, we introduce siamese graphs, which are homogeneous relations of user-user (U-U) similarity and item-item (I-I) correlations. A contrastive consensus optimization process is also adopted to learn effective features for user-item ratings, user-user similarity, and item-item correlation. Finally, we employ the self-supervised learning coupled with the siamese item-item/user-user graph relationships, which ensures unpopular users/items are well preserved in the embedding space. Different from existing studies, SGCCL performs well on both overall and debiasing recommendation tasks resulting in a balanced recommender. Experiments on four benchmark datasets demonstrate that SGCCL outperforms state-of-the-art methods with higher accuracy and greater long-tail item/user exposure.

CCS CONCEPTS

• **Information systems → Recommender systems**; **Collaborative filtering**; **Social recommendation**.

*Corresponding Author

WSDM '23, February 27–March 3, 2023, Singapore, Singapore.

ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570422

KEYWORDS

Graph Contrastive Learning, Recommender System, Consensus Learning, Popularity Bias

ACM Reference Format:
Boyu Li, Ting Guo, Xingquan Zhu, Qian Li, Yang Wang, and Fang Chen. 2023. SGCCL: Siamese Graph Contrastive Consensus Learning for Personalized Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27–March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570422

1 INTRODUCTION

Recommender system, as one of the most sought-after applications, is tempting to recommend items interesting to potential users, which has been widely used to help users discover contents and alleviate information overload in the era of information explosion [25]. Recently, graph learning-based recommender system becomes an emerging topic in this field that utilizes advanced graph learning to model users' preferences and intentions for personalized recommendation [35]. Neural Graph Collaborative Filtering framework (NGCF) [29] integrates user-item interactions as a bipartite graph, *e.g.* the left panel in Fig. 1, and adopts Graph Convolutional Networks (GCNs) to propagate information among neighbors to learn embedding features for recommendation. Later, LightGCN was proposed [13] to simplify the GCN, as feature transformation and nonlinear activation in GCNs are found to have limited contribution to collaborative filtering [30]. However, observed purchase behaviors may be caused by one-off consumption or impulse spending, which usually contain noise and misleading information. The message passing scheme rooted in GCNs tends to magnify the impact of interactions on representation learning, making the learning process inherently vulnerable to interaction noise.

Graph Contrastive Learning (GCL) has received considerable attention because it can leverage self-supervised learning to alleviate noise disturbance in purchase behavior and improve recommendation robustness [22, 32]. The theme of GCL-based recommendation systems is to apply the graph augmentation strategies on user-item bipartite graphs, as shown in the upper panel in Fig. 2, and then maximize the agreement between different views of the same node and the disagreement among different nodes [31]. To further alleviate the selection bias in graph augmentation, debiased contrastive loss is also proposed to provide sufficient negative samples and applies

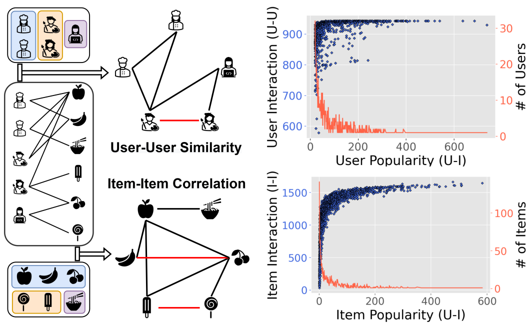

Figure 1: A toy example of user-item bipartite graph (left panel), the constructed siamese graphs (middle panel), and popularity bias in the ML-100K dataset (right panel). In the right panel, X-axis denotes the user/item popularity (node degrees) in user-item (U-I) bipartite graph. Item interactions are the number of items appeared together with a target item in the whole dataset (*i.e.* shopping mix). User interactions denote number of users selected one or more items same as a target user. Red item/user popularity curves follow scale-free distributions, meaning very few items/users have high degrees. Using item-item or user-user interactions, unpopular items/users show lifted interactions (blue dots) which will increase their exposure in personalized recommendation.

a bias correction probability to alleviate the sample bias [20]. Nevertheless, existing GCL methods focus on the user-item bipartite graph and neglect the direct-neighboring correlations among users (or items), which fall short of utilizing the potential of contrastive learning for recommendation.

Indeed, one of the outstanding problems in recommender systems is the *popularity bias*: from the data perspective, items always exhibit long-tail distributions on the user-item interactions due to the Pareto principle (*i.e.* 80-20 rule) [1], as shown in Fig. 1; from the modeling perspective, recommender systems tend to amplify the bias by over-recommending popular items, leading to the phenomenon that popular items receive a lot of exposure while the majority of other (less popular) items receive very little attention and hard to be recommended to users [40]. This is because high-order connectivity is built up based on the relationship between heterogeneous nodes of user-item bipartite graph, causing any two adjacent users sharing similar consumption habits are at least two-hop away (user-item-user). The indirect connection between homogeneous nodes (user-user/item-item) hinders the neighbor-aggregation-based models from exploring less popular preferences from homogeneous neighbors. Despite several studies adopted homogeneous graphs to model the user-user and item-item relationships separately, these methods all forcibly fuse the node embeddings to reconstruct the user-item bipartite graph for final recommendation. However, the nature of spatial inconsistency of different graphs makes these algorithms uneasy to converge and hard to balance different relationships [26, 27]. Meanwhile, recent studies have attempted to alleviate the bias problem by emphasizing the long-tail items/users in the recommender training: either to downweight the influence of popular items or to form a causal graph that takes cause-effect into account [24, 37]. However, these debiasing methods may cause another type of "bias": The deliberate operation of impact reduction

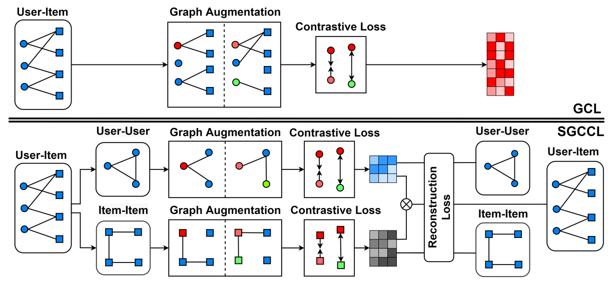

Figure 2: GCL *vs.* SGCCL. Existing GCL methods (upper panel) focus on user-item bipartite graph for representation learning. SGCCL (lower panel), introduces siamese (user-user and item-item) graphs into graph contrastive learning. A consensus loss is used to enforce that learned representation can reconstruct network structures from multiple views.

on popular items contradicts actual observations and will reversely downplay representation learning for popular users/items.

To remedy popularity bias, we introduce the *Siamese graphs* with homogeneous relations: user-user similarity and item-item correlation. Siamese graphs can be extracted from side information and historical interactions to boost recommendation performance. A unique advantage of Siamese graphs, compared to user-item bipartite graph, is that they can directly capture and model homogeneous relationships between user-user and item-item. Particularly, different from bipartite graph, two uncommon items sharing some attributes (*e.g.* same genres of movies) or being preferred by the same user group will form a connection in the item-item graph. This relation, hardly observable in the user-item graph, helps capture unique patterns shared by a group of items or the unexpected retail correlation of products (*e.g.* the classic case study of "beer and diapers" in marketing). Likewise for users, their distinctive preferences of unpopular items can be easily propagated through user-user connections if there is high similarity on their historical behaviors. As shown in Fig. 1 (right panel), the majority of long-tail unpopular items also have high active item interactions which can help increase their exposure and reduce popularity bias.

Instead of pure exploration of user-item bipartite graph in existing GCL methods, siamese graphs make it possible to establish a novel consensus learning principle integrating both homogeneous (user-user and item-item) and heterogeneous (user-item) interactions into a unified learning objective as shown in Fig. 2. Different from only exploring information from homogeneous graphs, the unified use of three views of graphs can greatly ensure the harmony of embeddings and the consistency of the final recommendation. And benefits from the multi-view information acquisition, our model can learn the nodes embedding which can handle both the overall recommendation and popularity bias eliminating, thus forming a balanced recommender. In addition, we put forward a contrastive consensus learning framework for personalized recommendation. The idea is to adopt a self-supervised learning scheme to pre-train the user and item representations in parallel based on the siamese graphs, and then reinforce the learned embeddings by minimizing the reconstruction losses of homogeneous and heterogeneous graphs with a consensus learning manner. A dynamic weighting strategy is adopted to consider each component in iterations. The key contributions of the paper are given as below:

- *Balanced Recommendation*: Instead of sacrificing the accuracy of popular items to reduce the popularity bias, like existing methods do, SGCCL is the first work that takes three types of relationships to generate consensus embedding and achieve high accuracy and balanced long-tail item/user coverage. Different from existing studies which always focus on either general performance or popularity bias, SGCCL performs well on both overall and debiasing recommendation tasks resulting in a balanced recommender.
- *Multi-view fusion*: We construct siamese graphs from homogeneous relations and propose a factorization-based consensus learning principle to explore multi-relationships from users and items simultaneously for personalized recommendation. Different from the state-of-the-art approaches focus on user-item bipartite graph for recommendation, our model can fully explore homogeneity relations with a multi-view fusion.
- *Siamese Contrastive Consensus Learning*: We propose a consensus learning principle, along with a graph learning model, to contrastively explore siamese graphs in a self-supervised manner and enforce the reconstruction losses from pre-trained embeddings to homogeneous and heterogeneous relations. Our contrastive consensus learning approach can be extended to many other applications with a rich set of relationships in the data.

2 CONSENSUS LEARNING PRINCIPLE

In most general settings, a standard recommender system has two sets of entities: a user set $\mathcal{U}$ with m users ($|\mathcal{U}| = m$) and an item set $\mathcal{V}$ with n items ($|\mathcal{V}| = n$). User and item interactions are recorded in a rating matrix $\mathcal{R} \in \mathbb{R}^{m\times n}$, where $\mathcal{R}_{ij} = 1$ denotes that user i interacted with (*e.g.* purchased or liked) item i, or 0 otherwise. With the rating matrix, learning good feature embeddings to precisely represent users and items is a focus for accurate recommendation. With the increasing success of deep graph learning, researchers attempted to formulate recommendation as a user-item bipartite graph, and adapted GCNs for recommendation [4, 36].

As discussed above, user-item bipartite graph cannot fully reveal the direct connection between homogeneous nodes, where two adjacent users can only interact through their shared items. It severely restricts the target users from learning potential preferences from similar users with particular likes or purchases. To overcome the aforementioned limitation, we formulate a novel learning framework with the exploration of multi-relationships from users and items for better collaborative filtering, *i.e.*, *consensus learning*. To take special care of the interactions between homogeneous nodes, we introduce the **siamese graphs** constructed by considering homogeneous relations: user-user similarity and item-item correlation, in addition to the heterogeneous relations in the user-item ratings. In this section, we first advocate a factorization-based consensus learning principle to guide the learning process that can simultaneously explore the valuable interactions under three relationship settings and leverage them for boosting recommendation performance. This principle will be practiced using a novel graph learning scheme (SGCCL) introduced in the following sections.

2.1 Factorization of User-Item Ratings

The rating matrix $\mathcal{R} \in \mathbb{R}^{m\times n}$ provides tabular relationships between m users and n items. We can adopt Non-negative Matrix Factorization (NMF) to factorize $\mathcal{R}$ into two compressed matrices $\mathcal{G} \in \mathbb{R}^{m\times c}$ and $\mathcal{F} \in \mathbb{R}^{n\times d}$ with the objective of minimizing squared errors between $\mathcal{R}$ and its approximation,

$$\underset{\mathcal{G},\mathcal{F}}{\operatorname{argmin}}\, J_{U\cdot I} = ||\mathcal{R} - \mathcal{G}\mathcal{F}^\top||_F^2,\ \ s.t.\ \mathcal{G},\mathcal{F} \geq 0, \tag{1}$$

Where $||\cdot||_F^2$ is the Frobenius norm of the matrix [8]. In reality, because two-factor NMF in Eq. (1) is restrictive, in which the compressed dimensions c and d have to be equal, one can introduce an additional factor $\mathbf{S} \in \mathbb{R}^{c\times d}$ to absorb the different scales of $\mathcal{R}$, $\mathcal{G}$ and $\mathcal{F}$. This leads to an extension of NMF, named NMTF [7]

$$\underset{\mathcal{G},\mathcal{F}}{\operatorname{argmin}}\, J'_{U\cdot I} = ||\mathcal{R} - \mathcal{G}\mathbf{S}\mathcal{F}^\top||_F^2,\ \ s.t.\ \mathcal{G},\mathcal{F} \geq 0, \tag{2}$$

In Eq. (2), $\mathbf{S}$ provides increased degrees of freedom such that the low-rank matrix representation remains accurate, while c and d can have different values. The optimized $\mathcal{G}$ and $\mathcal{F}$ by applying Eq. (2) can be considered as the learned embedding matrices based on User-Item ratings for users and items respectively, and this objective function is consistent with MF-based collaborative filtering [21].

In recommendation, it is important to directly characterize user-user and item-item relationships, such that we can learn embedding to ensure users sharing similar items are close to each other in the embedding space, regardless of whether the shared items are popular or not. Therefore, we introduce siamese graphs, *user-user similarity graph* and *item-item correlation graph*, to explore the distinctive features from homogeneous relations. In practice terms, the user-user similarity graph can be constructed based on users' attributes, like users' social states. Meanwhile, the item-item correlation graph can be derived from items' properties, such as their commodity use and manufacturers. This information can provide auxiliary support to comparability learning and performance-boosting. The construction of homogeneous graphs will be detailed in Sec. 3.

2.2 Factorization of User-User Similarity

User-user similarity graph $\Theta_U =< \mathcal{U}, \mathcal{A}_U, \mathcal{X}_U >$ contains pairwise user relations in the structure space. It provides information to characterize similarities between users for boosting preference learning. Thus, we can factorize the adjacency matrix $\mathcal{A}_U$ as an $m \times d$ matrix $\mathcal{G}_U$ which is the embedding matrix showing potential similarity of users by only considering User-User interactions:

$$\underset{\mathcal{G}_U}{\operatorname{argmin}}\, J_{U\cdot U} = ||\mathcal{A}_U - \mathcal{G}_U\mathcal{G}_U^\top||_F^2,\ \ s.t.\ \mathcal{G}_U \geq 0, \tag{3}$$

It is noteworthy that $\mathcal{G} \in \mathbb{R}^{m\times d}$ in $J_{U\cdot I}$ and $\mathcal{G}_U \in \mathbb{R}^{m\times d}$ in $J_{U\cdot U}$ each contains separated factorization results for the user set $\mathcal{U}$. By using this approach, we allow factorization for $\mathcal{R}$ and $\mathcal{A}_U$ to have maximum freedom to explore its optimal results, respectively.

2.3 Factorization of Item-Item Correlation

To enhance item embedding results, we also use $\Theta_I =< \mathcal{V}, \mathcal{A}_I, \mathcal{X}_I >$ to capture pairwise item correlations. Intuitively, if items v_i and v_j are highly correlated, they should be more likely being picked up together in the future. Similar to Eq. (3), the factorization of item correlations $\mathcal{A}_I$ is as follows,

$$\underset{\mathcal{F}_I}{\operatorname{argmin}}\, J_{I\cdot I} = ||\mathcal{A}_I - \mathcal{F}_I\mathcal{F}_I^\top||_F^2,\ \ s.t.\ \mathcal{F}_I \geq 0, \tag{4}$$

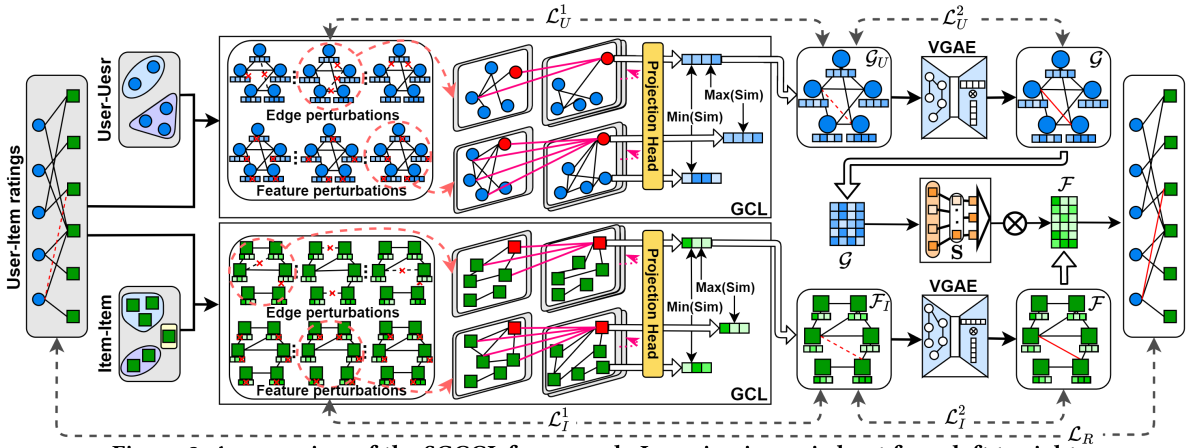

Figure 3: An overview of the SGCCL framework. Learning is carried out from left to right.

2.4 Consensus Factorization

In the above factorization processes, the objective functions $J'_{U \cdot I}$, $J_{U \cdot U}$ and $J_{I \cdot I}$ each provides embedding results from different aspects (user-item ratings, user-user similarity, and item-item correlation). To ensure that final results are consistent, we propose a consensus factorization objective function to jointly formulate $J'_{U \cdot I}$, $J_{U \cdot U}$ and $J_{I \cdot I}$ into a unified objective:

$$\begin{aligned} J =& ||\mathcal{R} - \mathcal{G}\mathbf{S}\mathcal{F}^\top||_F^2 + \alpha||\mathcal{A}_U - \mathcal{G}_U\mathcal{G}_U^\top||_F^2 + \beta||\mathcal{A}_I - \mathcal{F}_I\mathcal{F}_I^\top||_F^2 \\ &+ \rho(||\mathcal{G} - \mathcal{G}_U||_F^2 + ||\mathcal{F} - \mathcal{F}_I||_F^2), \\ & s.t.\ \mathcal{G} \geq 0, \mathcal{F} \geq 0, \mathcal{G}_U \geq 0, and\ \mathcal{F}_I \geq 0 \end{aligned} \tag{5}$$

The objective function in Eq. (5) is to factorize $\mathcal{R}$, $\mathcal{A}_U$, and $\mathcal{A}_I$ separately, and enforce the factorization consensus among all three aspects. For instance, $\mathcal{G}$ and $\mathcal{G}_U$ provide embedding results from User-Item and User-User relations, respectively. $||\mathcal{G} - \mathcal{G}_U||_F^2$ enforces that $\mathcal{G}$ should be maximally consistent with $\mathcal{G}_U$. Similarly, $||\mathcal{F} - \mathcal{F}_I||_F^2$ makes $\mathcal{F}$ and $\mathcal{F}_I$ close to each other. α and β in Eq. (5) are regularization parameters to balance each factorization part. ρ trade-offs the consistent degree. Intuitively, a very large ρ value will make $\mathcal{G} = \mathcal{G}_U$ and $\mathcal{F} = \mathcal{F}_I$, while a small ρ would make $\mathcal{G}$ and $\mathcal{G}_U$ totally independent (*e.g.*, $\rho = 0$). As a result, the designed objective function provides increased degrees of freedom to exploit different information encoded in multi-relationships.

Latent matrix $\mathbf{S}$ not only absorbs the different scales of $\mathcal{R}$, $\mathcal{F}$ and $\mathcal{G}$, but also reveals the corresponding relationships between the user and item embedding results. $\mathbf{S}_{ij}$ uncovers the relative weight between item embedding feature i and user embedding feature j.

3 SIAMESE GRAPH CONTRASTIVE CONSENSUS LEARNING

Minimizing Eq. (5) is respect to $\mathcal{G}$, $\mathcal{F}$, $\mathcal{G}_U$, $\mathcal{F}_I$ and $\mathbf{S}$, and the function is not convex in all variables together. Traditionally, this kind of Matrix Factorization (MF) task is always optimized by using Stochastic Gradient Descent (SGD) or Alternating Least Squares (ALS) algorithm [6, 38]. The objectives can be optimized *w.r.t* one variable while fixing the other variables. This procedure repeats until convergence. However, existing MF methods are not suitable for consensus factorization. The limitation of these MF methods are multi-fold: (1) The use of a simple and fixed inner product is hard for the estimation of complex node interactions in the low-dimensional latent space [14]. (2) Lack of the expressive modeling of high-order connectivity in graph-based representations which can effectively inject the collaborative signal into the embedding process in an explicit manner. The embedding propagation among similar users/items is important for learning better user and item representations [29]. (3) The introduction of multiplex sub-objects and optimization constraints in Eq. (5) makes it a tremendous hassle using matrix factorization methods, not to mention letting the algorithm converge to the global optimum.

In this work, we propose **SGCCL** which is a novel graph learning scheme followed the consensus learning principle. With the consideration of homogeneous relations, factorized embeddings ($\mathcal{G}_U$ and $\mathcal{F}_I$) are generated respectively by applying siamese graph contrastive learning scheme on user-user similarity graph Θ_U and item-item correlation graph Θ_I for robust pre-training. A nonlinear transformation of $\mathcal{G}_U$ and $\mathcal{F}_I$ is then applied to reconstruct the User-Item ratings ($\mathcal{R}$) for consensus optimization. With delicate design, the portfolio of the loss functions from different components is nicely consistent with J in Eq. (5) and the overall framework is as shown in Fig. 3.

3.1 Siamese Graph Generation

Homogeneous relation is the key concept introduced to boost recommendation performance and alleviate the popularity bias in this paper. The siamese graphs (Θ_U and Θ_I) can be generated from side information. Take movie recommendation network as an example [10], the movies contain different genres, and the spectators can be classified based on gender, age, and occupation. While for the Amazon datasets [11], the audiences' reviews can be considered as the side information for relation construction. By given the feature embedding of users/items in side information space $\mathcal{E} = \{\mathcal{E}_U, \mathcal{E}_I\}$, the adjacency matrices of siamese graphs can be formulated as:

$$[\mathcal{A}_U^S]_{ij} = \begin{cases} 1, & \mathcal{E}_{Ui}\mathcal{E}_{Uj}^\top \geq \delta_{\mathrm{U}} \\ 0, & \text{otherwise} \end{cases}; \ [\mathcal{A}_I^S]_{ij} = \begin{cases} 1, & \mathcal{E}_{Ii}\mathcal{E}_{Ij}^\top \geq \delta_{\mathrm{I}} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where δ refers to the threshold that controls the degree of graph connectivity. At the same time, homogeneous relations can also be extracted from user-item rating, as user preference and the audience for items show strong homogeneity in nature. Specifically,

$$[\mathcal{A}_U^R]_{ij} = \begin{cases} 1, & \mathcal{R}_{i\cdot}\mathcal{R}_{j\cdot}^\top \geq \delta_U \\ 0, & \text{otherwise} \end{cases};\ [\mathcal{A}_I^R]_{ij} = \begin{cases} 1, & \mathcal{R}_{\cdot i}^\top \mathcal{R}_{\cdot j} \geq \delta_I \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

As user preferences and shopping mix can be considered as a kind of side information which can be extracted from User-Item rating, we integrate them into the siamese graph generation in practice when side information is available. Specifically, $\mathcal{A}_U = \mathcal{A}_U^S \cup \mathcal{A}_U^R$ and $\mathcal{A}_U = \mathcal{A}_I^S \cup \mathcal{A}_I^R$, where "∪" is element-wise union operation. The node attribute matrices are formulated as $\mathcal{X}_U = \mathcal{E}_U || \mathcal{R}$ and $\mathcal{X}_I = \mathcal{E}_I || \mathcal{R}^\top$, in which "||" is a concatenation operator.

3.2 Siamese Contrastive Learning with Graph Augmentation

To learn user/item embeddings from the siamese graphs (Θ_U and Θ_I), we propose a siamese contrastive learning paradigm, which reinforces user/item representation learning from siamese graphs via self-discrimination as shown in Fig. 3. Specifically, we introduce how to perform graph augmentation that generates multiple representation views, followed by the contrastive learning based on the generated graph representations to build the adversarial task.

3.2.1 Graph Augmentation. To perform contrastive learning on siamese graphs, we conduct graph augmentation to generate a set of augmented graphs from the original graphs as the candidate space for further representation learning. We hence devise two operators on the graph data, edge perturbation and feature perturbation, to create different views of the graph.

Edge Perturbation: It drops out the edges in graph with a perturbation ratio γ. The augmentation processes for user-user and item-item graphs are represented as:

$$\mathrm{EP}(\Theta_U) =< \mathcal{U}, M_U \odot \mathcal{A}_U, \mathcal{X}_U >;\ \mathrm{EP}(\Theta_I) =< \mathcal{V}, M_I \odot \mathcal{A}_I, \mathcal{X}_I > \tag{8}$$

where $[M_U]_{ij} \in \{0,1\}$ and $[M_I]_{ij} \in \{0,1\}$ are two masking symmetrical matrices on adjacency matrices of user-user and item-item graphs respectively. With the random generation of M_U and M_I, we can augment the candidate space with edge perturbation.

Feature Perturbation: With the probability γ, we randomly drop out features from the nodes:

$$\mathrm{FP}(\Theta_U) =< \mathcal{U}, \mathcal{A}_U, N_U \odot \mathcal{X}_U >;\ \mathrm{FP}(\Theta_I) =< \mathcal{V}, \mathcal{A}_I, N_I \odot \mathcal{X}_I > \tag{9}$$

where $[N_U]_{ij} \in \{0,1\}$ and $[N_I]_{ij} \in \{0,1\}$ are two masking matrices on node attributes of user-user and item-item graphs respectively. With the random generation of N_U and N_I, we can augment the candidate space with feature perturbation.

We apply these augmentations on siamese graphs to generate the candidate space. For each training epoch, we randomly select two different views (with different perturbation operations as shown in Fig. 3) of each node at the beginning. For each node, the coupling of these two augmented graphs aims to capture useful patterns of both local structures and attributes, and further endows the representations with more robustness against noisy interactions.

3.2.2 Contrastive Learning. After expanding the candidate space with the augmented views of siamese graphs, we employ a discriminator to distinguish the deep representations of the same nodes from different augmented graphs with the aim to learn robust semantic information. For user-user similarity graph, we adopt an L-layer GCN with the following layer-wise propagation rule:

$$H_U^{(l+1)} = \sigma\left(\tilde{D}_U^{-\frac{1}{2}} \tilde{\mathcal{A}}_U \tilde{D}_U^{-\frac{1}{2}} H_U^l \mathbf{W}_U^{(l)}\right) \tag{10}$$

Here, $\tilde{\mathcal{A}}_U$ is the normalized adjacency matrix of $EP(\Theta_U)$ or $FP(\Theta_U)$ with added self-connections. $\tilde{D}_{ii} = \sum_j \tilde{\mathcal{A}}_{ij}$, and $\mathbf{W}^{(l)}$ is a layer-specific trainable weight matrix. To be consistent with consensus factorization introduced in Eq. (3), $\sigma(\cdot)$ is an non-negative activation function, such as the $\mathbf{ReLU}(\cdot) = max(0, \cdot)$. $H^{(l)} \in \mathbb{R}^{m \times k}$ is the embedding matrix in the l^{th} layer; $H^{(0)} = \mathcal{X}_U$ which refers to the node attribute. Following this, a projection head is designed to eliminate some noisy information [5]. Therefore, the learned embeddings $\mathcal{G}_U^{EP}$ and $\mathcal{G}_U^{FP}$ by using the shared GCN component can capture the robust homogeneous relations through iterative training. It is worth noting that the GCN components could be any GNN model, which remains a lot of potential for expansion.

The aim of contrastive learning is to discriminate representations by contrasting positive and negative pairs. For positive pairs, we treat two representations (views) (i.e., $[\mathcal{G}_U^{EP}]_{i\cdot}$ and $[\mathcal{G}_U^{FP}]_{i\cdot}$) of the same nodes as random variables and maximize their mutual information, which can enhance the consistency of the nodes for a better embedding quality. For negative pairs, learned embeddings of different nodes (i.e., $[\mathcal{G}_U^{EP}]_{i\cdot}$ and $[\mathcal{G}_U^{FP}]_{j\cdot}$) should be far away from each other by minimizing the mutual information. In practice, we adopt the InfoNCE [9] as the contrastive loss to maximize the agreement of the positive pairs and minimize the negative pairs:

$$[\mathcal{L}_U^1]^{cl} = \sum_{i \in \mathcal{U}} -\log \frac{\exp\left(sim\left([\mathcal{G}_U^{EP}]_{i\cdot}, [\mathcal{G}_U^{FP}]_{i\cdot}\right)/\tau\right)}{\sum_{j \in \mathcal{U}} \exp\left(sim([\mathcal{G}_U^{EP}]_{i\cdot}, [\mathcal{G}_U^{FP}]_{j\cdot})/\tau\right)} \tag{11}$$

Here, $sim(\cdot)$ refers to the cosine similarity function and τ is the temperature parameter in the softmax. Similarly, we can generate and boost the item embedding from Item-Item correlation graph Θ_I with the contrastive learning as:

$$H_I^{(l+1)} = \sigma\left(\tilde{D}_I^{-\frac{1}{2}} \tilde{\mathcal{A}}_I \tilde{D}_I^{-\frac{1}{2}} H_I^l \mathbf{W}_I^{(l)}\right), \tag{12}$$

$$[\mathcal{L}_I^1]^{cl} = \sum_{i \in \mathcal{V}} -\log \frac{\exp\left(sim\left([\mathcal{F}_I^{EP}]_{i\cdot}, [\mathcal{F}_I^{FP}]_{i\cdot}\right)/\tau\right)}{\sum_{j \in \mathcal{V}} \exp\left(sim([\mathcal{F}_I^{EP}]_{i\cdot}, [\mathcal{F}_I^{FP}]_{j\cdot})/\tau\right)} \tag{13}$$

The siamese GCN components are optimized iteratively in parallel by minimizing $[\mathcal{L}_U^1]^{cl}$ and $[\mathcal{L}_I^1]^{cl}$ with each training epoch. We set $\mathcal{G}_U = \mathcal{G}_U^{EP}$ and $\mathcal{F}_I = \mathcal{F}_I^{EP}$ as the updated embeddings fed to the following layers of SGCCL for consensus optimization.

3.3 Consensus Optimization

Under consensus learning setting, the user/item embeddings should consider both homogeneous and heterogeneous relations. From the perspective of graph learning, it means the learning goal is to minimize the reconstruction losses from the user/item embeddings to siamese graphs and the user-item bipartite graph, simultaneously.

3.3.1 Siamese Graph Reconstruction. The contrastive learning of $\mathcal{G}_U$ and $\mathcal{F}_I$ can be considered as a pre-training scheme for robust representation learning. To better explore homogeneous relations from a topology perspective, we aim to find the optimal embeddings by minimizing the reconstruction loss of siamese graphs,

$$[\mathcal{L}_U^1]^{re} = \left\|\mathcal{A}_U - \sigma\left(\mathcal{G}_U \mathcal{G}_U^\top\right)\right\|_F^2,\ [\mathcal{L}_I^1]^{re} = \left\|\mathcal{A}_I - \sigma\left(\mathcal{F}_I \mathcal{F}_I^\top\right)\right\|_F^2 \tag{14}$$

It is obvious that $[\mathcal{L}_U^1]^{re}$ and $[\mathcal{L}_I^1]^{re}$ are the non-linear variations to $J_{U \cdot U}$ (Eq. 3) and $J_{I \cdot I}$ (Eq. 4), respectively. By integrate the reconstruction loss into the contrastive loss in Eq. (11) and Eq. (13), we have the extended loss for siamese contrastive learning scheme,

$$\mathcal{L}_U^1 = [\mathcal{L}_U^1]^{cl} + [\mathcal{L}_U^1]^{re}, \ \mathcal{L}_I^1 = [\mathcal{L}_I^1]^{cl} + [\mathcal{L}_I^1]^{re} \tag{15}$$

3.3.2 Bipartite Graph Reconstruction. For the factorization of User-Item ratings, we aim to generate embedding matrices $\mathcal{G}$ and $\mathcal{F}$ that can reconstruct the rating matrix $\mathcal{R}$ with a minimal loss $J'_{U \cdot I}$. Traditionally, $J'_{U \cdot I}$ can be optimized by using SGD or ALS. However, This kind of MF methods does not apply to consensus learning as: (1) The collaborative signal, which is latent in local communities, is not encoded in the embedding process. (2) The embedding results by barely optimizing $J'_{U \cdot I}$ do not consider the relationships between homogeneous nodes.

Accordingly, we adopt Variational Graph Auto-Encoders (VGAEs) [16] on $\Theta'_U =< \mathcal{G}_U, \mathcal{A}_U >$ and $\Theta'_I =< \mathcal{F}_I, \mathcal{A}_I >$ to generate $\mathcal{G}$ and $\mathcal{F}$ in this work to consider the local communities and homogeneous relations. And at the same time, we force $\mathcal{G}$ and $\mathcal{F}$ to satisfy the objective function $J'_{U \cdot I}$ during the embedding process to take the User-Item heterogeneous relation into consideration.

For user perspective, VGAEs extended the variational auto-encoder framework to graph structure, which uses a probabilistic model involving latent variables g_i for each node $i \in \mathcal{U}$, interpreted as node representations in an embedding space [15]. The inference model, *i.e.* the encoding part of VAE, is defined as:

$$q(\mathcal{G}|\mathcal{G}_U, \mathcal{A}_U) = \prod_{i=1}^{m} q(g_i|\mathcal{G}_U, \mathcal{A}_U) \tag{16}$$

where $q(g_i|\mathcal{G}_U, \mathcal{A}_U) = \mathcal{N}(g_i|\mu_i, diag(\sigma_i^2))$. Gaussian parameters are learned from two GCNs, i.e. $\mu = GCN_\mu(\mathcal{G}_U, \mathcal{A}_U)$, with μ the matrix stacking up mean vectors μ_i; likewise, $log\sigma = GCN_\sigma(\mathcal{G}_U, \mathcal{A}_U)$. Latent vectors g_i are samples drawn from this distribution. From these vectors, a generative model aims at decoding $\mathcal{A}_U$, leveraging inner products: $p(\mathcal{A}_U|\mathcal{G}) = \prod_{i=1}^m \prod_{j=1}^m p([\mathcal{A}_U]_{ij}|g_i, g_j)$, where $p([\mathcal{A}_U]_{ij} = 1|g_i, g_j) = \sigma(g_i^\top g_j)$. GCN weights are tuned by maximizing a tractable variational lower bound (ELBO) of the model's likelihood by gradient descent, with a Gaussian prior on the distribution of latent vectors, and using the reparameterization trick from [15]. Formally, for VGAE, we minimize the reconstruction error from $\mathcal{G}$ to $\mathcal{G}_U$ by:

$$\mathcal{L}_U^2 = \mathbb{E}_{q(\mathcal{G}|\mathcal{G}_U, \mathcal{A}_U)}[\log p(\mathcal{A}_U|\mathcal{G})] - D_{KL}[q(\mathcal{G}|\mathcal{G}_U, \mathcal{A}_U)||p(\mathcal{G})] \tag{17}$$

where $D_{KL}(\cdot||\cdot)$ is the KL divergence of the approximate from the true posterior. It is obvious that $\mathcal{L}_U^2$ is a variation of the objective function $J_{U \cdot U}$ under the graph-based inference setting.

Similarly, the loss of the reconstruction from $\mathcal{F}$ to $\mathcal{F}_I$ is:

$$\mathcal{L}_I^2 = \mathbb{E}_{q(\mathcal{F}|\mathcal{F}_I, \mathcal{A}_I)}[\log p(\mathcal{A}_I|\mathcal{F})] - D_{KL}[q(\mathcal{F}|\mathcal{F}_I, \mathcal{A}_I)||p(\mathcal{F})] \tag{18}$$

To ensure the embedding matrices $\mathcal{F}$ and $\mathcal{G}$ are consistent with User-Item ratings, we aim to minimize the reconstruction loss introduced in Eq. (2). Here, we present a neural-network-based NMF module that can be integrated into our SGCCL framework by replacing the inner product in $J'_{U \cdot I}$ with a neural architecture. It can learn an arbitrary function from data. Specifically,

$$\hat{\mathcal{R}} = \mathcal{G}\mathbf{S}\mathcal{F}, \text{ and } \mathcal{L}_R = ||\mathcal{R} - \hat{\mathcal{R}}||_F^2 \tag{19}$$

Table 1: Data statistics & parameter setting.

Dataset	#User	#Items	Interactions	δ_U	δ_I	c	d
ML-100k	943	1,349	100,000	2	4	16	32
Automotive	2,928	1,835	20,473	2	3	16	32
Movies & TV	44,439	25,047	1,070,860	5	5	32	32
Gowalla	29,858	40,981	1,027,370	3	5	32	64

S can be delicately considered as a weight matrix under the neural network framework (as shown in Fig. 3) and it can be updated using typical backpropagation during the embedding process.

The reconstruction of siamese graph and bipartite graph introduced above enable us to learn embeddings by taking care of different aspects (user-user similarity, item-item correlations and user-item ratings). To ensure that the final results are consistent, we jointly formulate all loss functions as the unified loss of SGCCL, which is a nonlinear variation of the objective function J of consensus learning principle in Eq. (5):

$$\mathcal{L} = \lambda_1 \mathcal{L}_R + \lambda_2 \mathcal{L}_U^1 + \lambda_3 \mathcal{L}_I^1 + \lambda_4 (\mathcal{L}_U^2 + \mathcal{L}_I^2) \tag{20}$$

Therefore the whole SGCCL network parameters are jointly optimized by minimizing the loss $\mathcal{L}$ and the final output embedding matrices for users and items are $\mathcal{G}$ and $\mathcal{F}$.

3.4 Dynamic Loss Fusion

For most multi-task learning networks, optimizing multiple objectives is difficult without finding the correct balance among those objectives (*i.e.* λ in Eq. 20). In this paper, we adopt a simple yet effective adaptive weighting method, named Dynamic Weight Average (DWA) proposed in [19], which learns to average task weighting over time by considering the rate of change of loss for each task as:

$$w_k(t-1) = \frac{\mathcal{L}_k(t-1)}{\mathcal{L}_k(t-2)}, \lambda_k(t) = \frac{K \exp(w_k(t-1)/T)}{\sum_i \exp(w_i(t-1)/T)} \tag{21}$$

where $\mathcal{L}_k \in \{\mathcal{L}_R, \mathcal{L}_U^1, \mathcal{L}_I^1, (\mathcal{L}_U^2 + \mathcal{L}_I^2)\}$ and $\lambda_k \in \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$. t is an iteration index and T represents a temperature which controls the softness of task weighting.

4 EXPERIMENTS

4.1 Experimental Settings

4.1.1 Benchmark Datasets. We conduct experiments on four public benchmark datasets commonly used in recommendation tasks.

- **MovieLens-100k**: The movielens datasets are collected from MovieLens review websites [10]. In this study, we select the MovieLens-100k and use the 5-core setting to ensure that users and items have at least 5 interactions [12].
- **Amazon-Automotive/Movies & TV**: Amazon review datasets are widely used for product recommendation [11]. Here we select the Automotive and Movies & TV collections in this study. And we adopt 5-core setting for automotive and 10-core for Movies & TV. The rating information is used for connection building.
- **Gowalla**: Gowalla is a popular location-based social network [18]. We treat locations as items to capture user preferences based on the check-in history. 10-core setting is used to ensure the data quality. No side information is provided in Gowalla dataset.

The details of the datasets are reported in Table 1, where δ that controls the degree of connectivity (Eq. 6) and the dimensions of user/item embeddings (c and k) are used to regulate the graphs and output embeddings for graph-based collaborative filtering.

Table 2: Performance comparison between SGCCL and baseline methods.

Tasks	Models	ML-100k		Automotive		Movies & TV		Gowalla	
		recall@20	ndcg@20	recall@20	ndcg@20	recall@20	ndcg@20	recall@20	ndcg@20
Overall Recommendation	MF	0.2897	0.3223	0.0742	0.0298	0.0625	0.0422	0.1291	0.1109
	NeuMF	0.3168	0.3647	0.0762	0.0311	0.0820	0.0511	0.1399	0.1212
	GC-MC	0.2544	0.3025	0.0990	0.0440	0.0638	0.0401	0.1395	0.1204
	NGCF	0.3382	0.4016	0.1127	0.0455	0.0866	0.0555	0.1569	0.1327
	Gemini	0.2724	0.3108	0.0874	0.0400	0.0718	0.0487	0.1351	0.1178
	DICE-LightGCN	0.3131	0.3568	0.1202	0.0513	0.0863	0.0549	0.1648	0.1433
	LightGCN	0.3229	0.3805	0.1384	0.0598	0.0915	0.0599	0.1821	0.1537
	LightGCN-CL	0.3234	0.3878	0.1412	0.0617	0.0926	0.0600	0.1816	0.1528
	SGL	0.3238	0.3923	0.1401	0.0611	0.0932	0.0606	0.1824	0.1540
	SGCCL	0.3655	0.4221	0.1543	0.0658	0.0984	0.0643	**0.1869**	**0.1610**
	SGCCL-SI	**0.3682**	**0.4244**	**0.1557**	**0.0672**	**0.0993**	**0.0652**	N/A	N/A
Popularity Bias Eliminating	LightGCN	0.1052	0.0844	0.0301	0.0187	0.0334	0.0256	0.0610	0.0401
	SGL	0.1063	0.0856	0.0304	0.0189	0.0340	0.0264	0.0617	0.0409
	CausE-LightGCN	0.1035	0.0822	0.0293	0.0176	0.0334	0.0257	0.0606	0.0397
	IPW-LightGCN	0.1067	0.0863	0.0302	0.0189	0.0338	0.0260	0.0609	0.0401
	DICE-LightGCN	0.1198	0.1087	0.0308	0.0190	0.0344	0.0283	0.0619	0.0425
	SGCCL	0.1225	0.1116	0.0364	0.0230	0.0364	0.0301	**0.0631**	**0.0433**
	SGCCL-SI	**0.1246**	**0.1129**	**0.0372**	**0.0244**	**0.0375**	**0.0307**	N/A	N/A

4.1.2 Evaluation Metrics. We adopt two widely used evaluation metrics to evaluate the recommendation performance: *recall@k* and *ndcg@k* (Normalized discounted cumulative gain) [34]. In this study, the k is set as 20 which are used by other baseline methods for a fair comparison. For every user in each dataset, we randomly select 70% and 10% of their interacted items as training and validation sets. And the rest 20% of data are set as the ground truth and used for testing. While for the popularity debiasing task, we adopt the skewed split rule in [17]: A test set is sample with 20% of the total item exposures, which ensures the uniform probability of items. Training and validation sets are then created from the remaining data (as a regular split) with 70/10 proportions.

4.1.3 Baseline Methods. We compare the SGCCL with following baseline methods: *Vanilla CF methods*: **MF** [23] & **NeuMF** [14]: These are naive matrix factorization methods to factorize the User-Item rating matrix directly. *Graph-based CF methods*: **GC-MC** [2]; **NGCF** [29]; **Gemini** [33] & **LightGCN** [13]: These methods introduce the GCNs to learn the high-order node signal and model the complex neighborhood relationships between users and items. Gemini adopts the user-user and item-item correlations separately. **LightGCN-CL** [20] & **SGL** [31]: These are the latest GCL-based methods with injecting self-supervised learning into LightGCN. *Debiasing methods*: **IPW** [17]; **CausE** [3] & **DCIE** [37]: These debiasing methods introduced the inverse propensity weight or causal effect to alleviate the popularity bias problem. For a fair comparison, all of them adopt LightGCN as the backbone in their original settings *Proposed methods* - **SGCCL**: We generate the siamese graphs from user-item rating matrix only for a fair competition. **SGCCL-SI**: We exploit the side information in siamese graph generation.

4.2 Experimental Results

4.2.1 Performance Comparison. We report overall performance of SGCCL compared with baselines methods in Table 2.

Overall Recommendation: The experiments demonstrate that the SGCCL consistently yields the best performance on all the datasets. With considering the side information (**SGCCL-SI**), the performance can be further improved with **12.06%** (ML-100k), **10.27%** (Automotive) and **6.14%** (MoviesTV) in recall@k comparing with baseline methods. It proves the effectiveness of introducing and distinguishing both heterogeneous and homogeneous information from the original user-item network under a consensus principle.

Table 3: Popularity bias analysis on Movies & TV dataset: HD/LD refers to high-degree/low-degree and U/I refers to users and items, e.g., HD-U refers to high-degree users.

Model	HD-U & HD-I	HD-U & LD-I	LD-U & HD-I	LD-U & LD-I
LightGCN	0.3740	0.1203	0.6556	0.2513
SGL	0.3715	0.1163	0.6525	0.2422
CausE-LightGCN	0.4207	0.1359	0.6932	0.2744
IPW-LightGCN	0.3731	0.1151	0.6723	0.2486
DCIE-LightGCN	0.3812	0.1321	0.6601	0.2541
SGCCL	0.3542	0.0439	0.6481	0.1139
SGCCL-SI	**0.3478**	**0.0420**	**0.6473**	**0.1124**

To better demonstrate the improvements, we also visualize the ML-100k data in a two-dimensional space by adopting the t−SNE algorithm [28] on the learned embedding $\mathcal{F}$ and $\mathcal{G}$ as shown in Fig. 4. By applying k-means clustering algorithms to 10 clusters, we can see that the user and item embeddings generated based on SGCCL have the shortest intra cluster distance comparing with the counterparts. It validates the advantages of considering both user-user and item-item homogeneous relationships to infer user preferences and item correlations in SGCCL.

Popularity Bias Eliminating: In Table 2 (lower section), we compare methods for popularity bias elimination. The results show that baseline debiasing methods can alleviate bias to a certain degree but lead to worse performance on overall recommendation tasks (DCIE-LightGCN). It indicates that these debiasing methods excessively emphasize on unpopular items and cause another type of bias. SGCCL can still achieve best performance on all datasets on bias eliminating tasks. By applying SGCCL, majority unpopular items boost their exposure to users through item-item correlations and the distinctive preference of users' homogeneous neighbors, without losing the attention of popular items/users.

In order to delineate how SGCCL alleviates the popularity bias for both users and items, we define four different combinations of users and items according to their popularity: the high-degree nodes (users/items) as their degrees are at the top 20% while low-degree nodes are at the bottom 20%. The embedding quality reported in Table 3 is measured as $P = \frac{1}{m \times n} ||\mathcal{R}^* - \widehat{\mathcal{R}^*}||^2$, where $\mathcal{R}^* \in \mathbb{R}^{m \times n}$ refers to a submatrix of $\mathcal{R}$ that only contains the selected m users and n items. $\widehat{\mathcal{R}^*}$ is the reconstructed rating matrix from $\mathcal{G}$, $\mathcal{F}$ and $\mathbf{S}$.

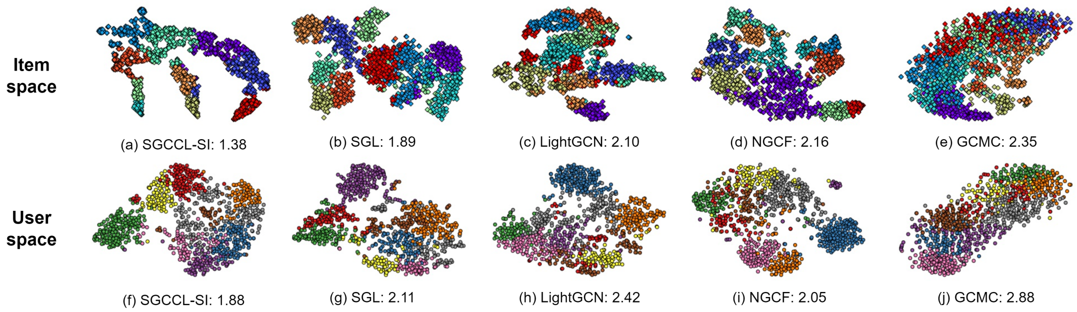

Figure 4: Embedding feature comparisons on ML-100k using t-SNE. Each point denotes a node, which is color-coded based on the clusters the node belonging to (there are 10 groups in total). (a)-(e) refers to the item nodes clustering, (f)-(j) refers to the user nodes clustering, and the indicators refer to the intra cluster distances.

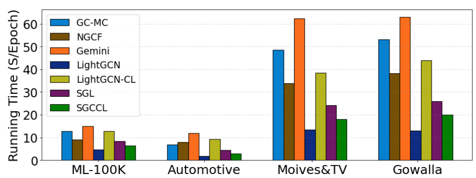

Figure 5: Comparative analysis of running time.

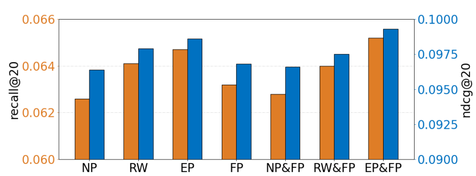

Figure 6: Comparative analysis of graph augmentation effect on Movies & TV dataset, where NP/EP/FP refers to node/edge/feature perturbations (dropout), and RW refers to random walk [20, 39].

SGCCL performs extremely well on unpopular item recommendation with **264%** in Column 2 and **212%** in Column 4, which indicates the introduction of homogeneous relations can significantly help de-biasing the user preference ratings.

4.2.2 Time Efficiency Analysis. Fig. 5 illustrates the training time comparison among different methods and shows that SGCCL is an efficient recommendation model even with a complex model structure. It is achieved by the adoption of parallel computation in siamese graph learning of $\mathcal{G}_U$ and $\mathcal{F}_I$. And the introduction of VGAE and DWA makes the reconstruction of $\mathcal{R}$ from Θ_U and Θ_I more flexible and therefore speeding up the convergence process. LightGCN is faster because the authors claim that feature transformation and nonlinear activation contribute little to the performance, thus it significantly simplifies the GCN model. However, our experiments suggest that nonlinear transformation is indispensable in better capturing the complex relations of users/items.

4.2.3 Comparison of Graph Augmentation Operations. As one of the most important components in contrastive learning, graph augmentation directly influences the node embedding quality. To explore the difference among augmentation operations, we conduct experiments based on Movie&TV as shown in Fig. 6. SGCCL that adopts the combination of edge and feature perturbations achieve the best performance. Coupling two augmentation operations together enable SGCCL to capture the robust patterns from both

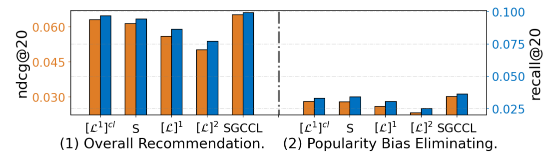

Figure 7: Effective of Each Components: the label in x-axis refers to w/o the components, e.g., S refers to W/O S.

local structures and attributes of a node, and further endows the representations more robustness against noisy interactions.

4.2.4 Ablation Study. To better understand the effectiveness of the components in SGCCL, we compare the performance of SGCCL and its variants on the Movie&TV dataset. The 4 variants is defined as: (1) W/O contrastive learning module and $[\mathcal{L}^1]^{cl}$; (2) W/O Latent matrix $\mathbf{S}$; (3) W/O $[\mathcal{L}]^1$ (Siamese Graph Reconstruction); (4) W/O $[\mathcal{L}]^2$ (Bipartite Graph Reconstruction), which their performance in two different recommendation tasks is shown in Fig. 7. We can observe that each component contributes to the model performance. The contrastive learning part makes an improvement on the popularity bias problem as it can learn the high-quality representations of unpopular nodes with sparse interactions. The latent matrix $\mathbf{S}$ can fuse different scales of node embeddings, and $[\mathcal{L}]^1$ improves the robustness of node embeddings with the constraint of siamese graphs. Moreover, consensus learning $[\mathcal{L}]^2$ forces the final recommendation to be consistent with the bipartite graph.

5 CONCLUSION

In this paper, we proposed a novel graph learning method for personalized recommendation. We introduced siamese graphs which consider user-user similarity and item-item correlations as two types of homogeneous relations to capture distinctive nodes information, and proposed a consensus learning principle to simultaneously factorize homogeneous and heterogeneous information. A siamese graph contrastive consensus learning framework (SGCCL) is further proposed to model high-hop connectivity with robustness between nodes and optimize them based on the consensus learning principle. Experiments demonstrated that SGCCL outperforms state-of-the-art methods. Case studies further confirmed its effectiveness for unpopular items.

ACKNOWLEDGMENTS

This research is partially sponsored by the U.S. National Science Foundation under grant No. IIS-1763452.

REFERENCES

[1] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The unfairness of popularity bias in recommendation. *arXiv preprint arXiv:1907.13286* (2019).

[2] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[3] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *Proceedings of the 12th ACM conference on recommender systems*. 104–112.

[4] Hsinchun Chen, Xin Li, and Zan Huang. 2005. Link prediction approach to collaborative filtering. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05)*. IEEE, 141–142.

[5] 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.

[6] Pierre Comon, Xavier Luciani, and André LF De Almeida. 2009. Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics: A Journal of the Chemometrics Society* 23, 7-8 (2009), 393–405.

[7] Chris Ding and Tao Li. 2010. Convex and semi-nonnegative matrix factorizations. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 32, 1 (2010), 45–55.

[8] GH Golub and CF Van Loan (Eds.). 2013. *Matrix computations 4th edition*. Vol. 3. Johns Hopkins University Press, Baltimore Maryland.

[9] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 297–304.

[10] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm Trans. on Interactive Intelligent Systems* 5, 4 (2015), 1–19.

[11] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[12] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[13] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. 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*. 639–648.

[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[15] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[16] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[17] Dawen Liang, Laurent Charlin, and David M Blei. 2016. Causal inference for recommendation. In *Causation: Foundation to Application, Wksp at UAI. AUAI*.

[18] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of the 25th international conference on World Wide Web*. 951–961.

[19] Shikun Liu, Edward Johns, and Andrew J Davison. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1871–1880.

[20] Zhuang Liu, Yunpu Ma, Yuanxin Ouyang, and Zhang Xiong. 2021. Contrastive Learning for Recommender System. *arXiv preprint arXiv:2101.01317* (2021).

[21] Xin Luo, Mengchu Zhou, Yunni Xia, and Q. Zhu. 2014. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Trans. on Industrial Informatics* 10, 2 (2014), 1273–1284.

[22] Yuqi Qin, Pengfei Wang, and Chenliang Li. 2021. The world is binary: Contrastive learning for denoising next basket recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 859–868.

[23] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[24] Masahiro Sato, Janmajay Singh, Sho Takemori, Takashi Sonoda, Qian Zhang, and Tomoko Ohkuma. 2020. Modeling user exposure with recommendation influence. In *Proc. of the 35th ACM Symposium on Applied Computing*. 1461–1464.

[25] Lalita Sharma and Anju Gera. 2013. A survey of recommendation system: Research challenges. *International Journal of Engineering Trends and Technology (IJETT)* 4, 5 (2013), 1989–1992.

[26] Jianing Sun, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, Xiuqiang He, Chen Ma, and Mark Coates. 2020. Neighbor interaction aware graph convolution networks for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1289–1298.

[27] Jianing Sun, Yingxue Zhang, Chen Ma, Mark Coates, Huifeng Guo, Ruiming Tang, and Xiuqiang He. 2019. Multi-graph convolution collaborative filtering. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1306–1311.

[28] Laurens van der Maaten. 2014. Accelerating t-SNE using tree-based algorithms. *The Journal of Machine Learning Research* 15, 1 (2014), 3221–3245.

[29] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proc. of the 42nd Intl. ACM SIGIR Conf. on Information Retrieval*. 165–174.

[30] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.

[31] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proc. of the 44th ACM SIGIR Conf.* 726–735.

[32] Zhe Xie, Chengxuan Liu, Yichi Zhang, Hongtao Lu, Dong Wang, and Yue Ding. 2021. Adversarial and contrastive variational autoencoder for sequential recommendation. In *Proceedings of the Web Conference 2021*. 449–459.

[33] Jixing Xu, Zhenlong Zhu, Jianxin Zhao, Xuanye Liu, Minghui Shan, and Jiecheng Guo. 2020. Gemini: A Novel and Universal Heterogeneous Graph Information Fusing Framework for Online Recommendations. In *Proc. of the 26th ACM SIGKDD Intl. Conf. on Knowledge Discovery & Data Mining*. 3356–3365.

[34] Jheng-Hong Yang, Chih-Ming Chen, Chuan-Ju Wang, and Ming-Feng Tsai. 2018. HOP-rec: high-order proximity for implicit recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 140–144.

[35] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proc. of the 24th ACM SIGKDD Conf.* 974–983.

[36] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *Advances in Neural Info. Proc. Systems* 31 (2018), 5165–5175.

[37] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling User Interest and Conformity for Recommendation with Causal Embedding. In *Proceedings of the Web Conference 2021*. 2980–2991.

[38] Yu Zheng, Ruiqi Hu, Sai-fu Fung, Celina Yu, Guodong Long, Ting Guo, and Shirui Pan. 2020. Clustering social audiences in business information networks. *Pattern Recognition* 100 (2020), 107126.

[39] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. 2021. An empirical study of graph contrastive learning. *arXiv preprint arXiv:2109.01116* (2021).

[40] Ziwei Zhu, Jianling Wang, and James Caverlee. 2020. Measuring and mitigating item under-recommendation bias in personalized ranking systems. In *Proc. of the 43rd Intl. ACM SIGIR Conf. on Information Retrieval*. 449–458.