Towards Unified Modeling for Positive and Negative Preferences in Sign-aware Recommendation

Yuting Liu[1], Yizhou Dang[1], Yuliang Liang[1], Qiang Liu[3], Guibing Guo[1](✉), Jianzhe Zhao[1](✉), and Xingwei Wang[2]

[1] Software College, Northeastern University, China
{guogb,zhaojz}@swc.neu.edu.cn
[2] School of Computer Science and Engineering, Northeastern University, China
[3] New Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, China

Abstract. Recently, sign-aware graph recommendations have drawn attention as they learn users' negative preferences in addition to positive ones. Nevertheless, due to adopting two independent encoders for positive and negative interactions, existing approaches fail to learn the users' comprehensive negative preferences and holistic collaborative signals from high-order heterogeneous interactions formed by multiple links with different signs. To compensate for this drawback, we devise a novel unified modeling approach to capture complete collaborative information and comprehensive user preferences. In this paper, we first explore the relationship between negative preferences and find that propagating both positive and negative high-order preferences along positive edges is feasible. Based on the observation, a Light Signed Graph Convolution Network for Recommendation (LSGRec) is proposed to comprehend user preferences within signed user-item interaction graphs. Then, recommendation results are generated based on positive preferences and optimized with negative ones. Finally, representations of users and items are trained through different auxiliary tasks. Extensive experiments on three real-world datasets demonstrate that our method outperforms existing baselines regarding performance and computational efficiency.

Keywords: Recommender System · Collaborative Filtering · Signed Graph Neural Network.

1 Introduction

Due to its powerful ability to model user-item interactions in a bipartite graph, graph-based collaborative filtering [20, 6, 15, 22, 17, 26] has become the mainstream method for recommendation systems. Graph-based approaches usually adopt a message-passing and neighborhood aggregation mechanism in user-item bipartite graphs to capture high-order collaborative signals that model users' preferences and learn effective user and item representations. Most existing works

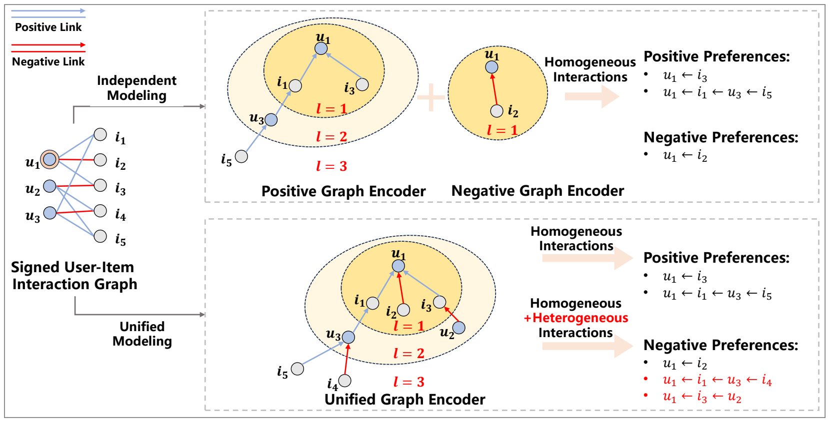

Fig. 1. An illustration of the user-item signed interaction graph and its high-order connectivity through unified and independent modeling, respectively. The u_1 is the target user that needs recommendations. The rightmost side shows the paths of positive and negative preferences in different modeling approaches.

in graph-based recommendations treat all user-item interactions as positive, ignoring actual negative interactions between users and items that also reflect users' personalized preferences in real-world recommendation systems (e.g., a user gives a negative review after purchasing an item). Without considering this type of interaction, these methods are likely to mistakenly treat items that users dislike as they like, resulting in inaccurate user preferences.

Recently, a few studies have paid attention to explicitly modeling users' negative preferences besides traditional positive ones in the graph-based recommendation [8, 16, 18, 9]. Since the negative links inherently have different semantics and principles compared to positive links [3, 10, 2], they split interactions into positive and negative and model users' positive and negative preferences with corresponding independent encoders, respectively. Although effective, these methods can disrupt high-order collaborative signals in the signed graph. As shown in Fig 1, u_1 and u_2 interact with the same item i_3, but u_1 likes i_3, and u_2 does not, we can assume that the preference of u_1 is negative correlative with the preference of u_2. This collaborative signal that the preferences of u_1 and u_2 are dissimilar cannot be captured by existing independent modeling approaches but can be captured by unified modeling. Furthermore, independent modeling for users and items on two separate graphs ignores the preferences within high-order heterogeneous interactions (i.e., the links formed by multiple edges with different signs). Taking u_1 in Fig 1 as an example, we can observe that there are two additional paths for high-order negative preferences when adopting unified modeling for users and items (i.e., $u_1 \leftarrow i_1 \leftarrow u_3 \leftarrow i_4$ and $u_1 \leftarrow i_3 \leftarrow u_2$ in red font) compared to adopting independent modeling. Consequently, users' preferences will be incomplete and inaccurate with these independent encoders, which prompts us to propose a new unified modeling approach.

However, we cannot directly use negative edges to calculate higher-order negative representations like the positive ones because of the failure of the balance theory in the recommendation [8, 16, 18, 7, 23]. To find an appropriate passing rule, we conduct a statistical analysis and explore the relationship between positive and negative user preferences. We discover that users who like the same items tend to dislike the same items more, but the converse is not necessarily true. Based on this observation, we propose passing both positive and negative preferences along positive edges to design a recommendation-specific unified modeling approach. Specifically, for the negative preferences within high-order heterogeneous interactions, first-order negative preferences are captured by the negative links, while high-order negative preferences are propagated along positive edges based on homophily.

Based on the unified modeling method, we propose **LSGRec**, a **L**ight **S**igned **G**raph Convolution Network for **Rec**ommendation, including an effective sign-aware graph neural network to calculate user/item representations, a negative preference filter to generate satisfied recommendations and multiple training objectives to optimize the parameters.

To sum up, the main contributions of our work are the following:

(1) We highlight the importance of unified modeling users and items in sign-aware graph recommendations. To find an appropriate modeling method for recommendation scenarios, we conduct an empirical study on real-world datasets. We devise a unified modeling approach to aggregate positive and negative preferences in a whole signed graph simultaneously based on our experimental analysis.

(2) We propose LSGRec, a novel graph-based model that utilizes positive and negative user-item interactions to provide more accurate recommendations, which consists of a unified modeling graph neural network, a negative preference filter, and a training strategy with multiple auxiliary tasks.

(3) We perform extensive experiments on three real-world datasets (Amazon-Beauty, Amazon-Book, Yelp2021) to demonstrate that our method outperforms the state-of-the-art graph-based recommendation methods. Our method outperforms the best baseline by up to 10.64%, 16.21%, and 15.60% in terms of Precision@10, Recall@10, and NDCG@10, respectively. Our code is available at https://github.com/VanillaCreamer/LSGRec.

2 Negative Preferences in Positive Neighbors

Although signed graph neural networks in other areas have been widely explored [13, 19, 10, 7, 3], there are two obstacles induced by the principles of negative edges that need to be avoided when calculating along high-order heterogeneous edges: (1) the balance theory no longer holds in recommender systems, and (2) there is no homophily between nodes linked by negative edges.

The assumption of balance theory [1], which implies that "the friend of my friend tends to be my friend" and "the enemy of my enemy tends to be my friend". However, "the enemy of my enemy tends to my friend" no longer holds

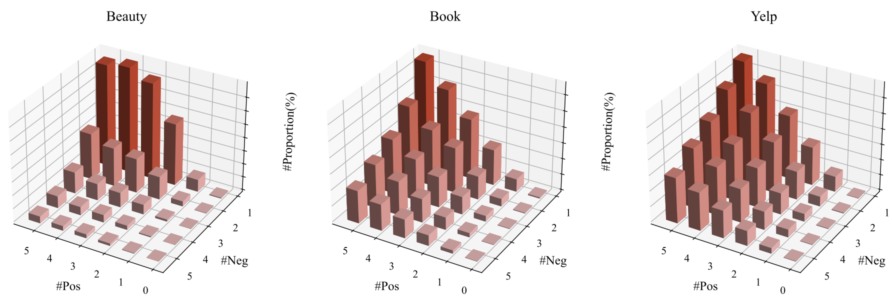

Fig. 2. The proportion(%) of negative similar user pairs in selected positive similar user pairs on three datasets.

in the context of recommendation [18] since users' preferences cannot be dichotomous. More specifically, we conduct a statistical analysis on three real-world datasets: *Beauty*, *Book*, and *Yelp*. We calculate the proportion of two users who dislike the same item (i.e., the enemy of my enemy) and prefer at least one identical item (i.e., my friend), finding that there are only 0.48%, 0.35%, and 1.27% of interactions on these datasets that meet this theory (i.e., "the enemy of my enemy is my friend"), respectively. We cannot directly utilize negative edges to propagate high-order negative representations. To avoid these limitations, we assume that the negative representations of positive neighbors should be similar since the attributes of adjacent nodes are similar (i.e., the idea of collaborative filtering), where the negative representation is one of the attributes, and validate this through an empirical study on three datasets.

We define a pair of users who like at least Pos same items as *positive similar user pair*, a pair of users who dislike at least Neg same items as *negative similar user pair*. We conduct a data analysis in two steps: (1) Randomly sample 10,000 positive similar user pairs. (2) Calculate the proportion (%) of negative similar user pairs in these positive similar user pairs. As shown in Fig 2, we can find that as users' positive preferences become more similar (i.e., as Pos increases), their negative preferences tend to be more similar (i.e., the proportion increases at all Neg values). Therefore, we propose passing both positive and negative collaborative signals to capture the representations with the homophily between positive neighbors. Specifically, for the negative preferences within high-order heterogeneous interactions, first-order negative representations are captured by the direct negative neighbors. In contrast, high-order negative representations are propagated along positive neighbors, thus avoiding the limitations mentioned.

3 Related Work

3.1 Graph-based Recommendation

Using graph neural networks (GNNs) [4] in modeling interactions between users and items has gained broad acknowledgment as potent architectures. Numerous recommendation models, built upon GNNs as their foundations, prove that they have attained state-of-the-art performance across various sub-fields [12, 6, 20]. Influenced and inspired by LightGCN, a number of improvements have been made to achieve more competitive performance[17, 14, 24, 25]. However, negative interactions and preferences are overlooked in most cases, which leads to inaccurate user and item representations learning.

3.2 Signed Graph Neural Networks

Signed graphs have been widely explored in social networks, focusing on node-level and graph-level tasks [13, 19, 10, 7, 3]. These methods are built upon the assumption of balance theory [1], which no longer holds in the context of recommender systems, so these methods cannot be applied to the graph-based recommendation tasks. Recent efforts have focused on better modeling users/items with positive and negative edges in signed user-item bipartite graphs. SiReN [18] constructs a signed bipartite graph and generates positive and negative embeddings for the partitioned graphs. PANE-GNN [16] employs contrastive learning on the negative graph to reduce noise and filter out items with high disinterest scores, ensuring the relevance of the recommended results. SiGRec [8] investigates three kinds of negative feedback and defines a new sign cosine loss function to adaptively capture differences among them.

Discussion. These methods adopt independent encoders for positive and negative interactions due to the limitations of the balance theory, which corrupt high-order collaborative information in the signed graph. Unlike existing works, our approach aims to explore negative preference relationships between users with similar preferences. By optimizing the computation of graph convolution, we use positive edges to convey users' negative preferences, addressing the limitations of balance theory and involving high-order heterogeneous interactions.

4 Methodology

4.1 Preliminary

The basic input in recommendation methods is the historical user-item interactions with ratings, which is modeled as a weighted bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{I}, \mathcal{E})$, where $\mathcal{U}$ and $\mathcal{I}$ are the set of M users and N items, respectively, and $\mathcal{E}$ is the set of weighted edges between $\mathcal{U}$ and $\mathcal{I}$. A weighted edge $(u, i, \omega_{ui}) \in \mathcal{E}$ represents that a user $u \in \mathcal{U}$ gives a rating ω_{ui} to an item $i \in \mathcal{I}$. To simplify the setting, we assume $\mathcal{G}$ as a static network without repeated edges.

In our study, to fully utilize ω_{ui} (i.e., ratings), we set a threshold δ to split the original ratings into binary signals $\mathcal{E}^+$ and $\mathcal{E}^-$, where

$$\begin{aligned} \mathcal{E}^+ &= \{(u, i, 1) | \text{sign}(\omega_{ui} - \delta) > 0, (u, i, \omega_{ui}) \in \mathcal{E}\}, \\ \mathcal{E}^- &= \{(u, i, -1) | \text{sign}(\omega_{ui} - \delta) < 0, (u, i, \omega_{ui}) \in \mathcal{E}\}. \end{aligned} \tag{1}$$

Here, $\mathcal{E}^+ \subset \mathcal{U} \times \mathcal{I}$ and $\mathcal{E}^- \subset \mathcal{U} \times \mathcal{I}$ denote the sets of positive and negative edges, respectively. The function $\mathrm{sign}(\cdot)$ outputs the sign of the input, and δ can be determined according to the characteristics of a given dataset. Unlike existing works that divide the whole graph into two edge-disjoint sub-graphs according to the sign of edges, we process these two signals in the original whole graph with a unified encoder; that is, a user-item weighted bipartite graph becomes a user-item signed bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{I}, \mathcal{E}^+, \mathcal{E}^-)$. Note that $\mathcal{E}^+ \cap \mathcal{E}^- = \emptyset$ and $\mathcal{E}^+ \cup \mathcal{E}^- = \mathcal{E}$, in other words, a user cannot have both positive and negative preferences for an item simultaneously. Hence, given a user-item interaction graph $\mathcal{G} = (\mathcal{U}, \mathcal{I}, \mathcal{E}^+, \mathcal{E}^-)$, our task is to generate top$K$ recommendations for each user.

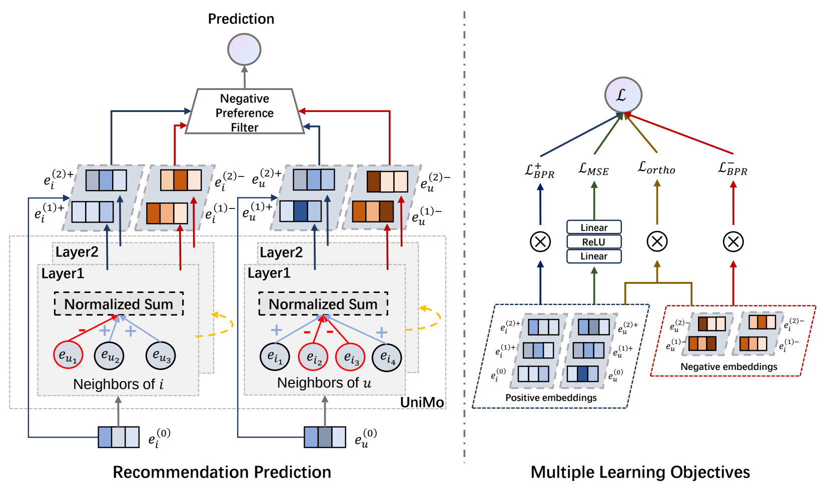

Fig. 3. An illustration of our LSGRec framework, consisting of a unified modeling approach, a negative preference filter, and multiple auxiliary tasks. Trainable parameters are learned by the multiple learning objectives.

4.2 Model Overview

The architecture of the LSGRec model is depicted in Fig 3, consisting of a unified modeling (UniMo) approach, a negative preference filter, and multiple auxiliary tasks to train parameters. The UniMo calculates the final embeddings $\mathbf{e}_u^+$, $\mathbf{e}_i^+$, $\mathbf{e}_u^-$, $\mathbf{e}_i^-$ corresponding to the positive and negative preferences of users and items from their initial embeddings $\mathbf{e}_u^{(0)}$ and $\mathbf{e}_i^{(0)}$ by aggregating high-order neighborhood information iteratively. The negative preference filter utilizes negative preferences to filter out items that users dislike, generating more accurate recommendations. The multi-task learning objectives aim to train user/item embeddings better by optimizing different loss functions.

4.3 Unified Modeling for Positive and Negative Preferences

We build upon the message-passing to capture collaborative signals along the signed user-item bipartite graph, unified modeling of the positive and negative embeddings of users and items in the whole graph. We first illustrate the design of first-order propagation and then generalize it to high-order recursively. Note that there are some minor differences between the first-order and high-order propagation due to the particularity of negative edges.

Direct Neighbors. Intuitively, the positively interacted items provide direct evidence of a user's positive preference (i.e., what a user likes), and the negatively interacted items imply a user's negative preference (i.e., what a user dislikes). In the first order, we calculate positive and negative representations of each node by aggregating messages passed by the direct neighbors through positive and negative edges. We adopt a simple weighted sum aggregator and abandon the user of feature transformation and nonlinear activation with the *AGG* function, which could be burdensome for collaborative filtering. The signed graph convolution operation (i.e., propagation rule) in the first layer in UniMo can be defined as:

$$\mathbf{e}_u^{(1)+} = \frac{1}{\sqrt{|\mathcal{N}_u^+|}\sqrt{|\mathcal{N}_i^+|}} \sum_{i \in \mathcal{N}_u^+} \mathbf{e}_i^{(0)}, \ \mathbf{e}_u^{(1)-} = \frac{1}{\sqrt{|\mathcal{N}_u^-|}\sqrt{|\mathcal{N}_i^-|}} \sum_{i \in \mathcal{N}_u^-} \mathbf{e}_i^{(0)} \tag{2}$$

$$\mathbf{e}_i^{(1)+} = \frac{1}{\sqrt{|\mathcal{N}_i^+|}\sqrt{|\mathcal{N}_u^+|}} \sum_{u \in \mathcal{N}_i^+} \mathbf{e}_u^{(0)}, \ \mathbf{e}_i^{(1)-} = \frac{1}{\sqrt{|\mathcal{N}_i^-|}\sqrt{|\mathcal{N}_u^-|}} \sum_{u \in \mathcal{N}_i^-} \mathbf{e}_u^{(0)} \tag{3}$$

The symmetric normalization term $\frac{1}{\sqrt{|\mathcal{N}_i^+|}\sqrt{|\mathcal{N}_u^+|}}$ and $\frac{1}{\sqrt{|\mathcal{N}_i^-|}\sqrt{|\mathcal{N}_u^-|}}$ evolve from the design of standard GCN, which can avoid the scale of embeddings increasing with graph convolution operations. L_1 norm and some other choices can also be applied here, while we chose this according to the performance.

High-order Propagation. With the representation in the first layer, we can stack more embedding propagation layers to explore the high-order neighborhood information. Such high-order connectivities are crucial to encoding the collaborative signal to model the preferences of users and items.

As elaborated in Sec 2, we utilize homophily between positive neighbors to pass both positive and negative collaborative signals to capture the preferences within high-order heterogeneous interactions since users are likely to have similar negative preferences if they are similar. As illustrated in Fig 3, by stacking l propagation layers, a user (and an item) receives the positive and negative messages passed from its l-hop neighbors. The representations of user u and item i in the l-th layer can be recursively formulated as:

$$\mathbf{e}_u^{(l+1)+} = \frac{1}{\sqrt{|\mathcal{N}_u^+|}\sqrt{|\mathcal{N}_i^+|}} \sum_{i \in \mathcal{N}_u^+} \mathbf{e}_i^{(l)+}; \mathbf{e}_u^{(l+1)-} = \frac{1}{\sqrt{|\mathcal{N}_u^+|}\sqrt{|\mathcal{N}_i^+|}} \sum_{i \in \mathcal{N}_u^+} \mathbf{e}_i^{(l)-} \tag{4}$$

$$\mathbf{e}_i^{(l+1)+} = \frac{1}{\sqrt{|\mathcal{N}_i^+|}\sqrt{|\mathcal{N}_u^+|}} \sum_{u \in \mathcal{N}_i^+} \mathbf{e}_u^{(l)+}; \mathbf{e}_i^{(l+1)-} = \frac{1}{\sqrt{|\mathcal{N}_i^+|}\sqrt{|\mathcal{N}_u^+|}} \sum_{u \in \mathcal{N}_i^+} \mathbf{e}_u^{(l)-} \tag{5}$$

Layer Combination. In UniMo, the only trainable model parameters are the embeddings at 0-th layer, i.e., $e_u^{(0)}$ and $e_i^{(0)}$. The positive and negative representations of users and items can be calculated via Eq. 2 - Eq. 5. After computing the high-order preference embedding at top-L layers, we respectively stack the positive and negative preference embeddings at each layer and take unweighted arithmetic mean to obtain the final positive and negative representations:

$$\mathbf{e}_u^{+} = \frac{1}{L+1}\sum_{l=0}^{L}\mathbf{e}_u^{(l)+};\ \mathbf{e}_u^{-} = \frac{1}{L}\sum_{l=1}^{L}\mathbf{e}_u^{(l)-} \tag{6}$$

$$\mathbf{e}_i^{+} = \frac{1}{L+1}\sum_{l=0}^{L}\mathbf{e}_i^{(l)+};\ \mathbf{e}_i^{-} = \frac{1}{L}\sum_{l=1}^{L}\mathbf{e}_i^{(l)-} \tag{7}$$

where $\mathbf{e}_u^{(0)+}$ is equivalent to $\mathbf{e}_u^{(0)}$. It is worth noting that the arithmetic mean terms in positive and negative calculations are not completely consistent since the initial embedding contains positive preference [15] while negative preference is obtained from negative interactions.

4.4 Negative Preference Filter and Recommendation Prediction

To generate satisfactory recommendations for users, we first utilize filters to remove elements that users dislike before the final recommendation. Then, we calculate the positive ranking score and obtain the top$K$ items to be recommended. For user u, the final recommendations can be expressed as:

$$Rec(u) = Filter(topK(\hat{y}_{ui}^{+}), topK(\hat{y}_{ui}^{-})), \tag{8}$$

where $\hat{y}_{ui}^{+} = \mathbf{e}_u^{+\top}\mathbf{e}_i^{+}$ and $\hat{y}_{ui}^{-} = \mathbf{e}_u^{-\top}\mathbf{e}_i^{-}$ are the predicted positive and negative ranking score between user u and item i, respectively. $Filter(\cdot)$ is the negative preference filter. Specifically, the filtering function $Filter(\cdot)$ is given by:

$$Filter(\cdot) = \{(u,i)|\hat{y}_{ui}^{+} \in topK(\hat{y}_{ui}^{+}), \hat{y}_{ui}^{-} \notin topK(\hat{y}_{ui}^{-}))\}. \tag{9}$$

With the negative preference filter, LSGRec considers the negative preference, excluding elements that users dislike.

4.5 Multiple Learning Objectives

To optimize the parameters in our method, we construct an end-to-end training strategy to jointly optimize the different tasks.

Positive BPR. The original Bayesian Personalized Ranking (BPR) loss cannot reflect the difference between positive and negative interactions since it is a pairwise loss based on the relative ranking between observed and unobserved interactions by encouraging the prediction of an observed user-item pair to be higher than its unobserved counterparts. In our setting, the objective function should consider three types of relations: positive interactions, negative interactions, and unobserved interactions. Following SiReN, we propose a positive BPR loss, which encourages the predicted positive ranking score of an observed interaction to be higher than an unobserved one, along with the induced difference between positive and negative interactions:

$$\mathcal{L}_{BPR}^{+} = -\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{N}_u}\sum_{j\notin\mathcal{N}_u} \ln\sigma(c_1\hat{y}_{ui}^{+} - \hat{y}_{uj}^{+}), \tag{10}$$

where $\sigma(\cdot)$ is the sigmoid function, $sign(\cdot)$ is the sign function, and c_1 is the induced term to distinguish positive and negative interactions. In our setting, $c_1 = (-1/2 * sign(w_{ui} - \delta) + 3/2)$, that is, $c_1 = 1$ for positive items and $c_1 = 2$ for negative items. In this definition, $-1/2$ and $3/2$ are hyperparameters. In this way, when minimizing this loss, the positive prediction score (i.e., positive preference) relations of positive, negative, and unobserved items will be $\hat{y}_{ui_+}^{+} > \hat{y}_{ui_-}^{+} > \hat{y}_{uj}^{+}$.

Negative BPR. Similar to positive BPR, we propose negative BPR loss to constrain the proportion of negative preferences among three types of interaction:

$$\mathcal{L}_{BPR}^{-} = -\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{N}_u}\sum_{j\notin\mathcal{N}_u} \ln\sigma(\hat{y}_{uj}^{-} - c_2\hat{y}_{ui}^{-}). \tag{11}$$

where $c_2 = 1/2 * sign(w_{ui} + 3/2)$ is the induced term to distinguish positive and negative interactions in negative BPR loss, that is $c_2 = 2$ for positive items and $c_2 = 1$ for negative items. When minimizing this loss, the negative prediction score (i.e., negative preference) relations in items will be $\hat{y}_{ui_+}^{-} < \hat{y}_{ui_-}^{-} < \hat{y}_{uj}^{-}$.

Through these calculations, the difference in predicted ratings among positive interactions, unobserved interactions, and negative interactions will be greater according to the signs of interactions.

Rating Prediction. In addition, we adopt the rating prediction to distinguish fine-grained preferences at each level, which evolves from link prediction, a widely applied task in signed graph representation learning [13]. We utilize an MLP module to predict the rating of user u on item i:

$$\mathcal{L}_{MSE} = \frac{1}{|\mathcal{E}|}\left(\mathrm{ReLU}([\mathbf{e}_u^{+}, \mathbf{e}_i^{+}]\mathbf{W}_{MLP}^{(1)})\mathbf{W}_{MLP}^{(2)} - \omega_{ui}\right)^2 \tag{12}$$

where $\mathbf{W}_{MLP}^{(1)} \in \mathbb{R}^{2d\times 2d}, \mathbf{W}_{MLP}^{(2)} \in \mathbb{R}^{2d\times 1}$ are the trainable weight matrices of MLP layers, $ReLU(\cdot)$ is the activation function.

Orthogonality Constraint We apply an orthogonality constraint to force the positive and negative representation of each user and each item to share none common information:

$$\mathcal{L}_{ort} = \|\mathbf{e}_u^{+} \cdot \mathbf{e}_u^{-}\|^2 + \|\mathbf{e}_i^{+} \cdot \mathbf{e}_i^{-}\|^2 \tag{13}$$

Overall. We simultaneously optimize the above losses. That is, the overall objective function can be written as follows:

$$\mathcal{L} = \mathcal{L}_{BPR}^{+} + \mathcal{L}_{BPR}^{-} + \mathcal{L}_{MSE} + \mathcal{L}_{ort} + \lambda\|\Theta\|_2^2 \tag{14}$$

where λ and Θ represent the strengths of L_2 regularization and the learnable parameters of the model, respectively.

Table 1. Statistics of the datasets. "Ratio" denotes the ratio of positive and negative ratings in the training set.

Dataset	Amazon-Beauty	Amazon-Book	Yelp2021
# Users	22,363	35,736	41,772
# Items	12,101	38,121	30,037
# Interactions	172,188	1,960,674	2,116,215
Density(%)	0.064	0.14	0.16
Ratio	1:0.13	1:0.07	1:0.16

5 Experiments and Analysis

We conduct a series of experiments on three real-world datasets to address the following research questions: **RQ1:** How does LSGRec perform compared with the state-of-the-art traditional CF methods and sign-aware recommendation methods? **RQ2:** How does the performance of LSGRec change when varying the value of the threshold δ when splitting the rating and the number of propagation layers in the UniMo module? **RQ3:** What is the effectiveness of each task in the optimization and the negative preference filter?

5.1 Experimental Settings

In this section, we describe datasets, metrics, baselines, hyperparameters, and other details used in the experiments.

Datasets. Three public datasets are used in our experiments, including (a) Amazon-Beauty[4], (b) Amazon-Book[4], and (c) Yelp2021[5]. We refer to them as *Beauty*, *Book*, and *Yelp* in brief, respectively. Table 1 summarizes the statistics of three datasets. All datasets are split into training, validating, and testing subsets with a ratio of 8:1:1. For the three datasets, we use the threshold δ of 3.5, 3.5, and 2.5 to split the original ratings as binary signals and remove users/items that have less than 5, 20, 20 interactions for Beauty, Book, Yelp respectively. Three common metrics – Precision@K, Recall@K, and NDCG@K are used to evaluate the effectiveness of our method, and K is set to $10, 20$ by default. We additionally calculate the average training time per epoch. 5-fold cross-validation is adopted to ensure the reliability of the experimental results.

Baselines. We compare LSGRec with six competing methods, including three unsigned GCN-based methods, a standard Signed Graph Convolutional Network (SGCN), and two state-of-the-art methods in the sign-aware recommendation.

– **NGCF** [20] explicitly integrates the user-item interactions into the embedding process, learning the topology by graph convolution, and effectively harvests the high-order collaborative signals for recommendation.

[4] https://cseweb.ucsd.edu/ jmcauley/datasets/amazon/links.html
[5] https://www.yelp.com/dataset

- **LightGCN** [6] abandons the feature transformation and nonlinear activation in standard GCN, only retaining the neighbor aggregation and layer combination for collaborative filtering.
- **CAGCN** [21] propose a recommendation-oriented topological metric, Common Interacted Ratio (CIR), and a recommendation-tailored GNN.
- **SGCN** [3] utilizes balance theory to aggregate and propagate the information, modeling positive and negative edges in a whole signed graph.
- **SiReN** [18] models users' preferences and generates positive and negative embeddings for the partitioned graphs.
- **PANE-GNN** [16] is one of the state-of-the-art sign-aware recommendation methods, which incorporates users' positive and negative preferences by distinct message-passing mechanisms and contrastive learning.

We do not include some methods in signed graph neural networks like GS-GNN [13] and SIGAT [10] since they are designed for link prediction.

Hyperparameters. For a fair comparison, we set the embedding dimension to 64, batch size to 1024, and initialized all model parameters with Xavier initializer [5], which is optimized by Adam optimizer [11]. The learning rate is set to 0.005, and *MultiStepLR* is utilized to schedule the learning rate. We test the number of propagation layers L in $[1, 2, 3, 4]$. Besides, we train all methods for 200 epochs and record the best performance according to Recall@10. For all baselines, we follow the original settings in their papers for the best performance.

Table 2. Performance of all comparison methods. The second-best score is underlined, and the top score is highlighted in bold. The final column indicates the percentage of performance improvement relative to the second-best one. The *Secs/Epoch* is the average training seconds per epoch.

Dataset	Metrix(%)	NGCF	LightGCN	CAGCN	SGCN	SiReN	PANEGNN	Ours	Imp.(%)
Beauty	Precision@10	1.286	1.892	1.931	1.149	<u>1.996</u>	1.831	**2.108**	5.611
	Recall@10	5.025	7.673	8.071	4.525	<u>8.762</u>	8.017	**9.265**	5.741
	NDCG@10	3.348	5.017	5.263	2.887	<u>5.802</u>	5.225	**6.143**	5.877
	Precision@20	0.990	1.260	1.317	0.921	<u>1.491</u>	1.346	**1.535**	2.951
	Recall@20	7.726	10.06	11.94	7.160	<u>12.76</u>	11.59	**13.22**	3.605
	NDCG@20	4.187	6.221	6.991	3.706	<u>7.036</u>	6.328	**7.365**	4.676
	Secs/Epoch↓	3.0	2.0	38.5	3.5	4.0	33.5	2.0	-
Book	Precision@10	5.499	6.055	<u>6.081</u>	3.646	5.646	5.921	**6.699**	10.16
	Recall@10	6.676	6.985	7.012	4.237	6.858	<u>7.025</u>	**8.164**	16.21
	NDCG@10	7.937	8.012	<u>8.108</u>	4.822	7.755	8.018	**9.269**	14.21
	Precision@20	4.275	4.842	<u>4.936</u>	3.104	4.588	4.819	**5.367**	8.732
	Recall@20	10.63	11.05	11.12	7.069	10.81	<u>11.14</u>	**12.63**	13.38
	NDCG@20	8.783	9.294	<u>9.473</u>	5.736	8.974	9.261	**10.62**	12.00
	Secs/Epoch↓	39.5	34.0	123.5	40.0	83.5	739.5	38.5	-
Yelp	Precision@10	2.721	2.919	3.700	2.677	<u>4.213</u>	4.059	**4.411**	4.699
	Recall@10	4.033	4.234	4.094	3.336	<u>5.304</u>	5.098	**5.516**	3.997
	NDCG@10	3.872	4.308	4.740	3.705	<u>5.787</u>	5.643	**6.107**	5.529
	Precision@20	2.566	2.614	3.145	2.234	<u>3.487</u>	3.359	**3.641**	4.416
	Recall@20	5.825	6.682	6.920	5.512	<u>8.726</u>	8.282	**8.996**	3.094
	NDCG@20	4.522	4.833	5.747	4.388	<u>6.908</u>	6.643	**7.188**	4.053
	Secs/Epoch↓	65.0	41.5	79.5	76.5	89.5	892.0	69.0	-

5.2 Performance Comparison (RQ1)

The comparative results are summarized in Table 2, from which we can find that our proposed LSGRec outperforms both unsigned and signed baselines. Generally, sign-aware recommendation methods perform better than unsigned methods, implying the value of modeling positive and negative user preferences from the sign of interactions.

For unsigned recommendation methods, LightGCN performs better than NGCF, thanks to its abandonment of feature transformation and nonlinear activation. CAGCN* performs best in unsigned methods because it alleviates the propagation of unreliable interactions based on its CIR value, which removes noisy interactions from the source. For sign-aware baselines, SiReN surpasses PANE-GNN on these two datasets because PANE-GNN adopts graph convolution on the negative graph, while negative edges cannot convey high-order similarity [2, 10]. SGCN obtains the worst performance due to its dependency on the balance theory, which mistakenly treats some nodes as positive neighbors.

It is worth noting that the performances of unsigned methods on *Book* are comparable to the sign-aware methods. This is because the ratio of negative interactions in this dataset is lower than in others, as shown in Table 1. Independent modeling neglects high-order heterogeneous interactions after separating the whole graph, and the negative graph will be a forest instead of a connected graph, resulting in difficulty for negative encoders in capturing effective negative preferences. That is, existing sign-aware methods heavily depend on adequate first-order negative interactions, which may not be met in all real-world recommendation scenarios.

Finally, our LSGRec beats the second-best baselines in terms of three metrics by around 2.9-5.8% on *Beauty*, 8.7-16.2% on *Book*, 3.1-5.5% on *Yelp*, respectively. We attribute these significant improvements mainly to learning the complete negative preferences from direct neighbors and high-order heterogeneous interactions. The comprehensive employment of positive and negative interactions in the signed graph provides valuable positive and negative preferences. Meanwhile, we propose a negative preference filter which can help ensure satisfactory recommendation results. On *Book* dataset, LSGRec outperforms all baselines by a large margin, indicating that even with a small number of negative interactions, our unified modeling method captures correct and precise high-order collaborative signals, improving the recommendations by utilizing comprehensive representations of users and items.

5.3 Hyperparameter Analysis (RQ2)

The Value of Threshold δ. Fig 4 reports the results of the performance comparison. $\delta = 0.5$ means that we do not split the original ratings into binary signals and treat all interactions as positive. On the contrary, $\delta = 4.5$ means that only 5 points are positive interactions.

When $\delta = 0.5$, there are no negative interactions after splitting. The performance on *Beauty* and *Book* reaches its poorest value, emphasizing the importance of modeling negative preferences. When $\delta = 4.5$, only 5 points are positive

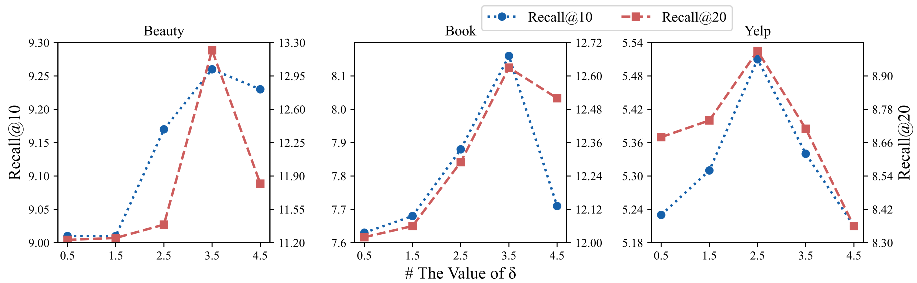

Fig. 4. Recall@10 and Recall@20 across various threshold values (i.e., the value of δ).

interactions. The performance on *Yelp* reaches its lowest point since the overall ratings in this dataset are relatively lower than the other two datasets. A too-high value of δ causes fewer positive interactions, and the positive preferences of users are incomplete and inaccurate.

On each dataset, there is a significant difference in performance between different δ values, indicating that it is important to properly partition positive and negative interactions. This further demonstrates that in addition to positive preferences, modeling user negative preferences is also necessary and effective.

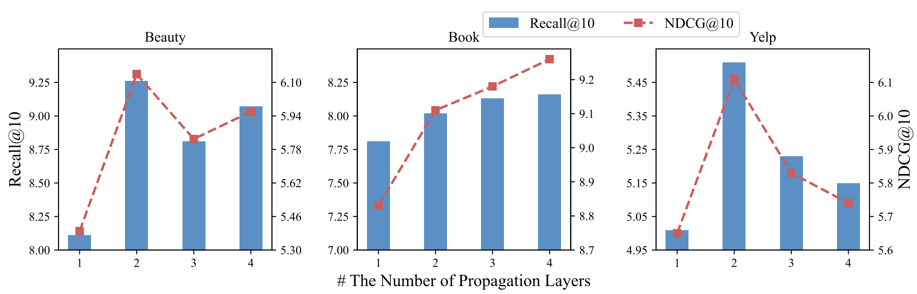

Fig. 5. Recall@10 and NDCG@10 across various numbers of propagation layers.

The Number of Propagation Layers L. We search various numbers of propagation layers (i.e., the value of L) in the range of $\{1, 2, 3, 4\}$. Fig 5 reports the results of the performance comparison. The performance on all datasets reaches its lowest value when L is set to 1, which considers the first-order neighbors only, indicating that higher-order collaborative signals are necessary for capturing user preferences.

When further stacking the propagation layer, we find that UniMo leads to overfitting on *Beauty* and *Yelp* datasets. This might be caused by a too-deep architecture, resulting in over-smoothing. The marginal improvements on these two datasets verify that conducting two propagation layers is sufficient to model the collaborative signals. On *Book* dataset, the performance improves as the number of layers increases. Since the ratio of negative interactions is lower, increasing the number of layers will significantly refine user's negative preferences.

Increasing the depth of UniMo enhances the recommendation cases. When L is adjusted to 2, 4, and 2, respectively, performance on *Beauty*, *Book*, *Yelp* yields the best results. Appropriate L can obtain better representations of users and items by aggregating collaborative signals.

5.4 Ablation Study (RQ3)

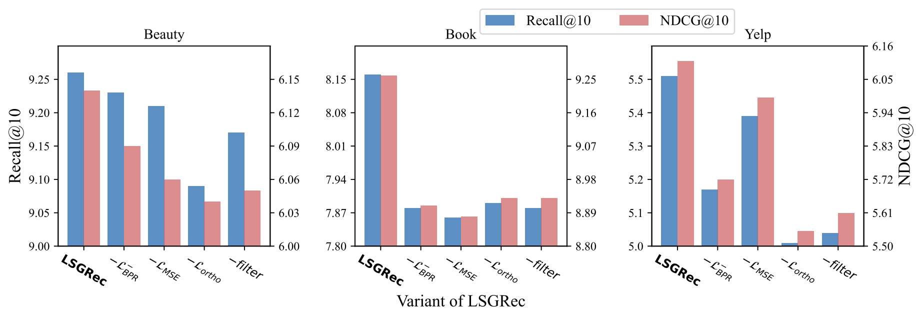

Fig. 6. Recall@10 and NDCG@10 across different variants of our LSGRec. '-' represents 'w/o' for brief.

To explore the effects of each auxiliary task and the negative preference filter, we compare the results on four variants: w/o $\mathcal{L}^{-}_{BPR}$, w/o $\mathcal{L}_{MSE}$, w/o $\mathcal{L}_{ortho}$ and *w/o filter*. Fig 6 intuitively presents Recall@10 and NDCG@10 change in different variants of LSGRec.

Without the negative BPR task, the performance of w/o $\mathcal{L}^{-}_{BPR}$ declines on all three datasets, indicating that controlling the proportion of negative preference among different types of interactions is essential. When discarding the rating prediction task, the performance ofmw/o $\mathcal{L}_{MSE}$ drops on all datasets. It helps the model capture the fine-grained level of interactions and improve the representations' perception of each score on top of the two BPR tasks. In addition, we find that the orthogonality constraint has almost the largest impact on the performance among these tasks (w/o $\mathcal{L}_{ortho}$ has a more significant performance decline). This is because, without the constraint, positive and negative representations may be mixed since they are learned from the same initial embedding. The orthogonality constraint can force the decoupling of positive and negative embeddings during graph convolution.

Due to the lack of the negative preference filter, the performance of *w/o filter* decreases compared to the original model. We believe not filtering before the recommendation may lead to recommending items that users dislike. For example, for a user who likes sports but is disgusted with soccer, the recommender system should filter soccer-related items out before predicting.

6 Conclusion

In this paper, we revisit the limitations of neglecting high-order heterogeneous interactions in previous works and adapting existing signed graph neural networks to recommender systems. Then, we explore the negative preferences between positive neighbors and propose adopting a unified graph neural network to simultaneously model users' positive and negative preferences in a whole signed graph based on the homophily instead of the balance theory. We also introduced a negative preference filter to filter items that users dislike and a multi-task training strategy to optimize learnable parameters. Experiments on three real-world datasets demonstrated that our LSGRec outperforms state-of-the-art sign-aware recommendation methods. To highlight the conceptual design, we have simplified model architectures as much as possible. We leave these more detailed designs and model designs for future work.

7 Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under Grant (No. 62032013, 62102074), the Science and Technology projects in Liaoning Province (No. 2023JH3/10200005).

References

1. Cartwright, D., Harary, F.: Structural balance: A generalization of heider's theory1. In: Psychological Review (1977)
2. Chen, C., Ma, W., Zhang, M., Wang, Z., He, X., Wang, C., Liu, Y., Ma, S.: Graph heterogeneous multi-relational recommendation. In: Proc. of AAAI (2021)
3. Derr, T., Ma, Y., Tang, J.: Signed graph convolutional networks. In: Proc. of ICDM (2018)
4. Franco, S., Marco, G., Chung, T.A., Markus, H., Gabriele, M.: The graph neural network model. IEEE Transactions on neural networks (2008)
5. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proc. of AISTATS (2010)
6. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proc. of SIGIR (2020)
7. Huang, J., Shen, H., Cao, Q., Tao, S., Cheng, X.: Signed bipartite graph neural networks. Proceedings of the 30th ACM International Conference on Information & Knowledge Management (2021)
8. Huang, J., Xie, R., Cao, Q., Shen, H., Zhang, S., Xia, F., Cheng, X.: Negative can be positive: Signed graph neural networks for recommendation. Information Processing & Management (2023)

9. Huang, W., Hao, F., Shang, J., Yu, W., Zeng, S., Bisogni, C., Loia, V.: Duallightgcn: Dual light graph convolutional network for discriminative recommendation. Comput. Commun. (2023)
10. Junjie, H., Huawei, S., Liang, H., Xueqi, C.: Signed graph attention networks. In: Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions (2019)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR (2015)
12. Li, C., Xia, L., Ren, X., Ye, Y., Xu, Y., Huang, C.: Graph transformer for recommendation. In: Proc. of SIGIR (2023)
13. Liu, H., Zhang, Z., Cui, P., Zhang, Y., Cui, Q., Liu, J., Zhu, W.: Signed graph neural network with latent groups. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (2021)
14. Liu, J., Li, D., Gu, H., Lu, T., Zhang, P., Shang, L., Gu, N.: Personalized graph signal processing for collaborative filtering. In: Proceedings of the ACM Web Conference 2023 (2023)
15. Liu, Y., Yang, E., Dang, Y., Guo, G., Liu, Q., Liang, Y., Jiang, L., Wang, X.: Id embedding as subtle features of content and structure for multimodal recommendation. ArXiv (2023)
16. Liu, Z., Wang, C., Xu, J., Wu, C., Zheng, K., Song, Y., Mou, N., Gai, K.: Pane-gnn: Unifying positive and negative edges in graph neural networks for recommendation. arXiv preprint arXiv:2306.04095 (2023)
17. Mao, K., Zhu, J., Xiao, X., Lu, B., Wang, Z., He, X.: Ultragcn: Ultra simplification of graph convolutional networks for recommendation. In: Proc. of CIKM (2021)
18. Seo, C., Jeong, K.J., Lim, S., Shin, W.Y.: Siren: Sign-aware recommendation using graph neural networks. IEEE Transactions on Neural Networks and Learning Systems (2022)
19. Wang, S., Tang, J., Aggarwal, C.C., Chang, Y., Liu, H.: Signed network embedding in social media. In: Proc. of SDM (2017)
20. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: Proc. of SIGIR (2019)
21. Wang, Y., Zhao, Y., Zhang, Y., Derr, T.: Collaboration-aware graph convolutional network for recommender systems (2023)
22. wei Wei, Y., Wang, X., Nie, L., He, X., Hong, R., Chua, T.S.: Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video. In: Proc. of ACM MM (2019)
23. Xu, P., Hu, W., Wu, J., Liu, W., Yang, Y., Yu, P.S.: Signed network representation by preserving multi-order signed proximity. IEEE Transactions on Knowledge and Data Engineering (2023)
24. Yu, J., Xia, X., Chen, T., Cui, L., Hung, N.Q.V., Yin, H.: Xsimgcl: Towards extremely simple graph contrastive learning for recommendation. IEEE Transactions on Knowledge and Data Engineering (2023)
25. Yu, J., Yin, H., Xia, X., Chen, T., zhen Cui, L., Nguyen, Q.V.H.: Are graph augmentations necessary?: Simple graph contrastive learning for recommendation. In: Proc. of SIGIR (2021)
26. Zhang, J., Zhu, Y., Liu, Q., Wu, S., Wang, S., Wang, L.: Mining latent structures for multimedia recommendation. In: Proc. of ACM MM (2021)