Interest-Aware Contrastive-Learning-Based GCN for Recommendation

CHUAN LIN¹, WEI ZHOU², AND JUNHAO WEN²

¹School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China
²School of Big Data and Software Engineering, Chongqing University, Chongqing 400044, China

Corresponding author: Wei Zhou (zhouwei@cqu.edu.cn)

Abstract

Graph convolutional networks (GCNs) have shown great potential in recommender systems. GCN models contain multiple layers of graph convolutions to exploit signals from higher-order neighbors. In each graph convolution, the embedding of a user or item is influenced by its directly connected neighbors. Some of the main problems with this approach are as follows. First, too many graph convolutional layers make different users or items have similar embeddings. Second, the obtained interaction data have some unfavorable characteristics, such as the sparsity of the data, the noise inside the data, and the distribution skewness of the data, that may impair the model’s performance. This paper proposes an interest-aware contrastive-learning-based GCN (IC-GCN) model. IC-GCN applies an interest-aware mechanism, divides users into different subgraphs according to their interests, and performs multilayer graph convolution on the subgraphs, where all collaborative signals received from multi-hop neighbors are positive. Furthermore, IC-GCN takes the contrastive learning task as an auxiliary task, where the interest-aware encoder receives two modified graphs generated by applying the node dropout operator on the full interaction graph. These two graphs generate two sets of embeddings as two additional views of the nodes. The contrastive learning loss function compares these two sets of embeddings. Extensive experiments are conducted to demonstrate the effectiveness of our model.

Index Terms

Neural networks, graph neural networks, big data applications.

I. INTRODUCTION

The focus of collaborative recommendation is to effectively and efficiently learn high-quality vector representations, i.e., embeddings, of users or items from historical interaction data. Graph convolutional networks (GCNs) [1], [2], [3], [4], [5] have been proposed to exploit the collaborative signals acquired from multi-hop neighbors. In each graph convolution layer, the embedding of a node is influenced by the embeddings of its directly connected neighbors. For example, neural graph-based collaborative filtering (NGCF) [2] has proven that exploiting high-order connectivity can help alleviate the sparsity problem in a recommendation system. LightGCN [1] has shown that both feature transformation and nonlinear activation can overcomplicate GCN models. The attribute-aware attentive GCN model [3] can widely exploit the attributes of users. The contrastive knowledge graph attention network [4] can integrate both unsupervised learning and supervised learning to improve the robustness of the constructed model.

However, the traditional GCN approaches encounter over-smoothing problems [6] because of some negative signals derived from multi-hop neighbors. The neighbors that generate these signals are very different from the source node. To utilize the positive collaborative signals of multi-hop neighbors and reject negative collaborative signals, an interest-aware mechanism was proposed for the first time in the interest-aware message passing GCN (IMP-GCN) model [7] so that the multi-hop collaborative signal transmission process can perceive the interests of its source and destination nodes. If the interests of its source node and destination node are the same, the given signal is considered positive, and transmission is allowed. Otherwise, the signal is considered negative, and transmission is rejected. To achieve
this goal, the interest-aware mechanism groups users into different subgraphs according to their interests, adds items directly related to the users in a subgraph to this subgraph, and conducts multilayer graph convolutions inside the subgraphs. As a result, all collaborative signals transferred from one user to another are positive because the two users are in the same subgraph and have the same interests. This alleviates the problem of oversmoothing to some extent.

However, another problem arises when focusing on solutions to the oversmoothing problem. When dividing a whole complete graph into subgraphs, the interest-aware mechanism exacerbates the unfavorable characteristics of interaction data. These characteristics include data sparsity, data noise, and data distribution skewness [8]. (1) Sparsity of data. Interaction data are very sparse, which means that the observed interaction data occupy only a small part of the potential interaction space. As a result, collaborative signals are also sparse, making embedding learning difficult. However, the interest-aware mechanism makes the collaborative signals sparser while dividing the complete graph into smaller subgraphs. Users with the same interest connected in the original full graph may not be connected in a subgraph and will thus receive fewer positive collaborative signals during the graph convolution stage. (2) Noise in the data. The observed interaction data may contain some noise. For example, a user may be misled into clicking on an item and finding it uninteresting. In general, if our interaction graph is sufficiently large, the noise does not impact the correctness of the learned embeddings. However, interest-aware mechanisms generate many subgraphs much smaller than the original graph, making the embedding learning process susceptible to these noises. Specifically, an item node in a subgraph may retain far fewer edges than the same node in the full graph, causing the embedding of this item node to be more susceptible to noisy edges connected to this node. (3) Skewed data distribution. A power-law distribution exists for nodes of all degrees, where a long tail contains many low-degree nodes. Nodes with high degrees account for only a small proportion of all nodes and have too much influence on the learned embeddings of other nodes. Such a skewed data distribution persists when dividing the entire full graph into subgraphs. Interest-aware mechanisms cannot solve this problem and may even amplify it. Note that when we generate subgraphs from the full interaction graph, many nodes lose edges and become part of the long tail in the skewed distribution.

To address the problem of oversmoothing without exacerbating the unfavorable features of interaction data, we propose a new GCN model called the interest-aware contrastive-learning-based GCN (IC-GCN). To handle oversmoothing, we apply an interest-aware mechanism similar to IMP-GCN [7], which divides a complete graph into subgraphs according to the user’s interests and performs multilayer graph convolutions on these subgraphs to generate the final embeddings. Furthermore, our model applies the ideas of multitasking and contrastive learning in graph convolution, which was first proposed in the self-supervised graph learning (SGL) model [8]. In our auxiliary contrastive learning task, our model applies this interest-aware mechanism to two modified graphs to generate two sets of embeddings as two additional node views. These two modified graphs are constructed by performing random node dropping on the original interaction data. For these two additional sets of embeddings, we apply the information-noise contrastive estimation (InfoNCE) loss, which attempts to maximize the similarity between two embeddings of the same node while minimizing the similarity between the embeddings of different nodes. This alleviates the unfavorable characteristics of interaction data to a certain extent. Since the auxiliary contrastive learning task can provide some auxiliary supervision signals for learning, the problem regarding sparse interaction data in subgraphs is weakened [8]. Since each node has multiple views (one view in the main task and two views in the auxiliary contrastive learning task), the data noise problem is alleviated [8]. Furthermore, as a data augmentation operator, node dropout helps us mitigate the influences of high-degree nodes on the embeddings of other nodes [8]. We summarize the contributions of this work as follows.

- We find and explain the shortcoming of the interest-aware mechanism, which exacerbates the problems posed by unfavorable features such as data sparsity, data noise, and data distribution skewness.
- We design a new learning model, interest-aware contrastive-learning-based GCN (IC-GCN), which can address the problem of oversmoothing without amplifying the unfavorable features of interaction data.
- We conduct extensive experiments on three benchmark datasets to demonstrate the superiority of our model IC-GCN and deeply study the properties of our model.

The rest of this article is organized as follows. We first introduce the motivation of our model in the introductory section. Then, we summarize our model’s related work in the related work section. We explain our model in the methodology section. We present our experimental results in the experimental section. Finally, we summarize our model in the conclusion section.

II. RELATED WORK
A. RECOMMENDER SYSTEMS
Recommender systems are some of the most critical information retrieval systems and have yielded outstanding achievements in recent years. Collaborative filtering (CF) [4], [9], [10], [11], [12] can effectively learn user and item embeddings from historical interaction matrices and has become the mainstream approach for recommender systems. Early studies focused on shallow models such as Bayesian personalization ranking (BPR) [13]. In the aspect-aware latent factor model [10], an aspect rating is weighted by aspect importance, which is dependent on the targeted user’s preferences and the targeted item’s features. Later, recommender systems exploited additional information underlying learning...
different tasks. Deep learning is also a promising direction for recommender system research, as it can enhance the expressiveness of models and introduce nonlinearities. Neural network matrix factorization (NeuMF) [14] and Wide&Deep [15] are good examples of applications in the field of deep learning systems. A knowledge-enhanced graph attention network [4] can exploit knowledge about items to generate a knowledge-enhanced session graph (KESG). The multimodal attentive metric learning (MAML) [12] method can model diverse user preferences for various items. In particular, for each user-item pair, MAML applies an attention neural network, which exploits the item’s multimodal features.

B. GRAPH-BASED RECOMMENDATION
Graph-based recommendation is another vital research direction that can exploit collaborative signals from high-order neighbors. Graph convolution networks (GCN) [1], [2], [7], [16], [17], [18], [19] are good examples of representation learning methods for non-Euclidean structures. There are many variants of GCN models. For example, the grand canonical Monte Carlo (GCMC) model [16] is the most straightforward approach, with only one convolutional layer; it can effectively utilize collaborative signals from directed connected nodes. PinSage [18] combines random walks with multiple graph convolution layers on an item-item graph for the image-of-interest recommendation. MEIRRec [17] utilizes metapath-guided neighbors to exploit rich structure information for an intent recommendation. The stacked mixed-order GCN for CF (SMOG-CF) [19] was proposed to directly capture the high-order connectivity between neighboring nodes at any order. After a comprehensive reflection on the complex structures of commonly used GCN models, it is noted that two widely used model designs, feature transformation and nonlinear activation, have limited significance to the final performance of the constructed model. Therefore, Light-GCN [1] was proposed to simplify the traditional GCN model and significantly improve neural graph collaborative filtering (NGCF). It was later noted that indiscriminately exploiting high-order collaborative signals would bring some adverse effects to the GCN model, such as oversmoothing problems, and make different users have similar embeddings. Therefore, an interest-aware message-passing GCN (IMP-GCN) [7] was proposed to ensure that all signals from multi-hop neighbors are positive by grouping users according to their interests before performing high-order graph convolution.

C. SELF-SUPERVISED GRAPH LEARNING
Although there are many GCN-based models, the main research focus concerns supervised learning paradigms for training. However, self-supervised learning based on GCN models has also made significant progress in the orthogonal direction. Self-supervised graph learning (SGL) [8] is a self-supervised GCN model. In SGL, data augmentation operators such as node dropout, edge dropout, or random walks are applied to the original full interaction graph to generate two modified interaction graphs. The preset encoder receives these two modified graphs to generate two sets of embeddings. By comparing these two sets of embeddings, the similarity between multiple embeddings of the same node is maximized, while the similarity between the embeddings of different nodes is minimized to alleviate the problem regarding the unfavorable characteristics of interactive data. These features include data sparsity, data noise, and data distribution skewness.

III. METHODOLOGY
The overall framework of our proposed model is shown in Figure 1. The whole framework can be summarized as one encoder and two learning tasks. We apply an interest-aware mechanism as the only encoder in the model, which receives a user-item bipartite interaction graph as input and generates a set of embeddings as output for a specific learning task to optimize the trainable parameters. Implementing the idea of multitask learning, our model has two tasks. The main task provides the encoder with the original complete interaction graph to generate a set of embeddings and applies a bayesian personalized ranking (BPR) loss [13] to this set of embeddings. The auxiliary contrastive learning task provides the encoder with two modified interaction graphs to generate two sets of embeddings and finally applies the information noise contrastive learning (InfoNCE) loss [5].

A. INTEREST-AWARE ENCODER
1) SUBGRAPH GENERATION MODULE
The overall framework of the interest-aware encoder is shown in Figure 2. The subgraph generation module is the first module of the interest-aware mechanism. This module tries to group different users into different subgraphs according to their interests. It ensures that any two users in a subgraph have the same interests, thus guaranteeing their collaborative signaling is positive. Noting that we do not have explicit ground-truth user attributes that can represent users’ interests, the subgraph generation module essentially solves an unsupervised learning task. Therefore, before the training phase, we need to specify the number of subgraphs $N_s$. In our work, this hyperparameter $N_s$ is always set to 3. In this module, we use a multilayer perceptron (MLP) structure with three layers, receive the sum of the user’s first-layer and second-layer embeddings as input, and generate a predictor vector of length $N_s$ as output. The predictor vector of a user determines the final group of this user.

$$
\begin{align}
O_1 &= \sigma \left( W_1 e^0_u + e^1_u \right) + b_1 \\
O_2 &= \sigma \left( W_2 O_1 + b_2 \right) \\
O_3 &= W_3 O_2 + b_3
\end{align}
$$

where $e^0_u$ is the ID embedding of a user, and $e^1_u$ is the feature embedding obtained for the user through the aggregation of the local neighbors in the graph. In other words, $e^0_u$ and $e^1_u$ are the user’s first-level embedding and second-level embedding, respectively. $O_1$, $O_2$, and $O_3$ are the output vectors of the first-layer, second-layer and third-layer perceptron, respectively,
with lengths of $d$, $d$, and $N_s$. $W_1$, $W_2$, and $W_3$ are the trainable weights for the first-, second- and third-layer perceptrons, respectively. $W_1 \in \mathbb{R}^{d \times d}$, $W_2 \in \mathbb{R}^{d \times d}$, and $W_3 \in \mathbb{R}^{d \times N_s}$. $b_1$, $b_2$, and $b_3$ are the bias vectors of the first-, second- and third-layer perceptrons, respectively. $b_1 \in \mathbb{R}^{1 \times d}$, $b_2 \in \mathbb{R}^{1 \times d}$, and $b_3 \in \mathbb{R}^{1 \times N_s}$. $\sigma$ is the activation function. The leaky rectified linear unit (LeakyReLU) function is used because it can encode both positive and negative signals. Finally, we use the output vector $O_3$ of the last layer to determine the group to which this user belongs. $O_3$ is a predictor vector, where the position of the largest element represents the group to which the user belongs. After dividing all users into different subgraphs, we add all items directly related to the users of one subgraph to this subgraph, forming the final subgraph. Therefore, each user belongs to only one subgraph. An item may appear in different subgraphs because it may be connected to multiple users with different interests in the original interaction graph.

2) INTEREST-AWARE MESSAGE PASSING STRATEGY

After applying the subgraph generation module, we obtain multiple subgraphs. Let $G_s$ ($s \in \{1, \ldots, N_s\}$) denote a subgraph, where $N_s$ is the total number of subgraphs.

The graph convolution of the first layer is directly performed on the complete interaction graph since all the collaborative signals transmitted in the first layer of graph convolution are positive. When only one layer of graph convolution is performed, all signals from a node can only be transmitted to its directly connected nodes. Generally, the signals acquired from directly connected neighbors are the most reliable and important signals. The following is a detailed introduction to the first-order graph convolution:

$$
\begin{align*}
    e_{iu}^1 &= \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|\sqrt{|\mathcal{N}_i|}}} e_{ui}^0 \\
    e_{ii}^1 &= \sum_{i \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|\sqrt{|\mathcal{N}_i|}}} e_{ii}^0
\end{align*}
$$

(2)

where $e_{iu}^0$ and $e_{ii}^0$ represent the ID embeddings of user $u$ and item $i$, respectively. $e_{iu}^1$ and $e_{ii}^1$ represent the first-level embeddings of target user $u$ and item $i$, respectively. $\mathcal{N}_i$ represents the set of all neighbor users of item $i$, and $\mathcal{N}_u$ represents the set of all neighbor items of user $u$.

High-order graph convolutions are performed independently in different subgraphs to prevent oversmoothing issues. Because any two users in a subgraph have the same interests, all multi-hop collaborative signals transmitted in our higher-order graph convolutions are positive. Note that each user belongs to only one subgraph, and an item may appear in different subgraphs. For graph convolutions in the second layer or later, there is only one embedding per user, and there may be multiple embeddings per item. Let $e_{is}^k$...
C. Lin et al.: Interest-Aware Contrastive-Learning-Based GCN for Recommendation

FIGURE 2. Overall framework of the interest-aware encoder with four layers and two subgraphs for illustration purposes. The first-order graph convolution is based on the complete graph, while the high-order graph convolutions are based on the subgraphs. Note that a user can belong to only one subgraph, while an item can belong to multiple subgraphs.

denote the embedding of item $i$ in subgraph $G_s$ after $k$ convolution layers, and let $e_u$ denote the embedding of user $u$ after $k$ convolution layers. High-order propagation in IMP-GCN is defined as:

$$
\begin{align*}
    e_{u}^{k+1} &= \sum_{i \in N_u} \frac{1}{\sqrt{|N_u| \sqrt{|N_i|}}} e_{i}^{k} \\
    e_{i}^{k+1} &= \sum_{u \in N_i} \frac{1}{\sqrt{|N_i| \sqrt{|N_u|}}} e_{u}^{k}
\end{align*}
$$

where $N_u$ represents the set of all neighbor items of user $u$, and $N_i$ represents the set of all neighbor users of item $i$ in subgraph $G_s$.

As a result, all signals transmitted in our high-order graph convolutions are positive, thus preventing the problem of oversmoothing to some extent. Furthermore, it is reasonable for items to have different embeddings in different subgraphs, since in reality, an item may simultaneously exhibit different characteristics when serving different populations. After the graph convolution at layer $k$, the final representation embedding of item $i$ is the combination of its embeddings in different subgraphs:

$$
\mathcal{E}_{i}^{k} = \sum_{s \in N_i} \mathcal{E}_{is}^{k}
$$

We need to combine all layers’ embeddings to form the final embedding for prediction or loss function optimization.

The combination formula is as follows:

$$
\begin{align*}
    e_{u} &= \sum_{k=0}^{K} \alpha_k e_{u}^{k} \\
    e_{i} &= \sum_{k=0}^{K} \alpha_k e_{i}^{k}
\end{align*}
$$

In our work, $\alpha_k$ is uniformly set to $1/(K + 1)$. We may explore other options for $\alpha_k$ in future work.

To take better advantage of the high computational performance of GPUs, we reimplement our algorithm in matrix form. After we treat each embedding as a row vector and aggregate all embeddings (including users and items) in a row-by-row manner, we can derive the matrix form of the embedding set, which has a shape of $(N_{users} + N_{items}) \times d$. Let $E^0$ be the ID embedding matrix of all users and items, $E^k$ be the embedding matrix after $k$ graph convolution layers, and $E_s^k$ be the embedding matrix after $k$ graph convolution layers in subgraph $G_s$. Then, the first graph convolution layer can be expressed as:

$$
E^1 = L \times E^0
$$

where $L$ is the Laplacian matrix of the full user-item interaction graph. More specifically, the shape of the Laplacian matrix is $(N_{users} + N_{items}) \times (N_{users} + N_{items})$, which is the same shape as the adjacency matrix of the user-item interaction graph. If user $u$ and item $i$ are not adjacent, the element at the
corresponding position in the Laplacian matrix $L$ is 0. If they are adjacent, the element at the corresponding position in the Laplacian matrix is $\frac{1}{\sqrt{|N_u| \cdot |N_i|}}$.

High-order graph convolutions can be expressed as:

$$E^{k+1} = L \times E^k$$

(7)

where $L_s$ is the Laplacian matrix of subgraph $G_s$. Specifically, if user $u$ and item $i$ is not adjacent or user $u$ does not belong to this subgraph, the element at the corresponding position in the Laplacian matrix is 0. If they are adjacent and user $u$ belongs to this subgraph, the element at the corresponding position in the Laplacian matrix is $\frac{1}{\sqrt{|N_u| \cdot |N_i|}}$. Such a matrix can be obtained by processing the original Laplacian matrix with the mask matrix $M$:

$$L_s = L \odot M_s$$

(8)

The shape of the mask matrix is also $(N_{users} + N_{items}) \times (N_{users} + N_{items})$. For subgraph $G_s$, the mask matrix is $M_s$. If the row number of a row corresponds to a user who does not belong to this subgraph, then the values in this row in matrix $M_s$ are all 0; if the column number of a column corresponds to a user who does not belong to this subgraph, then the values in this column in matrix $M_s$ are also all 0. The values in the positions left over are all 1.

At each graph convolutional layer $k$, we aggregate the embedding matrices of all subgraphs to obtain the final embedding matrix of layer $k$:

$$E = \sum_{k \in K} E^k$$

(9)

Finally, we combine the embedding matrices of all convolutional layers to obtain the final embedding matrix for the entire interest-aware encoder:

$$E = \alpha_k E^k + \alpha_{k-1} E^{k-1} + \cdots + \alpha_2 E^2 + \alpha_1 E^1 + \alpha_0 E^0$$

(10)

With graph convolution in this matrix form, we can put the embedding matrix into the GPU when running the model. As a result, the algorithm has not changed. Still, we can take advantage of the parallelism of matrix computations in the GPU, which significantly improves the model’s efficiency and reduces the time required to train the model during each use.

For prediction, we utilize the inner product of the learned final embeddings of the users and items:

$$\hat{r}_{ui} = e_u^T e_i$$

(11)

Note that there are other interactive prediction functions, such as the Euclidean distance. We may verify their effectiveness in the future.

**B. MAIN SUPERVISED LEARNING TASK**

Similar to other top-$n$ recommendation efforts that attempt to extract a set of top-ranked items that best match user preferences from all item spaces, IC-GCN applies a pairwise learning approach, namely, the BPR loss, as our main supervised learning task, which attempts to rank observed interactions higher than unobserved interactions. Each factor of the BPR loss is based on a triple $u, i^+, i^-$. Interactions are observed between $u$ and $i^+$ but not between $u$ and $i^-$. The objective function formula of the main supervision task is:

$$L_{main} = \sum_{(u, i^+, i^-) \in O} -\ln \frac{\exp(s_{ui}^+)/\tau}{\sum_{v \in U} \exp(s_{ui}^v)/\tau} + \lambda_{reg} ||\Theta||_2$$

(12)

$O = \{(u, i^+, i^-) \in R^+, (u, i^-) \in R^-\}$ denotes the training set, $R^+$ denotes the observed interaction between users $u$ and $i^+$ in the training dataset, and $R^-$ is a collection of sampled unobserved interactions. $\lambda_{reg}$ and $\Theta$ denote the regularization weight and all model parameters, respectively. $L_2$ regularization prevents overfitting.

**C. AUXILIARY CONTRASTIVE LEARNING TASK**

To perform the auxiliary contrastive learning task, at the beginning of each epoch, we apply the node dropout operator on the original full interaction graph as a data augmentation method and generate two modified graphs, which are then passed to the interest-aware encoder to generate two sets of additional embeddings. The node dropout operator drops each node and its connected edges from the graph with a probability of $p_{drop}$. Specifically, the node dropout process can be modeled as follows:

$$s_1(G) = (M^0 \circ V, E), s_2(G) = (M'' \circ V, E)$$

(13)

where $M', M'' \in \{0, 1\}^{V}$ are two mask vectors applied to node set $V$ for generating two subgraphs. Note that the edge set $E$ also changes as some nodes are dropped. Therefore, this augmentation step is expected to identify influential nodes from different augmented views and make the representation learning process less sensitive to structural changes.

In each epoch, once we have established two sets of additional embeddings, we apply a pairwise learning method to them, treating the embeddings of the same node as positive pairs (i.e., $(e_u^+, e_v^+)[u \in U]$), where $U$ is the set of user nodes) and the embeddings of different nodes as negative pairs (i.e., $(e_u^+, e_v^-)[u \in U, v \notin v]$). Auxiliary supervision signals from the positive pairs encourage agreement between views of the same node, while auxiliary supervision signals from the negative pairs encourage divergence between views of different nodes. Thus, we take the contrastive loss, InfoNCE, as follows:

$$L_{cl}^{user} = \sum_{u \in U} -\log \frac{\exp(s(e_u^+, e_v^+))/\tau}{\sum_{v \in U} \exp(s(e_u^+, e_v^v))/\tau}$$

(14)

where $s()$ measures the similarity between two vectors and is set as a cosine similarity function; $\tau$ is a hyperparameter called temperature in the softmax function. Likewise, we construct a contrastive learning loss $L_{cl}^{item}$ for items. Combining these two losses, we obtain the objective function of the auxiliary contrastive learning task:

$$L_{cl} = L_{cl}^{user} + L_{cl}^{item}$$

(15)
TABLE 1. Basic statistics of the experimental datasets.

| Dataset | Users | Items | Interactions | Density |
|---------|-------|-------|--------------|---------|
| KS10    | 14356 | 15883 | 365797      | 0.00161 |
| H&K     | 66519 | 28237 | 551681      | 0.00029 |
| Gowalla | 29858 | 40981 | 1027370     | 0.00084 |

D. MULTITASK LEARNING

Our multitask learning paradigm includes two tasks: the main supervised task and the auxiliary contrastive learning task. In the main supervised task, the interest-aware mechanism acts as an encoder that generates embeddings by processing the complete user-item bipartite graph and utilizes this set of embeddings to generate a BPR loss. In the auxiliary contrastive learning task, the interest-aware mechanism again acts as an encoder by processing two incomplete user-item bipartite graphs to generate two additional sets of embeddings. It utilizes these two sets of embeddings to generate a contrastive learning loss (CL loss) as our auxiliary contrastive learning task. The final objective loss function is as follows:

\[ L = L_{main} + \lambda_{cl}L_{cl} \]

where \( \lambda_{cl} \) is the coefficient of contrastive learning.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

1) DATA DESCRIPTION

To evaluate the effectiveness of our interest-aware contrastive-learning-based GCN (IC-GCN) model, we conduct extensive experiments on three benchmark datasets, KS10, H&K (Home&Kitchen) and Gowalla, which have been widely used in previous works. As in previous studies, each dataset is preprocessed under a 10-core setting, retaining users and items with at least 10 interactions. The statistics of the two datasets are shown in Table 1. We can see that these datasets exhibit significant sparsity and scale differences, making the findings more general.

We randomly split each dataset into a training set and a test set for each user at a ratio of 8:2. As in previous work, observed interactions are considered positive interactions. In contrast, unobserved interactions are considered negative interactions.

2) EVALUATION METRICS

For each user, we consider the items that the user has interacted with as positive items and the items that the user has not interacted with as negative items. Under this principle, we apply four widely used top-n recommendation metrics for evaluation purposes: recall, NDCG, precision, and hits. The performance in terms of each metric is evaluated against the top-1 and top-20 recommendation results. We average over all users to generate the final values in the table.

3) PARAMETER SETTINGS

For a fair evaluation, all hyperparameters are as identical as possible. For example, both IC-GCN and IMP-GCN implement an interest-aware mechanism and have the same parameters for this mechanism. All models’ embedding sizes are fixed at 64, and the embedding parameters are all initialized using the Xavier method [20]. In all models, we use Adam [21] to optimize the parameters and use a default learning rate of 0.001, a default minibatch size of 2048 and a \( L_2 \) regularization coefficient \( \lambda_{reg} \) of \( 1e^{-4} \). The early stopping and verification strategy is the same as in LightGCN.

Compared to IMP-GCN, our IC-GCN model has only three additional hyperparameters: the contrastive learning coefficient \( \lambda_{cl} \), the temperature coefficient \( \tau \) in the InfoNCE loss function, and the drop ratio \( p_{drop} \). After carefully adjusting the parameters, the temperature coefficient \( \tau \) is 0.5, the drop rate \( p_{drop} \) is 0.05, and the contrastive learning coefficient \( \lambda_{cl} \) is \( 1e^{-8} \).

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

1) DATA DESCRIPTION

To evaluate the effectiveness of our interest-aware contrastive-learning-based GCN (IC-GCN) model, we conduct extensive experiments on three benchmark datasets, KS10, H&K (Home&Kitchen) and Gowalla, which have been widely used in previous works. As in previous studies, each dataset is preprocessed under a 10-core setting, retaining users and items with at least 10 interactions. The statistics of the two datasets are shown in Table 1. We can see that these datasets exhibit significant sparsity and scale differences, making the findings more general.

We randomly split each dataset into a training set and a test set for each user at a ratio of 8:2. As in previous work, observed interactions are considered positive interactions. In contrast, unobserved interactions are considered negative interactions.

2) EVALUATION METRICS

For each user, we consider the items that the user has interacted with as positive items and the items that the user has not interacted with as negative items. Under this principle, we apply four widely used top-n recommendation metrics for evaluation purposes: recall, NDCG, precision, and hits. The performance in terms of each metric is evaluated against the top-1 and top-20 recommendation results. We average over all users to generate the final values in the table.

3) PARAMETER SETTINGS

For a fair evaluation, all hyperparameters are as identical as possible. For example, both IC-GCN and IMP-GCN implement an interest-aware mechanism and have the same parameters for this mechanism. All models’ embedding sizes are fixed at 64, and the embedding parameters are all initialized using the Xavier method [20]. In all models, we use Adam [21] to optimize the parameters and use a default learning rate of 0.001, a default minibatch size of 2048 and a \( L_2 \) regularization coefficient \( \lambda_{reg} \) of \( 1e^{-4} \). The early stopping and verification strategy is the same as in LightGCN.

Compared to IMP-GCN, our IC-GCN model has only three additional hyperparameters: the contrastive learning coefficient \( \lambda_{cl} \), the temperature coefficient \( \tau \) in the InfoNCE loss function, and the drop ratio \( p_{drop} \). After carefully adjusting the parameters, the temperature coefficient \( \tau \) is 0.5, the drop rate \( p_{drop} \) is 0.05, and the contrastive learning coefficient \( \lambda_{cl} \) is \( 1e^{-8} \).

4) BASELINES

To demonstrate the effectiveness of our proposed approach, we compare it with four competing baseline models: IMP-GCN, SGL, LightGCN, NGCF, and MF.

- **MF**: Matrix factorization (MF) is the cornerstone of graph-based collaborative filtering models. It is also the simplest version of these models, where the trainable parameters are passed directly to the loss function for optimization. No graph convolutions or any other operations are performed on the trainable parameters.
- **NGCF [2]**: Neural graph collaborative filtering (NGCF) is a standard graph-based collaborative filtering model that exploits multi-hop collaborative filtering signals by performing multilayer graph convolutions.
- **LightGCN [1]**: LightGCN is a simplified version of NGCF with the feature transformation and nonlinear activation modules removed. This approach makes GCN-based methods more compact and efficient in making recommendations.
- **SGL [8]**: Self-supervised graph learning (SGL) model is a new learning paradigm, which takes node self-discrimination as the self-supervised task to offer auxiliary signal and increases robustness to interaction noises. There are three variants of SGL. SGL based on node dropout operator (SGL-ND), SGL based on edge dropout operator (SGL-ED) and SGL based on random walk operator (SGL-RW).

- **IMP-GCN [7]**: Interest-aware message-passing GCN (IMP-GCN) classifies nodes into different subgraphs and propagates the collaborative signals inside these subgraphs to avoid negative collaborative signals and ensure that different nodes have different embeddings. Because the flaws of IMP-GCN inspire our IC-GCN model, it must serve as a baseline model for comparison.

B. COMPARISON WITH THE BASELINE METHODS

Table 2 displays the results. The best results are highlighted in bold. From the results, we make the following observations.

- **IC-GCN** consistently outperforms all baselines on all datasets and in terms of all metrics. The significant
TABLE 2. Performance of IC-GCN and the competitors on three datasets. The symbol * denotes that the improvement is significant with \( p - \text{value} < 0.05 \) based on a two-tailed paired t-test.

| Dataset  | IC-GCN | H&K | NGCF | LightGCN | SGL-ND | SGL-ED | SGL-RW | IMP-GCN |
|----------|--------|-----|------|---------|-------|-------|-------|--------|
| KS10     |        |     |      |         |       |       |       |        |
| Recall@1 | 0.08702 | 0.05602 | 0.01644 | 0.02013 | 0.1360 | 0.15610 | 0.05381 | 0.04227 |
| Recall@20| 0.08777 | 0.05805 | 0.01643 | 0.02082 | 0.23663 | 0.15798 | 0.05461 | 0.04367 |
| Precision@1| 0.02220 | 0.02131 | 0.02283 | 0.26321 | 0.26321 | 0.17543 | 0.05876 | 0.04663 |
| Precision@20| 0.23663 | 0.15798 | 0.05461 | 0.04367 | 0.05461 | 0.04367 | 0.05876 | 0.04663 |
| NDCG@1 | 0.09875 | 0.06406 | 0.03257 | 0.02357 | 0.26581 | 0.17944 | 0.06097 | 0.04812 |
| NDCG@20|        |      |      |         |       |       |       |        |

C. ABLATION STUDY

1) EFFECT OF THE DATA AUGMENTATION OPERATOR

For contrastive learning, our model first applies the node dropout operator to the original full interaction graph, generating two modified interaction graphs. The two generated graphs can be seen as different views of the interaction instance. In an ablation study, we explore the impacts of other data augmentation operators, including the edge dropout operator and random walk operator, by recording the performance achieved by IC-GCN after changing the operator. The results are shown in Figure 3. We can see that the node dropout operator is the best of the three.

2) EFFECT OF THE NUMBER OF GRAPH CONVOLUTION LAYERS

To study the effect of the number of layers on our model, we change the number of graph convolution layers from 2 to 6. To maintain the univariate principle and ensure valid experimental results, all other hyperparameters and model structure settings are always set to their default values. In addition, to help the experimental results reveal more truth, we add the LightGCN and IMP-GCN models to our comparison. The experimental results are shown in Figure 4. We omit the results obtained on Gowalla due to space limitations because they show exactly the same trend.

First, both IMP-GCN and IC-GCN greatly outperform LightGCN when the number of layers is 6, the highest option for this parameter, which can tell us that the interest-aware message passing strategy applied in IMP-GCN and IC-GCN helps prevent oversmoothing problems when we perform high-order graph convolution. Grouping users according to their interests before executing high-order graph convolution
helps to ensure that all multi-hop collaborative signals in the transmission step are positive and prevents very different users from having similar embeddings, even if we stack up to 6 layers of graph convolutions. Second, both IC-GCN and IMP-GCN improve or maintain their performance when we stack more graph convolutional layers. In contrast, when we continue to add more graph convolution layers after we already have 5 layers, LightGCN starts to greatly deteriorate. This shows the same idea as the first point: an interest-aware messaging strategy can help alleviate the oversmoothing problem and improve model performance with the help of positive multi-hop collaborative signals. In contrast, when we stack too many graph convolution layers in the LightGCN model, although it can obtain the same positive multi-hop collaborative signals as those of IC-GCN and IMP-GCN, it also obtains many negative multi-hop collaborative signals, which may cause the oversmoothing problem and degrade the overall model performance. Third, IC-GCN outperforms IMP-GCN in most situations, suggesting that the contrastive learning strategy helps to compensate for the shortcomings of IMP-GCN. Regardless of the number of graph convolution layers in the LightGCN model, although it can obtain the same positive multi-hop collaborative signals as those of IC-GCN and IMP-GCN, it also obtains many negative multi-hop collaborative signals, which may cause the oversmoothing problem and degrade the overall model performance. Third, IC-GCN outperforms IMP-GCN in most situations, suggesting that the contrastive learning strategy helps to compensate for the shortcomings of IMP-GCN. Regardless of the number of graph convolution layers, IMP-GCN exacerbates the unfavorable features of interaction data, such as data sparsity, data noise, and distribution skewness, because the interest-aware encoder divides the complete interaction graph into smaller subgraphs. Fourth, IC-GCN and IMP-GCN do not outperform LightGCN when stacking only 2 or 3 graph convolution layers. Therefore, the better performance of LightGCN can show that when we do not try to utilize the multi-hop collaborative signals, a simpler model structure is a better choice.

3) THE EFFECT OF THE NUMBER OF SUBGRAPHS
To investigate the effect of the number of subgraphs, we set the number of subgraphs to 1, 2, 3, and 4 and record the obtained results. The results are shown in Table 3 for KS10 and Table 4 for Gowalla. The best results are highlighted in bold. To ensure that our experiments follow the univariate principle and to guarantee the validity of our conclusions, all other hyperparameters and model structures are set to their default values.

As seen from the results, in this case, 2 is the best choice on KS10, and 3 is the best choice on Gowalla. Using too many subgraphs or too few subgraphs can hurt model performance. It should be noted that when we set the number of subgraphs to \(k\), we assume that there are \(k\) groups of people in total, and an item can exhibit at most \(k\) different characteristics to serve other groups of people. As a result, we can conclude...
As mentioned in the paper introducing SGL, the temperature of subgraphs is the number that can best balance these goals, the number of subgraphs is more significant, there are fewer is smaller, there are more positive collaborative signals. When we can reject more negative collaborative signals because each person is connected to more people, there are many negative collaborative signals. Conversely, when the number of subgraphs is too large, we distinguish people too finely. People with similar interests are grouped into different subgraphs. Although we ensure we can apply data augmentation to subgraphs for performing contrastive learning. Moreover, in the future, we may explore the effect of simultaneously using different numbers of subgraphs instead of fixing the number of subgraphs to 3. Each number of subgraphs involves a corresponding grouping strategy. Two people who are connected in a coarse grouping strategy may not be connected in a fine grouping strategy. As long as they are connected in any grouping strategy, the positive collaborative signal between them will be passed for any two people. However, when they are linked in more grouping strategies, their collaborative signal is stronger. Under this method of simultaneously applying different subgraphs, we can avoid the tedious selection of the appropriate number of subgraphs while ensuring that the transmitted strong multi-hop collaborative signals contain more positive and fewer negative collaborative signals.

**D. HYPERPARAMETER STUDY**

As mentioned in the paper introducing SGL, the temperature parameter \( \tau \) plays a crucial role in hard negative mining. The quality of \( \tau \) can determine whether our model can distinguish between similar nodes. Our hyperparameter study focuses on the effect of different \( \tau \) values in our IC-GCN model. We set the values of \( \tau \) to 0.1, 0.3, 0.5, 0.7, and 1.0 and study the performance metrics of our IC-GCN model at these values. The results are shown in Figure 5. We can see that 0.5 is the best choice for \( \tau \), which aligns with the theoretical proof in the SGL paper.

**V. CONCLUSION AND FUTURE WORK**

In this paper, we propose an interest-aware contrastive-learning-based GCN (IC-GCN) model, which has both an interest-aware mechanism and a contrastive learning task. The interest-aware mechanism performs high-order graph convolutions inside the subgraphs and generates final embeddings for users and items by aggregating the results of all subgraphs. Subgraphs are generated by a designed subgraph generation module that groups users with similar interests into the same subgraph and adds items directly related to a subgraph’s users. The auxiliary contrastive learning task augments the complete user-item graph with a node dropout operator. It runs our previous interest-aware encoder independently on these two modified graphs, thereby constructing multiple views for each node. The contrastive learning task can compare the two sets of additionally generated embeddings. Therefore, our model can solve the oversmoothing problem without exacerbating the problems caused by unfavorable features such as the sparsity, noise, and distribution skewness of the interaction data. The interest-aware mechanism can ensure that all transmitted multi-hop collaborative signals are positive, and the contrastive learning task can counteract the effects of the unfavorable characteristics of the interaction data. Extensive experiments are conducted to demonstrate the superiority of our model.

In future work, we may try to apply contrastive learning at a more fine-grained level. For example, instead of directly using data augmentation for the entire interaction graph, we can apply data augmentation to subgraphs for performing contrastive learning. Moreover, in the future, we may explore the effect of simultaneously using different numbers of subgraphs instead of fixing the number of subgraphs to 3. Each number of subgraphs involves a corresponding grouping strategy. Two people who are connected in a coarse grouping strategy may not be connected in a fine grouping strategy. As long as they are connected in any grouping strategy, the collaborative signal between them will be passed for any two people. However, when they are linked in more grouping strategies, their collaborative signal is stronger. Under this method of simultaneously applying different subgraphs, we can avoid the tedious selection of the appropriate number of subgraphs while ensuring that the transmitted strong multi-hop collaborative signals contain more positive and fewer negative collaborative signals.

**REFERENCES**

[1] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, “LightGCN: Simplifying and powering graph convolution network for recommendation,” in Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., China, J. X. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J. Wen, and Y. Liu, Eds., Jul. 2020, pp. 639–648.

[2] X. Wang, X. He, M. Wang, F. Feng, and T. S. Chua, “Neural graph collaborative filtering,” in Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., 2019, pp. 165–174.

[3] F. Liu, Z. Cheng, L. Zhu, C. Liu, and L. Nie, “An attribute-aware attentive GCN model for attribute missing in recommendation,” IEEE Trans. Knowl. Data Eng., vol. 34, no. 9, pp. 4077–4088, Sep. 2022.

[4] X. Ma, Z. Gao, Q. Hu, and M. Abdelhady, “Contrastive knowledge graph attention network for request-based recipe recommendation,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Singapore, May 2022, pp. 3278–3282.

[5] M. Gutmann and A. Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in Proc. 13th Int. Conf. Artif. Intell. Statist., vol. 9, Y. W. Teh and D. M. Titterington, Eds. Sardinia, Italy: Chia Laguna Resort, 2010, pp. 297–304.

[6] S. Rendle and C. Freudenthaler, “Improving pairwise learning for item recommendation from implicit feedback,” in Proc. 7th ACM Int. Conf. Web Search Data Mining, New York, NY, USA, B. Carterette, F. Diaz, C. Castillo, and D. Metzler, Eds. Feb. 2014, pp. 273–282.
[7] F. Liu, Z. Cheng, L. Zhu, Z. Gao, and L. Nie, “Interest-aware message-passing GCN for recommendation,” in Proc. Web Conf., S. Ljubljana, J. Leskovec, M. Grobelnik, M. Najork, J. Tang, and L. Zia, Eds. Apr. 2021, pp. 1296–1305.

[8] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, “Self-supervised graph learning for recommendation,” in Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., Toronto, ON, Canada, F. Diaz, C. Shah, T. Suel, P. Castells, R. Jones, and T. Sakai, Eds. Jul. 2021, pp. 726–735.

[9] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He, “Bias and debias in recommender system: A survey and future directions,” 2020, arxiv:2010.03240.

[10] Z. Cheng, Y. Ding, L. Zhu, and M. S. Kankanhalli, “Aspect-aware latent factor model: Rating prediction with ratings and reviews,” 2018, arxiv:1802.07938.

[11] Z. Li, Z. Cui, S. Wu, X. Zhang, and L. Wang, “Fi-gnn: Modeling feature interactions via graph neural networks for CTR prediction,” in Proc. 28th ACM Int. Conf. Inf. Knowl. Manage., Beijing, China, W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He, and J. X. Yu, Eds. Nov. 2019, pp. 539–548.

[12] F. Liu, Z. Cheng, C. Sun, Y. Wang, L. Nie, and M. Kankanhalli, “User diverse preference modeling by multimodal attentive metric learning,” in Proc. 27th ACM Int. Conf. Multimedia, Nice, France, L. Amsaleg, B. Huet, M. A. Larson, G. Gravier, H. Hung, C. Ngo, and W. T. Ooi, Eds. Oct. 2019, pp. 1526–1534.

[13] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: Bayesian personalized ranking from implicit feedback,” 2012, arxiv:1205.2618.

[14] H. Morise, K. Atarashi, S. Oyama, and M. Kurihara, “Neural collaborative filtering with multicriteria evaluation data,” Appl. Soft Comput., vol. 119, Apr. 2022, Art. no. 108548.

[15] H. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, “Wide & deep learning for recommender systems,” in Proc. 1st Workshop Deep Learn. Recommender Syst., Boston, MA, USA, A. Karatzoglou, B. Hidasi, D. Tikk, O. S. Shalom, H. Rotiman, B. Shapira, and L. Rokach, Eds. Sep. 2016, pp. 7–10.

[16] C. Su, M. Chen, and X. Xie, “Graph convolutional matrix completion via relation reconstruction,” in Proc. 10th Int. Conf. Softw. Comput. Appl., Kuala Lumpur, Malaysia, Feb. 2021, pp. 51–56.

[17] S. Fan, J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, and Y. Li, “Metapath-guided heterogeneous graph neural network for intent recommendation,” in Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Anchorage, AK, USA, A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, Eds. Jul. 2019, pp. 2478–2486.

[18] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for web-scale recommender systems,” in Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, London, U.K., Y. Guo and F. Farooq, Eds. Jul. 2018, pp. 974–983.

[19] H. Zhang and J. J. McAuley, “Stacked mixed-order graph convolutional networks for collaborative filtering,” in Proc. SIAM Int. Conf. Data Mining (SDM), Cincinnati, OH, USA, C. Demeniconi and N. V. Chawla, Eds. May 2020, pp. 73–81.

[20] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proc. 13th Int. Conf. Artif. Intell. Statist. (AISTATS), Y. W. Teh and D. M. Titterington, Eds. Sardinia, Italy: Chia Laguna Resort, May 2010, pp. 249–256.

[21] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, Y. Bengio and Y. LeCun, Eds. May 2015, pp. 1–15.

CHUAN LIN was born in Putian, Fujian, China, in 2001. He is currently pursuing the B.S. degree in data science with Lanzhou University, Lanzhou, China.

His research interests include data analysis, data mining, and machine learning.

WEI ZHOU received the B.S. and Ph.D. degrees in computer science from Chongqing University, Chongqing, China, in 2010 and 2015, respectively.

He is currently an Associate Professor at the School of Big Data and Software Engineering, Chongqing University. His research interests include recommender systems, machine learning, deep learning, and natural language processing.

JUNHAO WEN received the B.S. and Ph.D. degrees in computer science from Chongqing University, Chongqing, China, in 2003 and 2008, respectively.

He is currently a Professor at the College of Big Data and Software Engineering, Chongqing University. He has authored over 80 refereed journals and conference papers in the above areas. He holds over 30 research and industrial grants and has developed many commercial systems and software tools. His research interests include service computing, cloud computing, and dependable software engineering.