Applied Soft Computing Journal 174 (2025) 113008

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Attention-aware graph contrastive learning with topological relationship for recommendation

Xian Mo [a,c], Jun Pang [b], Zihang Zhao [a,*]

[a] School of Information Engineering, Ningxia University, Yinchuan, 750021, China
[b] Faculty of Science, Technology and Medicine, University of Luxembourg, Esch-sur-Alzette, L-4364, Luxembourg
[c] Ningxia Key Laboratory of Artificial Intelligence and Information Security for Channeling Computing Resources from the East to the West, Ningxia University, Yinchuan, 750021, China

ARTICLE INFO

Dataset link: https://github.com/ZZHCodeZera/AteGCL

Keywords:
Contrastive learning
Attention-aware recommendation
Graph embedding

ABSTRACT

Recommender systems are a vital tool to guide the overwhelming amount of online information for users, which has been successfully applied to online retail platforms, social networks, etc. Recently, contrastive learning has revealed outstanding performance in recommendation by data augmentation strategies to handle highly sparse data. Most existing work fails to leverage the original network's topology to construct attention-aware modules that identify user–item interaction importance for guiding node aggregation while preserving key semantics and reducing noise in the reconstructed graph during data augmentation. In this paper, our work proposes an Attention-aware Graph Contrastive Learning architecture with Topological Relationship (AteGCL) for recommendation. In particular, our AteGCL proposes an attention-aware mechanism with topological relationships to learn the importance between users and items for extracting the local graph dependency, which identifies the importance between nodes by constructing an attention-aware matrix into graph convolutional networks using a random walk with a restart strategy for generating node feature aggregation. We then employ principal component analysis (PCA) for contrastive augmentation and utilize the attention-aware matrix to ease noise from the reconstructed graph generated by PCA and to generate a new view with global collaborative relationships and less noise. Comprehensive experiments on three real-world user–item networks reveal the superiority of our AteGCL over diverse state-of-the-art recommendation approaches. Our code is available at https://github.com/ZZHCodeZera/AteGCL.

1. Introduction

In recent years, various applications generates overwhelming information [1]. For example, online retail platforms [2] consistently generate a lot of purchase and review information from users. The overwhelming online information can easily cause the problem of information overload, which makes it very difficult for users to obtain target information. Recommender systems are a vital tool to guide the overwhelming amount of online information for users, which can enhance user experience by recommending items that users are interested in. It has been widely used in various applications, such as items recommendation on online retail platforms [2], posts recommendation on online social networks [3], and news recommendation on news websites [4], etc.

Graph neural networks (GNNs) [5] have shown outstanding performance in learning user and item representations for recommendation, which can capture high-order relationship on the user–item network by stacking numerous message passing layers. For example, LightGCN [6] adopts a simplified GNN embedding layer to learn user and item embeddings on user–item networks for improving efficiency. DGSR [7] learns the interactive behavior of nodes with preferences using Dynamic GNNs for sequential recommendation. Nevertheless, these approaches do not consider the different importance of user–item interactions on user–item networks, i.e., ignoring the attention mechanism [8] on the user–item network. Averagely aggregating feature information from all user–item interactions can easily lead to inaccurate node representations, which negatively affects recommendation performance. In addition, the multi-hop message passing layers in GNNs can seriously worsen the effect [9].

Drawing on the human visual attention mechanism, attention-based recommendation models [8,10] have emerged. The attention mechanism obtains the focus of the target area and then invests more attention resources to this target area to obtain more detailed information about the target area [11], which can learn the different

* Corresponding author.
E-mail address: zzhsolace@gmail.com (Z. Zhao).

https://doi.org/10.1016/j.asoc.2025.113008
Received 17 April 2024; Received in revised form 10 October 2024; Accepted 6 March 2025
Available online 16 March 2025

importance of user–item interactions and has been successfully applied to the recommendation system. For example, RetaGNN [8] adopts a relational attentive GNN to learn weight matrices on diverse relationships for sequential recommendations. DRAN [10] utilizes an attention architecture to embed the graph-based disentangled user representations for the next point-of-interest recommendation. Recently, TMFUN [12] utilizes an attention mechanism to identify the impact of user–item interaction for multimodal recommendations. However, these existing approaches mainly learn model parameters by adaptive training attention weight [12], which fails to utilize the topology of the original network to construct an attention-aware module into GNNs to identify the importance of user–item interaction for guiding node aggregation and leads to high time complexity [13]. In more details, a traditional GNN-based model mainly employs the Laplacian matrix [5] of the graph to aggregate node features and ignores building an attention-aware module based on the original network topology into the Laplacian matrix, which only reveals the link relationship between nodes and cannot identify the different importance between nodes. Although the GAT-based model [12] can identify the importance between nodes, adaptive training attention weight mechanisms for downstream tasks are not intuitively understandable [14] and has a high time complexity as well [13]. How to utilize the topology of the original network to construct an attention-aware module with topological relationships into GNNs to identify the importance of user–item interaction to improve efficiency and explanation remains as a challenge.

Most GNNs-based recommendation models require adequate high-quality labeled data for supervised model training. However, many real-world recommendation scenarios suffer from data sparsity when learning node representations from very limited interactive data [15]. Contrastive learning (CL) [16,17], a self-supervised learning paradigm (SSL), solves the problem of label scarcity for data augmentation. In detail, contrastive learning aims to enhance node representation by agreement between the created contrastive views by contrasting the defined positive pairs with negative instance counterparts to address data sparsity [18]. For example, Yu et al. [19] add uniform noises to construct contrastive views for recommendation. HCCF [20] generates a new self-supervised hypergraph-enhanced cross-view using local and global collaborative relationships for recommendation. Recently, CCFCRec [21] creates a content collaborative filtering (CF) view and a co-occurrence CF view to generate node embedding for cold-start item recommendation. However, above methods are mainly based on heuristic contrastive view generators, which limits the generality of the model [22]. CVAEs [16] present a disentangled conditional variational autoencoder with a contrastive learning loss function for explainable recommendation. IDCL [23] constructs a disentangled graph contrastive learning framework using intent-wise contrastive learning for intent-aware recommendations. However, these approaches mainly rely on graph augmentation with random perturbation, which may lose useful structural information, thus affecting the quality of the learned representations. To automatically augment data, AutoSSL [24] adopts multiple pretext tasks for automatically augmenting data, while AutoCF [25] executes data augmentation using a unified automated collaborative filtering. Nevertheless, the above methods do not consider the data noise of the reconstructed view. In real-life scenarios, due to the over-recommendation of popular content, users often click on irrelevant content, resulting in noise in the reconstructed view obtained from the original graph by data augmentation. Specially, after the graph convolution operations are performed on user–item graphs, all representations of users and items are explicitly infused with this noise information. As a result, the augmented contrastive views may be contaminated by noise, leading to inaccurate self-supervised signals and a deterioration in recommendation performance. Thus, it will limit the applicability of contrastive learning [9]. More recently, SHT [26] mines the global collaborative relation to augment user representations to denoise user–item interactions using a self-supervised hypergraph transformer framework, while AdaGCL [9] utilizes two trainable view generators for data augmentation to ease data sparsity and noise in recommendation. Subsequently, GO-GCL [27] adopt a Graphormer-based graph contrastive learning strategy to reduce the influence of noise while retaining the inherent semantic structure. However, both methods neglect to directly use original graph topological relationships to ease noise from the reconstructed graph, which may add extra computation and cannot accurately learn user interests, resulting in inaccurate recommendations. In more detail, the original graph topological relationships can directly reveal the intrinsic topological structure of graphs and are easier to obtain. Thus, we can easily use it to revise the topology of the reconstructed graph to reduce noise.

In light of the challenges identified above, our work proposes an effective augmentation method AteGCL (Attention-aware Graph Contrastive Learning Architecture with Topological Relationship) for recommendation. In particular, our AteGCL proposes an attention-aware mechanism with topological relationships to learn the importance between users and items for extracting the local graph dependency, which identifies the importance between nodes by constructing an attention-aware matrix into graph convolutional networks for generating node feature aggregation. In more detail, we count how many times a node walks to another node using a random walk with a restart strategy [28] as the similarity of these two nodes to construct the attention-aware matrix. In this way, the attention-aware matrix preserves the weight information between nodes and can generate weighted node aggregation. Compared with adaptive training attention weight mechanisms, this mechanism can ease the time complexity, benefiting from not requiring training model parameters. To differentiate it from the conventional attention mechanism, we refer to our proposed approach as the attention-aware mechanism. We then employ principal component analysis (PCA) [29] for contrastive augmentation, which injects a global collaborative context into the representational alignment of contrast learning. In this way, it can preserve the key semantic of user–item graph with global collaborative relationships. Further, because the attention-aware matrix reveals the intrinsic topological structure of graphs and are easier to obtain, we directly use it to ease noise from the reconstructed graph generated by PCA and generate a more robust view.

In summary, this paper makes the following contributions:

- We propose an effective augmentation method AteGCL for recommendation, which introduces an attention-aware mechanism with topological relationships to learn the importance between nodes for extracting the local graph dependency.
- We employ PCA for contrastive augmentation, which can preserve the key semantic of user–item graph with global collaborative relationships. Furthermore, we also utilize the attention-aware matrix to ease noise from the reconstructed graph and generate a more robust contrastive view.
- Comprehensive experiments on three real-world user–item networks reveal the superiority of the AteGCL over diverse state-of-the-art recommendation approaches.

2. Related work

GNN-based recommendations. Recently, GNN-based recommendation approaches have shown superior performance in learning user and item representations on user–item networks. NGCF [30] introduces a new Neural Graph Collaborative Filtering architecture to explore the graph topology for recommendations. DirectAU [31] demonstrates that uniformity or alignment can contribute to higher recommendation performance by measure the representation quality using collaborative filtering methods. To deal with the cold start problem, STGCN [32] utilizes a stack of GCNs encoder–decoders in recommender systems. To improve efficiency, LightGCN [6] adopts a simple GNN embedding layer to learn user and item embeddings on user–item networks. Some GNN-based recommendations for specific application domains also emerge. PinSage [33] adopts random walks to construct the convolutions to improve convergence for web-scale recommendations. SSNet [34] presents a self-rescaling graph neural networks to ease the scale distortion for friend recommendations. Recently, DCCF [35] learns global disentangled user representations to distill finer-grained latent factors based on GNNs collaborative filtering models. DGSR [7] learns the interactive behavior of nodes with preferences using Dynamic GNNs for sequential recommendation. However, the above approaches ignore the attention mechanism [8] on the user–item network.

DRAN [10] utilizes an attention architecture to embed the graph-based disentangled user representations for the next point-of-interest recommendation. SOTA [36] uses a multi-head self-attention mechanism and a meta-path guided model for explainable session-based recommendations. To deal with sequential recommendations, RetaGNN [8] adopts a relational attentive GNNs to learn weight matrices on diverse relationships, STOSA [37] designs a self-attention framework to describe item-item position-wise connections, and PMAN [38] learns the sparse pattern of attention using a probabilistic masked attention network. Recently, TMFUN [12] utilizes an attention mechanism to identify the impact of user–item interaction for multimodal recommendations. MOJITO [39] applies attention mixtures strategies that employs the temporal context and past actions to predict items for user. However, these methods mainly learn model parameters by adaptive training attention weight [12], which can lead to high time complexity [13] and are not intuitively understandable explanations [14].

CL-based recommendations. Recent studies have introduced contrastive learning (CL) into graph-based recommendation systems to solve the label sparsity problem with self-supervised signals. SGL [40] enhances GCN using self-supervised learning on user–item networks for recommendations. HCCF [20] generates a new self-supervised hypergraph-enhanced cross-view using local and global collaborative relationships for recommendation. Yu et al. [19] add uniform noises to construct contrastive views for recommendation. To deal with bundle recommendations, CrossCBR [41] aligns the two views to model the cooperative association using cross-view contrastive learning. Recently, Wang et al. [42] apply self-supervised learning to display unique data poisoning vulnerabilities in recommender systems. CCFCRec [21] creates a content collaborative filtering (CF) view and a co-occurrence CF view to generate node embedding for cold-start item recommendation. However, above methods are mainly based on heuristic contrastive view generators, which have a limited ability to adapt to different recommendation tasks.

NCL [43] combines neighbors of a node with graph structure and semantic space into contrastive pairs. AutoSSL [24] automatically augments data using multiple pretext tasks, while AutoCF [25] executes data augmentation using a unified automated collaborative filtering for recommendation. Some CL-based recommendations for specific application domains emerge. CVAEs [16] present a disentangled conditional variational autoencoder with a contrastive learning loss function for explainable recommendation, while IDCL [23] constructs a disentangled graph contrastive learning framework using intent-wise contrastive learning for intent-aware recommendations. Recently, DCRec [44] adopts adaptive conformity-aware augmentation using a debiased contrastive learning for recommendations, while VGCL [45] adopts variational graph reconstruction to yield multiple contrastive views for recommendations. Subsequently, LightGCL [22] uses singular value decomposition to generate a more robust recommendation model for graph contrastive learning. More Recently, GO-GCL [27] reduces the influence while retaining the inherent semantic structure using a Graphormer-based graph contrastive learning method. Nevertheless, these above methods either rely on graph augmentation with random perturbation or ignore data noise in reconstructed views, which may perform limited recommended performance.

3. Problem definition

A user–item interaction graph can be represented as a heterogeneous graph $G = (U, V, Y)$, where $U = \{u_1, \ldots, u_i, \ldots, u_I\}$ with $(|U| = I)$ defines the set of users and $V = \{v_1, \ldots, v_j, \ldots, v_J\}$ with $(|V| = J)$ defines the set of items, respectively. I and J represent the number of users and items. Y defines the user–item interaction matrix, and $Y = [y_{i,j}]_{I\times J} \in \{0, 1\}$ indicates the interaction between user u_i and item v_j. If $y_{ij} = 1$, it means there exists an interaction between user u_i and item v_j, and $y_{ij} = 0$ otherwise.

As discussed in Section 1, it remains a challenge to apply the topology of the original network to construct an attention-aware module into GNNs to identify the importance of user–item interaction to improve efficiency and explanation in recommender systems. Furthermore, how to design data augmentation methods to preserve the key semantic of user–item graph and ease noise from the reconstructed graph to generate a more robust view is also a challenge. Hence, the recommendation problem is formally defined as follows in this paper:

Give a user–item interaction graph $G = (U, V, Y)$, we first construct an attention-aware module with topological relationships S into GNNs to identify the importance of user–item interaction, where S can be obtained by counting the number of times that node u_i walks to node v_j and node v_j walks to node u_j using the random walk with a restart strategy. Then, PCA-guided augmentation generates a new view and original topologies-guided denoising generates a more robust contrastive view for learning user and item representations. Finally, the goal of our recommendation task is to predict the unobserved user–item interactions y_{ij} with the encoded corresponding representations.

4. Methodology

We introduce our proposed AteGCL model in this section. AteGCL is a graph contrastive learning paradigm with an attention-aware mechanism with topological relationships, as shown in Fig. 1. We first introduce an attention-aware mechanism with topological relationships to learn the importance between users and items for extracting the local graph dependency. Then, PCA-guided augmentation generates a new view with global collaborative relationships. The attention-aware matrix is utilized to ease noise from the reconstructed graph generated by PCA.

4.1. Attention-aware graph learning module

In this section, we follow the common collaborative filtering paradigm with attention-aware mechanism to embed user and item nodes into a d-dimensional latent space. In general, the common collaborative filtering method embeds user u_i and item v_j as embedding vectors e_i and e_j with dimension R^d. The embedding matrices for users and items are usually defined as $E^{(u)} \in R^{I\times d}$ and $E^{(v)} \in R^{J\times d}$, respectively. Thus, we construct a local graph embedding propagation layer with attention-aware mechanism to embed users and items as in Formula (1):

$$z^u_{i,l} = \sigma(p(L_{i,*}) \cdot E^{(v)}_{l-1}), \quad z^v_{j,l} = \sigma(p(L_{*,j}) \cdot E^{(u)}_{l-1}), \tag{1}$$

where $z^u_{i,l}$ and $z^v_{j,l}$ denote embedding of user u_i and item v_j in the lth layer. $\sigma(\cdot)$ defines the LeakyReLU [22] with a negative slope of 0.5, and $p(\cdot)$ represents edge dropouts to mitigate the overfitting. L is a Laplacian matrix with attention-aware, which can be constructed by Formula (2):

$$L = D^{-\frac{1}{2}}_{(u)}(A + (A \odot \varepsilon S))D^{-\frac{1}{2}}_{(v)}, \tag{2}$$

where A represents the adjacency matrix of the user–item interaction graph, and $D_{(u)}$ and $D_{(v)}$ represent the diagonal degree matrice for users and items. S is an attention-aware matrix with topological relationships to define the similarity on user–item interaction, which can be constructed by Formula (3). $\odot$ represents the element-wise product. We

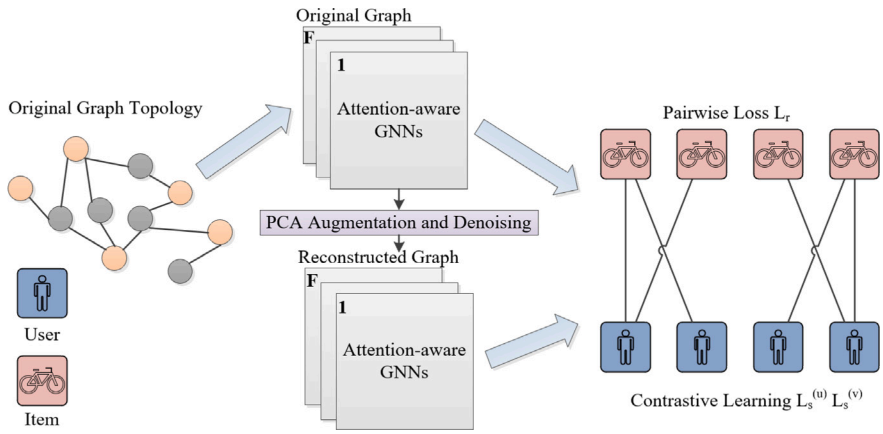

Fig. 1. Overall architecture of AteGCL.

utilize $A \odot \epsilon S$ to eliminate the noisy edges in S. Besides, we employ ϵ to control the contribution of the attention-aware matrix.

$$S(u,v) = |W_{M,R}(u \to v)| + |W_{M,R}(v \to u)| \tag{3}$$

In Formula (3), $S(u,v)$ represents the number of times that node u walks to node v and node v walks to node u using the random walk with a restart strategy [28]. M represents the length of the sampled paths and R represents the number of sampled paths. $|W_{M,R}(u \to v)|$ represents the number of times that from a starting node u walks to the node v through MR steps and $|W_{M,R}(v \to u)|$ represents the number of times that from a starting node v walks to the node u through MR steps. Thus, the $S(u,v)$ preserves the sum of the number of mutual times that from a starting node u (v) walks to the node v (u). To preserve the original node information, we adopt the residual connections, which can be constructed by Formula (4):

$$e^{(u)}_{i,l} = z^{(u)}_{i,l} + e^{(u)}_{i,l-1}, \quad e^{(v)}_{j,l} = z^{(v)}_{j,l} + e^{(v)}_{j,l-1} \tag{4}$$

We sum the embeddings of a node across all layers to get the final embedding. We then adopt the inner product between the final embedding of a user u_i and an item v_j to predict u_i's preference towards v_j, as shown in Formula (5).

$$e^{(u)}_i = \sum_{l=0}^{L} e^{(u)}_{i,l}, \quad e^{(v)}_j = \sum_{l=0}^{L} e^{(v)}_{j,l}, \quad y_{i,j} = e^{(u)\mathrm{T}}_i e^{(v)}_j \tag{5}$$

4.2. Contrastive learning module

As discussed in Section 1, most existing graph contrastive learning approaches either utilize the heuristic-based augmentation techniques or perform stochastic augmentation on the user–item network to generate contrastive views, which cannot well preserve the intrinsic semantic information and are easily biased by noise perturbation. PCA-guided augmentation injects a global collaborative context utilizing Laplacian matrix decomposition operations, which can preserve the intrinsic semantic information and improve the generality and robustness of CL-based recommenders.

In detail, PCA aims to reduce dimensionality while minimizing information loss. Specifically, PCA maps the n-dimensional global features to the k-dimension ($k < n$) orthogonal features, which is called the principal component corresponding to the intrinsic semantic information of networks. It should be emphasized that the new k-dimensional orthogonal features contains as much information as possible from the global n-dimensional features. Hence, we adopt the PCA to extract important global collaborative information for graph augmentation. For the Laplacian matrix with attention-aware L, we perform the de-averaging operation and calculate its covariance matrix $\mathbf{L}$, similar to Ref. [29]. Then, we employ SVD [46] to compute the eigenvalues and eigenvectors of the covariance matrix $\mathbf{L}$, which can be constructed by Formula (6):

$$\mathbf{L} = U \Sigma V^{\mathrm{T}}, \tag{6}$$

where U is an $I \times I$ orthonormal matrix and V is a $J \times J$ orthonormal matrix. Σ stores the singular values of $\mathbf{L}$ and is a diagonal matrix with dimension $I \times J$. The principal components of the matrix Σ usually correspond to the largest singular values. We preserve the largest k value by truncating the list of singular values to reconstruct the matrix $\mathbf{L}$, which can be constructed by Formula (7) (see below). Due to the high cost of performing SVD on large matrices $\hat{\mathbf{L}}$, we use the randomized SVD algorithm [47] to approximate the range of the input matrix with a low-rank orthonormal matrix, which can improve the efficiency.

$$\hat{\mathbf{L}} = U_k \Sigma_k V_k^{\mathrm{T}} \approx ApproxSVD(\hat{\mathbf{L}}, k) = \hat{U}_k \hat{\Sigma}_k \hat{V}_k^{\mathrm{T}} \tag{7}$$

In Formula (7), U_k is an $I \times k$ orthonormal matrix and V_k is a $J \times k$ orthonormal matrix. Σ stores the k largest singular values of $\hat{\mathbf{L}}$ and is a diagonal matrix with dimension $k \times k$. $\hat{U}_k$, $\hat{\Sigma}_k$ and $\hat{V}_k$ are the approximated versions of U_k, Σ_k and V_k. $\hat{\mathbf{L}}$ is a low-rank approximation reconstructed matrix of the matrix L, which indicates the principal components of the graph and preserve the global collaborative information by considering user–item interactions. As discussed in Section 1, reconstructed view obtained from the original graph by data augmentation are always noisy. Thus, we directly explore the attention-aware matrix S to ease noise from the reconstructed graph $\hat{\mathbf{L}}$, which can be constructed by Formula (8)

$$\hat{\mathbf{LD}} = \hat{\mathbf{L}} \odot Random(S), \tag{8}$$

where $\odot$ represents the element-wise product and $\hat{\mathbf{LD}}$ is the de-noised Laplacian matrix. The straightforward idea to ease noisy edges with the least assumptions about $\hat{\mathbf{LD}}$ is to penalize the number of non-zero entries in S. In detail, the $S(u,v)$ preserves the importance between users and items. Therefore, when the zero entries in S is inconsistent with the zero entries in $\hat{\mathbf{L}}$ at the corresponding position, the corresponding position in $\hat{\mathbf{L}}$ is intuitively a noise edge. Hence, we use $Random(\cdot)$ to randomly select a zero entries in S from each row to penalize the corresponding term in the matrix $\hat{\mathbf{L}}$. Thus, we construct a graph embedding propagation layer on the reconstructed user–item relation graph to embed users and items as defined in Formula (9).

$$h^u_{i,l} = \sigma(p(\hat{\mathbf{LD}}_{i,*}) \cdot E^{(v)}_{l-1}), \quad h^v_{j,l} = \sigma(p(\hat{\mathbf{LD}}_{*,j}) \cdot E^{(u)}_{l-1}) \tag{9}$$

Time complexity analysis: Let M denote for the route length of the random walk, R for the number of the random walk, I and J for the number of users and items, respectively. The time complexity to calculate $S(u,v)$ is $O(R \cdot M \cdot (I+J))$. Although AteGCL needs to calculate S, this computational cost is lower than in model training stages since it only requires to be performed once. Our AteGCL model avoids the repetitive updating of parameters in the training process and improves the efficiency of the model compared with adaptive training attention weight mechanisms.

GCL models often suffer from a high computational cost due to the construction of extra views and the convolution operations performed on them during training. However, the low-rank nature of the PCA-reconstructed graph enables the training of our AteGCL to be highly efficient. Let k be the rank of PCA, E be the number of the interactions, the time complexity of performing PCA is $O(k \cdot E)$. Although the AteGCL needs to perform PCA, the computational cost is negligible compared to model training stages since it only requires to be performed once. Our AteGCL models avoid the repetitive graph augmentation during training, which improves model efficiency.

Interpretability analysis: Intuitively, the more times between node u and node v, the more similar they are. The attention-aware matrix S uses the topology of the original network to construct the attention-aware module to generate attention-aware node feature aggregation, which preserves the importance between users and items. Compared with adaptive training attention weight mechanisms, this mechanism can ease the time complexity and increase the interpretability of the model. In more detail, the graph embedding propagation layer aggregates information from the node and its neighboring nodes, and attention-aware matrix S clarify the graph embedding propagation layer on node level with the contributions of each neighboring node. Thus, the interpretability of the model is improved.

4.3. Model optimization

To train the model parameters, we directly adopt InfoNCE loss [48] to contrast PCA-augmented view embeddings with main-view embeddings:

$$L_s^{(u)} = \sum_{i=0}^{I} \sum_{l=0}^{L} -log \frac{exp(s(z_{i,l}^u, h_{i,l}^u/\tau))}{\sum_{i'=0}^{I} exp(s(z_{i,l}^u, h_{i',l}^u/\tau))}, \tag{10}$$

where τ and $s(\cdot)$ represent the temperature and the cosine similarity respectively. We define item InfoNCE loss $L_s^{(v)}$ in the same way. We jointly optimize the contrastive loss with main objective function for the recommendation, which can be constructed by Formula (11):

$$L = L_r + \theta_1 (L_s^{(u)} + L_s^{(v)}) + \theta_2 \cdot \|\Theta\|_2^2, \tag{11}$$

where θ_1 and θ_2 are hyper-parameter for control contribution and Θ are the model parameters. L_r is the main objective function and we adopt a random node dropout [22] to alleviate overfitting, which can be constructed by Formula (12). For a pair of positive items of user i, we define the predicted scores as y_{i,p_s}. Similarly, we define the predicted scores as y_{i,n_s} for a pair of negative items of user i.

$$L_r = \sum_{i=0}^{I} \sum_{s=1}^{S} max(0, 1 - y_{i,p_s} + y_{i,n_s}) \tag{12}$$

5. Evaluation

In this section, we design experiments to answer the following research questions for evaluating the effectiveness and effective of the AteGCL.

- How does our proposed model perform compared to various state-of-the-art recommendation methods?
- How do the key components of the AteGCL contribute to its overall performance?
- How robust is the AteGCL in dealing with noise data?
- How effective is the AteGCL compared to the traditional GAT-based model?
- How do key hyperparameters affect the performance of the AteGCL?

Table 1
Datasets.

Dataset	User	Item	Interaction
Yelp	29,601	24,734	1,517,326
Gowalla	50,821	57,440	1,172,425
Tmall	47,939	41,390	2,357,450

5.1. Experimental settings

5.1.1. Datasets

We select three datasets from the latest version of online applications to organize our experiment, i.e., Yelp, Gowalla, and Tmall.

Table 1 shows some statistical properties of these three datasets.

- **Yelp:** It is a user ratings on business venues dataset from the Yelp platform, which includes 29,601 users, 24,734 items, and 1,517,326 interactions.
- **Gowalla:** It is a check-in dataset collected from Gowalla platform, which includes 50,821 users, 57,440 items, and 1,172,425 interactions.
- **Tmall:** It is an E-commerce dataset collected from the Tmall platform, which includes 47,939 users, 41,390 items, and 2,357,450 interactions.

5.1.2. Baselines

We select various baselines to verify the performance of our proposed AteGCL models, which are described in detail below.

- **DGCF** [49]: It is a graph collaborative filtering method that can disentangle user–item relationships at the finer granularity.
- **HyRec** [50]: It is a hypergraph-based graph collaborative filtering method that employs hypergraph to describe the item relationship.
- **LightGCN** [6]: It adopts a simplified GNN embedding layer to learn user and item embeddings on user–item networks for improved efficiency.
- **SGL** [40]: It enhance GCN using self-supervised learning on user–item networks for recommendations.
- **HCCF** [20]: It generates a new self-supervised hypergraph-enhanced cross-view using local and global collaborative relationships for recommendation.
- **SHT** [26]: It mines the global collaborative relation to augment user representations using a self-supervised hypergraph transformer framework.
- **SimGCL** [19]: It adds uniform noises to construct contrastive views for recommendation.
- **LightGCL** [22]: It eases noise perturbation and generates a more robust recommendation model using graph contrastive learning.
- **AdaGCL** [9]: It utilizes two trainable view generators for data augmentation to ease data sparsity and noise in recommendation.

5.1.3. Parameter settings

We set the layer number of AteGCL L=2; the length of the sampled paths $M = 300$ and the number of sampled paths $R = 20$ for Gowalla and Tmall datasets; $M = 1,000$ and $R = 20$ for the Yelp dataset. the embedding dimension $d = 64$; the rank $k = 5$; the parameter θ_1 is searched from $\{10^{-6}, 10^{-7}, 10^{-8}\}$; the temperature τ is searched from $\{0.3, 0.5, 1, 3, 10\}$; the control contribution parameter ε is searched from $\{0.5, 0.6, 0.7, 0.8, 0.9, 1\}$; the learning rate was set 0.001. The parameters of baselines are set to their default values. We utilize the NDCG@N and Recall@N [9] with the default $N = 20/40$ to evaluate our proposed model and execute 10 times experiments reporting average performance. The experiments are conducted on the Ubuntu 20.04.6 operating system with a Intel(R) Xeon(R) Gold 5317 CPU @ 3.00 GHz machine, 512 GB memory, Tesla A100 80G, and Python 3.9.

Table 2
Recommended performance (Recall@20 and NDCG@20 values).

Datasets	Yelp		Gowalla		Tmall	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
DGCF	0.047	0.040	0.094	0.052	0.024	0.016
HyRec	0.047	0.040	0.090	0.050	0.023	0.016
HCCF	0.063	0.053	0.107	0.064	0.031	0.021
LightGCN	0.048	0.041	0.099	0.059	0.023	0.015
SHT	0.065	0.055	0.123	0.073	0.038	0.026
SGL	0.053	0.044	0.103	0.062	0.027	0.018
SimGCL	0.072	0.062	0.134	0.082	0.047	0.033
LightGCL	0.079	0.067	0.158	0.094	0.053	0.036
AdaGCL	0.085	0.073	0.134	0.083	0.004	0.003
AteGCL	**0.096**	**0.085**	**0.211**	**0.121**	**0.083**	**0.059**

Table 3
Recommended performance (Recall@40 and NDCG@40 values).

Datasets	Yelp		Gowalla		Tmall	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
DGCF	0.077	0.051	0.140	0.067	0.039	0.021
HyRec	0.079	0.052	0.135	0.066	0.035	0.019
HCCF	0.104	0.068	0.153	0.076	0.051	0.028
LightGCN	0.080	0.052	0.143	0.071	0.037	0.020
SHT	0.109	0.070	0.180	0.088	0.064	0.035
SGL	0.086	0.057	0.150	0.074	0.044	0.024
SimGCL	0.116	0.077	0.195	0.097	0.076	0.042
LightGCL	0.129	0.085	0.224	0.110	0.085	0.047
AdaGCL	0.136	0.092	0.192	0.098	0.005	0.003
AteGCL	**0.159**	**0.103**	**0.301**	**0.144**	**0.129**	**0.075**

5.2. Recommended performance

The experimental results show that our model AteGCL achieves the best performance, as shown in Tables 2 and 3. To enhance typographic clarity, we utilize two distinct tables—Tables 2 and 3—to present the recommended performance metrics. In the experiments, we find that the self-supervised models is superior to the CF models. Since the self-supervised model adopts augmented learning tasks, it provides beneficial regularization of the original graph. For example, SGL generate views by stochastic data augmentation and SHT uses the global collaborative relation to augment user representations. while DGCF adopts collaborative filtering method that to disentangle user–item relationships. However, stochastic data augmentation is easily affected by noisy data, resulting in misleading self-supervised signals. In contrast, AteGCL has following main benefits. First, AteGCL introduces an attention-aware mechanism with topological relationships to learn the importance between nodes for extracting the local graph dependency. Second, AteGCL utilizes PCA for contrastive augmentation to preserve the key semantic of user–item graph with global collaborative relationships. And more importantly, AteGCL applies the attention-aware matrix to ease noise and generate a more robust contrastive learning architecture. Overall, the experimental results demonstrate the effectiveness of the proposed contrastive learning in designing data augmentation techniques. As a result, our model beats baselines.

5.3. Ablation study

In this section, we design an ablation study to demonstrate the effect of attention-aware mechanism, denoising strategy, and PCA augmentation. We execute ten experiments to display the average Recal@20 and NDCG@20 values on three datasets.

5.3.1. Effectiveness of attention-aware mechanism

In this section, we design an ablation study to verify the effect of the attention-aware mechanism. We organize 10 experiments to report the average performance on the three datasets. Specifically, we remove the attention-aware matrix S from AteGCL (denoted as AteGCL_1) to verify its contribution to the model. As displayed in Table 4, the average NDCG of AteGCL is 0.4% higher than AteGCL_1 and the average Recall value is 0.2% higher than AteGCL_1 on the Yelp dataset; the average NDCG of AteGCL is 0.3% higher than AteGCL_1 and the average Recall value is 0.9% higher than AteGCL_1 on the Gowalla dataset; the average NDCG and Recall of AteGCL are both 0.3% higher than AteGCL_1 on the Tmall dataset. The experimental results show the effectiveness of the attention-aware mechanism. The reason may be that the AteGCL_1 does not consider the attention mechanism, and the average aggregation of user–item interactions does not effectively learn node embedding. Our AteGCL introduces an attention-aware mechanism with topological relationships identifies the importance between nodes, which can effectively learn user and item embeddings to improve recommendation performance.

Table 4
Ablation study (Recall@20 and NDCG@20 values).

Dataset	Yelp		Gowalla		Tmall	
Variants	Recall	NDCG	Recall	NDCG	Recall	NDCG
AteGCL_1	0.094	0.081	0.202	0.118	0.080	0.056
AteGCL_2	0.091	0.077	0.202	0.117	0.083	0.058
AteGCL_3	0.079	0.067	0.203	0.118	0.079	0.055
Ours	**0.096**	**0.085**	**0.211**	**0.121**	**0.083**	**0.059**

5.3.2. Effectiveness of denoising strategy

In this section, we design a robustness analysis to verify the effectiveness of the denoising strategy $\hat{\mathbf{L}} \odot Random(S)$. Specifically, we remove the component $Random(S)$ (denoted as AteGCL_2) from AteGCL to verify its contribution to the model. As displayed in Table 4, the average NDCG of AteGCL is 0.8% higher than AteGCL_2 and the average Recall value is 0.5% higher than AteGCL_2 on the Yelp dataset; the average NDCG of AteGCL is 0.4% higher than AteGCL_2 and the average Recall value is 0.9% higher than AteGCL_2 on the Gowalla; the average NDCG of AteGCL is 0.1% higher than AteGCL_2 on the Tmall. The experimental results indicate that our denoising strategy can successfully ease noise from the reconstructed graph and generate a more robust contrastive view, resulting in significantly improved recommendation performance.

5.3.3. Effectiveness of PCA augmentation

To verify the effectiveness of the PCA augmentation, we replace the PCA augmentation with the SVD augmentation [46] (denoted as AteGCL_3) to verify its contribution to the AteGCL model. The experimental result as displayed in Table 4 show that the average NDCG of AteGCL is 1.8% higher than AteGCL_3 and the average Recall value is 1.7% higher than AteGCL_3 on the Yelp dataset; the average NDCG of AteGCL is 0.3% higher than AteGCL_3 and the average Recall value is 0.8% higher than AteGCL_3 on the Gowalla dataset; the average NDCG of AteGCL is 0.4% higher than AteGCL_3 and the average Recall value is 0.4% higher than AteGCL_3 on the Tmall dataset. These experimental results demonstrate the effectiveness of the PCA augmentation.

5.4. Efficiency analysis

We evaluate the efficiency of our AteGCL model on three datasets in this section. For fairness, we mainly adopt the baselines based on GAT and GCL models to explore the effectiveness of AteGCL. We select the latest architecture AdaGCL [9] and SHT [26] to compare our model because they use an adaptive training attention weight based on graph contrastive learning to train models. The experimental results show that our model takes about 1,191 s, while the AdaGCL model takes about 9,577 s and the SHT model takes about 1640 s for the Yelp dataset; our model takes about 1,962 s, while the AdaGCL model takes about 12,496 s and the SHT model takes about 2117 s for the Gowalla dataset;

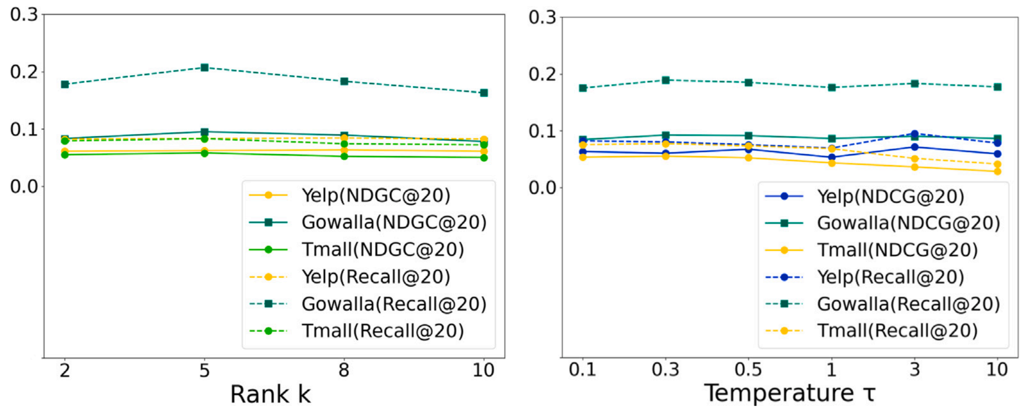

Fig. 2. Hyperparameter analysis for k and τ.

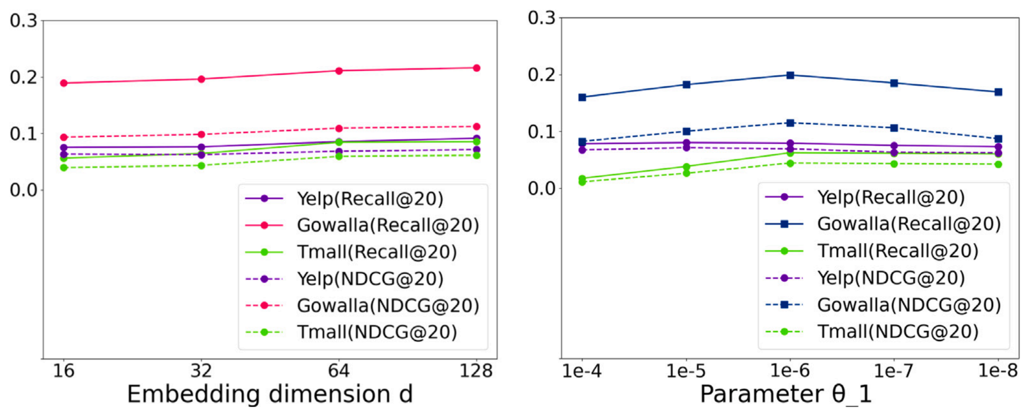

Fig. 3. Hyperparameter analysis for d and θ_1.

our model takes about 3,105 s, while the AdaGCL model takes about 82,800 s and the SHT model takes about 3791 s for the Tmall dataset. Compared with the latest architecture AdaGCL and SHT, our model has demonstrated excellent efficiency.

Furthermore, our model runs 100 epochs to train the model. The attention matrix S takes about 15 epochs of training time, and the PCA augmentation takes about 1 epoch of training time. From this, we can conclude that the computational cost of the PCA augmentation can be negligible. Our model utilizes the topology of the original network to construct an attention-aware module into GNNs, which does not require training model parameters. In denoising, our model directly employs the attention-aware matrix to ease noise from the reconstructed graph, which does not require extra computation. The main reason may be that the PCA augmentation and the attention-aware module with topological relationships can improve the efficiency of recommendation tasks compared to GAT models.

5.5. Hyperparameter analysis

In this section, we perform hyperparameter analysis, and the results are displayed in Figs. 2 and 3. Especially, we evaluate how different the rank k in PCA, the temperature τ in the contrastive loss, the embedding dimension d, and the parameter θ_1 can impact the recommended performance. We perform 10 experiments to report the average Recal@20 and NDCG@20 values on three datasets.

Rank k: We set the rank k to 2, 5, 8, and 10 to verify the recommendation performance of our AteGCL. Our model utilizes k to decide the rank of SVD. In the experiment, we found that a small value of k can obtain the suitable results. The experimental results show in Fig. 2 that when $K = 5/8$, our model can well preserve the key semantic of user–item graph with global collaborative relationships. Considering the balance between performance and computational cost, we set the $K = 5$.

Temperature τ: We search from $\{0.1, 0.3, 0.5, 1, 3, 10\}$ to verify the recommendation performance of this parameter for our AteGCL. The experimental results in Fig. 2 indicate that the performance is relatively stable for different τ. The optimal configuration of τ values varies by datasets.

Embedding dimension d: We set the embedding dimension d to 16, 32, 64, and 128 to verify the recommendation performance of our AteGCL and report the average recall. The experimental results are shown in Fig. 3. As dimension d increases, the performance continues to increase, and the satisfactory result is obtained when $d = 64$. As the dimensions continue to increase, the performance slightly increase. However, the computational cost will also increase. Considering the balance between performance and computational cost, we set the embedding dimension $d = 64$.

Parameter θ_1: We search from $\{10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$ to verify the recommendation performance of this parameter for our AteGCL and report the average recall. This parameter controls the contribution of contrastive learning. The experimental results are shown in Fig. 3, the model achieves the best performance varies by datasets. When θ_1 is in the range $[10^{-6}, 10^{-8}]$, it is conducive to performance improvement. The experimental phenomenon indicates that a larger value may overemphasize the contrastive optimization loss.

6. Conclusion

In this paper, we have proposed an effective augmentation method AteGCL to the graph contrastive learning architecture for recommendation. In particular, our AteGCL advocates an attention-aware mechanism with topological relationships to learn the importance between users and items for extracting the local graph dependency, which identifies the importance between nodes by constructing an attention-aware matrix into graph convolutional networks for generating node feature aggregation. In more detail, the number of times a node walks to another node using a random walk with a restart strategy is used as the similarity of the two nodes to construct the attention-aware matrix. PCA is then applied for contrastive augmentation, which injects a global collaborative context into the representational alignment of contrast learning. Thus, it can preserve the key semantic of user–item graph with global collaborative relationships. Moreover, we utilize the attention-aware matrix to ease noise from the reconstructed graph generated by PCA and generate a more robust view. Comprehensive experiments on three real-world user–item networks reveal the superiority of the AteGCL over diverse state-of-the-art recommendation approaches. Our future work will explore semantic relationships [51–53] of user–item networks to construct attention-aware mechanisms into graph convolutional networks to guide node feature aggregation for a high-quality recommendation.

CRediT authorship contribution statement

Xian Mo: Writing – review & editing, Writing – original draft, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Jun Pang:** Writing – review & editing, Methodology, Formal analysis, Conceptualization. **Zihang Zhao:** Validation, Methodology, Data curation, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Xian Mo has patent licensed to ZL202410203434.5. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work is financed by the Natural Science Foundation of Ningxia (2024AAC05011) and the National Natural Science Foundation of China (62306157).

Data availability

Our code is available at https://github.com/ZZHCodeZera/AteGCL.

References

[1] C. Zhang, S. Xue, J. Li, J. Wu, B. Du, D. Liu, J. Chang, Multi-aspect enhanced graph neural networks for recommendation, Neural Netw. 157 (2023) 90–102, http://dx.doi.org/10.1016/j.neunet.2022.10.001.

[2] J. Wang, R. Louca, D. Hu, C. Cellier, J. Caverlee, L. Hong, Time to shop for valentine's day: Shopping occasions and sequential recommendation in E-commerce, in: Proceedings of the 13th ACM International Conference on Web Search and Data Mining, ACM, Houston, 2020, pp. 645–653, http://dx.doi.org/10.1145/3336191.3371836.

[3] F. Zhang, J. Tang, X. Liu, Z. Hou, Y. Dong, J. Zhang, X. Liu, R. Xie, K. Zhuang, X. Zhang, L. Lin, P.S. Yu, Understanding WeChat user preferences and "wow" diffusion, IEEE Trans. Knowl. Data Eng. 34 (12) (2022) 6033–6046, http://dx.doi.org/10.1109/TKDE.2021.3064233.

[4] J. Yi, F. Wu, B. Zhu, J. Yao, Z. Tao, G. Sun, X. Xie, UA-FedRec: Untargeted attack on federated news recommendation, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, Long Beach, 2023, pp. 5428–5438, http://dx.doi.org/10.1145/3580305.3599923.

[5] F. Terroso-Sáenz, F. Arcas-Túnez, A. Muñoz, Nation-wide touristic flow prediction with graph neural networks and heterogeneous open data, Inf. Fusion 91 (2023) 582–597, http://dx.doi.org/10.1016/j.inffus.2022.11.005.

[6] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Virtual Event, China, 2020, pp. 639–648, http://dx.doi.org/10.1145/3397271.3401063.

[7] M. Zhang, S. Wu, X. Yu, Q. Liu, L. Wang, Dynamic graph neural networks for sequential recommendation, IEEE Trans. Knowl. Data Eng. 35 (5) (2023) 4741–4753, http://dx.doi.org/10.1109/TKDE.2022.3151618.

[8] C. Hsu, C. Li, Retagnn: Relational temporal attentive graph neural networks for holistic sequential recommendation, in: Proceedings of the Web Conference, ACM / IW3C2, Ljubljana, 2021, pp. 2968–2979, http://dx.doi.org/10.1145/3442381.3449957.

[9] Y. Jiang, C. Huang, L. Huang, Adaptive graph contrastive learning for recommendation, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, Long Beach, 2023, pp. 4252–4261, http://dx.doi.org/10.1145/3580305.3599768.

[10] Z. Wang, Y. Zhu, H. Liu, C. Wang, Learning graph-based disentangled representations for next POI recommendation, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Madrid, 2022, pp. 1154–1163, http://dx.doi.org/10.1145/3477495.3532012.

[11] R. Tang, Z. Yong, Y. Mei, X. Li, J. Li, J. Ding, X. Mo, Degrading the accuracy of interlayer link prediction: A method based on the analysis of node importance, International Journal of Modern Physics C (2024) 2442004.

[12] Y. Zhou, J. Guo, H. Sun, B. Song, F.R. Yu, Attention-guided multi-step fusion: A hierarchical fusion network for multimodal recommendation, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Taipei, 2023, pp. 1816–1820, http://dx.doi.org/10.1145/3539618.3591950.

[13] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, Y. Bengio, End-to-end attention-based large vocabulary speech recognition, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, Shanghai, 2016, pp. 4945–4949, http://dx.doi.org/10.1109/ICASSP.2016.7472618.

[14] J. Stacey, Y. Belinkov, M. Rei, Supervising model attention with human explanations for robust natural language inference, in: Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI Press, Virtual Event, 2022, pp. 11349–11357, http://dx.doi.org/10.1609/aaai.v36i10.21386.

[15] W. Liu, J. Su, C. Chen, X. Zheng, Leveraging distribution alignment via stein path for cross-domain cold-start recommendation, in: Proceedings of the 34th Advances in Neural Information Processing Systems, MIT Press, Virtual Event, 2021, pp. 19223–19234.

[16] G. Sun, Y. Shen, S. Zhou, X. Chen, H. Liu, C. Wu, C. Lei, X. Wei, F. Fang, Self-supervised interest transfer network via prototypical contrastive learning for recommendation, in: Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI Press, Washington, 2023, pp. 4614–4622, http://dx.doi.org/10.1609/aaai.v37i4.25584.

[17] X. Mo, J. Pang, B. Wan, R. Tang, H. Liu, S. Jiang, Multi-relational graph contrastive learning with learnable graph augmentation, Neural Networks 181 (2025) 106757.

[18] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, J. Zhang, B. Ding, B. Cui, Contrastive learning for sequential recommendation, in: Proceedings of the 38th IEEE International Conference on Data Engineering, IEEE, Kuala Lumpur, 2022, pp. 1259–1273, http://dx.doi.org/10.1109/ICDE53745.2022.00099.

[19] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, Q.V.H. Nguyen, Are graph augmentations necessary?: Simple graph contrastive learning for recommendation, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Madrid, 2022, pp. 1294–1303, http://dx.doi.org/10.1145/3477495.3531937.

[20] L. Xia, C. Huang, Y. Xu, J. Zhao, D. Yin, J.X. Huang, Hypergraph contrastive collaborative filtering, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Madrid, 2022, pp. 70–79, http://dx.doi.org/10.1145/3477495.3532058.

[21] Z. Zhou, L. Zhang, N. Yang, Contrastive collaborative filtering for cold-start item recommendation, in: Proceedings of the ACM Web Conference 2023, ACM, Austin, 2023, pp. 928–937, http://dx.doi.org/10.1145/3543507.3583286.

[22] X. Cai, C. Huang, L. Xia, X. Ren, Lightgcl: Simple yet effective graph contrastive learning for recommendation, in: Proceedings of the 7th International Conference on Learning Representations, OpenReview.net, Kigali, 2023, pp. 1–15.

[23] Y. Wang, X. Wang, X. Huang, Y. Yu, H. Li, M. Zhang, Z. Guo, W. Wu, Intent-aware recommendation via disentangled graph contrastive learning, in: Proceedings of the 32th International Joint Conference on Artificial Intelligence, ijcai.org, Macao, 2023, pp. 2343–2351, http://dx.doi.org/10.24963/ijcai.2023/260.

[24] W. Jin, X. Liu, X. Zhao, Y. Ma, N. Shah, J. Tang, Automated self-supervised learning for graphs, in: Proceedings of the 10th International Conference on Learning Representations, OpenReview.net, Virtual Event, 2022, pp. 1–20.

[25] L. Xia, C. Huang, C. Huang, K. Lin, T. Yu, B. Kao, Automated self-supervised learning for recommendation, in: Proceedings of the ACM Web Conference 2023, ACM, Austin, 2023, pp. 992–1002, http://dx.doi.org/10.1145/3543507.3583336.

[26] L. Xia, C. Huang, C. Zhang, Self-supervised hypergraph transformer for recommender systems, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, Washington, 2022, pp. 2100–2109, http://dx.doi.org/10.1145/3534678.3539473.

[27] J. Wang, J. Ren, Graphormer based contrastive learning for recommendation, Appl. Soft Comput. 159 (2024) 111626.

[28] Y. Zhang, J. Shen, R. Zhang, Z. Zhao, Network representation learning via improved random walk with restart, Knowledge- Based- Syst. 263 (2023) 110255, http://dx.doi.org/10.1016/j.knosys.2023.110255.

[29] S. Fang, Z. Xu, S. Wu, S. Xie, Efficient robust principal component analysis via block krylov iteration and CUR decomposition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Vancouver, 2023, pp. 1348–1357, http://dx.doi.org/10.1109/CVPR52729.2023.00136.

[30] X. Wang, X. He, M. Wang, F. Feng, T. Chua, Neural graph collaborative filtering, in: Proceedings of the 42th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019, ACM, Paris, 2019, pp. 165–174, http://dx.doi.org/10.1145/3331184.3331267.

[31] C. Wang, Y. Yu, W. Ma, M. Zhang, C. Chen, Y. Liu, S. Ma, Towards representation alignment and uniformity in collaborative filtering, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, Washington, 2022, pp. 1816–1825, http://dx.doi.org/10.1145/3534678.3539253.

[32] J. Zhang, X. Shi, S. Zhao, I. King, STAR-GCN: stacked and reconstructed graph convolutional networks for recommender systems, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, ijcai.org, Macao, 2019, pp. 4264–4270, http://dx.doi.org/10.24963/ijcai.2019/592.

[33] R. Ying, R. He, K. Chen, P. Eksombatchai, W.L. Hamilton, J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, London, 2018, pp. 974–983, http://dx.doi.org/10.1145/3219819.3219890.

[34] X. Song, J. Lian, H. Huang, M. Wu, H. Jin, X. Xie, Friend recommendations with self-rescaling graph neural networks, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, Washington, 2022, pp. 3909–3919, http://dx.doi.org/10.1145/3534678.3539192.

[35] X. Ren, L. Xia, J. Zhao, D. Yin, C. Huang, Disentangled contrastive collaborative filtering, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Taipei, 2023, pp. 1137–1146, http://dx.doi.org/10.1145/3539618.3591665.

[36] J. Zheng, J. Mai, Y. Wen, Explainable session-based recommendation with meta-path guided instances and self-attention mechanism, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Madrid, 2022, pp. 2555–2559, http://dx.doi.org/10.1145/3477495.3531895.

[37] Z. Fan, Z. Liu, Y. Wang, A. Wang, Z. Nazari, L. Zheng, H. Peng, P.S. Yu, Sequential recommendation via stochastic self-attention, in: Proceedings of the ACM Web Conference 2022, ACM, Lyon, 2022, pp. 2036–2047, http://dx.doi.org/10.1145/3485447.3512077.

[38] H. Chen, K. Zhou, Z. Jiang, C.M. Yeh, X. Li, M. Pan, Y. Zheng, X. Hu, H. Yang, Probabilistic masked attention networks for explainable sequential recommendation, in: Proceedings of the 32th International Joint Conference on Artificial Intelligence, ijcai.org, Macao, 2023, pp. 2068–2076, http://dx.doi.org/10.24963/ijcai.2023/230.

[39] V. Tran, G. Salha-Galvan, B. Sguerra, R. Hennequin, Attention mixtures for time-aware sequential recommendation, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Taipei, 2023, pp. 1821–1826, http://dx.doi.org/10.1145/3539618.3591951.

[40] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, X. Xie, Self-supervised graph learning for recommendation, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Canada, 2021, pp. 726–735, http://dx.doi.org/10.1145/3404835.3462862.

[41] Y. Ma, Y. He, A. Zhang, X. Wang, T. Chua, CrossCBR: Cross-view contrastive learning for bundle recommendation, in: A. Zhang, H. Rangwala (Eds.), Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, Washington, 2022, pp. 1233–1241, http://dx.doi.org/10.1145/3534678.3539229.

[42] Y. Wang, Y. Liu, Q. Wang, C. Wang, C. Li, Poisoning self-supervised learning based sequential recommendations, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Taipei, 2023, pp. 300–310, http://dx.doi.org/10.1145/3539618.3591751.

[43] Z. Lin, C. Tian, Y. Hou, W.X. Zhao, Improving graph collaborative filtering with neighborhood-enriched contrastive learning, in: Proceedings of the 7th ACM Web Conference 2022, ACM, Lyon, 2022, pp. 2320–2329, http://dx.doi.org/10.1145/3485447.3512104.

[44] Y. Yang, C. Huang, L. Xia, C. Huang, D. Luo, K. Lin, Debiased contrastive learning for sequential recommendation, in: Proceedings of the ACM Web Conference 2023, ACM, Austin, 2023, pp. 1063–1073, http://dx.doi.org/10.1145/3543507.3583361.

[45] Y. Yang, Z. Wu, L. Wu, K. Zhang, R. Hong, Z. Zhang, J. Zhou, M. Wang, Generative-contrastive graph learning for recommendation, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Taipei, 2023, pp. 1117–1126, http://dx.doi.org/10.1145/3539618.3591691.

[46] H. Li, T. Liu, X. Wu, S. Li, Correlated SVD and its application in bearing fault diagnosis, IEEE Trans. Neural Networks Learn. Syst. 34 (1) (2023) 355–365, http://dx.doi.org/10.1109/TNNLS.2021.3094799.

[47] N. Halko, P. Martinsson, J.A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. 53 (2) (2011) 217–288, http://dx.doi.org/10.1137/090771806.

[48] A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, CoRR (2018) 1–13, URL http://arxiv.org/abs/1807.03748.

[49] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, T. Chua, Disentangled graph collaborative filtering, in: Proceedings of the 43th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Virtual Event, 2020, pp. 1001–1010, http://dx.doi.org/10.1145/3397271.3401137.

[50] J. Wang, K. Ding, L. Hong, H. Liu, J. Caverlee, Next-item recommendation with sequential hypergraphs, in: Proceedings of the 43th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Virtual Event, 2020, pp. 1101–1110, http://dx.doi.org/10.1145/3397271.3401133.

[51] S. Jiang, Y. Qiu, X. Mo, R. Tang, W. Wang, An effective node injection approach for attacking social network alignment, IEEE Transactions on Information Forensics and Security (2024).

[52] Y. Huang, Z. Li, Z. Chen, C. Zhang, H. Ma, Sentence salience contrastive learning for abstractive text summarization, Neurocomputing 593 (2024) 127808.

[53] R. Tang, Z. Yong, S. Jiang, X. Chen, Y. Liu, Y.-C. Zhang, G.-Q. Sun, W. Wang, Network alignment, Physics Reports 1107 (2025) 1–45.