https://doi.org/10.1007/s11704-025-50044-5

REVIEW ARTICLE

Graph contrastive learning view construction methods in recommender systems: a survey

Zhihang YI[1]✉, Hairong WANG[1,2], Fangping CHEN[1], Zhaojing XU[1], Jianling YANG[3]

1. School of Computer Science and Engineering, North Minzu University, Yinchuan 750021, China
2. The Key Laboratory of Images & Graphics Intelligent Processing of State Ethnic Affairs Commission, North Minzu University, Yinchuan 750021, China
3. Ningxia Institute of Meteorological Sciences, Ningxia Meteorological Bureau, Yinchuan 750002, China

Received January 15, 2025; accepted April 9, 2025

E-mail: cnhzyi@gmail.com

© The Author(s) 2025. This article is published with open access at link.springer.com and journal.hep.com.cn

Abstract

Recent advances in deep learning have significantly improved recommendation systems. However, these methods often rely heavily on labeled data, leaving challenges like data sparsity and the cold-start problem unresolved. Self-supervised learning, particularly Graph Contrastive Learning (GCL), has emerged as a powerful approach to mitigate these issues by generating informative views from unlabeled data, attracting considerable attention in recent years. This survey provides a timely and comprehensive review of current GCL-based recommendation methods. First, it introduces a comprehensive framework and taxonomy for view construction in GCL for recommendation systems, dividing it into three main types: structure generation, feature generation, and modality generation. Each category is analyzed in detail, offering insights into their methodologies, strengths, and limitations. Comparative experiments and visualization experiments are conducted on three public datasets, analyzing the complexity of various methods to guide the selection of appropriate approaches. The survey also highlights existing limitations and proposes future research directions along with potential roadmaps to inspire innovative solutions in recommendation systems.

Keywords

contrastive learning; view generation; recommendation; data augmentation

1 Introduction

Currently, “information overload” has become an issue that troubles humanity. To improve the efficiency of user selection when faced with homogenized information, personalized recommender systems have been proposed as an important means to alleviate information overload [1].

In recent years, thanks to breakthroughs in artificial intelligence technology, deep learning has been widely applied to recommender systems in various fields, such as movie recommendations [2], book recommendations [3], and music recommendations [4]. Most of these recommender systems adopt supervised learning methods based on historical interaction records between users and items. However, in practice, the data volume generated by a large number of users and items is substantial. This implies that, on one hand, users do not interact with all items and may only engage with a very small subset; on the other hand, new users lack historical interaction records, which hinders deep learning recommender systems from effectively learning user preferences. These issues lead to data sparsity [5] and cold start problems [6], significantly limiting the generalization ability and application scenarios of recommendation models. A feasible solution is to introduce enhanced or external information to assist recommender systems in collaborative filtering [7], which not only mitigates the problems mentioned above but also improves the explainability and accuracy of recommendation results [7,8]. It can even enable diversified cross-domain recommendations for users [9].

Self-supervised learning (SSL) [10], as an emerging learning paradigm, can utilize a large amount of unlabeled data for training, thereby reducing reliance on manual labels. This approach provides a new opportunity to address data sparsity and cold start problems, serving as a specific implementation strategy for the solutions mentioned above. The core idea of SSL is to design self-supervision tasks that extract potential invariant features from the data itself and apply these features to downstream tasks. Since these supervisory signals are semi-automatically generated, they can effectively overcome the problem of insufficient labels.

Currently, SSL has been successfully applied in multiple domains, including computer vision [11,12], pre-training of language models [13,14], and graph learning [15,16]. Given that the principles of SSL align well with the need for more annotated data in recommender systems, inspired by its significant success in other areas, an

increasing number of studies are exploring the application of SSL in recommender systems, aiming to bring about breakthroughs in this field. In these explorations, contrastive learning (CL) constructs positive and negative sample pairs, using a contrastive loss function to maximize the similarity between positive pairs while minimizing the similarity between negative pairs [17]. This process can effectively capture structural and semantic information within the data, thus generating high-quality representation features [18,19]. In these CL-based methods, Graph Contrastive Learning (GCL) is widely used due to its alignment with the application scenarios of recommendation systems.

Despite this, current GCL-based recommendation methods still have limitations: 1) the long-tail distribution of entities [20] hinders the learning of semantic representations; 2) the data itself contains noise irrelevant to the recommendation topic [21]. To address these problems, existing research works construct various information-enhanced views through GCL to achieve better feature modeling. This paper aims to provide a timely and comprehensive review of these methods. Existing related reviews [10,22] mostly focus on the application of SSL in broad fields, with a few also covering some CL-based recommendation methods [23,24].

However, these reviews offer limited introductions and lack in-depth discussions. Reviews specifically focused on contrastive learning in recommendation [25,26] often aim to summarize a unified framework structure for overall recommendation tasks, which leads to a lack of detailed discussion on the specific differences in view construction for contrastive learning techniques in recommender systems.

Moreover, due to new opportunities and challenges faced by recommender systems today, such as multimodal data [27,28] and large language models [29,30], the construction of views in GCL has found new ground, leading to the emergence of a series of novel methods. In light of this, there is an urgent need for a timely and systematic investigation to discuss the strengths and limitations of existing work, providing guidance and prospects for future research. The main contributions of this paper are as follows:

- We propose a general framework to unify the view construction methods of GCL techniques in recommender systems and categorize existing view generation methods into three types: structure generation, feature generation, and modality generation. For each category, we discuss its concept and representation, the involved methods, as well as its advantages and limitations.
- We present the latest and most comprehensive survey of contrastive learning techniques in recommendation systems, providing detailed descriptions and discussions of different implementation approaches. We introduce relevant background knowledge to help readers understand GCL for recommendations. Additionally, we conduct comparative experiments and visualization experiments on three publicly available datasets to guide the selection of appropriate methods.
- We analyze the complexity and limitations of existing research and discuss the challenges faced by GCL recommendation methods. Based on the identified issues, we propose potential future directions and feasible research roadmaps.

Paper collection: We constructed a comprehensive list of keywords based on the popularity and relevance to our research question, including “Recommender Systems”, “Contrastive Learning”, “Self-Supervised Learning”, “Graph Neural Networks”, “Multimodal”, “Negative Sampling”, and “Large Language Models”. Using Boolean operators “AND” and “OR”, we formulated detailed search strings to query authoritative knowledge bases in Computer Science and Artificial Intelligence, such as SpringerLink, Web of Science, IEEE Xplore, ACM Digital Library, and ScienceDirect. Additionally, we reviewed papers from top-tier conferences in recent years, such as WWW, SIGIR, KDD, AAAI, and CIKM. To avoid missing relevant work, we further examined the reference lists of each paper. Beyond authoritative publication channels, we also searched for preprints on arXiv to ensure that novel works not yet formally published were included when writing this article.

Survey organization: The remainder of this survey is organized as follows. In Section 2, we provide background knowledge on relevant techniques for newcomers to the field. Then, in Section 3, we introduce a unified framework and taxonomy. Section 4 reviews contrastive learning in recommendation systems and discusses existing works according to our taxonomy. In Section 5, we present commonly used datasets in the recommendation domain and conduct comparative experiments on three publicly available datasets. The experimental results demonstrate the applicability of different methods across various scenarios. In Section 6, we perform a complexity analysis of the proposed taxonomy, which helps evaluate their practical usability. Furthermore, we discuss the current limitations of contrastive learning techniques in recommendation systems and explore potential future directions, providing some possible roadmaps. Finally, we conclude the survey in Section 7.

2 Background

In this section, we will introduce the fundamental background knowledge of recommendation based on GCL. First, we clarify the definition of recommender systems and how GCL operates within them. Then, to lay the necessary groundwork for the subsequent content, we introduce GCL techniques and provide basic definitions for the following terms according to their common manifestations in recommender systems: Graph Neural Network (GNN), Knowledge Graph, Multi-modal and Multi-view.

2.1 Contrastive learning in recommender systems

The concept of recommender systems was first introduced by Resnick et al. [31], and it has since evolved into a distinct and highly regarded research area. The mechanism of recommender systems lies in analyzing users’ interest preferences based on their historical interaction behaviors, rating items to be recommended, and finally generating a recommendation list according to the ranking, thereby delivering the most relevant items to the user [32]. For $R \in \mathcal{R}^{|U|\times|V|}$,

where $R_{i,j}$ represents the degree of preference that user U_i has for item V_j, the problem that recommender systems aim to solve is to find the item V_k with the highest preference score for any given user U_i. Formally, a recommender system can be defined as:

$$\forall U_i \in U, V_k = \arg\max_{V_j \in V} f(U_i, V_j), \tag{1}$$

where U denotes the set of users, V denotes the set of items, and f is the scoring function.

From traditional content-based [33,34] and collaborative filtering-based [35,36] recommender systems, to the hybrid recommender systems that were later proposed to combine the advantages of both [37], and then to the novel recommender systems incorporating knowledge graphs [38,39], the underlying logic remains consistent. This logic aims to avoid issues such as data sparsity and cold start while learning as comprehensive feature representations of users and items as possible, to calculate the interaction probabilities between them. On this foundation, various representation learning methods, when integrated with recommender systems, have led to a series of different recommender models. Contrastive learning is one such method, and its general paradigm is illustrated in Fig. 1. The implementation of contrastive learning is grounded in the consensus that meaningful features can be derived from data through self-supervised learning [40]. Its core idea involves augmenting the original data to generate different views that include noise. By maximizing the consistency between different views, typically measured by mutual information (MI), the model learns to capture invariant features inherent in the original data, thereby enhancing its generalization capability. For recommender systems, this means achieving a more accurate understanding of user latent interests, as depicted in the processing flow in Fig. 2. Compared to discriminative and generative models, contrastive learning learns representations by comparing the relationships between input samples [17], rather than learning signals from individual data samples one by one. Let D denote the original data view, and through data augmentation, multiple augmented views D_i can be obtained. Subsequently, by employing a predefined contrastive task, i.e., a contrastive learning loss function, the goal is to maximize the consistency of representations for the same instance across different views while minimizing the consistency for different instances across these views. This process can be formulated as follows:

$$D_i = \mathcal{E}(D), i = 1, 2, \ldots, n, \tag{2}$$

$$\theta^*, c^* = \arg\max_{\theta, c} \mathcal{L}_{cl}(f_c(D_i)), \tag{3}$$

where $\mathcal{E}$ denotes different data augmentation methods, which will be detailed in Sections 3 and 4. θ and c are learnable parameters. f_c is the function for computing view consistency. $\mathcal{L}_{cl}$ represents the contrastive learning loss.

It is worth noting that by tracking the latest literature, we observe that the view construction in contrastive learning for recommender systems has evolved from early single-perspective approaches [41,42] towards a multi-perspectives direction [43–45].

2.2 Contrastive learning loss

As described in Section 2.1, the optimization objective of Contrastive Learning Loss (CL Loss) is typically to maximize the mutual information (MI) between two representations. Specifically, given a pair of positive and negative sample representations $(\mathbf{h}_i, \mathbf{h}_j)$, their MI can be defined as:

$$\mathcal{MI}(\mathbf{h}_i, \mathbf{h}_j) = \mathbb{E}_{P(\mathbf{h}_i, \mathbf{h}_j)} \left[\log \frac{P(\mathbf{h}_i, \mathbf{h}_j)}{P(\mathbf{h}_i) P(\mathbf{h}_j)} \right], \tag{4}$$

where $P(\mathbf{h}_i, \mathbf{h}_j)$ represents the joint probability distribution, while $P(\mathbf{h}_i)P(\mathbf{h}_j)$ denotes the product of the marginal probability distributions. By measuring the ratio of the joint distribution to the

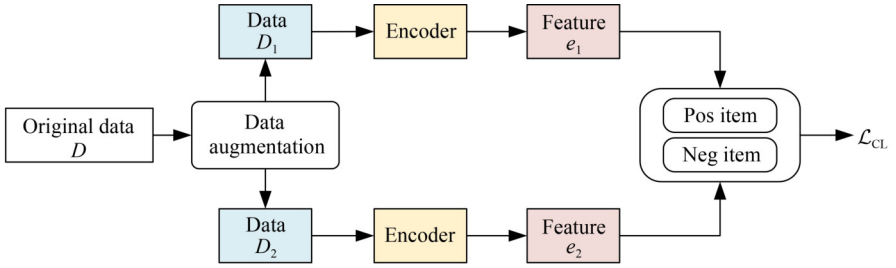

Fig. 1 General paradigm of contrastive learning

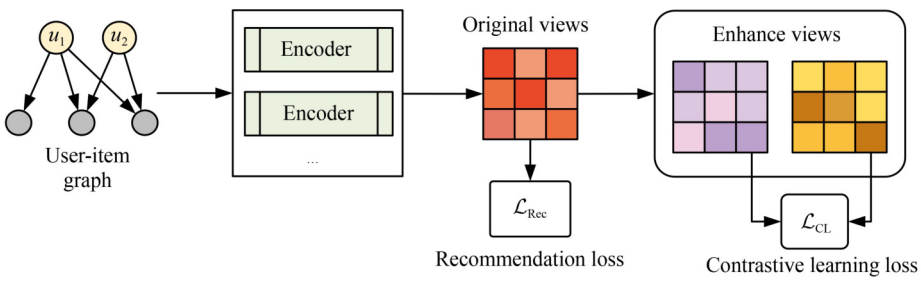

Fig. 2 Recommendation based on contrastive learning

independent distribution and performing a weighted average over all possible scenarios under the joint distribution, the dependency relationship between $\mathbf{h}_i$ and $\mathbf{h}_j$ can be established. Specifically, if $\mathbf{h}_i$ and $\mathbf{h}_j$ are independent, this ratio equals 1, resulting in zero mutual information; conversely, if they exhibit a dependency relationship, the ratio exceeds 1, yielding a positive mutual information value.

One of the CL losses is the InfoNCE loss [46], which is the most widely used lower bound of MI. InfoNCE is an improved version of the Noise-Contrastive Estimator (NCE) [47] that incorporates the softmax function. Given a set of samples $\{\mathbf{h}_i\}_{i=1}^{N}$, InfoNCE assists the model in identifying positive samples while treating all other samples as negative. The InfoNCE loss for the positive pair is given by the following formula:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(s(\mathbf{h}_j, \mathbf{h}_j))/\tau}{\sum_{k=1,k\neq i}^{N} \exp(s(\mathbf{h}_i, \mathbf{h}_k)/\tau)}, \tag{5}$$

where $s(\cdot)$ denotes the similarity function, typically represented by the cosine similarity. τ is the temperature parameter that controls the sharpness of the softmax distribution. N refers to the total number of samples in the batch.

Another approach is the Jensen-Shannon divergence (JSD) [48], which quantifies the similarity between two probability distributions. In the context of contrastive learning, JSD can be utilized to promote the generation of similar representations for positive pairs and dissimilar representations for negative pairs. This method offers a symmetric and smooth means of measuring the divergence between distributions. The JSD loss is defined as follows:

$$\begin{aligned}\mathcal{L}_{\text{JSD}} = &- \mathbb{E}_{P(\mathbf{h}_i,\mathbf{h}_j)}\left[\log(f(\mathbf{h}_i,\mathbf{h}_j))\right] \\ &- \mathbb{E}_{P(\mathbf{h}_i)P'(\mathbf{h}'_j)}\left[\log(1-f(\mathbf{h}_i,\mathbf{h}'_j))\right],\end{aligned} \tag{6}$$

where f is a discriminator that can be optimized to distinguish between sample pairs from the joint distribution (positive sample pairs) and sample pairs from the marginal product distribution (negative sample pairs). The formal difference between $\mathcal{L}_{\text{JSD}}$ and $\mathcal{L}_{\text{InfoNCE}}$ primarily lies in whether the expectation of the negative sample distribution is included within the positive sample distribution. This distinction implies that for JSD, a positive sample requires only one negative sample, whereas for InfoNCE, a positive sample necessitates N negative samples (where N is the batch size). Furthermore, the analysis presented in paper [12] demonstrates that $\mathcal{L}_{\text{JSD}}$ is insensitive to the number of negative samples, while the performance of $\mathcal{L}_{\text{InfoNCE}}$ declines as the number of negative samples decreases.

2.3 Concept definitions

2.3.1 Graph neural network and graph convolutional network

Graph is an abstract data structure composed of nodes and edges, which can be formally represented as $G = (V, E)$ where V is the set of nodes, and E is the set of edges. Graph Neural Network (GNN) is a generic neural network framework designed to handle data structured in any type of graph. They update the state (feature vector) of each node iteratively, involving not only the node's own features but also information from its adjacent nodes. After multiple rounds of iteration, the state of each node reflects its position within the network and information about its neighbors, making them suitable for tasks such as computer vision [49,50], cross-modal retrieval [51,52], and recommendation [53]. Particularly in recommendation systems, users and items are typically regarded as nodes, while their interactions are viewed as edges in an interaction graph. Therefore, graphs can effectively represent historical interactions.

Graph Convolutional Networks (GCNs) are one of the most popular variants of Graph Neural Networks (GNNs). They introduce convolution operations to handle graph-structured data. Traditional Convolutional Neural Networks (CNNs) are primarily used for processing grid-structured data, such as images, whereas GCNs extend this idea to graph structures. In GCNs, the new state of each node is calculated based on its own state and those of its directly connected neighbor nodes, similar to how filters aggregate information locally in traditional convolutions. GCNs achieve these convolution operations through spectral methods or spatial methods, enabling effective capture of relationships among nodes.

2.3.2 Knowledge graph

Knowledge Graph (KG) is a semantic network that reveals the relationships between entities, capable of formally describing real-world objects and their interrelations. A triplet is a common representation format for KGs, denoted as $G \in (E, R, S)$, where E is the set of entities, containing $|E|$ different entities; R is the set of relations, containing $|R|$ different relations; $S \subseteq E \times R \times E$ represents the set of triplets in the KG.

When dealing with collaborative filtering recommendation systems based on user-item interaction data, KGs can serve as a good source of feature enhancement data. Since they present in the form of triples, head entities or tail entities can be mapped to projects for knowledge expansion. Currently, newer works (such as KGIN [54], KGCL [21], KGRec [55], and DiffKG [56]) tend to use knowledge graphs to construct heterogeneous networks to enhance the pattern of preference mining.

2.3.3 Multi-modal and multi-view

These are two concepts that are easily confused in recommender systems. Multi-modal refers to a composite data form in deep learning, encompassing a unified term for multiple data types such as text, images, and audio. In recent years, explorations related to multimodality have been conducted in various fields, including recommendation systems. Multi-view [57–59] is also a commonly used concept in recommendation methods, often referring to perspectives of observed data (such as local *vs*. global views [60], or the initial *vs*. final layers in neural networks [61]). However, unlike multimodality, multiple views do not imply multiple modalities. There is no inherent connection between the two concepts.

3 Taxonomy

In this section, we first propose a unified framework for view construction methods based on GCL in recommender systems. Then, we categorize existing GCL-based recommendation models into three types according to the characteristics of their view construction methods: structure generation, feature generation, and modality generation.

3.1 Unified framework

As mentioned in Section 2, the core of recommendation methods based on GCL lies in executing view generation strategies to obtain multiple views and then performing contrastive tasks to maximize the consistency of positive pairs across these views. Specifically, given data D, N data views $\{D_i\}_{i=1}^{i=N}$ are obtained through N data-based augmentations $\{\mathcal{E}_i(\cdot)\}_{i=1}^{i=N}$, which can be formulated as:

$$D_i = \mathcal{E}_i(\cdot), i = 1, 2, \ldots, N. \tag{7}$$

Then, encoders $\{F_{encoder}(\cdot)\}_{i=1}^{i=N}$ are used to generate representations $\{\mathbf{e}_i\}_{i=1}^{i=N}$ for each data view. And these representations will be used for specific downstream recommendation tasks. Formally, this can be expressed as:

$$\mathbf{e}_i = F_{encoder}(D_i), i = 1, 2, \ldots, N. \tag{8}$$

The choice of augmentation is determined by the characteristics of the recommendation task, with common options including GNNs, pre-trained models, and more recently, encoding through Large Language Models (LLMs) [62]. To be specific, we have analyzed a substantial body of existing work and summarized the following strategies to serve as unified guidelines for our view construction framework:

$$\mathcal{E} = \begin{cases} F_{\mathrm{d}}, & \text{Structure Generation,} \\ F_{\mathrm{a}}, & \text{Feature Generation,} \\ F_{\mathrm{m}}, & \text{Modality Generation.} \end{cases} \tag{9}$$

In this context, F_d represents disturbance, F_a denotes augmentation, and F_m refers to modal. These three strategies will be discussed in Section 4. Note that we do not separately list the strategy for hybrid generation methods, as this approach is merely a combination of the three aforementioned methods.

3.2 Proposed taxonomy

We confine the differences among GCL-based recommendation methods to view generation strategies. Therefore, by identifying the variations in these strategies, one can determine the GCL-based recommendation methods. Consequently, we propose a taxonomy based on these strategies, as illustrated in Fig. 3. Table 1 presents the existing works covered by our taxonomy.

It should be noted that due to their complex architectures, real models often cannot be fully summarized by a single refined classification. For example, recommendation models designed for multimodal scenarios may use both modality data and interaction data to generate views. Therefore, the "Type" indicated in the table only describes the model's primary view generation strategy. More specifically, advanced models in recent years typically adopt a hybrid strategy for view generation.

4 View generation

The objective of contrastive learning can be seen as the acquisition of a specialized loss function that acts akin to an encoder, capable of generating similar representations for data from the same class, thereby reducing the semantic distance between positive sample pairs. Conversely, the mappings of negative sample pairs, or encodings of data from different classes, should be maximally dissimilar. Recent studies have highlighted that the crux of this method lies in constructing appropriate positive and negative sample contrastive views [57,98]. This section will review relevant literature within the domain to provide an overview of contrastive view generation strategies employed in recommendation methods.

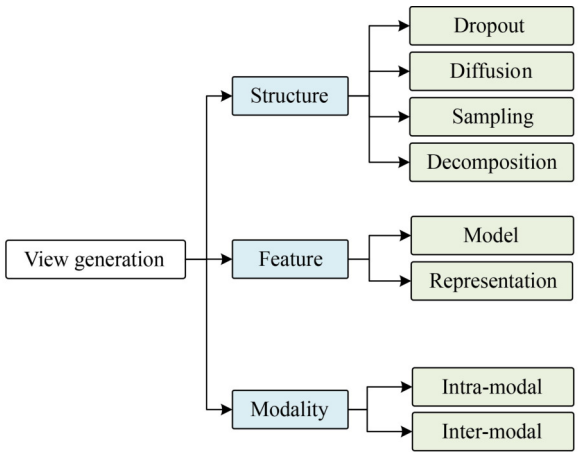

Fig. 3 Taxonomy of view generation methods in GCL-based recommendation

4.1 Structure generation methods

The structure refers to the interaction graph formed by users and items in the recommendation model. As introduced in Section 2, users and items are treated as nodes of the graph, while the interaction relationships are represented as edges. Within this framework, perturbations to the graph structure can be employed to disrupt the original topology and generate new semantic views. Through a review of existing work, we categorize the specific perturbation methods for graph $G = (V, E)$ into four types (Fig. 4): Drop, Sampling, Decompose, and Diffusion.

4.1.1 Drop methods

Such methods generate views by removing nodes or edges from the original view, hence they are also referred to as *Edge Drop* or *Node Drop*. The theoretical basis for this approach is that not all nodes and edges are relevant to the recommended topic. Therefore, discarding this noise can enhance the robustness of the representation. This method is formulated as:

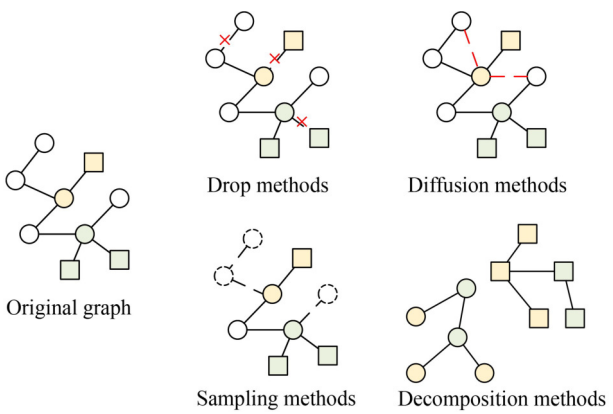

Fig. 4 Structure generation methods

Table 1 Summary of CL-based recommendation methods

Model	Year	Contrastive loss	Type	View generation
SGL [63]	2021	InfoNCE	Structure	Random dropout
CrossCBR [64]	2022	InfoNCE	Structure	Random dropout
KACL [65]	2023	InfoNCE	Structure	Selective dropout
SCL [66]	2022	InfoNCE	Structure	Random dropout
MMGCL [67]	2022	InfoNCE	Structure	Random dropout
KGCL [21]	2022	InfoNCE	Structure	Selective dropout
SEPTSEPT [68]	2021	InfoNCE	Structure	Selective dropout
CLHG [69]	2021	InfoNCE	Structure	Selective dropout
GDCL [70]	2022	InfoNCE	Structure	Graph diffusion
DiffKG [56]	2024	InfoNCE	Structure	Graph diffusion
BiGI [71]	2021	InfoNCE	Structure	Range sampling
KAUR [72]	2023	InfoNCE	Structure	Range sampling
RGCL [73]	2022	InfoNCE	Structure	Range sampling
BGCL [74]	2023	InfoNCE	Structure	Range sampling
KGCL-SBR [75]	2023	InfoNCE	Structure	Semantic sampling
CHEST [76]	2023	InfoNCE	Structure	Semantic sampling
LA-MPGCL [45]	2024	InfoNCE	Structure	Semantic sampling
HHGR [77]	2021	Jensen-Shannon	Structure	Semantic sampling
HMGCR [42]	2021	InfoNCE	Structure	Semantic sampling
LightGCL [78]	2022	InfoNCE	Structure	MF decomposition
MVCLF [58]	2024	InfoNCE	Structure	Heterogeneous decomposition
MKGCL [79]	2024	InfoNCE	Structure	Heterogeneous decomposition
ML-KGCL [80]	2023	InfoNCE	Structure	Heterogeneous decomposition
CKGC [81]	2022	InfoNCE	Structure	Semantic decomposition
DCLKR [82]	2023	InfoNCE	Structure	Semantic decomposition
KMCLR [83]	2023	InfoNCE	Feature	Model architecture-based
GCAUV [84]	2023	InfoNCE (g-EMD)	Feature	Model architecture-based
SimGRACE [85]	2022	InfoNCE	Feature	Model parameters-based
LMA4Rec [86]	2024	InfoNCE	Feature	Model parameters-based
MCLRec [87]	2023	InfoNCE (w/o τ)	Feature	Model parameters-based
SimGCL [88]	2022	InfoNCE	Feature	Representation noise-based
XSimGCL [61]	2024	InfoNCE	Feature	Representation noise-based
RKGCL [89]	2024	InfoNCE	Feature	Representation noise-based
SimDCL [90]	2023	InfoNCE	Feature	Representation dropout-based
MDGCL [91]	2024	InfoNCE	Feature	Representation dropout-based
MMSSL [92]	2023	InfoNCE	Modality	Inter modal
CI^{2}MG [93]	2023	InfoNCE	Modality	Intra&inter modal
MCCL [94]	2024	InfoNCE	Modality	Intra&inter modal
COSMOS [95]	2023	InfoNCE	Modality	Inter modal
MMCPR [96]	2022	InfoNCE	Modality	Intra&inter modal
SLMRec [97]	2023	InfoNCE	Modality	Inter modal

$$\tilde{A} = F_{\mathrm{d}}^{\mathrm{drop}}(G) = \begin{cases} \mathcal{T}_{\mathrm{edge}} = (V, E \circ M), \\ \mathcal{T}_{\mathrm{node}} = (V \circ M, E), \end{cases} \tag{10}$$

where $\tilde{A}$ is the enhanced view of the adjacency matrix A corresponding to graph G. $M \in (0,1)$ is a mask vector and its Hadamard product with the adjacency matrix A can be used to perform dropout operations on a set of nodes or edges.

SGL [63] was the first to introduce this method into GCL-based recommendation systems, and it has been widely utilized in numerous recommendation approaches [64,66,99]. However, this method of random perturbation poses risks, as indiscriminate random dropping is likely to discard valuable semantic information while removing irrelevant thematic data. Therefore, selective deletion can be employed alongside perturbation enhancement [21,65,68,69]. This approach not only generates robust augmented views but also reduces the introduction of irrelevant noise during the data augmentation process. For instance, KGCL [65] disrupts the input graph by randomly removing a certain proportion of edges while adaptively removing unimportant edges using two learnable view generators. In addition to perturbation through "deletion", "addition" can also serve as a perturbation method. For example, SCL [66] designed a node replication scheme that replaces a node's neighbor with a similar node's neighbor to facilitate view generation.

4.1.2 Diffusion methods

This method is based on the principle of information diffusion, which is a common operation in graph learning. The diffusion process can be used to generate new views by spreading information from a focal node to its neighbors. Formally, this method can add new edges to the original graph to complete user interaction behaviors. For example, GDCL [70] designed a diffusion matrix approximation algorithm to capture the importance of edges in the user-item interaction graph, thereby generating a diffusion-enhanced view. The diffusion method can be formulated as:

$$\tilde{A} = F_{\mathrm{d}}^{\mathrm{diff}}(G) = \sum_{k=0}^{\infty} \theta_k T^k, \tag{11}$$

where $T = D^{-1/2} A D^{-1/2}$ represents the normalized adjacency matrix of a graph, indicating the transition probabilities or information transmission relationships between nodes. The parameter θ_k denotes the weighting coefficients for different orders of diffusion.

The diffusion method can also serve as a downstream component for other view generation methods, where the original graph structure is first disrupted and then reconstructed through the diffusion process [56]. This combination of forward and backward operations can assist the model in learning more robust representations.

4.1.3 Sampling methods

This method refers to the selection of a subset of nodes and edges from the original graph $G = (V, E)$ to generate a new subgraph which can be formulated as:

$$\tilde{A} = F_{\mathrm{d}}^{\mathrm{samp}}(G) = (V', E'). \tag{12}$$

Here, $V' \in V$ and $E' \in E$ represent the selected sets of nodes and edges, respectively.

Similar to Drop Methods, random sampling is unreliable. Therefore, it is necessary to selectively sample from the original graph, obtaining subgraphs that contain key information while avoiding the disruption of structural semantics. However, the methods for sampling subgraphs are highly diverse, lacking a unified implementation form. We categorize the existing sampling methods into two main classes:

- Range sampling: This sampling method typically begins from the perspective of the sampling range, conducting uniform and consistent sampling. For example: 1) sampling the k-hop neighbors of nodes [71,72]; 2) sampling specific interaction terms of nodes [73,74].
- Semantic sampling: Strategically sampling with the aid of semantic information. For example: 1) constructing subgraphs using node attributes [75,76]; 2) sampling based on structural relationships [45,77]; 3) guiding sampling through meta-paths [1,42].

4.1.4 Decomposition methods

This method typically refers to matrix factorization (MF) [100], a popular approach in recommender systems. However, through our investigation of existing methods, we found that some works, although not explicitly stated as decomposition, align with the principles of decomposition in their view construction approach. Therefore, we include these works in this category as well. Formally, decomposition methods can be defined as follows:

$$\tilde{A} = F_{\mathrm{d}}^{\mathrm{deco}}(G) = \prod_{i=1}^{N} P_i, \tag{13}$$

where P_i is a decomposition of the interaction matrix, which is the mathematical representation of a graph, resulting in several submatrices.

Traditional MF methods decompose the user-item rating matrix R into two smaller matrices: $P \times Q^{\mathrm{T}} = R$. Here, P and Q serve as latent feature representations for users and items, respectively [101]. In GCL-based recommender systems, However, the focus is more on the reconstructed augmented views rather than the decomposed submatrices. Therefore, the reconstructed views can be considered as augmented views for contrastive learning [78,102].

Recently, in addition to decomposing from the user-item interaction graph, decomposition can also be conducted based on KGs to generate comparative views. Compared to interaction graphs without attributes, KGs, as heterogeneous information networks, contain rich semantic information. Therefore, mainstream decomposition methods typically obtain focal views through two types: heterogeneous and semantic. Heterogeneous decomposition [58,79,80] disassembles the heterogeneous networks within KGs to enhance them, while semantic decomposition [81,82] utilizes the semantic relationships (edges) between nodes in the KG for decomposition.

4.2 Feature generation methods

Feature generation method is a data enhancement technique that

focuses on obtaining augmented feature vectors from a results-oriented perspective, rather than transforming the graph structure. Accordingly, we categorize existing feature view generation methods into two types (Fig. 5): model-based methods and representation-based methods.

4.2.1 Model-based methods

Model-based methods generate views through various models (Fig. 5(a)). The advantage of this approach lies in its ability to focus directly on producing distinct features without the necessity to consider complex graph structural transformations. Consequently, model-based methods adopt a straightforward strategy: by propagating through different models, one can naturally obtain diverse features. This method can be defined as:

$$\mathbf{e}_i = F_{\mathrm{a}}^{\mathrm{model}}(G) = f_i(A), i = 1, 2, \ldots, K, \tag{14}$$

where f_i represents the ith model, regarded as an encoding function. K denotes the number of models.

It is important to note that the "model" typically refer to a localized computational module constructed using neural networks in deep learning, rather than an overarching model designed to address specific situational problems. Consequently, different feature outputs can be easily achieved by transforming the model. We categorize these approaches into two types:

- Adjust the neural network architecture. Augmented views with different information focus can be obtained by constructing different neural network models. For instance, [83,84] achieve different feature representations by employing distinct models as encoders.
- Modifying model parameters. There are two implementation strategies for this method. One strategy involves perturbing the model parameters to obtain models with varying sensitivities. For example, Simgrace [85] achieves model-level view augmentation by adding noise to the parameters of the GNN encoder. The other strategy [86,87] entails training a learnable sub-model to generate feature-enhanced views, essentially acquiring a set of model parameters for model-level view augmentation.

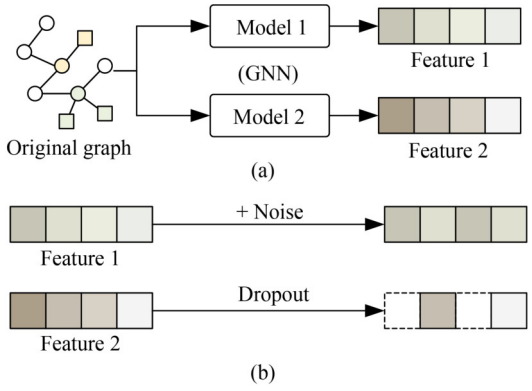

Fig. 5 Feature generation methods. (a) Model-based methods; (b) representation-based methods

4.2.2 Representation-based methods

The representation-based method (Fig. 5(b)) completely circumvents the complexities associated with graph data and deep models by simply enhancing the output feature representations to obtain contrasting views. This method can be expressed as follows:

$$\tilde{\mathbf{e}} = F_{\mathrm{a}}^{\mathrm{feature}}(\mathbf{e}) = f_{\mathrm{drop}}(\mathbf{e}) + \epsilon \boldsymbol{\Delta}. \tag{15}$$

The function $f_{\mathrm{drop}} = \mathbf{m} \circ \mathbf{e}$ represents the feature dropout, where $\mathbf{m} \in \{0,1\}^n$ is the mask vector used to set certain feature dimensions to zero. $\boldsymbol{\Delta}$ denotes the added noise, which is typically random and uniformly distributed. ϵ is a hyperparameter that controls the magnitude of the noise addition.

It is important to note that the dropout operation and the noise addition do not necessarily occur simultaneously. It simply indicates that for a final feature, either information is removed or added. Therefore, some methods that enhance features by shuffling them are also categorized in this way, as the shuffling operation essentially represents the linear transformation indicated by Eq. (15).

Based on the aforementioned discussion, we categorize representation-based methods into two types: 1) Adding noise [61,88,89]. Different views are generated by incorporating various types of noise into the embedding representations output by the model. Similar to model-based methods, the addition of multiple or diverse types of noise can also yield multiple contrasting views. However, considering that the added noise may mislead the model, the focus is typically on contrasting the enhanced views without involving the original view. 2) Information dropout [86,90,91]. This is achieved by randomly masking features across certain dimensions (usually by setting them to zero) to obtain multiple enhanced views from the same data source. This operation can mitigate overfitting during model training and enhance the model's robustness.

4.3 Modality generation methods

The modality generation method (Fig. 6) is a recently emerging approach for view generation. Given the abundance of multimodal information [27,28] in the current internet interactions, such as text, images, audio, and video, it is natural to leverage these diverse modalities to produce different views. Furthermore, each modality's view can capture information from a specific modal space, which not only enhances the overall performance of the model but also alleviates issues related to data sparsity and cold start to some extent [2,103]. Formally, this method can be defined as:

$$\mathbf{e}^{M \in \{\mathrm{t,v,a}\}} = F_{\mathrm{m}}(G) = f_{\mathrm{m}}^{\mathrm{t}}(g_{\mathrm{t}}), f_{\mathrm{m}}^{\mathrm{v}}(g_{\mathrm{v}}), f_{\mathrm{m}}^{\mathrm{a}}(g_{\mathrm{a}}). \tag{16}$$

In this context, t, v, and a represent the text, visual, and audio

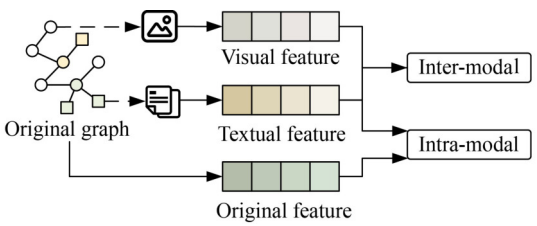

Fig. 6 Modality generation methods

modalities, respectively. f_m corresponds to the respective encoders for each of the three modalities, typically utilizing a pre-trained model [13,104,105].

Equation (16) lists only three mainstream modalities. However, there are additional works addressing other modalities [106,107]. Although modality generation methods may appear to be approaches that do not require data augmentation, this is not the case. Different modalities inherently possess significant semantic differences, making direct comparisons using multimodal data views unfeasible. Without a rigorous understanding of which data views can provide information gain, the comparison tasks are likely to fail.

In multi-modal contexts, in addition to the features of users and the items themselves, both intra-modal and inter-modal features are present. Both types of features are essential for modeling user preferences for items. Ignoring intra-modal features may lead to less accurate recommendations [92], as they reflect modality-specific information about the items. Inter-modal features, on the other hand, encompass shared semantic information across different modalities, such as the characteristics of the items. Therefore, we summarize modality generation methods [92–96] into two parts: intra-modal comparison and inter-modal comparison.

The intra-modal comparison does not refer to separately enhancing the same type of multimodal data for comparison; instead, it involves comparing the modality views with the user-item interaction views. By constructing interaction views under a single modality, we can generate contrastive pairs. This approach fully leverages the self-supervised signals within each modality and enhances the expressive capability of intra-modal representations through contrastive learning tasks. In contrast, inter-modal comparison constructs contrastive learning tasks between different multimodal data views. Given the inherent heterogeneity between different modal information, it is essential to bridge the heterogeneous gap through contrastive learning, thereby assisting the model in obtaining multimodal representations within a similar semantic space.

It should be clarified that not all GCL recommendation models utilizing multimodal datasets fall under the category of modality generation methods. This is because some works merely introduce multimodal data to enhance item features, while the contrastive learning task itself is not constructed using multimodal data. This implies that the view generation for contrastive learning does not rely on multimodal data but instead depends on structure or feature-based generation.

5 Experiments

Due to the varying view generation approaches of existing GCL recommendation methods, inconsistent datasets, and non-uniform evaluation metrics, there is a lack of uniformly designed comparative experiments to demonstrate the specific performance of different methods in real-world scenarios. This is crucial for balancing different view generation strategies.

In this section, we conduct an experimental analysis of some mainstream GCL recommendation methods under a unified dataset. Considering that the original papers of these baselines have provided sufficient details, our experiments will focus on their practical applicability across different recommendation scenarios, without exploring the range of hyperparameter values or module ablation studies.

5.1 Datasets

To facilitate readers' access to domain-related information, Table 2 lists commonly used benchmark datasets (with URL links) for recommendation models, even though we did not conduct experiments on them. However, due to limitations in computational resources and time constraints within research environments, it is often not feasible to use the full-scale dataset. Instead, a subset of the data is sampled for experiments. This is particularly common in multimodal datasets.Specifically, we used three publicly available recommendation benchmark datasets: Yelp, iFashion, and TikTok. Table 3 lists the detailed statistical information of these datasets.

5.2 Comparison and analysis

The results are presented in Table 4, where bold indicates the best performance. By categorizing methods into structure generation, feature generation, and modality generation (consistent with our taxonomy in Section 3), we conducted a unified comparison of open-source and reproducible works. To provide actionable insights for

Table 2 Commonly used benchmark datasets for recommendation models

Dataset	Description	Modality	Interactions
Amazon	Comment and metadata from the Amazon e-commerce platform, which is the latest version as of September 2023	V&A&T	Rating
iFashion	The fashion clothing dataset from Alibaba's e-commerce platform, collected and organized by POG [108]	V&T	Click
Book-Crossing	Collected by Cai-Nicolas Ziegler in a 4-week crawl (August / September 2004) from the Book-Crossing community with kind permission from Ron Hornbaker, CTO of Humankind Systems	V&T	Rating
Gowalla	A location-based social networking website collected by Stanford University. Users share their locations by checking in	T	Check in
Last.FM	Social networks, tags, and music artist information from 2,000 users of the Last.fm online music system	T	Click
MovieLens	Movie rating data collected by GroupLens Research from the MovieLens website, including user ratings, movie information, and user information	T	Rating
Yelp	Real-world data related to businesses collected by Yelp, including attributes such as reviews, photos, and ambiance	V&T	Rating

Table 3 Statistics of benchmark datasets

Dateset	Modality	Dim	Users	Items	Interactions	Sparsity
Yelp	V, T	512, 1024	37,397	32,491	707,178	99.94%
iFashion	V, T	512, 1024	38,403	20,000	382,765	99.95%
TikTok	V, A, T	128, 128, 768	9,319	6,710	59,541	99.90%

practitioners and assist them in selecting appropriate methods based on data characteristics, we avoided cross-comparisons within each category and instead emphasized data formats in the table. This decision is based on two key observations:

- Data heterogeneity: There are fundamental differences in input information and architectural complexity between single-modal and multi-modal models. Even within single-modal models, whether or not to incorporate KGs for enhancement can lead to significant variations, especially in cases of high data sparsity.
- Generational differences: Advanced single-modal techniques may outperform early multi-modal baseline models. It reflects technological advancements rather than inherent superiority. Aggregating results by category could conflate temporal progress with methodological effectiveness.

According to the results in Table 4, it can be seen that in practical recommendation scenarios, the number of observable interactions between users and items is often extremely sparse. In this case, it is difficult for the model to obtain the real preferences of users from the sparse interaction data, which is the direct reason for the low numerical results of the experiment.

Compared with relying on pure interaction data (Collaborative Filter, CF) for recommendation, adding KG for enhancement often achieves good results. This is because, compared with single dimensional interaction data, the nodes and edges in the KG can form various types of relationships. This means that structure generation methods can generate views based on semantics rather than simple node or edge perturbations.

Modal generation methods obviously only exist in multimodal data scenarios, but this does not mean that using multimodal data is equivalent to using modality generation methods. For example, MMGCL, although using multimodal data, its contrastive view generation is still based on structure generation methods (dropout and masking). This leads to an increase in the impact of data noise while introducing multimodal information. This is also the reason why its performance is on par with only CF-based methods (such as NCL). In other words, for multimodal recommendation scenarios, it is necessary to leverage intra-modal/inter-modal contrastive learning to narrow the distance in the semantic space between different modalities in order to achieve better results.

5.3 Visualization

To demonstrate how the GCL method works, we selected two classic baseline recommendation models. LightGCN uses a traditional graph convolution approach, while SGL employs GCL based on structure generation methods. The final user and item embeddings obtained from model training are reduced in dimensionality using the t-SNE method and then visualized using a 2D feature distribution map. Specifically, 1,000 users were randomly selected from the Yelp dataset, and then the top 500 most popular items and the bottom 500 least popular items were selected based on the number of item interactions.

As can be seen from Fig. 7(a), the method that solely uses GCN inevitably encounters the over-smoothing problem. This is specifically manifested as the feature distribution of the three types of nodes showing a highly clustered distribution. The trend where popular items and unpopular items converge to the same state indicates that stacking multiple layers of CNNs eliminates differences between nodes, causing the learned feature embeddings to exhibit popularity bias. On the other hand, the results in Fig. 7(b) present a relatively uniform distribution. Popular items and unpopular items are no longer distinctly separated but are distributed around the user nodes. Of course, the low numerical experimental results in Section 5.2 indicate that the feature distribution map cannot be absolutely ideal. However, by comparison, this sufficiently demonstrates the effectiveness of the GCL method for learning more robust feature representations.

Table 4 Performance comparison of GCL-based recommendation methods

Method	Type	Data format	Yelp		iFashion		TikTok	
			NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20
SGL	Structure	CF	0.0683	0.1071	0.0240	0.0344	0.0248	0.0641
NCL	Structure	CF	0.0732	0.1143	0.0246	0.0411	0.0255	0.0653
XSimGCL	Feature	CF	0.0752	0.1201	0.0265	0.0437	0.0288	0.0694
KGCL	Structure	CF+KG	0.0796	0.1268	0.0246	0.0403	–	–
KGRec	Structure	CF+KG	0.0829	0.1336	0.0272	0.0453	–	–
MMGCL	Structure	CF+MM	0.0721	0.1162	0.0275	0.0458	0.0256	0.0678
SLMRec	Modality	CF+MM	0.0854	0.1357	0.0305	0.0492	0.0353	0.0745
MMSSL	Modality	CF+MM	**0.0908**	**0.1436**	**0.0333**	**0.0536**	**0.0371**	**0.0823**

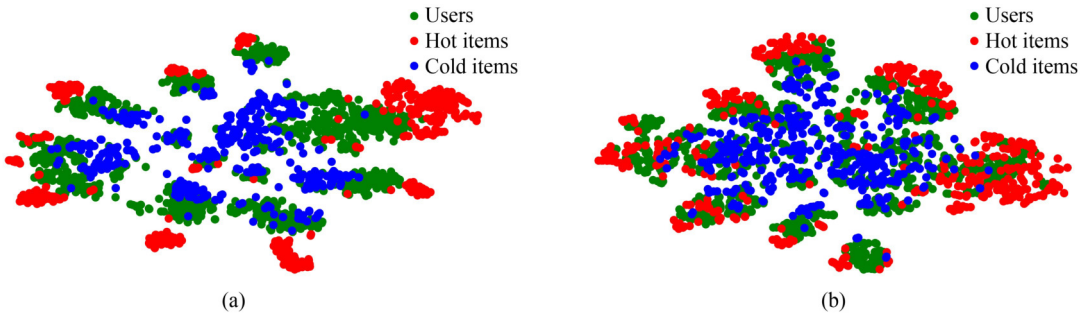

Fig. 7 Visualization of the recommendation results. (a) LightGCN; (b) SGL

6 Discussion

In the realm of GCL for recommender systems, the construction of effective views is pivotal for enhancing the performance and robustness of the models. Through the detailed exploration of contrastive learning view generation methods, we delineate a comprehensive landscape of the current state-of-the-art techniques. The following discussion aims to synthesize the insights gained and provide a critical analysis of the existing methodologies, their implications, and potential future directions.

6.1 Effectiveness of view generation

The effectiveness of view generation strategies depends on their ability to capture inherent features of the data while introducing sufficient diversity to promote the learning of invariant features. It has been proven that structure generation methods can effectively disrupt graph layouts to create new semantic views. These methods are particularly useful in scenarios with rich and complex interaction graphs because they help discover potential patterns that may not be obvious in the original data. Feature generation methods focus on expanding the feature space to produce diverse representations. When the feature space is multi-dimensional and sparse, these methods are advantageous because they help enrich feature representations and improve the model's generalization capabilities. Modal generation methods leverage the multimodal nature of the data to create views that capture different aspects of user preferences and item characteristics. This approach is especially promising in the era of rich multimedia content, allowing for a more comprehensive understanding of user interests by utilizing various forms of information. Additionally, many works [57,109–111] have employed hybrid strategies combining multiple view generation methods for data augmentation. The advantage of this approach is that it combines the strengths of different strategies, achieving complementarity between different methods. However, designing such a hybrid structure is challenging, and optimizing model performance will become difficult.

6.1.1 Robustness of GCL to noisy data

Noisy data poses a significant challenge for recommendation systems, often manifesting as inaccurate user feedback (e.g., user u interacts with item i due to accidental touch rather than genuine interest). Such noise can mislead the learning process and lead to increasing errors over iterations of training models, resulting in suboptimal representations of users and items. However, it is noteworthy that the intrinsic design of GCL provides advantages in mitigating noise. A key idea is that by generating multiple views from the same underlying graph, the model is forced to identify common patterns while ignoring inconsistent signals (i.e., noise). This multiview approach naturally encourages robustness, as noise tends to be random and less likely to persist across different views. GCL methods primarily enhance robustness to noisy data through the following principles:

- Multiview consistency. GCL generates multiple views from the same base graph, where each view captures different aspects of the data. Contrastive loss encourages alignment of consistent signals appearing across these views while discouraging representations driven by random, isolated interactions. Since noise is typically inconsistent and unlikely to appear in all views, it is naturally suppressed during contrastive learning.
- Filtering inherent noise via contrastive objectives. The contrastive loss function aims to pull together representations of the same node or user-item pair across different views while pushing apart irrelevant pairs. This forces the model to focus on invariant features that remain robust under perturbations. Noisy data may not exhibit consistency across views and contribute less to the final learned representations. In fact, the optimization process acts as an implicit noise filter, eliminating the need for additional mechanisms at the view generation stage.
- Utilizing higher-order connectivity. GCL methods typically leverage the connectivity structure of graphs. Even if some interactions are noisy, higher-order relationships (such as neighborhood structures and global patterns) still provide reliable signals. Aggregating multi-hop information within the graph helps dilute the impact of false interactions, thereby improving the overall robustness of the representations.

In summary, the robustness of GCL in handling noisy data mainly stems from its multiview alignment and contrastive learning objectives. These properties ensure that the model emphasizes consistent and invariant signals while reducing the influence of random or incorrect interactions. Based on the aforementioned foundational insights, there are several general strategies in the design of GCL methods for processes.

1) One widely adopted strategy is joint training with

complementary objectives. For example, combining contrastive learning objectives (Section 2.2) with supervised pairwise ranking losses like Bayesian Personalized Ranking (BPR) [112] can offer dual benefits. Contrastive loss encourages the model to learn invariant and consistent representations by aligning multiple views, inherently suppressing the effects of noisy interactions. Meanwhile, BPR loss directly optimizes the ranking performance of the recommendation task, ensuring that the learned representations are well-aligned with the ultimate prediction goals. This joint training not only enhances robust feature learning but also provides regularization against noise.

2) Another general strategy is further implementing noise filtering through the design of contrastive loss functions [110,113]. By leveraging temperature scaling in the loss calculation, the model can maximize the distance between positive pairs while simultaneously pushing negative pairs. As a result, the structure of the embedding space emphasizes consistent signals over random noise. Through this approach, even the presence of some noisy data reduces its effect, as it does not constructively contribute to cross-view representation alignment.

3) Regularization mechanisms integrated into the GCL framework (e.g., dropout, weight decay) can also further enhance the model's generalization capability [90,91]. These mechanisms help prevent overfitting to noise or irrelevant interactions, ensuring that the model focuses on potentially robust patterns in the data.

6.1.2 Mitigation of the cold start problem

Cold start remains a significant challenge in recommendation systems due to the scarcity of historical interaction data for new users or items. In our proposed GCL taxonomy framework, the view generation process is divided into three distinct strategies—structure generation, feature generation, and modality generation. Each of these strategies provides complementary signals that, when aligned through contrastive learning, help alleviate the cold start problem.

Structure-based views are obtained by imposing perturbations on the graph topology (e.g., via edge dropout or subgraph sampling). The intrinsic connection patterns between nodes persist even when direct interactions are limited. These views allow the model to capture latent relational patterns and neighborhood structures. Under cold-start conditions, this structural cue enables the system to infer potential associations between users or items based on their connectivity, thus compensating for the lack of explicit interaction history.

Feature-based views leverage the intrinsic attributes of users or items, such as profile information, item descriptions, or engineered features, which are independent of interaction data. For new users or items with sparse interactions, these features can serve as an alternative source of supervisory signals. By contrasting feature-based views, the model learns to associate similar intrinsic characteristics, enabling robust representation learning even when interaction data is insufficient.

Modality-based views incorporate additional auxiliary information from different data sources, such as textual reviews, images, or other contextual signals. These diverse modalities provide rich semantic content that can significantly enhance the learned representations. In cold-start scenarios where historical data is scarce, modality-based views enrich the representation space by highlighting external cues relevant to the semantics. Aligning these views through contrastive objectives ensures that the model captures consistent signals across heterogeneous sources.

6.2 Complexity analysis

In this section, we will conduct a complexity analysis of the three view generation methods proposed in the taxonomy. This is very helpful for providing guidance in real-world production environments.

Structure generation methods. Based on different perturbation approaches, we divide these methods into four subcategories. However, there are significant differences between the different sub-methods.

1) Drop methods. For a graph $G = (V, E)$, referring to Eq. (10), the main operation of this method involves performing Hadamard multiplication on the adjacency matrix A, with a time complexity of $O(m)$ or $O(n)$. Here, $m = |E|$ and $n = |V|$, meaning that the computational complexity of this method depends on the larger value between the number of nodes and edges.

2) Diffusion methods. Referring to Eq. (11), where T is the normalized adjacency matrix, K is the diffusion order adopted, and θ_k represents the weight coefficients. If sparse matrix multiplication is utilized, the computational complexity for each calculation of T^k is approximately $O(m)$, making the overall complexity $O(K \cdot m)$.

3) Sampling methods. As shown in Eq. (12), for range sampling, the sampling complexity for a single node is typically $O(d^k)$ (where d is the average degree, k is the sampling step), and the overall complexity is approximately $O(n \cdot d^k)$. For semantic sampling, if it involves similarity calculations between nodes, the complexity may reach $O(n^2)$ in the worst case. However, in practical applications, approximations or local computations are often used to reduce the complexity.

4) Decomposition methods. According to Eq. (13), the complexity of matrix factorization methods is typically $O(n \cdot d^2)$ or $O(m \cdot d)$ (matrix sparsity), where d represents the dimension of latent factors. However, although KG-based decomposition is categorized under decomposition methods, in terms of computational complexity, it is consistent with the other three types of structure generation methods.

Feature generation methods. This approach focuses on enhancing existing embeddings and is mainly divided into Model-based methods and Representation-based methods.

1) Model-based methods. Referring to Eq. (14), if the computational complexity of a single model's forward propagation is set as $O(f(n, m))$, then the total complexity for generating features using K different models is $O(K \cdot f(n, m))$. Here, $f(n, m)$ is a function used to describe the relationship between the main computational cost during the model's forward propagation and the number of nodes n and edges m in the graph. In other words, the computational cost required during forward propagation depends on the scale of the graph.

2) Representation-based methods. As shown in Eq. (15), this

method primarily performs element-wise operations on feature vectors. The complexity for each node's embedding vector (d dimension) is $O(d)$, and the overall complexity is approximately $O(n \cdot d)$, which is relatively low.

Modality generation method. The computational process of this method is shown in Eq. (16), where t, v, and a represent the text, visual, and audio modalities, respectively. Assuming the complexity of the pre-trained encoders corresponding to each modality is $O(C_{\mathrm{t}})$, $O(C_{\mathrm{v}})$, and $O(C_{\mathrm{a}})$, respectively, then the total complexity of modality generation is: $O(\sum_{m=1}^{M} C_m + C_{fusion})$. Here, M is the number of modalities, and C_{fusion} represents the complexity of multimodal information fusion (common methods include attention mechanisms or fully connected layers), which typically reaches $O(n \cdot d^2)$.

6.3 Challenges and future directions

Despite the progress in view generation technology, many challenges remain in practical applications. These issues also represent important directions for future research. This section will discuss these problems and propose some possible roadmaps to address them.

6.3.1 Balancing view generation strategies

Table 5 presents a comparison among different view generation methods. Specifically, in GCL approaches for recommendation systems, one of the key challenges lies in balancing the preservation of important information and the introduction of diversity. For structure generation methods, excessive perturbation may lead to the loss of critical semantic information, while insufficient augmentation might fail to provide the necessary diversity for effective contrastive learning. Moreover, recent studies [61,88] suggest that augmentations involving perturbations of graph structures may not be necessary. However, structure generation methods are still advantageous due to their simplicity, which allows them to be easily transferred to other domains with good portability.

In feature generation methods, model-based approaches benefit from the gradient optimization capabilities of deep learning models, enabling adaptive adjustments. However, this comes at the cost of limited generalizability, as the models are often designed for specific task scenarios. Representation-based methods, similar to structure generation methods, are easy to implement and even simpler. However, they lack the ability for adaptive adjustment [114,115], relying heavily on manual trial and error. Additionally, these methods face challenges in semantic expansion [116,117], such as incorporating KGs.

Table 5 Difference between view generation methods

	Structure	Feature		Modality
		Model-based	Representation-based	
Portability	✓	×	✓	×
Auto-adjustment	×	✓	×	✓
Semantic Expansion	×	✓	×	✓

Another trade-off that needs consideration is computational complexity, especially in modality generation-based methods. These methods involve processing multi-modal data (e.g., text, images), and the conventional approach is to use pre-trained models for multi-modal feature extraction. However, since each modality has different feature representations and distributions, a key challenge for recommendation systems is how to fuse these features without losing the advantages of each modality. This is particularly problematic in GCL-based recommendation methods, where different modalities may introduce their own noise and inconsistencies. Such issues can lead to information redundancy or errors in the fused views, affecting the stability and effectiveness of contrastive learning. Additionally, computations involving multi-modal data are often resource-intensive, which can limit their applicability in large-scale recommendation scenarios.

To address the aforementioned challenges, we propose two potential research directions: 1) Feature Fusion and Augmentation Mechanisms. When generating views by leveraging the inherent attributes of nodes and edges, adaptive hyperparameter tuning and meta-learning methods can be introduced to make the generation process adaptable to different task requirements. Additionally, external knowledge (e.g., KGs) can be used to enrich the semantic information of nodes, enhancing both the depth and breadth of the generated views; 2) Cross-Modal Alignment Mechanisms. Design fusion modules based on attention mechanisms or adversarial training to achieve efficient alignment of different modalities within a unified embedding space. This ensures that the semantic information from each modality can complement one another, thereby jointly improving the quality of the generated views.

In summary, the appropriate choice of view generation strategy is highly dependent on the specific characteristics of the dataset and the recommendation task, and there is no one-size-fits-all solution. The effectiveness of a method in one scenario does not guarantee its success in another. This calls for a more nuanced understanding of the interactions between data characteristics, view generation strategies, and recommendation performance.

6.3.2 Handling noisy data

In the view generation process of contrastive learning, an ideal scenario would involve dynamically adjusting the view generation strategy based on user behavior patterns and preferences. This would allow the model to learn more valuable information and provide more personalized recommendations. However, in real-world scenarios, recommendation systems often face a significant amount of noise, which manifests as false-positive and false-negative samples [118]. False-positive samples refer to observed interactions that are not actually driven by interest, often due to accidental clicks. False-negative samples refer to unobserved interactions that are actually of interest to the user but remain latent. Particularly in the multi-hop information propagation process within graph structures, the issue of noise becomes even more pronounced. Therefore, recent studies have focused extensively on removing noise from recommendation data.

Existing works leverage knowledge-aware approaches [21,119],

attention mechanisms [116], and other methods to filter out critical information for "adaptive" information fusion, thereby avoiding excessive noise. However, these methods still rely on implicit denoising, which reduces the impact of noise by learning latent invariant features, but their adaptability is quite limited. There is currently no mature solution for how to perform active denoising. Thus, dynamically adjusting the level of data augmentation based on data characteristics and model performance will be a key direction for future development, involving learning to balance the preservation of essential information with the introduction of diversity.

Based on the sources of noisy data, they can be divided into two categories: inherent noise in the raw data and noise introduced during the propagation of information aggregation. On this basis, we propose the following two possible solutions.

1) **View generation strategy based on learnable modules.** Traditional structure-based, feature-based, and modality-based generation methods often rely on predefined manual strategies. These methods lack adaptability when dealing with inherent noise in the raw data and can result in significant human and computational costs. To address this, end-to-end learnable view generation modules can be designed. By introducing noise metrics as regularization terms during the generation process, the model can adaptively adjust its generation strategy. Specifically, a noise-aware mechanism could assign different weights to each sample or local subgraph in the input data, thereby automatically filtering out potential noise during view generation. Through joint training strategies, view generation and subsequent contrastive learning tasks can be optimized together, ensuring that the generated views not only reflect users' real behavior patterns but also maintain diversity while reducing noise. This approach avoids the limitations of manually selecting fixed strategies and allows for continuous adaptive correction of noise effects during training.

2) **Controlling noise introduced by multi-hop information propagation.** In graph structures, multi-hop information propagation is an important means of capturing global structure. However, as the number of hops increases, irrelevant information and noise tend to accumulate gradually. To address this, hop-sensitive gating mechanisms or decay functions can be designed to automatically attenuate information passed from distant neighbors, thereby limiting meaningless information transfer. Such mechanisms can dynamically adjust information fusion weights based on neighbor distance, reducing the cumulative effect of noise. Combined with attention-based aggregation strategies, the model can automatically determine which neighbor information is more reliable during multi-hop aggregation, assigning higher attention weights accordingly. This helps better suppress noise introduced by distant nodes during contrastive learning. Additionally, dynamic negative sample adjustment strategies can be introduced in contrastive learning to identify and remove potential false negatives, further enhancing the model's robustness against noise.

6.3.3 High-quality negative sampling

One of the core aspects of contrastive learning techniques is the mining of positive and negative sample pairs, which determines whether the model can truly learn latent preferences and invariant features. Currently, many methods employ uniform negative sampling, where negative samples are randomly selected from the raw data. However, this approach has several limitations:

- Sampling scope limited by computational cost [120]. Uniform sampling often focuses only on the distribution of nodes in discrete space, failing to fully exploit the fine-grained information embedded in implicit user-item feedback. Due to the vast candidate negative sample space, calculating similarity scores for all negative samples is computationally expensive. As a result, only a subset of negative samples is typically sampled, which may lead to insufficient representativeness of the negative samples.
- Ignoring inherent differences among negative samples [121]. Uniform sampling does not differentiate the importance of different negative samples, potentially including semantically similar but false-negative samples into the negative sample set. This can interfere with the model's ability to learn correct discriminative features.

To solve these problems, we propose exploring more efficient and differentiated negative sampling strategies from the following perspectives.

1) **Dynamic negative sampling.** Utilize the current model's embedding representations to dynamically select "hard negatives" and "soft negatives." A similarity-based sampling mechanism can be designed, where the similarity between candidate negative samples and anchor points is first calculated. Then, based on predefined thresholds or distributions, samples that are discriminative yet not too close are selected. This avoids mistakenly sampling false negatives. This strategy can be implemented through an online updating mechanism, allowing the negative sampling process to interactively feedback with model training.

2) **Negative sampling with memory bank or candidate set mechanism.** Establish a dynamic memory bank to store high-quality negative samples that have been validated historically, and regularly update it based on the latest model state. This mechanism ensures sampling efficiency while reducing redundant computations, enhancing the diversity and representativeness of negative samples. Samples in the memory bank can be ranked according to previous gradient information, prediction confidence, and other metrics, selecting the most challenging samples as negative samples.

3) **Fusing external information for weighted negative sampling.** In addition to intrinsic model features, auxiliary information (e.g., user behavior logs, content similarity, contextual information, etc.) can be leveraged to weight and filter negative samples. By fusing multi-source information, negative sample selection can rely not only on distances in the embedding space but also on semantic or behavioral differences. This approach can more accurately distinguish true negative samples from potential false negatives.

4) **Designing robust contrastive loss functions.** Besides improving negative sampling strategies, robustness to negative

sample noise can be considered in loss function design. For example, using adaptive temperature parameters, margin adjustments, or weighted InfoNCE losses can ensure that the model maintains stable gradient signals even when facing potential mis-sampling, thereby reducing the adverse impact of false negatives on representation learning.

6.3.4 Explainable mechanisms

Although recommendation models based on GCL have achieved significant performance improvements, the underlying mechanisms and principles behind these enhancements remain unclear. Specifically, how the model internally utilizes different views for contrastive enhancement, and why certain views are more critical for improving performance, still lack intuitive explanations. This "black-box" nature not only limits the in-depth understanding of the latent relationships between user preferences and item characteristics, but also hinders the clear guidance needed to enhance the model's generalization capabilities. Therefore, enhancing the explainability and transparency of recommendation systems is an urgent problem to be addressed. To address the issue of explainability, we believe that future work can focus on the following aspects:

1) **Semantic path explanation using KGs [122].** Since KGs are heterogeneous information networks, external knowledge can be leveraged to align the views generated in recommendation systems with rich semantic information, thereby revealing the differences in how different views capture user preferences and item features. Specifically, a semantic path network can be constructed, where each view corresponds to one or more explanatory paths in the KG, intuitively demonstrating why that view better reflects users' interests or key attributes of items.

2) **Enhancing explainability with multi-modal data [123].** Multi-modal data (e.g., image information and user reviews) often contain rich semantic and emotional information. By aligning multi-modal data with the views generated through GCL, the weight and role of different modalities in decision-making can be revealed. By comparing views generated from different modalities, it is possible to analyze which modality information guided users' choices, thereby providing intuitive explanations.

3) **Exploration through visualization [61].** Visualization techniques (e.g., clustering, dimensionality reduction, and heatmaps) can be used to display representations generated from different views, allowing for intuitive observation of which views exhibit more distinct separations in data distribution. This method can reveal the specific contributions of different views and their fusion in a particular recommendation decision, thus providing intuitive explanations for model decisions.

6.3.5 Incorporating LLMs

Existing recommendation models are generally classified as deep neural network models, which can fully exploit collaborative signals within the dataset for recommendations. However, their limitation lies in the inability to leverage knowledge from the open domain. From this perspective, LLMs and recommendation models can form a strong complementary relationship. We believe that current recommendation systems incorporating large language models can be structurally divided into two categories:

- LLM as Recommender (LaR): These works [29,124,125] focus on large language models, where the LLM directly provides the final recommendation results without relying on the traditional deep learning approach of computing dot-product scores between user and item embeddings. The core of this method lies in prompt engineering, where recommendation tasks are completed using few-shot or even zero-shot learning.
- LLM with Recommender (LwR): These works do not entirely discard traditional recommendation models but instead enhance the recommendation pipeline by leveraging large language models, for example, in feature engineering [126,127] and scoring/ranking [128].

LaR is undoubtedly a promising approach. This method can directly utilize knowledge from the open domain, thereby capturing richer semantic information. Moreover, users can interact with the recommendation system in a natural, conversational manner, which can significantly enhance user experience. For example, P5 [124] constructs prompts for different recommendation tasks on a unified LLM and then fine-tunes for various downstream tasks using Supervised Fine-Tuning (SFT). However, the biggest challenge currently faced by this type of method lies in the LLM itself. An LLM capable of meeting real-world application requirements demands enormous computational costs. Additionally, LLMs are prone to hallucination [129], where they generate fictitious or inaccurate information, potentially leading to recommendations that do not align with users' actual needs. LLMs also pose privacy and security risks when handling user data, potentially leaking sensitive information. Furthermore, LLMs are highly sensitive to prompt design and can be easily affected by contextual noise, impacting the stability and accuracy of their outputs.

In contrast, LwR retains the architecture of traditional recommendation models, ensuring technical maturity while leveraging LLMs to enhance the recommendation process and achieve good results. For instance, LLMRec [127] uses LLMs to augment user profiles and item attributes, addressing the long-standing issue of data sparsity in recommendation systems. The advantage of this approach lies in its ability to incorporate open-domain knowledge from LLMs as a supplement, achieving complementary effects, or utilizing the NLP capabilities of LLMs to make deeper use of existing data.

The main challenge for LwR lies in coordinating and aligning the training tasks. Traditional recommendation models are typically trained using supervised or self-supervised loss functions, whereas LLMs rely on pre-training with large-scale language data and then fine-tuning for specific tasks. Jointly training these two types of models can lead to inconsistent gradient updates during training, affecting the convergence speed and overall performance of the integrated model. Additionally, the increase in parameter count may significantly raise the demand for training time and computational

resources. Therefore, designing an effective fusion mechanism that allows the two components to complement each other rather than interfere will be the primary technical challenge for LwR.

Although some existing works have explored the application of LLMs in recommendation systems—for example, studies [62,130] attempt to combine LLMs with contrastive learning in sequential recommendation scenarios—the integration of LLMs into GCL-based recommendation methods remains an unexplored area. Therefore, combining LLMs with GCL methods can be explored from the following directions:

1) **Semantic-enhanced view generation.** LLMs possess strong natural language understanding and generation capabilities, enabling them to generate semantically rich views based on item descriptions, user reviews, or other textual information. These views not only capture collaborative signals present in traditional graph structures but also incorporate knowledge from open domains, making the generated views more nuanced and semantically enriched.

2) **Multi-modal view alignment and fusion.** In traditional GCL frameworks, structure generation, feature generation, and modality generation form the three basic types of views. By introducing LLMs, the semantic views generated by LLMs can be treated as a new modality and fused with other views. The key to this approach lies in designing cross-modal alignment mechanisms, such as shared contrastive loss or attention mechanisms, to align the text-based views generated by LLMs with graph-structured views at the semantic level, allowing different views to complement each other. This not only helps alleviate the data sparsity problem but also provides more accurate semantic supplementation in cold-start scenarios.

3) **Hard negative sampling.** In Section 6.3.3, we discussed the main challenges and potential solutions for negative sampling in GCL-based recommendation methods. Since LLMs, through probabilistic modeling, can develop near-human logical reasoning capabilities, they are able to understand users' true preferences. This implies that contrastive learning's negative sampling process could leverage LLMs to perform hard negative sampling—selecting items that users genuinely dislike as negative samples. By enhancing the model's ability to distinguish users' true preferences, it reduces noise interference and achieves superior overall recommendation performance.

7 Conclusion

This survey systematically reviews the current state of GCL view construction methods in recommender systems. It provides a taxonomy for categorizing existing methods and a framework for unifying different techniques. We highlight significant findings and offer a detailed discussion to guide the selection of appropriate methods for enhancing recommendation performance. Finally, we outline future research directions to address the limitations of current studies. We hope this survey will inspire further innovations, tackle the limitations of existing research, and pave the way for more effective and robust recommendation systems.

Acknowledgements

This survey was supported by the following grants and projects: the Graduate Education Quality Improvement Project of North Minzu University (Grant No. YJZT202424), the Natural Science Foundation of Ningxia Hui Autonomous Region (Grant No. 2023AAC03316), the Joint Fund Project of the National Natural Science Foundation of China (Grant No. U22A20577). The funding bodies had no role in the design of the study, the collection, analysis, or interpretation of data, or in the writing of the manuscript.

Competing interests

The authors declare that they have no competing interests or financial conflicts to disclose.

Open Access

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

[1] Yao Z, Ji M, Xing T, Fu R, Li S, Yin F. Multi-component graph collaborative filtering using auxiliary information for TV program recommendation. Neural Computing and Applications, 2023, 35(30): 22737−22754

[2] Mu Y, Wu Y. Multimodal movie recommendation system using deep learning. Mathematics, 2023, 11(4): 895

[3] Li Y, Li X, Zhao Q. Multimodal deep learning framework for book recommendations: harnessing image processing with VGG16 and textual analysis via LSTM-enhanced Word2Vec. Traitement du Signal, 2023, 40(4): 1367−1376

[4] Wang W. Personalized music recommendation algorithm based on hybrid collaborative filtering technology. In: Proceedings of 2019 International Conference on Smart Grid and Electrical Automation. 2019, 280−283

[5] Nanthini M, Pradeep Mohan Kumar K. Cold start and data sparsity problems in recommender system: a concise review. In: Proceedings of International Conference on Innovative Computing and Communications. 2023, 107−118

[6] Zhang Y, Yin G, Chen D. A dynamic cold-start recommendation method based on incremental graph pattern matching. International Journal of Computational Science and Engineering, 2019, 18(1): 89−100

[7] Zhang F, Yuan N J, Lian D, Xie X, Ma W Y. Collaborative knowledge base embedding for recommender systems. In: Proceedings of

the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016, 353−362

[8] Chang C, Zhou J, Weng Y, Zeng X, Wu Z, Wang C D, Tang Y. KGTN: knowledge graph transformer network for explainable multi-category item recommendation. Knowledge-Based Systems, 2023, 278: 110854

[9] Zhao Y, Li C, Peng J, Fang X, Huang F, Wang S, Xie X, Gong J. Beyond the overlapping users: cross-domain recommendation via adaptive anchor link learning. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023, 1488−1497

[10] Liu X, Zhang F, Hou Z, Mian L, Wang Z, Zhang J, Tang J. Self-supervised learning: generative or contrastive. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(1): 857−876

[11] Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning. 2020, 1597−1607

[12] Hjelm R D, Fedorov A, Lavoie-Marchildon S, Grewal K, Bachman P, Trischler A, Bengio Y. Learning deep representations by mutual information estimation and maximization. In: Proceedings of the 7th International Conference on Learning Representations. 2019

[13] Devlin J, Chang M W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2019, 4171−4186

[14] Gao T, Yao X, Chen D. SimCSE: simple contrastive learning of sentence embeddings. In: Proceedings of 2021 Conference on Empirical Methods in Natural Language Processing. 2021, 6894−6910

[15] Wu Y, Wang X, Zhang A, He X, Chua T S. Discovering invariant rationales for graph neural networks. In: Proceedings of the 10th International Conference on Learning Representations. 2022

[16] Qiu J, Chen Q, Dong Y, Zhang J, Yang H, Ding M, Wang K, Tang J. GCC: graph contrastive coding for graph neural network pre-training. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020, 1150−1160

[17] Le-Khac P H, Healy G, Smeaton A F. Contrastive representation learning: a framework and review. IEEE Access, 2020, 8: 193907−193934

[18] Shen X, Zhang Y. A knowledge graph recommendation approach incorporating contrastive and relationship learning. IEEE Access, 2023, 11: 99628−99637

[19] Jiang L, Yan G, Luo H, Chang W. Improved collaborative recommendation model: integrating knowledge embedding and graph contrastive learning. Electronics, 2023, 12(20): 4238

[20] Shu H, Huang J. Multi-task feature and structure learning for user-preference based knowledge-aware recommendation. Neurocomputing, 2023, 532: 43−55

[21] Yang Y, Huang C, Xia L, Li C. Knowledge graph contrastive learning for recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022, 1434−1443

[22] Khan A, AlBarri S, Manzoor M A. Contrastive self-supervised learning: a survey on different architectures. In: Proceedings of the 2nd International Conference on Artificial Intelligence. 2022, 1−6

[23] Liu Y, Jin M, Pan S, Zhou C, Zheng Y, Xia F, Yu P S. Graph self-supervised learning: a survey. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(6): 5879−5900

[24] Xie Y, Xu Z, Zhang J, Wang Z, Ji S. Self-supervised learning of graph neural networks: a unified review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(2): 2412−2429

[25] Yu J, Yin H, Xia X, Chen T, Li J, Huang Z. Self-supervised learning for recommender systems: a survey. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(1): 335−355

[26] Jing M, Zhu Y, Zang T, Wang K. Contrastive self-supervised learning in recommender systems: a survey. ACM Transactions on Information Systems, 2024, 42(2): 59

[27] Liu Y, Lyu C, Liu Z, Cao J. Exploring a large-scale multimodal transportation recommendation system. Transportation Research Part C: Emerging Technologies, 2021, 126: 103070

[28] Li J, Yang C, Ye G, Nguyen Q V H. Graph neural networks with deep mutual learning for designing multi-modal recommendation systems. Information Sciences, 2024, 654: 119815

[29] He Z, Xie Z, Jha R, Steck H, Liang D, Feng Y, Majumder B P, Kallus N, Mcauley J. Large language models as zero-shot conversational recommenders. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023, 720−730

[30] Wu L, Zheng Z, Qiu Z, Wang H, Gu H, Shen T, Qin C, Zhu C, Zhu H, Liu Q, Xiong H, Chen E. A survey on large language models for recommendation. World Wide Web, 2024, 27(5): 60

[31] Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J. GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of 1994 ACM Conference on Computer Supported Cooperative Work. 1994, 175−186

[32] Ko H, Lee S, Park Y, Choi A. A survey of recommendation systems: recommendation models, techniques, and application fields. Electronics, 2022, 11(1): 141

[33] Pazzani M J, Billsus D. Content-based recommendation systems. In: Brusilovsky P, Kobsa A, Nejdl W, eds. The Adaptive Web. Berlin: Springer, 2007, 325−341

[34] Bogaards N, Schut F. Content-based book recommendations: personalised and explainable recommendations without the cold-start problem. In: Proceedings of the 15th ACM Conference on Recommender Systems. 2021, 545−547

[35] Su X, Khoshgoftaar T M. A survey of collaborative filtering techniques. Advances in Artificial Intelligence, 2009, 2009: 421425

[36] He M, Wang B, Du X. HI2Rec: exploring knowledge in heterogeneous information for movie recommendation. IEEE Access, 2019, 7: 30276−30284

[37] Wang H, Zhang F, Wang J, Zhao M, Li W, Xie X, Guo M. RippleNet: propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018, 417−426

[38] Wang X, He X, Cao Y, Liu M, Chua T S. KGAT: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019, 950−958

[39] Wang H, Zhao M, Xie X, Li W, Guo M. Knowledge graph convolutional networks for recommender systems. In: Proceedings of the World Wide Web Conference. 2019, 3307−3313

[40] Silva T, Rivera A R. Representation learning via consistent assignment of views to clusters. In: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing. 2022, 987−994

[41] Lin F, Jiang W, Zhang J, Yang C. Dynamic popularity-aware contrastive learning for recommendation. In: Proceedings of the 13th Asian Conference on Machine Learning. 2021, 964−968

[42] Yang H, Chen H, Li L, Yu P S, Xu G. Hyper meta-path contrastive learning for multi-behavior recommendation. In: Proceedings of 2021 IEEE International Conference on Data Mining. 2021, 787−796

[43] Lin Z, Tian C, Hou Y, Zhao W X. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In: Proceedings of the ACM Web Conference 2022. 2022, 2320−2329

[44] Meng Z, Ounis I, Macdonald C, Yi Z. Knowledge graph cross-view contrastive learning for recommendation. In: Proceedings of the 46th European Conference on Information Retrieval on Advances in Information Retrieval. 2024, 3−18

[45] Qian Z S, Huang H, Zhu H, Liu J P. Multi-perspective graph contrastive learning recommendation method with layer attention mechanism. Journal of Computer Research and Development, 2025, 62(1): 160−178

[46] van den Oord A, Li Y, Vinyals O. Representation learning with contrastive predictive coding. 2019, arXiv preprint arXiv: 1807.03748

[47] Gutmann M, Hyvärinen A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics. 2010, 297−304

[48] Nowozin S, Cseke B, Tomioka R. f-GAN: training generative neural samplers using variational divergence minimization. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016, 271−279

[49] Shi Y, Cai J X, Shavit Y, Mu T J, Feng W, Zhang K. ClusterGNN: cluster-based coarse-to-fine graph neural network for efficient feature matching. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 12507−12516

[50] Bai L, Cui L, Wang Y, Li M, Li J, Yu P S, Hancock E R. HAQJSK: hierarchical-aligned quantum Jensen-Shannon kernels for graph classification. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(11): 6370−6384

[51] Yu H, Yao F, Lu W, Liu N, Li P, You H, Sun X. Text-image matching for cross-modal remote sensing image retrieval via graph neural network. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2023, 16: 812−824

[52] Li M, Zhou S, Chen Y, Huang C, Jiang Y. EduCross: dual adversarial bipartite hypergraph learning for cross-modal retrieval in multimodal educational slides. Information Fusion, 2024, 109: 102428

[53] Li M, Li Z, Huang C, Jiang Y, Wu X. EduGraph: learning path-based hypergraph neural networks for MOOC course recommendation. IEEE Transactions on Big Data, 2024, 10(6): 706−719

[54] Wang X, Huang T, Wang D, Yuan Y, Liu Z, He X, Chua T S. Learning intents behind interactions with knowledge graph for recommendation. In: Proceedings of the Web Conference 2021. 2021, 878−887

[55] Yang Y, Huang C, Xia L, Huang C. Knowledge graph self-supervised rationalization for recommendation. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023, 3046−3056

[56] Jiang Y, Yang Y, Xia L, Huang C. DiffKG: knowledge graph diffusion model for recommendation. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining. 2024, 313−321

[57] Lin Y, Gou Y, Liu X, Bai J, Lv J, Peng X. Dual contrastive prediction for incomplete multi-view representation learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(4): 4447−4461

[58] Xia B, Qin J, Han L, Gao A, Ma C. Knowledge filter contrastive learning for recommendation. Knowledge and Information Systems, 2024, 66(11): 6697−6716

[59] Li M, Yang X, Chen Y, Zhou S, Gu Y, Hu Q. ReFNet: rehearsal-based graph lifelong learning with multi-resolution framelet graph neural networks. Information Sciences, 2025, 700: 121856

[60] Li M, Zhang L, Cui L, Bai L, Li Z, Wu X. BLoG: bootstrapped graph representation learning with local and global regularization for recommendation. Pattern Recognition, 2023, 144: 109874

[61] Yu J, Xia X, Chen T, Cui L, Hung N Q V, Yin H. XSimGCL: towards extremely simple graph contrastive learning for recommendation. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(2): 913−926

[62] Li Y, Zhai X, Alzantot M, Yu K, Vulić I, Korhonen A, Hammad M. CALRec: contrastive alignment of generative LLMs for sequential recommendation. In: Proceedings of the 18th ACM Conference on Recommender Systems. 2024, 422−432

[63] Wu J, Wang X, Feng F, He X, Chen L, Lian J, Xie X. Self-supervised graph learning for recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021, 726−735

[64] Ma Y, He Y, Zhang A, Wang X, Chua T S. CrossCBR: cross-view contrastive learning for bundle recommendation. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022, 1233−1241

[65] Wang H, Xu Y, Yang C, Shi C, Li X, Guo N, Liu Z. Knowledge-adaptive contrastive learning for recommendation. In: Proceedings of the 16th ACM International Conference on Web Search and Data Mining. 2023, 535−543

[66] Yang C, Zou J, Wu J, Xu H, Fan S. Supervised contrastive learning for recommendation. Knowledge-Based Systems, 2022, 258: 109973

[67] Yi Z, Wang X, Ounis I, Macdonald C. Multi-modal graph contrastive learning for micro-video recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022, 1807−1811

[68] Yu J, Yin H, Gao M, Xia X, Zhang X, Hung N Q V. Socially-aware self-supervised tri-training for recommendation. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021, 2084−2092

[69] Li H, Luo X, Yu Q, Wang H. Session-based recommendation via contrastive learning on heterogeneous graph. In: Proceedings of 2021 IEEE International Conference on Big Data. 2021, 1077−1082

[70] Zhang L, Liu Y, Zhou X, Miao C, Wang G, Tang H. Diffusion-based graph contrastive learning for recommendation with implicit feedback. In: Proceedings of the 27th International Conference on Database Systems for Advanced Applications. 2022, 232−247

[71] Cao J, Lin X, Guo S, Liu L, Liu T, Wang B. Bipartite graph embedding via mutual information maximization. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. 2021, 635−643

[72] Ma Y, Zhang X, Gao C, Tang Y, Li L, Zhu R, Yin C. Enhancing recommendations with contrastive learning from collaborative knowledge graph. Neurocomputing, 2023, 523: 103−115

[73] Shuai J, Zhang K, Wu L, Sun P, Hong R, Wang M, Li Y. A review-aware graph contrastive learning framework for recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022, 1283−1293

[74] Zhu J, Li B, Wang J, Li D, Liu Y, Zhang Z. BGCL: Bi-subgraph network based on graph contrastive learning for cold-start QoS prediction. Knowledge-Based Systems, 2023, 263: 110296

[75] Zhang X, Ma H, Yang F, Li Z, Chang L. KGCL: a knowledge-enhanced graph contrastive learning framework for session-based recommendation. Engineering Applications of Artificial Intelligence, 2023, 124: 106512

[76] Wang H, Zhou K, Zhao X, Wang J, Wen J R. Curriculum pre-training heterogeneous subgraph transformer for top-*N* recommendation. ACM Transactions on Information Systems, 2023, 41(1): 19

[77] Zhang J, Gao M, Yu J, Guo L, Li J, Yin H. Double-scale self-supervised hypergraph learning for group recommendation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021, 2557−2567

[78] Cai X, Huang C, Xia L, Ren X. LightGCL: simple yet effective graph contrastive learning for recommendation. In: Proceedings of the 11th International Conference on Learning Representations. 2023, 1−15

[79] Zhang Y, Zhu J, Chen R, Liao W, Wang Y, Zhou W. Mixed-curvature knowledge-enhanced graph contrastive learning for recommendation. Expert Systems with Applications, 2024, 237: 121569

[80] Chen G, Xie X. ML-KGCL: multi-level knowledge graph contrastive learning for recommendation. In: Proceedings of the 28th International Conference on Database Systems for Advanced Applications. 2023, 253−268

[81] Cao X, Shi Y, Wang J, Yu H, Wang X, Yan Z. Cross-modal knowledge graph contrastive learning for machine learning method recommendation. In: Proceedings of the 30th ACM International Conference on Multimedia. 2022, 3694−3702

[82] Huang S, Hu C, Kong W, Liu Y. Disentangled contrastive learning for knowledge-aware recommender system. In: Proceedings of the 22nd International Semantic Web Conference on the Semantic Web. 2023, 140−158

[83] Xuan H, Liu Y, Li B, Yin H. Knowledge enhancement for contrastive multi-behavior recommendation. In: Proceedings of the 16th ACM International Conference on Web Search and Data Mining. 2023, 195−203

[84] Guo Y, Liu Y. A graph contrastive learning framework with adaptive augmentation and encoding for unaligned views. In: Proceedings of the 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining. 2023, 92−104

[85] Xia J, Wu L, Chen J, Hu B, Li S Z. SimGRACE: a simple framework for graph contrastive learning without data augmentation. In: Proceedings of the ACM Web Conference 2022. 2022, 1070−1079

[86] Hao Y, Zhao P, Xian X, Liu G, Zhao L, Liu Y, Sheng V S, Zhou X. Learnable model augmentation contrastive learning for sequential recommendation. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(8): 3963−3976

[87] Qin X, Yuan H, Zhao P, Fang J, Zhuang F, Liu G, Liu Y, Sheng V. Meta-optimized contrastive learning for sequential recommendation. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023, 89−98

[88] Yu J, Yin H, Xia X, Chen T, Cui L, Nguyen Q V H. 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. 2022, 1294−1303

[89] Wang B, Yang X, Zhang X, Zhao C, Wang C, Feng W. Recommendation algorithm based on refined knowledge graphs and contrastive learning. In: Proceedings of the 17th International Conference on Knowledge Science, Engineering and Management. 2024, 205−217

[90] Xu Y, Wang Z, Wang Z, Guo Y, Fan R, Tian H, Wang X. SimDCL: dropout-based simple graph contrastive learning for recommendation. Complex & Intelligent Systems, 2023, 9(5): 4751−4763

[91] Xu Q, Li W, Chen J. MDGCL: message dropout graph contrastive learning for recommendation. In: Proceedings of the 20th International Conference on Advanced Intelligent Computing Technology and Applications. 2024, 60−71

[92] Wei W, Huang C, Xia L, Zhang C. Multi-modal self-supervised learning for recommendation. In: Proceedings of the ACM Web Conference 2023. 2023, 790−800

[93] Lin Z, Tan Y, Zhan Y, Liu W, Wang F, Chen C, Wang S, Yang C. Contrastive intra-and inter-modality generation for enhancing incomplete multimedia recommendation. In: Proceedings of the 31st ACM International Conference on Multimedia. 2023, 6234−6242

[94] Wei Y, Xu Y, Zhu L, Ma J, Peng C. Multi-level cross-modal contrastive learning for review-aware recommendation. Expert Systems with Applications, 2024, 247: 123341

[95] Won H, Oh B, Yang H, Lee K H. Cross-modal contrastive learning for aspect-based recommendation. Information Fusion, 2023, 99: 101858

[96] Liu Z, Ma Y, Schubert M, Ouyang Y, Xiong Z. Multi-modal contrastive pre-training for recommendation. In: Proceedings of 2022 International Conference on Multimedia Retrieval. 2022, 99−108

[97] Tao Z, Liu X, Xia Y, Wang X, Yang L, Huang X, Chua T S. Self-supervised learning for multimedia recommendation. IEEE Transactions on Multimedia, 2023, 25: 5107−5116

[98] Li J, Qiang W, Zheng C, Su B, Xiong H. MetAug: contrastive learning via meta feature augmentation. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 12964−12978

[99] Liu Z, Ma Y, Ouyang Y, Xiong Z. Contrastive learning for recommender system. 2021, arXiv preprint arXiv: 2101.01317

[100] Isinkaye F O. Matrix factorization in recommender systems: algorithms, applications, and peculiar challenges. IETE Journal of Research, 2023, 69(9): 6087−6100

[101] Liu H, Zheng C, Li D, Shen X, Lin K, Wang J, Zhang Z, Zhang Z, Xiong N N. EDMF: efficient deep matrix factorization with review feature learning for industrial recommender system. IEEE Transactions on Industrial Informatics, 2022, 18(7): 4361−4371

[102] Liu Z, Ma Y, Li H, Hildebrandt M, Ouyang Y, Xiong Z. Debiased

contrastive loss for collaborative filtering. In: Proceedings of the 16th International Conference on Knowledge Science, Engineering and Management. 2023, 94−105

[103] Zhao Z, Yang Q, Lu H, Weninger T, Cai D, He X, Zhuang Y. Social-aware movie recommendation via multimodal network learning. IEEE Transactions on Multimedia, 2018, 20(2): 430−440

[104] Radford A, Kim J W, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, Krueger G, Sutskever I. Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning. 2021, 8748−8763

[105] Ao J, Wang R, Zhou L, Wang C, Ren S, Wu Y, Liu S, Ko T, Li Q, Zhang Y, Wei Z, Qian Y, Li J, Wei F. SpeechT5: unified-modal encoder-decoder pre-training for spoken language processing. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. 2022, 5723−5738

[106] Lian D, Zheng K, Ge Y, Cao L, Chen E, Xie X. GeoMF++: scalable location recommendation via joint geographical modeling and matrix factorization. ACM Transactions on Information Systems (TOIS), 2018, 36(3): 33

[107] Wei C, Hu C, Wang C D, Huang S. Time-aware multibehavior contrastive learning for social recommendation. IEEE Transactions on Industrial Informatics, 2024, 20(4): 6424−6435

[108] Chen W, Huang P, Xu J, Guo X, Guo C, Sun F, Li C, Pfadler A, Zhao H, Zhao B. POG: personalized outfit generation for fashion recommendation at Alibaba iFashion. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019, 2662−2670

[109] Bin C, Li W, Wu F, Chang L, Wen Y. Multi-behavior-based graph contrastive learning recommendation. Knowledge and Information Systems, 2024, 66(6): 3477−3496

[110] Dong Z, Yang Y, Zhong Y. Mixed augmentation contrastive learning for graph recommendation system. In: Proceedings of the 8th International Joint Conference on Web and Big Data. 2024, 130−143

[111] He Y, Wu G, Cai D, Hu X. Cross-view sample-enriched graph contrastive learning network for personalized micro-video recommendation. In: Proceedings of 2023 ACM International Conference on Multimedia Retrieval. 2023, 48−56

[112] Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L. BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence. 2009, 452−461

[113] Li B, Jin B, Song J, Yu Y, Zheng Y, Zhou W. Improving micro-video recommendation via contrastive multiple interests. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022, 2377−2381

[114] Jiang Y, Huang C, Huang L. Adaptive graph contrastive learning for recommendation. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023, 4252−4261

[115] Jing M, Zhu Y, Zang T, Yu J, Tang F. Graph contrastive learning with adaptive augmentation for recommendation. In: Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases. 2022, 590−605

[116] Tao S, Qiu R, Cao Y, Zhao H, Ping Y. Intent with knowledge-aware multiview contrastive learning for recommendation. Complex & Intelligent Systems, 2024, 10(1): 1349−1363

[117] Yu W, Bin C, Liu W, Chang L. Contrastive learning-based multi-behavior recommendation with semantic knowledge enhancement. In: Proceedings of 2023 IEEE International Conference on Data Mining. 2023, 1511−1516

[118] Wei Z, Wu N, Li F, Wang K, Zhang W. MoCo4SRec: a momentum contrastive learning framework for sequential recommendation. Expert Systems with Applications, 2023, 223: 119911

[119] Chen F, Kang Z, Zhang C, Wu C. Multi-contrastive learning recommendation combined with knowledge graph. In: Proceedings of 2023 International Joint Conference on Neural Networks. 2023, 1−8

[120] Miao R, Yang Y, Ma Y, Juan X, Xue H, Tang J, Wang Y, Wang X. Negative samples selecting strategy for graph contrastive learning. Information Sciences, 2022, 613: 667−681

[121] Yan R, Bao P. ConCur: self-supervised graph representation based on contrastive learning with curriculum negative sampling. Neurocomputing, 2023, 551: 126525

[122] Jiang H, Li C, Cai J, Wang J. RCENR: a reinforced and contrastive heterogeneous network reasoning model for explainable news recommendation. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023, 1710−1720

[123] Liu Z, Ma Y, Schubert M, Ouyang Y, Rong W, Xiong Z. Multimodal contrastive transformer for explainable recommendation. IEEE Transactions on Computational Social Systems, 2024, 11(2): 2632−2643

[124] Geng S, Liu S, Fu Z, Ge Y, Zhang Y. Recommendation as language processing (RLP): a unified pretrain, personalized prompt & predict paradigm (P5). In: Proceedings of the 16th ACM Conference on Recommender Systems. 2022, 299−315

[125] Liao J, Li S, Yang Z, Wu J, Yuan Y, Wang X, He X. LLaRA: large language-recommendation assistant. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024, 1785−1795

[126] Liu Q, Chen N, Sakai T, Wu X M. A first look at LLM-powered generative news recommendation. 2023, arXiv preprint arXiv: 2305.06566

[127] Wei W, Ren X, Tang J, Wang Q, Su L, Cheng S, Wang J, Yin D, Huang C. LLMRec: large language models with graph augmentation for recommendation. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining. 2024, 806−815

[128] Kang W C, Ni J, Mehta N, Sathiamoorthy M, Hong L, Chi E, Cheng D Z. Do LLMs understand user preferences? Evaluating LLMs on user rating prediction. 2023, arXiv preprint arXiv: 2305.06474

[129] Shanahan M. Talking about large language models. Communications of the ACM, 2024, 67(2): 68−79

[130] Wu Y, Xie R, Zhu Y, Zhuang F, Zhang X, Lin L, He Q. Personalized prompt for sequential recommendation. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(7): 3376−3389

Zhihang YI is currently pursuing a master's degree at North Minzu University, China. He obtained a bachelor's degree in engineering from Wuhan Business University, China in 2022. His main research focus is multimodal self-supervised recommendation methods.

Hairong WANG is currently a professor at North Minzu University, China. She received her PhD from Northeastern University, China. Her main research areas are big data knowledge engineering and intelligent information processing. She has published more than 40 papers in international journals and conferences, led and completed over 10 scientific research projects, and applied for 8 invention patents.

Fangping CHEN is currently pursuing a master's degree at North Minzu University, China. She received a bachelor's degree in engineering from Longdong University, China in 2022. Her research focus is on multimodal knowledge extraction methods.

Zhaojing XU is currently pursuing a master's degree at North Minzu University, China. She received her bachelor's degree in engineering from Changchun Institute of Technology, China in 2024. Her main research direction is on knowledge graph-based recommendation methods.

Jianling YANG is currently a senior researcher at the Ningxia Institute of Meteorological Science, China. She earned her PhD from the Ocean University of China, China. Her primary research areas include climate and climate change, the anomaly patterns, causes, and mechanisms of meteorological disasters, prediction technologies, and the ecological impacts of climate change. She has published over 50 papers in international journals and conference proceedings, co-authored 3 monographs, and led or completed more than 10 scientific research projects.