Neurocomputing 626 (2025) 129539

Contents lists available at ScienceDirect

Neurocomputing

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

TPGRec: Text-enhanced and popularity-smoothing graph collaborative filtering for long-tail item recommendation

Chenyun Yu, Junfeng Zhao, Xuan Wu, Yingle Luo, Yan Xiao *

Shenzhen Campus of Sun Yat-sen University, Shenzhen, 518107, Guangdong, China

ARTICLE INFO

Communicated by Q. Liu

Keywords:
Recommendation systems
Long-tail recommendation
GNNs
Representative learning
Contrastive learning

ABSTRACT

GNN-based graph collaborative filtering methods have shown significant potential in recommendation systems, but they are often challenged by the long-tail effect due to exposure bias. While existing methods utilize techniques like contrastive learning, data augmentation and resampling as countermeasures, their reliance on ID-based embeddings can result in less informative representations and limit the model's grasp of intricate neighbor relationships. Recent researches have attempted to improve overall recommendation performance by incorporating text information for items, but they usually rely on extra graph structures or complex calculations, increasing computational costs and lacking adequate consideration for long-tail items. In this paper, we propose TPGRec, a novel Graph collaborative filtering method jointly from the text enhancement and popularity smoothing perspectives, which simultaneously improves both overall and long-tail recommendation performance. Initially, we introduce a balancing mechanism applied to the graph structure and training set to reduce the influence of popular items. Upon this, a structural-level contrastive learning technique is proposed for graph representation learning, which captures complex structural relationships without introducing excessive noise to node representations. Furthermore, we develop a semantic-level contrastive learning strategy that effectively and economically integrates ID embeddings with textual data, establishing implicit semantic relationships and deepening the model's understanding of items. Ultimately, we develop a popularity-balanced BPR optimization module to facilitate fair recommendation opportunities for items of varying popularity and promote the model's discriminative power over hard negative samples. Comprehensive experiments on four real-world datasets have demonstrated the superiority of TPGRec compared with the state-of-the-art baselines. Our codes and datasets are available at Github: https://github.com/ycy89/MyTPGRec.

1. Introduction

As a powerful tool for information acquisition and dissemination, recommendation systems are widely used across a spectrum of online applications, such as e-commerce, social media, and video-sharing websites. Collaborative filtering (CF) [1], one of the core algorithms behind recommendation systems, operates on the basic concept of leveraging similarities between users or items to provide recommendation services to similar users/items. Benefiting from advancements in graph representation learning, recent studies have integrated Graph Neural Networks (GNNs) [2–6] into collaborative filtering techniques. These GNN-based methods excel at capturing intricate high-order neighborhood information for representation learning, which significantly enhance the performance of recommendation systems.

Despite the effectiveness of applying GNNs to CF, recommendation systems are often challenged by the long-tail issue [7,8] in practical applications. This issue arises when a small number of popular products dominate the majority of recommendation slots, leaving a large number of less popular products with limited exposure, making them difficult for users to discover. The fundamental cause of this phenomenon lies in the exposure bias during the model's training process, which leads the model to favor recommending products that have already received considerable attention. Moreover, this issue is exacerbated in scenarios with sparse user–item interaction data, further hindering the recommendation of new products and services.

To address the long-tail challenge, exiting approaches typically employ techniques like reweighting [9–11], resampling [12–15], and transfer learning [16,17] to reduce popularity bias. Recognizing the sparsity of interactions between long-tail items and users in observed data, some studies [4,6,18] further enhance node embeddings by integrating graph augmentation strategies with contrastive learning methodologies. However, these methods still suffer from two major obstacles. Firstly, random or subjective alterations to the original

* Corresponding author.
E-mail addresses: yuchy35@mail.sysu.edu.cn (C. Yu), zhaojf6@mail2.sysu.edu.cn (J. Zhao), xiaoy367@mail.sysu.edu.cn (Y. Xiao).

https://doi.org/10.1016/j.neucom.2025.129539
Received 5 July 2024; Received in revised form 10 December 2024; Accepted 25 January 2025
Available online 1 February 2025

user–item interaction graph could disrupt its intrinsic structure and inadvertently introduce noise, potentially affecting the effectiveness of recommendation models. Additionally, most existing methods rely on ID-based embeddings for users and items, which often lack rich semantic information for accurately establishing high-order connections between nodes, leading to limited performance improvements.

Recent studies have redirected to explore and establish the implicit semantic connections among items. Representative methods include MMGCN [19], RGCN [20], LATTICE [21], MICRO [22], and MGCN [23], which aim to integrate multimodal information to enrich item representations and capture more accurate user preferences. Nonetheless, to effectively fuse ID embeddings with multimodal data, these approaches often require extra network layers or graph structures. Take LATTICE and MICRO for instance, they construct an additional K-Nearest-Neighbor item-item graph for representation learning, incurring prohibitive computational and memory costs. In addition, these methods prioritize the enhancement of overall recommendation performance and tackle the cold start challenge, rather than being specifically crafted to address the long-tail problem. Consequently, popularity bias still persists and demands further attention and resolution.

In this paper, we concentrate on improving the overall recommendation performance while balancing the exposure opportunities for both popular and long-tail items. To tackle the limitations of existing approaches, we propose a novel **G**raph collaborative filtering **Rec**ommendation method jointly from the **T**ext enhancement and **P**opularity smoothing perspectives, which we term **TPGRec** for brevity. This framework composes of four components: Graph Construction, Graph Representation Learning, Feature Fusion & Alignment, and Popularity-Balanced BPR optimization.

First of all, the Graph Construction module refines the user–item bipartite graph using a degree-aware edge pruning technique and incorporates a balancing mechanism into the training set, so that we can amplify the visibility and influence of long-tail items during model training. Following this process, the Graph Representation Learning module is introduced. We propose a structural-level contrastive learning technique for graph representation learning, enabling the learning of complex structural relationships while avoiding the introduction of excessive noise into node representations. To further improve the quality of embeddings, the Feature Fusion & Alignment module integrates text information of items into the graph model. Specifically, we organize the title, category, brand and description into a text sequence for each item, and subsequently, utilize an advanced language model to transform them into semantically rich embeddings. Due to the fact that ID embeddings and text data are separate modalities, we propose a semantic-level contrastive learning strategy to reduce the gap between their distributions for effective feature fusion. Building on the refined embeddings, we implicitly establish semantic relationships between popular and tail items, as well as deepening the model's understanding of item characters and alleviating the long-tail issue. Recognizing the limitations of traditional BPR (Bayesian Personalized Ranking) optimization [24] in addressing long-tail items and handling negative samples, we develop a popularity-balanced BPR optimization module. This innovative loss function facilitates equitable recommendation opportunities for all items and aids the model in adaptively identifying and handling negative samples of varying difficulty. Consequently, our method promotes the model's capacity to distinguish hard negative samples, contributing to a more robust and fair recommendation system.

The main contributions of this work can be summarized as follows:

- We propose an effective GNN-based collaborative filtering approach, which incorporates both text enhancement and popularity smoothing to concurrently promote overall recommendation performance and mitigate the long-tail issue.
- To capture informative user/item representations, we propose a structural-level contrastive learning technique for graph representation learning, and employ a semantic-level contrastive learning strategy to effectively fuse ID embeddings with textual data.
- To increase the exposure and recommendation opportunities for long-tail items, we propose a degree-aware edge dropout strategy, a balancing mechanism applied for training set, along with a popularity-balanced BPR loss to optimize the recommendation model comprehensively.

Extensive experiments conducted on four real-world datasets have demonstrated the superiority of our proposed TPGRec method. Compared to the best performance across a variety of state-of-the-art baselines, TPGRec achieves average improvement on HR@10, HR@20, NDCG@10, and NDCG@20 of 10.61%, 9.01%, 15.39%, and 14.85%, respectively, when evaluated on Amazon datasets. Furthermore, ablation studies and visualization analysis have confirmed the effectiveness of TPGRec in improving the exposure chances for long-tail items.

2. Related work

2.1. Graph-based collaborative filtering

Graph-based Collaborative Filtering (GCF) technique employs bipartite graphs to model user–item interactions and stacks multiple layers of graph convolution neural networks (GCNs) to generate node embeddings. It can capture rich collaborative signals through high-order graph connectivity, offering a distinct advantage over matrix factorization methods enhanced by MLP [1,25–27]. Pioneering efforts like NGCF [28] and PinSAGE [29] laid the foundation for GCF, which was subsequently refined by SGCN [30] and LightGCN [2]. They simplified GCN architectures by removing unnecessary linear and nonlinear transformations, thereby enhancing model efficiency. MGDCF [31] establishes relationships between GCF and traditional deep learning, and further demonstrates that the effectiveness of GNNs can be attributed to optimization rather than regularization. Despite their capabilities, GCF still grapples with data sparsity and the cold start problems. Recent studies have focused on leveraging self-supervised signals from graphs [3–6,32–34]. For instance, SGL [3] performs node or edge dropout to generate positive examples via minor subgraph modifications. SimGCL [4] designs a simple but effective graph contrastive learning method by generating contrastive pairs with noisy node representations. XSimGCL [6] further simplifies SimGCL by contrasting the comprehensive representation with the first-layer representation, requiring only a single forward propagation process. NCL [5], enhances embeddings by aligning central nodes with both graph-homogeneous and semantically similar neighbors. Although contrastive learning and data augmentation methods offer some benefits, most existing approaches rely on ID-based user/item embeddings, which often lack the semantic depth for establishing higher-order connections, thus leading to limited performance improvement and falling short in addressing the long-tail issue in recommendation systems.

2.2. Long-tail recommendation

The long-tail problem in recommender systems is characterized by a few items acquiring excessive popularity, which overshadows the majority of less popular but valuable items [7,8]. In the fields of computer vision and natural language processing, effective strategies such as resampling [13,14,35], reweighting [9,10] and transfer learning [16], have been employed to tackle the long-tail problem. Inspired by these approaches, some researches have adapted similar methodologies to recommender systems. For instance, YouTube's Neural Retrieval Model [11] estimates the frequency of items and incorporates it into the batch softmax cross-entropy loss to reduce sampling bias. MIRec [17] proposes a dual transfer learning framework that transfers knowledge from head items to tail items at both the model and item levels. CDN [15] emphasizes the ID representations of popular items and side features of unpopular items respectively, and executes downsampling for interactions containing popular items. Google's SSL framework [18]

utilizes negative sampling to increase the update frequency for tail items during contrastive learning. Parallel to these efforts, some related researches have concentrated on cold-start recommendations [12,36,37], employing techniques like meta-learning to refine the embeddings of new items. It is important to note the distinction between cold-start and long-tail items, and our objective is to enhance the performance of long-tail items under the premise of maintaining overall performance, broadening our focus beyond new item recommendations. To achieve this, we integrate textual information of items into graph model and develop effective popularity smoothing strategies to improve the exposure and recommendation chances for long-tail items.

2.3. Text-integrated recommendation

Tail items inherently lack interaction data and require the establishment of relationships via alternative means, such as the utilization of textual information. M2GNN [38] establishes a heterogeneous graph with item-tag edges and designs a hierarchical aggregation framework of GNN to learn cross-domain interests via meta-paths [39]. HGCL [40] constructs user-user and item-item homogeneous auxiliary graphs based on shared keywords, achieving user-specific and item-specific knowledge transfer. Emerging multimodal methods integrate a variety of data types, including text, images, and audio, to capture user preferences comprehensively. Early works such as VBPR [41] and DeepStyle [42] expanded traditional matrix factorization by incorporating visual or style factors into the loss function. MMGCN [19] and GRCN [20] pioneered the use of GNNs in multimodal recommendations, creating user–item bipartite graphs for each modality and fusing ID with text data during neighbor aggregation. Despite these innovations, existing methods have not fully exploited the potential of textual features, often using semantic information implicitly. Later studies like LATTICE [21], MICRO [22] and MGCN [23], explicitly construct item-item KNN graphs based on intra-modality similarity and fuse item embeddings with an attention mechanism. Nevertheless, recent studies [43–45] have verified that multimodal integration does not invariably yield positive outcomes due to modality competition, sometimes even underperforming the best single modality. Moreover, the dependency on auxiliary graphs in existing methods can lead to prohibitive computational and memory costs, and they usually lack adequate consideration for the long-tail items. Consequently, in this paper, we concentrate on textual information and explore an economical and effective way to integrate ID features with textual data to optimize recommendation performance and alleviate the long-tail issue.

3. Problem formulation

Consider a set of users and items, denoted as $\mathcal{U}(|\mathcal{U}| = M)$ and $\mathcal{I}(|\mathcal{I}| = N)$, respectively. We use $R \in \{0,1\}^{M\times N}$ to represent the interactions between users and items, where any entry $R_{u,i} = 1$ indicates there is a connection between u and i, and vice versa. Generally, graph collaborative filtering methods organize users' behavior matrix R into an user–item bipartite graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{\mathcal{U} \cup \mathcal{I}\}$ refers to the set of nodes, and $\mathcal{E} = \{(u,i)|u \in \mathcal{U}, i \in \mathcal{I}, R_{u,i} = 1\}$ denotes the set of edges. Based on the user–item interaction matrix R, we can construct the adjacent matrix A as follow:

$$\mathrm{A} = \begin{bmatrix} 0^{M\times M} & \mathbf{R} \\ \mathbf{R}^T & 0^{N\times N} \end{bmatrix}. \tag{1}$$

Assume that $E^{(0)}$ denotes the initial node embeddings (usually are ID embeddings), and then those embeddings are calculated and updated through message propagation and neighborhood aggregation. Formally, node embeddings for users and items can be captured after performing L graph convolutional operations, denoted as follow:

$$\begin{aligned} E_u &= READOUT([E_u^{(0)}, E_u^{(1)}, \dots, E_u^{(L)}]), \\ E_i &= READOUT([E_i^{(0)}, E_i^{(1)}, \dots, E_i^{(L)}]), \end{aligned} \tag{2}$$

where READOUT refers to an aggregation function used in GNNs.

With the final node representations, we calculate the inner product between user u and each unobserved item i as the preference score, and then return the item with the highest score to u. To optimize the model training, the Bayesian personalized ranking (BPR) loss [24] is typically used as follow:

$$\mathcal{L}_{bpr} = -\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{N}_u}\sum_{j\notin\mathcal{N}_u} \log\sigma\left(e_u^T e_i - e_u^T e_j\right), \tag{3}$$

where $\mathcal{N}_u$ represents the neighborhood set of u, j denotes a randomly selected negative item that u has not interacted with before, $\sigma(\cdot)$ is the sigmoid function, e_u and e_i denote the embedding of user u and item i, respectively.

4. Methodology

In this section, we introduce TPGRec, a novel GNN-based collaborative filtering approach designed to concurrently enhance the overall recommendation performance and alleviate the long-tail effect. Its core idea lies in effectively and economically incorporating textual information into item embeddings, while smoothing the contribution of interactions of varying popularity during model training. Fig. 1 sketches the overview of TPGRec, which can be divided into four components: Graph Construction, Graph Representation Learning, Feature Fusion & Alignment, and Popularity-Balanced BPR optimization. Initially, TPGRec constructs an user–item bipartite graph and employs a degree-aware edge dropout strategy to balance the exposure chances of items. Subsequently, superior node embeddings are generated by the graph model through the structural-level contrastive learning. Building on those embeddings, we propose a semantic-level contrastive learning technique and effectively integrate textual information of items into the model through the Feature Fusion & Alignment module. Finally, we present a popularity-balanced BPR optimization strategy and describe details of objective functions for model training. In the subsequent subsections, we introduce each module of the model in details.

4.1. Graph construction

As introduced in Section 3, we represent user–item interactions through a bipartite graph $\mathcal{G}$, where the initial embeddings for nodes are derived from their ID indices. Specifically, our model incorporates an ID encoder for users $\mathcal{U}$ and items $\mathcal{I}$, respectively, to generate distinctive embeddings. This is realized using two ID embedding matrix $E_u^{(0)} \in \mathbb{R}^{|\mathcal{U}\times d|}$ and $E_i^{(0)} \in \mathbb{R}^{|\mathcal{I}\times d|}$, with d denoting the dimensionality of the embeddings. Note that each row in the matrix corresponds to the specific ID-based node embedding, while both matrices are trainable parameters within the network and continually refined throughout the learning process.

Furthermore, to mitigate the influence of popular interests on the model, we introduce a balancing mechanism applied to the graph structure and training set, respectively, with the objective of diminishing the learning frequency of popular preferences while enhancing that of less popular ones. Details are presented as follows.

(1) Degree-aware Edge Dropout

As introduced in [46], popular nodes or edges are usually prone to suffer from the interest overwhelming and over-smoothing problems. In response, existing methods often adopt Edge/Node dropout strategy to refine the user–item bipartite graph. For instance, DropEdge [47] randomly removes a portion of edges form the model training process, where each edge has the equal probability to be pruned no mater how important they are. Inspired by recent researches on model sparsification [46,47], we propose a degree-aware edge pruning strategy to alleviate the popularity bias problem. The core concept is to assign a higher probability to prune popular user–item interactions, so that we can decrease the model's tendency to recommend mainstream items and encourage a broader range of choices.

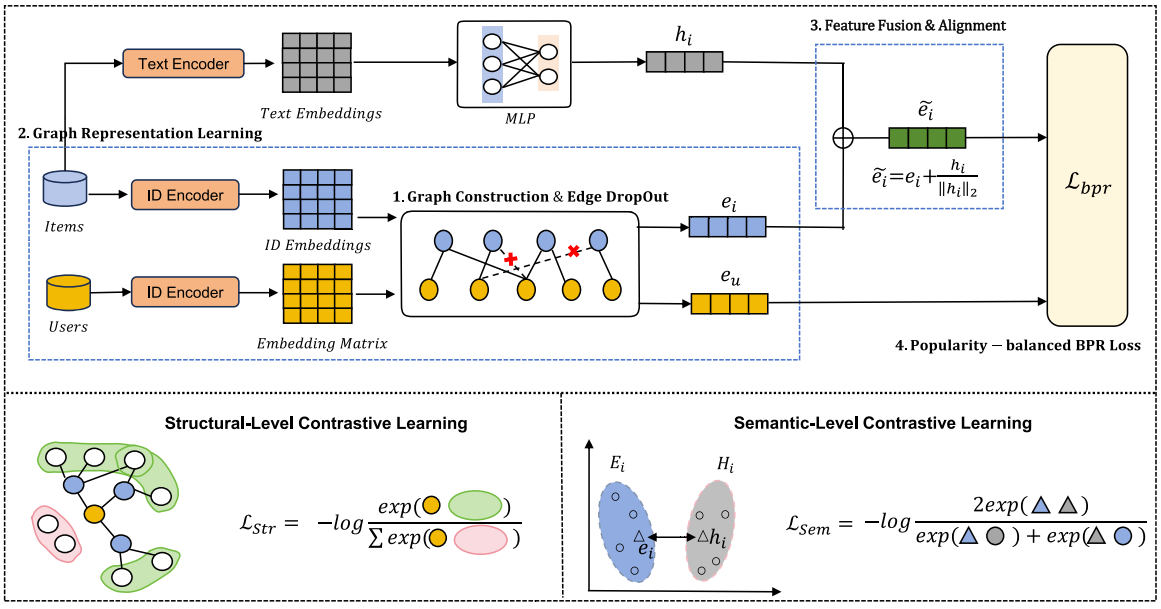


Fig. 1. Overview of the TPGRec method. The model mainly contains four components: Graph construction, Graph representation learning, Feature fusion & Alignment, and popularity-balanced BPR optimization. E_i denotes the ID-based embeddings for the item set, H_i represents the Text-based embeddings for the item set, and $\widetilde{e}_i$ refers to the item embedding generated by feature fusion. TPGRec is jointly optimized by $\mathcal{L}_{bpr}$, $\mathcal{L}_{str}$ and $\mathcal{L}_{sem}$.

Assume that $\epsilon_k \in \mathcal{E}(0 \le k < |\mathcal{E}|)$ represents an edge between nodes i and j, while ω_i and ω_j denote the degrees of i and j, respectively. For any ϵ_k, we calculate $p_k = \frac{1}{\sqrt{\omega_i}\sqrt{\omega_j}}$ as the retaining probability, which means edges connecting low-degree nodes have a higher chance to be reserved in the graph $\mathcal{G}$. As for model training, we follow DropEdge to refine the user–item graph by removing $\lfloor \rho_{drop}|\mathcal{E}| \rfloor$ edges according to their retaining probability using the multinomial sampling method, where ρ_{drop} is the proportion of discarded edges from $\mathcal{G}$. Afterwards, with the remained edges, we construct and normalize the adjacency matrix A iteratively in each training epoch.

(2) Boosting the training for Long-tail Items

Except for the degree-aware edge pruning strategy, we also try to amplify the effect of less popular user–item interactions. More precisely, we first set a proportion ρ_{add}, and sample n ($n = \rho_{add}|\mathcal{E}|$) edges with replacement from the multinomial distribution with the parameter vector $p = \{p_0, p_1, \ldots, p_{|\mathcal{E}-1|}\}$, where $p_k = \frac{1}{\sqrt{\omega_i}\sqrt{\omega_j}}$. In this manner, less popular edges could be sampled from the graph $\mathcal{G}$ with a greater probability. Considering that existing edges cannot be duplicated in the graph, we incorporate these sampled interactions into the training set for model updating, and this operation may ultimately enhance the visibility and influence of long-tail items during the training.

4.2. Graph representation learning

4.2.1. Graph collaborative filtering backbone

As an efficient and widely-used graph encoder, LightGCN [2] removes redundant operations such as transformation matrices and activation functions to achieve superior performance. In our implementation, we utilize LightGCN as the backbone of TPGRec, where the message passing and embedding propagation are as follows:

$$e_u^{(l+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e_i^{(l)}, \tag{4}$$

$$e_i^{(l+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} e_u^{(l)}. \tag{5}$$

Given each initial user representation $e_u^{(0)}$ and item representation $e_i^{(0)}$, after performing the linear propagation for L times on the user–item interaction graph, the final node embedding is the weighted sum of embeddings learned across all layers:

$$e_u = \frac{1}{L+1}\sum_{l=0}^{L} e_u^{(l)}, \quad e_i = \frac{1}{L+1}\sum_{l=0}^{L} e_i^{(l)}. \tag{6}$$

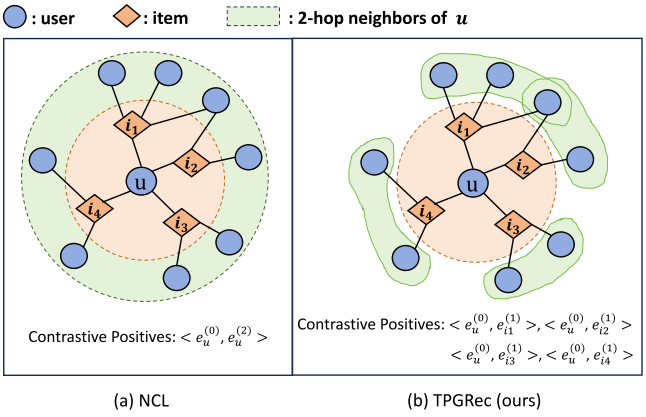


Fig. 2. Comparative analysis of structural contrastive strategies between NCL and TPGRec.

4.2.2. Structural-level contrastive learning

To obtain superior node representations, early GNN-based CF methods create contrastive views via random graph augmentation or node representation masking, which may result in structural information damage and undesirable noise. Recent work like NCL [5], proposes a layer-to-layer contrastive learning paradigm for representation learning. It employs the kth layer's output from GNN as the representation of k-hop neighbors for a node and devises a structure-aware contrastive learning strategy, which aims at aligning the representations of a node and its even-hop neighbors. As shown in Fig. 2(a), for a given user u, NCL creates a contrastive positive pair $\langle e_u^{(0)}, e_u^{(2)} \rangle$, where $e_u^{(0)}$ is the initial user embedding and $e_u^{(2)}$ denotes the aggregated information of 2-hop neighbors of u (located in the green zone).

Nevertheless, the preference information in the entire 2nd-layer neighbors may highly intricate and diverse, directly aggregating their representations as the positive sample may introduce excessive noise, potentially overwhelming the truly significant features with irrelevant ones. To address this problem, we partition the second-layer neighbors of u into multiple groups, and aggregate representations within each group as the positive sample corresponding to u. Take Fig. 2(b) as an example, user nodes in the green zone from each group interact with a same item i, which are more likely to exhibit a relatively unified preference. We use $e_i^{(1)}$ as the aggregated information of each group and create 4 contrastive positive pairs for u: $\langle e_u^{(0)}, e_{i_1}^{(1)}\rangle$, $\langle e_u^{(0)}, e_{i_2}^{(1)}\rangle$, $\langle e_u^{(0)}, e_{i_3}^{(1)}\rangle$, and $\langle e_u^{(0)}, e_{i_4}^{(1)}\rangle$. Our contrastive learning strategy avoids using the outputs of deep convolution, reducing the noises propagated from high-order neighbors.

Formally, our contrastive learning loss function is constructed by utilizing $e_u^{(0)}$ and $e_i^{(1)}$, as well as $e_i^{(0)}$ and $e_u^{(1)}$:

$$\mathcal{L}_{str}^{U} = -\frac{1}{M}\sum_{u\in\mathcal{U},\ R_{u,i}=1}\log\frac{\exp(s(\mathbf{e}_u^{(0)},\mathbf{e}_i^{(1)})/\tau)}{\sum_{j\in\mathcal{I}}\exp(s(\mathbf{e}_u^{(0)},\mathbf{e}_j^{(1)})/\tau)}, \tag{7}$$

where $e_i^{(1)}$ is output of GNN at layer 1. In one batch, item i is the direct one-hop neighbors of user u in a bipartite graph, whose representation can be obtained like:

$$e_i^{(1)} = \sum_{u\in\mathcal{N}_i}\frac{1}{\sqrt{|\mathcal{N}_u|\,|\mathcal{N}_i|}}e_u^{(0)}. \tag{8}$$

Similarly, the structural contrastive learning loss of the item side can be captured as:

$$\mathcal{L}_{str}^{I} = \frac{1}{N}\sum_{i\in\mathcal{I},\ R_{u,i}=1} -\log\frac{\exp(s(\mathbf{e}_i^{(0)},\mathbf{e}_u^{(1)})/\tau)}{\sum_{k\in\mathcal{U}}\exp(s(\mathbf{e}_i^{(0)},\mathbf{e}_k^{(1)})/\tau)}, \tag{9}$$

where $e_u^{(1)}$ is calculated as follow:

$$e_u^{(1)} = \sum_{i\in\mathcal{N}_u}\frac{1}{\sqrt{|\mathcal{N}_u|\,|\mathcal{N}_i|}}e_i^{(0)}. \tag{10}$$

Finally, the structural-level contrastive loss $\mathcal{L}_{str}$ is the sum of the user-side loss and item-side loss:

$$\mathcal{L}_{str} = \mathcal{L}_{str}^{U} + \mathcal{L}_{str}^{I}. \tag{11}$$

4.3. Feature fusion & alignment

Most existing graph-based collaborative filtering approaches utilize ID-based embeddings for items, which typically lack the semantic richness required to accurately capture high-order relationships between nodes. Furthermore, long-tail items receive significantly fewer exposure opportunities during model training, often resulting in less satisfactory recommendation performance compared to head items. To overcome these limitations, we leverage the text data of items to develop informative representations, and then propose a semantic-level contrastive learning strategy to effectively fuse these textual features with ID-based item embeddings. This approach not only enhances the model's understanding of item characteristics but also implicitly establishes semantic relationships between popular and long-tail items. Consequently, the exposure opportunities for long-tail items are naturally increased when they share similarities with popular head nodes, effectively alleviating the long-tail problem.

Specifically, for each item i, we first organize its title, category, brand and description into a text document, denoted by $\{w_1, w_2, \dots, w_c\}$, where c represents the length of the document. Subsequently, an advanced language model, Sentence-BERT [48], is utilized to generate a comprehensive embedding $x_i \in \mathbb{R}^{d_m}$ for each text sequence. For the efficiency and performance concerns, we use a MLP to transform the raw text vector into a high-level feature $\mathbf{h}_i$:

$$\mathbf{h}_i = \mathbf{W}x_i + \mathbf{b}, \tag{12}$$

$$x_i = Bert(\{w_1, w_2, \dots, w_c\}), \tag{13}$$

where $\mathbf{W} \in \mathbb{R}^{d_m\times d}$ and $\mathbf{b} \in \mathbb{R}^d$ are the trainable transformation matrix and bias vector. Finally, we calculate the text-enhanced representation $\widetilde{\mathbf{e}}_i$ for item i by fusing its ID embedding $\mathbf{e}_i$ and text embedding $\mathbf{h}_i$ as follow:

$$\widetilde{\mathbf{e}}_i = \mathbf{e}_i + \frac{\mathbf{h}_i}{\|\mathbf{h}_i\|_2}. \tag{14}$$

Based on the final node embeddings e_u and $\widetilde{e}_i$, we calculate their inner product $y_{u,i}$ as the preference score, denoted by:

$$y_{u,i} = e_u^T\widetilde{e}_i. \tag{15}$$

Acknowledging that $\mathbf{e}_i$ and $\mathbf{h}_i$ are generated from different perspectives, which can be regarded as separate modalities, we propose a semantic-level contrastive learning strategy to reduce the gap between their distributions. More precisely, we make e_i and h_i be positive samples for each other, and simultaneously conduct negative sampling in their own representation spaces, so as to avoid favoring either side in the embedding fusion process. The semantic contrastive learning loss $\mathcal{L}_{sem}$ can be calculated as:

$$\mathcal{L}_{sem} = -\frac{1}{N}\sum_{i\in\mathcal{I}}\log\frac{2\exp(s(\mathbf{e}_i,\mathbf{h}_i)/\tau)}{\sum_{j\in\mathcal{I}}\exp(s(\mathbf{e}_i,\mathbf{h}_j)/\tau)+\sum_{j\in\mathcal{I}}\exp(s(\mathbf{e}_j,\mathbf{h}_i)/\tau)}, \tag{16}$$

where $s(\cdot)$ is the cosine similarity function and τ is the temperature parameter.

4.4. Popularity-balanced BPR optimization

As detailed in Section 3, the standard BPR optimization approach constructs contrastive pairs via stochastic sampling, and then encourages the preference score of an observed item to be greater than those unobserved entries. From Eq. (3), we have identified two significant shortcomings of the naive BPR loss. At first, the training set is predominantly composed of less popular interactions, consequently increasing the likelihood of long-tail items being selected as negative samples. Since those negative samples are easier to distinguish, the BPR optimization process might ultimately make a limited contribution to enhance the model's discriminative power. Additionally, during the BPR optimization phase, negative samples are treated equally no matter how similar they are to the item that u has interacted with. This inevitably poses challenges for the model to distinguish hard negatives and may fail to accurately predict users' preference scores for items.

To address the first challenge, we categorize the candidate items into two groups: the head and tail, denoted by $\mathcal{I}^h$ and $\mathcal{I}^t$, respectively. For each user u in the training batch, we construct two contrastive training pairs by randomly selecting a negative sample from $\mathcal{I}^h$ and $\mathcal{I}^t$, respectively. After calculating the BPR loss from each group, we introduce a parameter γ to adaptively regulate the model's focus on these two BPR components. Moreover, to address the second challenge, we incorporate a weighting factor α into the BPR loss function. This aids the model in adaptively identifying and handling negative samples of varying difficulty. Specifically, for any user u, if a negative item j is similar to the current positive item i, we regard it as a hard negative sample. A higher penalty is then assigned to force the model to distinguish the sample when calculating the preference score for j. Conversely, a smaller weight is applied. In our approach, the similarity between the embeddings of item i and item j is used to determine the value of α.

Formally, the improved BPR loss is calculated as follow:

$$\begin{aligned}
\mathcal{L}_{bpr} &= (1-\gamma)\cdot\mathcal{L}_{bpr}^{h} + \gamma\cdot\mathcal{L}_{bpr}^{t},\\
\mathcal{L}_{bpr}^{h} &= -\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{N}_u}\sum_{j\notin\mathcal{N}_u\cap j\in\mathcal{I}^h}\log\sigma(y_{u,i}-\alpha_{i,j}\cdot y_{u,j}),\\
\mathcal{L}_{bpr}^{t} &= -\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{N}_u}\sum_{j\notin\mathcal{N}_u\cap j\in\mathcal{I}^t}\log\sigma(y_{u,i}-\alpha_{i,j}\cdot y_{u,j}),
\end{aligned} \tag{17}$$

where $\mathcal{N}_u$ is the neighborhood set of u, j is a randomly selected negative sample, and $\sigma(\cdot)$ is the sigmoid function. $\alpha_{i,j}$ is calculated by $\widetilde{e}_i^T \widetilde{e}_j$, where $\widetilde{e}_i$ and $\widetilde{e}_j$ denote the item embeddings calculated by Eq. (14), and $y_{u,i}$ refers to the inner product of e_u and $\widetilde{e}_i$ using Eq. (15).

Note that γ is a hyperparameter for controlling our focus on long-tail items, which is denoted by:

$$\gamma = (1 - \frac{t}{T}) \cdot \gamma_{max} + \frac{t}{T} \cdot \gamma_{min}, \tag{18}$$

where T refers to the total number of training epochs, and t is the current epoch. γ_{max} and γ_{min} are the margins that determine the value of γ. In our implementation, we set γ_{max} to 0.8 according to the 80/20 rule, while a γ_{min} of 0.5 indicates that equal attention is given to both the head and tail items. As the iteration index t increases in the training process, the value of γ gradually decays, promoting the model to shift its focus from easily distinguishable tail negative samples to head negative samples that pose a certain level of discernment difficulty. Consequently, this strategy amplifies the model's opportunity to refine its discriminatory capability.

4.5. Model training

To extract more significant node representations for TPGRec, we use a multi-task training strategy to jointly optimize the recommendation task and two contrastive learning tasks. Formally, the final loss function $\mathcal{L}$ is given by:

$$\mathcal{L} = \mathcal{L}_{bpr} + \lambda_1 \mathcal{L}_{str} + \lambda_2 \mathcal{L}_{sem} + \lambda_3 \|\Theta\|_2, \tag{19}$$

where $\mathcal{L}_{bpr}$ refers to the popularity-balanced BPR loss (Eq. (17)), λ_1, λ_2, and λ_3 are hyperparameters that control the strength of structural-level contrastive learning loss $\mathcal{L}_{str}$ (Eq. (11)), semantic-level contrastive learning loss $\mathcal{L}_{sem}$ (Eq. (16)), and L_2 regularization (Θ refers to the parameters of model), respectively.

5. Experiments

In this section, we have conducted a series of experiments to answer the following research questions:

- **RQ1:** How does TPGRec perform compared to the state-of-the-art GNN-based methods?
- **RQ2:** How do the text enhancement, graph contrastive learning and popularity smoothing modules contribute to the performance of TPGRec?
- **RQ3:** Does TPGRec improve the recommendation performance for long-tail items?
- **RQ4:** How do different hyperparameter settings affect TPGRec?
- **RQ5:** Does TPGRec learn embeddings effectively?

5.1. Experimental settings

Datasets. We conduct experimental evaluation on four categories of publicly available Amazon review datasets introduced in [49]: Baby, Sports, Clothing, and Beauty. Following existing research [22], we maintain the '5-core' setting for all datasets, where each user and each item has at least 5 interactions. Detailed statistics of these datasets are summarized in Table 1. In addition, we create the initial text embedding for each item by organizing its title, brand, category and description into a sentence, and then we use Sentence-BERT [48] to capture a 768-dimensional representation.

Evaluation Metrics. We adopt two widely-used top-k metrics to evaluate the model's performance: Recall@k and NDCG@k with $k = \{10, 20\}$. In the experiments, historical interactions within each dataset are partitioned into training, validation, and testing sets in an 8:1:1 ratio, respectively. We apply the all-ranking protocol for calculating the evaluation metrics. During the training process of each baseline method, we randomly select a negative sample that the user has not encountered before to form a training pair for BPR optimization. For our TPGRec method, we construct two training pairs by selecting a negative sample from both the head and the tail parts of item set, and then calculate the improved BPR loss using Eq. (17).

Table 1
Statistics of the datasets.

Dataset	#Users	#Items	#Interactions	Density
Baby	19,445	7050	160,792	0.00117
Sports	35,598	18,357	296,337	0.00045
Clothing	39,387	23,033	278,677	0.00031
Beauty	22,363	12,101	198,502	0.00073

Baselines. To verify the effectiveness of our proposed method, we have included a range of state-of-the-art ID-based graph collaborative filtering approaches (i.e., LightGCN, NCL, XSimGCL), and Text-fused graph collaborative filtering algorithms (i.e., MMGCN, GRCN, MICRO) for model comparison.

- **LightGCN** [2] simplifies the design of GCNs by eliminating some unnecessary operations like nonlinear activation, feature transformation, and self-joins, thereby maintaining a light-weight yet effective graph learning model.
- **MGDCF** [31] treats GNN as an untrainable Markov process that can construct constant context features of vertices for a fully-connected layer that encodes context features. It utilizes simple yet powerful InfoBPR loss to optimize the parameters.
- **NCL** [5] proposes a suite of contrastive learning strategies built upon LightGCN, which can capture the correlations between a node and its context.
- **XSimGCL** [6] is a very recent graph contrastive learning approach for recommendation, which presents a simple but effective noise-based embedding augmentation technique to create views for contrastive learning and can smoothly adjust the distribution of learned embeddings.
- **MMGCN** [19] is one of the representative multi-modal recommendation methods, which constructs a user–item graph for each modality to derive node embeddings via GCNs, and then fuses all modal-specific embeddings with ID-embeddings to capture the representations of users or items.
- **RGCN** [20] is also one of the representative multi-modal recommendation methods. It identifies and removes the false-positive edges from the user–item interaction graph, and then perform recommendation using GCNs on the refined graph.
- **MICRO** [22] is a very recent and state-of-the-art multimodal recommendation method. It constructs an extra item-item graph as Lattice [21] did, and proposes contrastive learning strategies to fuse the item embeddings learned from different modality using LightGCN.
- **MGCN** [23] enhances MICRO by employing gating networks to purify the multimodal data and splitting the representations into modality-shared and modality-specific parts.

Implementation Details. We have implemented all methods using PyTorch with Python 3.9 and conducted model training on NVIDIA RTX A6000 GPUs. To ensure a fair comparison, we standardized the use of 2 GNN layers across all methods and uniformly set the embedding dimension to 64. Model parameters are initialized using the Xavier initializer and all methods are optimized with the Adam optimizer, where the learning rate is set to 0.001 and the batch size is 2048. The optimal hyperparameters are identified through a grid search, and we terminate training when there is no performance improvement on the validation set for 10 consecutive epochs. As for specific parameters of our TPGRec method, the temperature parameter τ is set to 0.5 for the Baby dataset and 0.15 for the remaining datasets, tailored for graph contrastive learning and semantic contrastive learning. The values of λ_1 and λ_2 are 0.03 across the four datasets. For the Sports dataset,

Table 2
Overall performance comparison of different models: the bold and underlined results represent the best and second-best performances, respectively. Additionally, the best result among ID-based methods is marked with a dashed underline.

Dataset	Metric	ID-based method				Text-fused method					Improv.
		LightGCN	MGDCF	NCL	XSimGCL	MMGCN	GRCN	MICRO	MGCN	TPGRec	
Baby	R@10	0.0479	0.0485	0.0476	0.0480	0.0427	0.0542	0.0561	0.0611	**0.0644**	5.40%
	R@20	0.0759	0.0782	0.0772	0.0744	0.0672	0.0842	0.0860	0.0924	**0.1006**	8.87%
	N@10	0.0256	0.0258	0.0255	0.0260	0.0225	0.0290	0.0314	0.0323	**0.0343**	6.19%
	N@20	0.0327	0.0329	0.0330	0.0327	0.0288	0.0369	0.0393	0.0402	**0.0434**	7.96%
Sports	R@10	0.0571	0.0585	0.0573	0.0625	0.0412	0.0606	0.0629	0.0676	**0.0766**	13.31%
	R@20	0.0845	0.0871	0.0857	0.0898	0.0659	0.0910	0.0917	0.1011	**0.1126**	11.37%
	N@10	0.0316	0.0323	0.0317	0.0350	0.0215	0.0334	0.0351	0.0357	**0.0420**	17.65%
	N@20	0.0386	0.0394	0.0390	0.0420	0.0283	0.0418	0.0427	0.0442	**0.0512**	15.84%
Clothing	R@10	0.0352	0.0368	0.0333	0.0354	0.0240	0.0429	0.0488	0.0567	**0.0615**	8.47%
	R@20	0.0534	0.0556	0.0500	0.0517	0.0387	0.0673	0.0728	0.0856	**0.0912**	6.54%
	N@10	0.0192	0.0190	0.0177	0.0194	0.0128	0.0226	0.0270	0.0237	**0.0331**	22.59%
	N@20	0.0238	0.0241	0.0220	0.0233	0.0166	0.0292	0.0331	0.0304	**0.0406**	22.66%
Beauty	R@10	0.0855	0.0872	0.0880	0.0943	0.0634	0.0956	0.1003	0.1009	**0.1163**	15.26%
	R@20	0.1217	0.1248	0.1268	0.1301	0.0804	0.1377	0.1434	0.1492	**0.1630**	9.25%
	N@10	0.0463	0.0475	0.0494	0.0543	0.0315	0.0542	0.0569	0.0549	**0.0655**	15.11%
	N@20	0.0567	0.0586	0.0594	0.0634	0.0367	0.0657	0.0687	0.0674	**0.0776**	12.95%

Table 3
Overall, head and tail performance comparison between different models(Recall@20). The bold and the underline indicate the optimal and the runner-up results, respectively.

Dataset	Baby			Sports			Clothing			Beauty		
	Overall	Head	Tail	Overall	Head	Tail	Overall	Head	Tail	Overall	Head	Tail
LightGCN	0.0759	0.1248	0.0061	0.0845	0.1475	0.0098	0.0534	0.1110	0.0068	0.1217	0.2042	0.0300
MGDCF	0.0782	0.1288	0.0063	0.0871	0.1528	0.0061	0.0556	0.1124	0.0082	0.1248	0.2024	0.0253
XSimGCL	0.0744	0.1202	0.0056	0.0900	0.1602	0.0081	0.0522	0.1092	0.0061	0.1301	0.2252	0.0374
MICRO	0.0864	0.1334	0.0183	0.0963	0.1581	**0.0209**	0.0778	0.1460	0.0222	0.1448	0.2207	0.0545
MGCN	0.0924	0.1450	0.0149	0.1011	0.1730	0.0164	0.0856	0.1512	0.0230	0.1492	0.2262	0.0569
TPGRec-Str	0.0969	0.1565	0.0166	0.1013	0.1771	0.0169	0.0845	0.1650	0.0212	0.1533	0.2406	0.0502
TPGRec-Sem	0.0972	0.1598	0.0098	0.1080	0.1893	0.0172	0.0759	0.1553	0.0137	0.1512	0.2458	0.0452
TPGRec-T	0.0873	0.1461	0.0066	0.0863	0.1522	0.0113	0.0634	0.1290	0.0108	0.1198	0.2045	0.0305
TPGRec-P	0.0979	0.1611	0.0172	0.1109	**0.1945**	0.0173	**0.0910**	**0.1787**	0.0216	0.1621	**0.2595**	0.0554
TPGRec	**0.1006**	**0.1646**	**0.0204**	**0.1126**	0.1942	0.0204	0.0909	0.1752	**0.0253**	**0.1630**	0.2560	**0.0612**

the parameters ρ_{add} and ρ_{drop} are specifically adjusted to 0.1 and 0.3, while for the other datasets, these parameters were set to 0.2 and 0.3, respectively.

5.2. Overall performance (for RQ1)

Table 2 presents the performance of all compared methods on the four Amazon datasets. The optimal result is highlighted in bold, the runner-up is underlined, and the best result among ID-based methods is marked with a dashed underline. From the experimental results, we can summarize the following key observations:

- Our TPGRec method consistently and significantly outperforms all the baseline methods on the four datasets. Compared to the suboptimal results, TPGRec demonstrates average improvement ratios of 10.61% in Recall@10, 9.01% in Recall@20, 15.39% in NDCG@10, and 14.85% in NDCG@20. This suggests that by incorporating the proposed text enhancement, graph contrastive learning, and popularity smoothing modules into the LightGCN framework, we can effectively utilize the textual information of items to build a better graph-based collaborative filtering model and ultimately improve the recommendation performance.
- Among the ID-based methods, NCL and XSimGCL adopt the layer-to-layer contrastive learning strategy and outperform LightGCN, which verifies the effectiveness of representation alignment between layers in graph-based collaborative filtering. In addition, we notice that NCL generally underperforms XSimGCL. One possible reason is that important features in NCL may be excessively smoothed by a large number of second-order neighbors, which may lead to unsatisfactory second-order embeddings, thereby negatively impacting the final performance of graph contrastive learning.
- Among the text-based baselines, MICRO and MGCN outperform MMGCN and GRCN on four datasets. This superiority can be attributed to the fact that MMGCN and GRCN integrate textual information into the embedding learning process at a lower layer or an earlier stage, which may result in reduced information across layers. Moreover, they combine the ID embedding with text representations through summation or concatenation, potentially leading to coarse node embeddings. In contrast, MICRO and MGCN incorporates textual information at a higher layer or a stage closer to the final recommendation task, which avoids damaging the original distribution of representations through complicated models. Furthermore, when comparing MICRO to MGCN, MGCN exhibits superior performance in the majority of cases across the four datasets. This is attributed to its enhanced ability to refine multimodal data and improve representation learning.
- In general, text-based methods will benefit from the textual information and surpass the ID-based recommendation methods. However, we observe that MMGCN exhibits inferior performance than LightGCN. One reason is the way of utilizing text feature we just analyzed, while another one is that MMGCN is constructed on a GNN backbone that retains linear transformations, activation functions, and other operations that LightGCN has proven to be unfavorable to collaborative filtering. While GRCN is not built on LightGCN, it eliminates some unnecessary operations and fuses ID and text embeddings through a more sophisticated attention mechanism, ultimately yielding superior results.

5.3. Ablation study (for RQ2)

In this subsection, we conduct ablation studies to demonstrate the impact of text enhancement, graph contrastive learning, and popularity

smoothing modules separately to answer RQ2. To be specific, we design the following variants of TPGRec and compare their recommendation accuracy.

- **TPGRec-Str** means that the graph contrastive learning module is removed from our model. In other words, we discard the contrastive loss $\mathcal{L}_{Str}$.
- **TPGRec-SCL** refers to that the structural-level contrastive learning component in TPGRec has been replaced with the corresponding approach used in NCL.
- **TPGRec-Sem** denotes that we fuse text and ID embeddings for items without using the semantic contrastive loss $\mathcal{L}_{Sem}$.
- **TPGRec-T** represents that the whole text enhancement module is removed from our model.
- **TPGRec-B** indicates replacing the improved BPR loss with the commonly-used version using Eq. (3) as the baseline methods did.
- **TPGRec-P** represents that the popularity smoothing module is subtracted from TPGRec.

Fig. 3 illustrates the performance metrics, Recall@20 and NDCG@20, for all introduced variants of TPGRec on four Amazon datasets. A noticeable decline in overall recommendation performance is observed whenever a component is removed from TPGRec, thereby validating the efficacy of the text enhancement, graph contrastive learning, and popularity smoothing modules. Furthermore, we notice that the majority of contributions come from the text enhancement joint with semantic contrastive learning. This is because this operation incorporates more information into item embeddings and establishes implicit semantic correlations between items. Consequently, we not only boost the overall recommendation performance but also construct bridges to connect the popular and similar long-tail items, eventually alleviating the long-tail issue.

We can find that TPGRec outperforms TPGRec-SCL, and TPGRec-SCL generally surpasses TPGRec-Str. This demonstrates the effectiveness of layer-to-layer structural contrastive learning and highlights the superiority of our method over NCL. In fact, our method strikes a balance between the aggregation of the entire set of two-hop neighbors and each specific two-hop neighbor. It mitigates the impact of interest confusion caused by overall aggregation and avoids the extensive computation and prolonged convergence time required for aligning the central node with each individual two-hop neighbor.

From Fig. 3, we notice that the impact of the popularity smoothing module is not as remarkable as that of text enhancement and contrastive learning strategies. This is because textual features inherently enrich node representations, and contrastive learning markedly increases the training frequency for tail items, effectively enhancing overall recommendation quality. Nevertheless, it is crucial to emphasize that the popularity smoothing strategy does bring substantial improvements for tail items. As shown in Table 3, our TPGRec method achieves an average improvement rate of 16.03% for tail items compared to TPGRec-P, without any significant negative impact on overall and head metrics.

Given that the refined BPR loss modestly contributes to the overall performance of TPGRec, we have conducted additional experiments to explore its impact on model convergence. As detailed in Table 4, our optimized BPR module has significantly accelerated the model training, potentially reducing the required training epochs by one-quarter to one-half. This improvement is attributed to that we consider the difference between negative samples and increase the likelihood of selecting more challenging negative samples from both popular and long-tail items for BPR optimization.

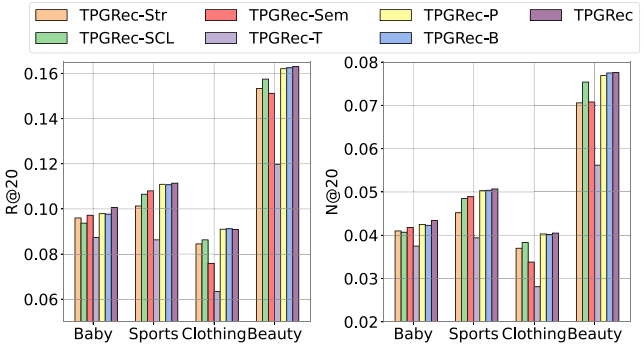


Fig. 3. Results of ablation studies on the four Amazon datasets.

Table 4
The number of epochs when TPGRec and TPGRec-B reach optimal performance on the four datasets.

	Baby	Sports	Clothing	Beauty
TPGRec-B	46	54	131	94
TPGRec	33	42	70	74
Accellerate	28%	22%	47%	21%

5.4. Long-tail item recommendation (for RQ3)

To evaluate the ability of our method in promoting long-tail items, we categorize the testing set items into 'head' and 'tail' segments, using a ratio of 2:8 according to their popularity across the entire dataset, with the head part comprising more popular items. Subsequently, we measure the Recall@20 metric for each group, comparing our algorithm to some leading competitors (XSimGCL, MICRO, MGDCF and MGCN).

As depicted in Table 3, in contrast to ID-based graph collaborative filtering methods like XSimGCL and MGDCF, our TPGRec, along with MICRO and MGCN, integrate textual data into the embedding learning process, creating implicit semantic associations between items and markedly enhancing the recommendation performance for long-tail items.

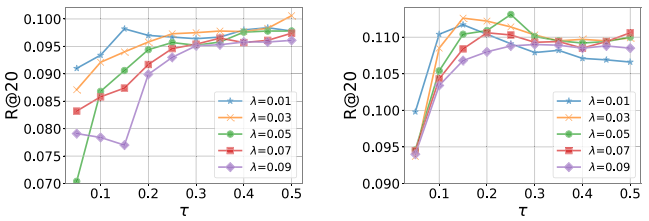
Furthermore, compared with MICRO and MGCN, our TPGRec method has achieved greater recall values in the context of promoting long-tail items upon most occasions. Regarding overall performance and the promotion of head items, TPGRec has demonstrated a consistent and significant advantage over MICRO and MGCN. This superiority is primarily attributed to our popularity smoothing strategies and the graph contrastive learning approach, which substantially enhance the representation learning for both users and items.

5.5. Hyperparameter analysis (for RQ4)

To streamline the hyperparameter tuning process, we assign the same value of λ as the weight for both the structural and semantic contrastive learning losses, i.e. $\mathcal{L}_{str}$ and $\mathcal{L}_{sem}$ in Eq. (19), where the temperature parameter τ for $\mathcal{L}_{str}$ and $\mathcal{L}_{sem}$ are also identical. Furthermore, TPGRec has two critical hyperparameters: ρ_{drop} denotes the ratio of edge dropout and ρ_{add} denotes the ratio of user–item interactions that repeatedly appear in the training set. Next, we investigate the model's sensitivity to these four hyperparameters using the Baby and Sports datasets.

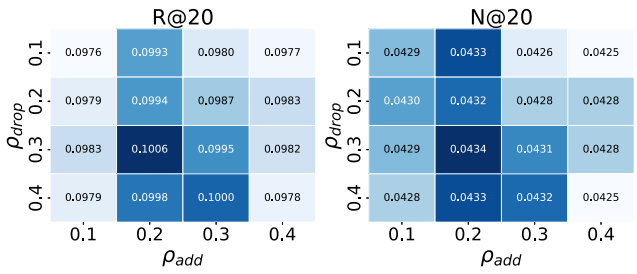
5.5.1. Effects of λ and τ

We conduct experiments with varying combinations of λ and τ, utilizing the set $[0.01, 0.03, 0.05, 0.07, 0.09]$ for λ and the set $[0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50]$ for τ. Fig. 4 depicts the results of Recall@20, revealing that our TPGRec method achieves the best performance on the Baby and Sports datasets when $\lambda = 0.03$. Experimental

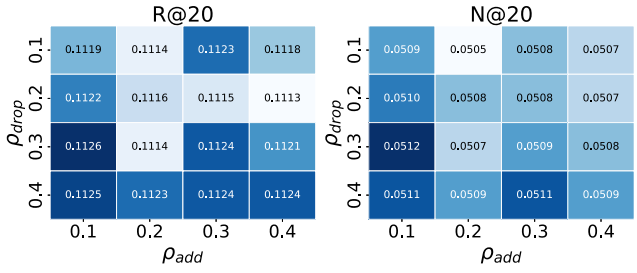


(a) Baby (b) Sports

Fig. 4. Performance comparison w.r.t. different temperature parameter τ and weight λ of contrastive learning loss on Baby and Sports datasets.



(a) Baby



(b) Sports

Fig. 5. Performance analysis for different combinations of ρ_{add} and ρ_{drop}.

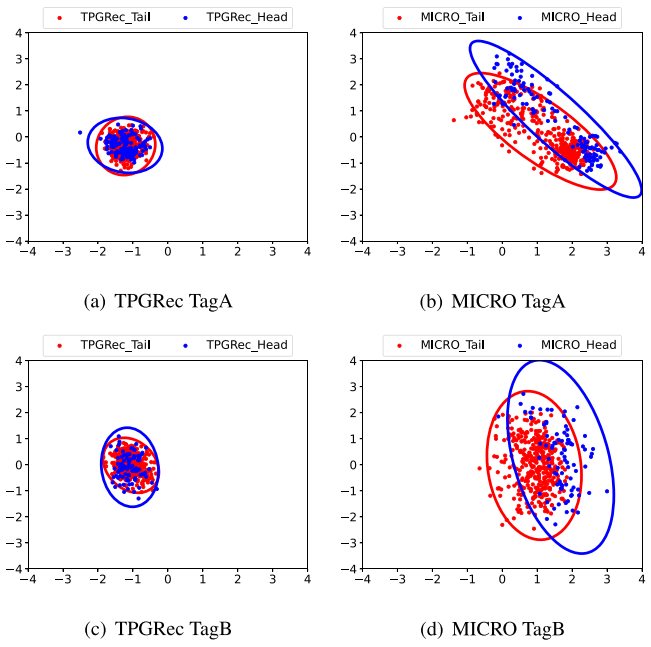
results suggest that TPGRec is more sensitive to τ compared to λ. Generally, a lower value of λ tends to yield superior results. As τ increases, Recall@20 for the Baby dataset exhibits a consistent upward trend, and reaches its peak at around $\tau = 0.5$, while the peak point on Sports lies at around $\tau = 0.15$, which is consistent with the conclusion in NCL and XSimGCL, i.e., the sparser the dataset, the smaller the appropriate temperature parameter should be.

5.5.2. Effects of ρ_{add} and ρ_{drop}

Similarly, we assign values to both ρ_{add} and ρ_{drop} within the range of $[0.1, 0.2, 0.3, 0.4]$. Subsequently, we display the Recall@20 results for the Baby and Sports datasets under various combinations of ρ_{add} and ρ_{drop} in Fig. 5. On both datasets, the best performance is achieved when ρ_{drop} = 0.3, whereas the best ρ_{add} settings for Baby and Sports are 0.2 and 0.1, respectively. It is evident that the model is more sensitive to ρ_{add}, as indicated by the significant variation in the color bar with changes in ρ_{add}. Additionally, it is inadvisable to set either ρ_{add} or ρ_{drop} too large, given that performance begins to deteriorate when ρ_{add} or ρ_{drop} reaches 0.4.

5.6. Visualization of node embedding (for RQ5)

Following the case study presented in [50,51], we conduct a qualitative assessment of the item representations created by our TPGRec method. Owing to space limitation, we choose the Sports dataset and include the best competitive approach, i.e. MICRO, for performance evaluation, where item embeddings are graphically represented in two dimensions using the PCA and t-SNE techniques. Specifically, we fist randomly select two categories within the Sports dataset, namely "Gun Scopes" (tag A) and "Archery" (tag B). Afterwards, we categorize items based on the presence of either tag A or tag B, and further divide the them into head and tail parts according to the criteria outlined in Section 5.4. Corresponding results are presented in Figs. 6 and 7, respectively, where head items are denoted by blue points, while the tail items are indicated by red. Clearly, item embeddings generated by TPGRec exhibit a more confined distribution range, signifying a more cohesive clustering of items that share similar characteristics. More significantly, when compared with MICRO, a notable merging of the head and tail item distributions under the same tag is observed using our method. This overlap highlights the effectiveness of our TPGRec method in tackling the long-tail challenge in recommendation systems.



(a) TPGRec TagA (b) MICRO TagA

(c) TPGRec TagB (d) MICRO TagB

Fig. 6. Visualization of the items representations generated by TPGRec and MICRO on sports using PCA.

6. Conclusion

In this paper, we have proposed TPGRec, a GNN-based collaborative filtering approach that boosts overall recommendation performance while tackling the long-tail issue. To capture insightful user and item representations, we propose a structural-level contrastive learning technique for graph representation learning, and develop a semantic-level contrastive learning method to effectively fuse ID embeddings with textual data. To balance the contributions of head and tail items, we present a degree-aware edge dropout technique, enhance training for long-tail items, and derive a popularity-balanced BPR loss for model optimization. Experimental results on four real-world datasets have demonstrated the superiority of TPGRec compared with the state-of-the-art baselines. For future enhancements, we intend to leverage LLMs to enrich text information for items and refine user–item interactions within the graph structure. Additionally, we plan to extend our research to more sophisticated recommendation scenarios, such as cross-domain and multi-modal recommendations.

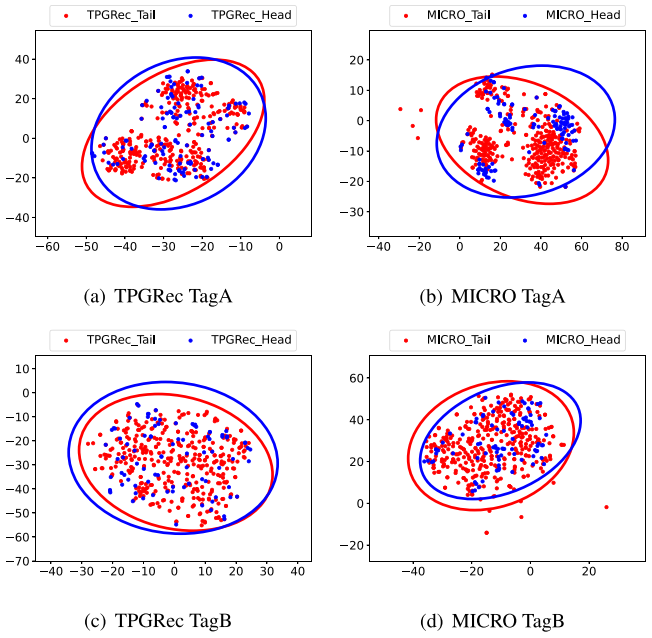


Fig. 7. Visualization of the item representations generated by TPGRec and MICRO on sports using t-SNE.

CRediT authorship contribution statement

Chenyun Yu: Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Junfeng Zhao:** Writing – original draft, Visualization, Software, Methodology, Investigation, Data curation. **Xuan Wu:** Validation, Software, Formal analysis, Data curation. **Yingle Luo:** Validation, Software, Formal analysis, Data curation. **Yan Xiao:** Writing – review & editing, Supervision, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by Shenzhen Science and Technology Program (Grant No. 202206193000001, 20220817180954005).

Data availability

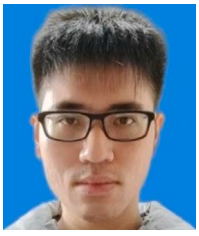
Data will be made available on request.

References

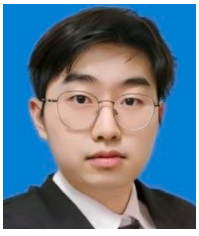
[1] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 173–182.
[2] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, LightGCN: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 639–648.
[3] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, X. Xie, Self-supervised graph learning for recommendation, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 726–735.
[4] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, Q.V.H. Nguyen, Are graph augmentations necessary? Simple graph contrastive learning for recommendation, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 1294–1303.
[5] Z. Lin, C. Tian, Y. Hou, W.X. Zhao, Improving graph collaborative filtering with neighborhood-enriched contrastive learning, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 2320–2329.
[6] J. Yu, X. Xia, T. Chen, L. Cui, N.Q.V. Hung, H. Yin, XSimGCL: Towards extremely simple graph contrastive learning for recommendation, IEEE Trans. Knowl. Data Eng. 36 (3) (2023) 913–926.
[7] Y. Ge, X. Zhao, L. Yu, S. Paul, D. Hu, C.C. Hsieh, Y. Zhang, Toward Pareto efficient fairness-utility trade-off in recommendation through reinforcement learning, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pp. 316–324.
[8] X. Niu, B. Li, C. Li, R. Xiao, H. Sun, H. Deng, Z. Chen, A dual heterogeneous graph attention network to improve long-tail performance for shop search in E-commerce, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2020, pp. 3405–3415.
[9] Y. Cui, M. Jia, T.Y. Lin, Y. Song, S. Belongie, Class-balanced loss based on effective number of samples, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9268–9277.
[10] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[11] X. Yi, J. Yang, L. Hong, D.Z. Cheng, L. Heldt, A. Kumthekar, E. Chi, Sampling-bias-corrected neural modeling for large corpus item recommendations, in: Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 269–277.
[12] W. Ouyang, X. Zhang, S. Ren, L. Li, K. Zhang, J. Luo, Y. Du, Learning graph meta embeddings for cold-start ads in click-through rate prediction, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 1157–1166.
[13] B. Zhou, Q. Cui, X.S. Wei, Z.M. Chen, Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9719–9728.
[14] B. Li, Y. Hou, W. Che, Data augmentation approaches in natural language processing: A survey, AI Open 3 (2022) 71–90.
[15] Y. Zhang, D.Z. Cheng, T. Yao, X. Yi, L. Hong, E.H. Chi, Empowering long-tail item recommendation through cross decoupling network (CDN), in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 5608–5617.
[16] J. Liu, Y. Sun, C. Han, Z. Dou, W. Li, Deep representation learning on long-tailed data: A learnable embedding augmentation perspective, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2970–2979.
[17] Y. Zhang, D.Z. Cheng, T. Yao, X. Yi, L. Hong, E.H. Chi, A model of two tales: Dual transfer learning framework for improved long-tail item recommendation, in: Proceedings of the Web Conference 2021, 2021, pp. 2220–2231.
[18] T. Yao, X. Yi, D.Z. Cheng, F. Yu, T. Chen, A. Menon, E. Ettinger, Self-supervised learning for large-scale item recommendations, in: Proceedings of the 30th ACM International Conference on Information and Knowledge Management, 2021, pp. 4321–4320.
[19] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, T.S. Chua, MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 1437–1445.
[20] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, T.S. Chua, Graph-refined convolutional network for multimedia recommendation with implicit feedback, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 3541–3549.
[21] J. Zhang, Y. Zhu, Q. Liu, S. Wu, S. Wang, L. Wang, Mining latent structures for multimedia recommendation, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 3872–3880.
[22] J. Zhang, Y. Zhu, Q. Liu, S. Wu, S. Wang, L. Wang, Latent structure mining with contrastive modality fusion for multimedia recommendation, IEEE Trans. Knowl. Data Eng. 35 (9) (2023) 9154–9167.
[23] P. Yu, Z. Tan, G. Lu, B. Bao, Multi-view graph convolutional network for multimedia recommendation, in: Proceedings of the 31st ACM International Conference on Multimedia, 2023, pp. 6576–6585.
[24] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, 2012, arXiv Preprint arXiv:1205.2618.
[25] Y. Zheng, B. Tang, W. Ding, H. Zhou, A neural autoregressive approach to collaborative filtering, in: International Conference on Machine Learning, 2016, pp. 764–773.
[26] R. Lara-Cabrera, A. Gonzalez-Prieto, F. Ortega, Deep matrix factorization approach for collaborative filtering recommender systems, Appl. Sci. 10 (14) (2020) 4926.
[27] D. Liang, R.G. Krishnan, M.D. Hoffman, T. Jebara, Variational autoencoders for collaborative filtering, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 689–698.

[28] X. Wang, X. He, M. Wang, F. Feng, T.S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 165–174.
[29] R. Ying, R. He, K. Chen, P. Eksombatchai, W.L. Hamilton, J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 974–983.
[30] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, K. Weinberger, Simplifying graph convolutional networks, in: International Conference on Machine Learning, 2019, pp. 6861–6871.
[31] J. Hu, B. Hooi, S. Qian, Q. Fang, C. Xu, MGDCF: Distance learning via Markov graph diffusion for neural collaborative filtering, IEEE Trans. Knowl. Data Eng. 36 (7) (2024) 3281–3296.
[32] W. He, G. Sun, J. Lu, X.S. Fang, Candidate–aware graph contrastive learning for recommendation, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 1670–1679.
[33] L. Xia, C. Huang, Y. Xu, J. Zhao, D. Yin, J. Huang, Hypergraph contrastive collaborative filtering, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 70–79.
[34] G. Zhu, W. Lu, C. Yuan, Y. Huang, Adamcl: Adaptive fusion multi-view contrastive learning for collaborative filtering, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 1076–1085.
[35] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, Y. Kalantidis, Decoupling representation and classifier for long-tailed recognition, in: 8th International Conference on Learning Representations, 2020, pp. 26–30.
[36] Y. Zhu, R. Xie, F. Zhuang, K. Ge, Y. Sun, X. Zhang, J. Cao, Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 1167–1176.
[37] Z. Zhu, J. Kim, T. Nguyen, A. Fenton, J. Caverlee, Fairness among new items in cold start recommender systems, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 767–776.
[38] Z. Huai, Y. Yang, M. Zhang, Z. Zhang, Y. Li, W. Wu, M2GNN: Metapath and multi-interest aggregated graph neural network for tag-based cross-domain recommendation, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 1468–1477.
[39] Y. Sun, J. Han, X. Yan, P.S. Yu, T. Wu, PathSim: Meta path-based top-k similarity search in heterogeneous information networks, Proc. VLDB Endow. 4 (11) (2011) 992–1003.
[40] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, R. Luo, Heterogeneous graph contrastive learning for recommendation, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 544–552.
[41] R. He, J. McAuley, VBPR: Visual Bayesian personalized ranking from implicit feedback, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 144–150.
[42] Q. Liu, S. Wu, L. Wang, Deepstyle: Learning user preferences for visual recommendation, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2017, pp. 841–844.
[43] W. Wang, D. Tran, M. Feiszli, What makes training multi-modal classification networks hard? in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12695–12705.
[44] Y. Huang, J. Lin, C. Zhou, H. Yang, L. Huang, Modality competition: What makes joint training of multi-modal network fail in deep learning?(provably), in: International Conference on Machine Learning, 2022, pp. 9226–9259.
[45] X. Peng, Y. Wei, A. Deng, D. Wang, D. Hu, Balanced multimodal learning via on-the-fly gradient modulation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8238–8247.
[46] M. Chen, Z. Wei, Z. Huang, B. Ding, Y. Li, Simple and deep graph convolutional networks, in: International Conference on Machine Learning, 2020, pp. 1725–1735.
[47] Y. Rong, W. Huang, T. Xu, J. Huang, Dropedge: Towards deep graph convolutional networks on node classification, in: Proceedings of the 8th International Conference on Learning Representations, 2020.
[48] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, 2019, arXiv Preprint arXiv:1908.10084.
[49] J. McAuley, C. Targett, Q. Shi, A.V.D. Hengel, Image-based recommendations on styles and substitutes, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 43–52.
[50] Y. Chen, Z. Liu, J. Li, J. McAuley, C. Xiong, Intent contrastive learning for sequential recommendation, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 2172–2182.
[51] Y. Xie, C. Yu, X. Jin, L. Cheng, B. Hu, Z. Li, Heterogeneous graph contrastive learning for cold start cross-domain recommendation, Knowl.-Based Syst. 299 (2024) 112054.

Chenyun Yu is an Assistant Professor at School of Intelligent Systems Engineering in Shenzhen Campus of Sun Yat-sen University. She received her Ph.D. degree from the City University of Hong Kong and served as a research fellow at the National University of Singapore from 2018 to 2020. Her research interests include large-scale data mining, recommendation systems, and cross-modal retrieval.



Junfeng Zhao received the B.E. degree in electronic engineering from Sun Yat-sen University in 2022. He is currently working toward the M.S. degree in electronic information at Sun Yat-sen University. His research interests include graph neural networks and recommendation systems.



Xuan Wu is currently a Master's student at Sun Yat-sen University, conducting academic research in the field of recommender algorithms. He also possesses practical experience in industrial applications of community-based multimedia content recommendation systems.

Yingle Luo received his Bachelor's Degree in Intelligent Science and Technology from Sun Yat-sen University. He is currently a graduate student at University of Sydney. His research interests focus on recommendation systems, graph neural networks and machine learning.

Yan Xiao is an Associate Professor at School of Cyber Science and Technology in Shenzhen Campus of Sun Yat-sen University. She received her Ph.D. degree from the City University of Hong Kong and held a research fellow position at the National University of Singapore. Her research focuses on the trustworthiness of deep learning systems and AI applications in software engineering.