Mind Individual Information! Principal Graph Learning for Multimedia Recommendation

Penghang Yu[1], **Zhiyi Tan**[1], **Guanming Lu**[1, 2], **Bing-Kun Bao**[3*]
[1]School of Communications and Information Engineering,
Nanjing University of Posts and Telecommunications, Nanjing, China
[2]Jiangsu Key Laboratory of Intelligent Information Processing and Communication Technology, Nanjing, China
[3]School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China
2022010201@njupt.edu.cn, tzy@njupt.edu.cn, lugm@njupt.edu.cn, bingkunbao@njupt.edu.cn

Abstract

Graph Neural Network (GNN)-based methods have recently emerged as effective approaches for multimedia recommendation. Typically, these methods employ message passing on the user-item interaction graph, and model user preferences by exploiting co-occurrence patterns. Despite their effectiveness, we argue that they insufficiently exploit the individual information, potentially limiting recommendation performance. To validate our argument, we first analyze existing methods from spectral graph theory. We identify that existing methods focus on capturing global structural features, but underutilize local structural features that convey individual information. Further detailed experiments reveal that such an underutilization leads to overly similar user preferences modeling. Furthermore, we propose a novel **P**rincipal **G**raph **L**earning (**PGL**) framework to address this issue. The idea is to enhance user preference modeling by effectively mining and utilizing principal local structural features. PGL first extracts the principal subgraph from the user-item interaction graph using two novel extraction operators: global-aware and local-aware subgraph extraction. It then employs message passing on the principal subgraph to comprehensively model user perference, with the aim of simultaneously capturing co-occurrence patterns and individual information. Compared to existing methods, PGL achieves an average performance improvement of 9%.

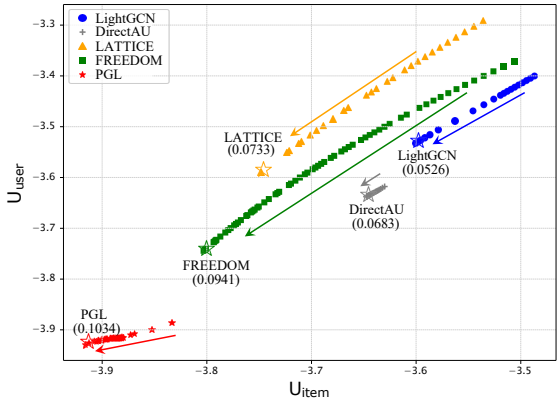

Figure 1: $U_{user} - U_{item}$ demonstrates the uniformity of the representations during the training phase. The stars denote the converged points. Lower Uniformity indicates a more uniform distribution and higher distinguishability. We also include the Recall@20 for each model in parentheses, where higher values indicate better performance.

Introduction

Recommender systems have been widely adopted in various domains. They analyze user historical behaviors to provide personalized item recommendations (Zhou et al. 2023a; Ding et al. 2023; Marinó et al. 2023). Yet, the behavioral data sparsity issue hinders the recommendation performance (Wu et al. 2021). To address this limitation, researchers begin to explore incorporating rich content information to enhance user preference modeling. Consequently, multimedia recommendation methods start to emerge (Chen et al. 2017).

Early multimedia recommendation methods (Shen et al. 2013; Zhang et al. 2016; He and McAuley 2016) primarily rely on matrix factorization (Koren, Bell, and Volinsky 2009). These studies combine content features with behavioral features obtained from matrix factorization, aiming to effectively model user preference. Recognizing that behavioral data can be naturally represented as a bipartite graph, recent researches have focused on leveraging Graph Neural Networks (GNNs) for more effective user preference modeling (Wu et al. 2022; Gao et al. 2023). For instance, (He et al. 2020) propose a linear message passing mechanism (LightGCN) to capture high-order connections between users and items. Due to its simplicity and effectiveness, this mechanism has been widely adopted in subsequent GNN-based methods (Zhang et al. 2021; Sharma et al. 2024). Considering that latent item-item connections may also be informative, (Zhang et al. 2021) introduce the LATTICE framework. LATTICE simultaneously aggregates information on both user-item graph and item-item graph, further boosting recommendation performance. This framework has inspired a series of effective methods such as MICRO (Zhang et al. 2022), FREEDOM (Zhou and Shen 2023), and MGCN (Yu et al. 2023c). Despite their effectiveness, existing methods model user preference heavily based on complex connection relationships (i.e., co-occurrence patterns). We argue that

*Corresponding author.

they underutilize individual information, which may limit the recommendation performance.

To validate our argument, we first analyze existing methods (Zhang et al. 2021; Zhou and Shen 2023; Yu et al. 2023c) from the spectral graph theory. According to spectral graph theory, low-frequency signals are normally associated with global structural features that reflect co-occurrence patterns, while high-frequency signals correspond to local structural features that convey individual information (Nica 2018; Guo et al. 2024a). Existing methods usually employ low-pass message passing on the complete use-item interaction graph, which capture global structural features and exploit high-order co-occurrence patterns. However, they lack capturing local structural features, resulting in decreased distinguishability of user preferences. For a more intuitive explanation, we use the Uniformity property (Wang et al. 2022) as the evaluation metric (Figure 1), and visualize user and item representations obtained from existing recommendation methods (Figures 2 and 3). The results indicate that the discriminability of user preferences obtained by existing methods is limited, and the representations of user preferences are over-similar. Thus, existing methods achieve suboptimal recommendation performance.

To address this issue, we propose the **P**rincipal **G**raph **L**earning (**PGL**) framework for multimedia recommendation. The idea is to enhance user preference modeling by preserving and utilizing more individual information, through mining local structural features. PGL first extracts the principal subgraph from the user-item interaction graph using two novel extraction operators: global-aware extraction and local-aware extraction. It then models user preference by mining the structural features on the principal subgraph, with the aim of simultaneously capturing co-occurrence patterns and individual information. Additionally, PGL introduces a feature discriminating auxiliary self-supervised task to further enhance the discriminability of learned representations. Extensive experiments on three public real-world datasets demonstrate that the proposed framework outperforms state-of-the-art methods in terms of recommendation accuracy.

Our main contributions can be summarized as follows:

• We identify that existing methods insufficiently capture individual information, and demonstrate that this leads to over-similar user preference modeling.

• We propose a novel principal graph learning framework for multimedia recommendation, which efficiently mines individual information.

• We conduct a comprehensive experimental study on three benchmark datasets, showing that PGL has distinct advantages in terms of recommendation accuracy.

Related Work

CF-based Multimedia Recommendation

Collaborative Filtering (CF) has emerged as a prominent recommendation method that leverages behavior similarity to make top-k recommendations (Su and Khoshgoftaar 2009). However, due to the sparsity issue of behavioral data, recommender systems struggle to accurately model user preferences. Therefore, researchers have begun incorporating item content information to enhance the recommendation performance (Shen et al. 2013; Deldjoo et al. 2018). Typically, these approaches first extract multimodal content features using pre-trained neural networks, and then fuse these features with behavioral features to better model user preferences. For instance, VBPR (He and McAuley 2016) and VPOI (Wang et al. 2017) use convolution neural networks (CNN) to extract visual features, and combine the visual features with item behavior features. Beyond using a single modality, some methods incorporate multimodal features into item behavior features (Kaššák, Kompan, and Bieliková 2016; Xu et al. 2021). For example, MDCF (Xu et al. 2021) maps multimodal features to a consensus Hamming space for cold-start recommendation. However, these methods can only explore the shallow connections between users and items, and lack the ability to exploit the high-order connections. This limits the recommendation performance.

GNN-based Multimedia Recommendation

As user behavior data can be naturally represented as a bipartite graph, recent researchers have favored Graph Neural Network (GNN) as a powerful tool to capture high-order connections between users and items (Yu et al. 2023b; Wang et al. 2022; Yu et al. 2023a). Specifically, NGCF (Wang et al. 2019) captures co-occurrence patterns by iteratively performing neighbor aggregation on the user-item graph. Based on NGCF, LightGCN (He et al. 2020) simplifies the classical message passing mechanism. It designs a parameter-free linear mechanism, which is more suitable for recommendation scenarios. Due to its simplicity, this message passing mechanism has been widely adopted in subsequent GNN-based methods (Wang et al. 2021; Mao et al. 2021). Recognizing that latent item-item connections may also be informative, (Zhang et al. 2021) introduce the LATTICE framework, which simultaneously aggregates information from both the user-item graph and the item-item graph. Given its effectiveness, LATTICE has become the most widely used framework for multimedia recommendation. Building upon this framework, FREEDOM (Zhou and Shen 2023) freezes the learning of the item-item graph, thereby promoting more robust results. Furthermore, MGCN (Yu et al. 2023c) adds a modality noise filter to remove preference-irrelevant noise, achieving state-of-the-art recommendation performance. Despite their effectiveness, they typically model user preference by complex connection relationships, focusing on mining global structural features. Although LGMRec (Guo et al. 2024b) designs hypergraph networks to simultaneously mine global and local structural features, it still over-emphasizes co-occurrence patterns. This neglect of individual information leads to a lack of differentiation among similar users, impeding comprehensive user preference modeling.

Preliminary

In this section, we first formulate the multimedia recommendation problem, and then summarize the common paradigm of multimedia recommendation models.

Notations and Terminology

Let $\mathcal{U}$ denote the set of users and $\mathcal{I}$ denote the set of items. $\mathcal{M}$ represent the set of modality information, which encompasses behavioral (ID), visual, and textual data.

User historical behavior data is represented by a sparse binary matrix $\mathbf{R} \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{I}|}$, where $r_{u,i} = 1$ if user u has interacted with item i, and $r_{u,i} = 0$ otherwise. Naturally, this historical behavior data can be represented as a sparse bipartite graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{\mathcal{U} \cup \mathcal{I}\}$ represents the set of nodes, and $\mathcal{E} = \{(u,i) \mid u \in \mathcal{U}, i \in \mathcal{I}, r_{u,i} = 1\}$ represents the set of edges. The adjacency matrix $\mathbf{A}$ of the user-item interaction graph $\mathcal{G}$ is given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^\intercal & \mathbf{0} \end{bmatrix}. \tag{1}$$

In this bipartite graph, we denote the neighbors of node k as $\mathcal{N}_k$, and its cardinality as $N_k = |\mathcal{N}_k|$.

We denote the all one column vector of any dimension as $\mathbf{1}$, and degree matrices as $\mathbf{D}_U = \text{Diag}(\mathbf{R} \cdot \mathbf{1})$ and $\mathbf{D}_I = \text{Diag}(\mathbf{1}^\intercal \cdot \mathbf{R})$. The normalized behavioral matrix $\tilde{\mathbf{R}}$ is denoted as

$$\tilde{\mathbf{R}} = \mathbf{D}_U^{-\frac{1}{2}} \mathbf{R} \mathbf{D}_I^{-\frac{1}{2}}, \tag{2}$$

with $\tilde{\mathbf{r}}_u$ as the u-th row of $\tilde{\mathbf{R}}$. Similarly, the normalized user-item adjacency matrix $\tilde{\mathbf{A}}$ is given by

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \tilde{\mathbf{R}} \\ \tilde{\mathbf{R}}^\intercal & \mathbf{0} \end{bmatrix}. \tag{3}$$

Definition 1. (Graph Frequency) $\tilde{\mathbf{A}}$ is a real, symmetric matrix. Its eigendecomposition is defined as follows:

$$\begin{aligned} \tilde{\mathbf{A}} &= \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\intercal, \\ \mathbf{\Lambda} &= \text{Diag}(\lambda_1, \cdots, \lambda_n), \end{aligned} \tag{4}$$

where $\mathbf{\Lambda} = \text{Diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the eigenvalue matrix, and $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. The columns of $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n]$ are the corresponding orthonormal eigenvectors, with $\mathbf{u}_i \in \mathbb{R}^n$ being the eigenvector associated with eigenvalue λ_i. In spectral graph theory (Nica 2018; Guo et al. 2024a; Ramakrishna, Wai, and Scaglione 2020), given a threshold eigenvalue λ_i, the low-frequency components ($\lambda_j < \lambda_i$) of the eigenvalue spectrum are usually associated with the global structural features that reflect co-occurrence patterns, while the high-frequency components relate to the local structural features that convey individual information.

Definition 2. (Message Passing) Message passing on a graph can capture high-order connections (He et al. 2020). From the spatial domain perspective, message passing of the input x is represented as:

$$\tilde{x} = H(\tilde{\mathbf{A}})x, \tag{5}$$

where $H(\cdot)$ is a filter. From the frequency domain perspective, it is expressed as:

$$\tilde{x} = \mathbf{U}\text{Diag}(h(\lambda_1), \cdots, h(\lambda_n))\mathbf{U}^\intercal x. \tag{6}$$

Here, the Graph Fourier Transform (GFT) of a signal $\mathbf{x}$ is defined as $\hat{\mathbf{x}} = \mathbf{U}^\intercal\mathbf{x}$, where the eigenvector matrix $\mathbf{U}$ serves as the GFT basis. Similar to the Fourier transform, GFT is a linear orthogonal transform and its inverse transform is given by $\mathbf{x} = \mathbf{U}\hat{\mathbf{x}}$. The function $h(\cdot)$ represents the corresponding filter $\mathbf{H}(\cdot)$ in the frequency domain.

Analysis of LATTICE

LATTICE (Zhang et al. 2021) is a state-of-the-art GNN-based framework for multimedia recommendation. It serves as the foundation for most multimedia recommendation methods (Zhang et al. 2022; Zhou and Shen 2023; Yu et al. 2023c). It performs message passing on the user-item interaction graph $\mathcal{G}$ to capture high-order co-occurrence patterns:

$$\tilde{\mathbf{E}}_b = H(\tilde{\mathbf{A}})\mathbf{E}_b, \tag{7}$$

where $\tilde{\mathbf{E}}_b$ represents the matrix of user and item representations, typically is the ID embeddings of users and items.

Meanwhile, LATTICE utilizes the similarity of item modality information to construct an item-item graph $\mathcal{S}_m$. Similar to $\tilde{\mathbf{A}}$, a normalized item-item adjacency matrix $\tilde{\mathbf{S}}_m$ can be obtained. Message passing on item-item graph $\mathcal{S}_m$ captures latent item-item connections:

$$\tilde{\mathbf{E}}_m^{(I)} = H(\tilde{\mathbf{S}})\mathbf{E}_b^{(I)}, \tag{8}$$

where $\mathbf{E}_b^{(I)}$ refers to the item ID embeddings. By aggregating the item ID representations within the item-item graph, the item multimodal embedding $\tilde{\mathbf{E}}_m^{(I)}$ is obtained. The final user and item representations are obtained by summing $\tilde{\mathbf{E}}_b$ and $\tilde{\mathbf{E}}_m^{(I)}$. (In LATTICE, user multimodal embeddings are represented as a zero matrix.)

Like most GNN-based recommendation methods (Mao et al. 2021; Wang et al. 2021), LATTICE adopts the message passing mechanism in LightGCN. Thus, the $H(\cdot)$ and $h(\cdot)$ is represented as:

$$\begin{aligned} H(\tilde{\mathbf{A}}) &= \alpha_0 + \alpha_1\tilde{\mathbf{A}} + \cdots + \alpha_K\tilde{\mathbf{A}}^K, \\ h(\lambda_i) &= \sum_{k=0}^{K-1} \alpha_k(1-\lambda_i)^k, \end{aligned} \tag{9}$$

where α_k is a predefined constant, K is the number of GNN layers used for message passing. Equation 9 indicates that the message passing mechanism captures low-frequency signals while suppressing high-frequency signals. Combined with Definition 1, LATTICE models user preferences by exploiting the low-frequency signals, which reflect co-occurrence patterns on the complete user-item interaction graph. (A detailed explanation is provided in the Appendix.) However, it is insufficient in fully capturing the high-frequency related to principal local structural features. It potentially results in the loss of individual information and a decrease in recommendation performance.

Visualization of representations

For a more intuitive explanation, we use the Uniformity property (Wang et al. 2022) as the evaluation metric (Figure 1), and visualize user and item representations obtained from existing recommendation methods (Figures 2 and 3). Lower Uniformity indicates a more uniform distribution and higher distinguishability. Specifically, we randomly sample 500 user representations and 500 item representations from the Amazon-Clothing dataset and map them to the unit hypersphere $\mathcal{S}^1$ (i.e., a circle with a radius of 1) by using t-SNE

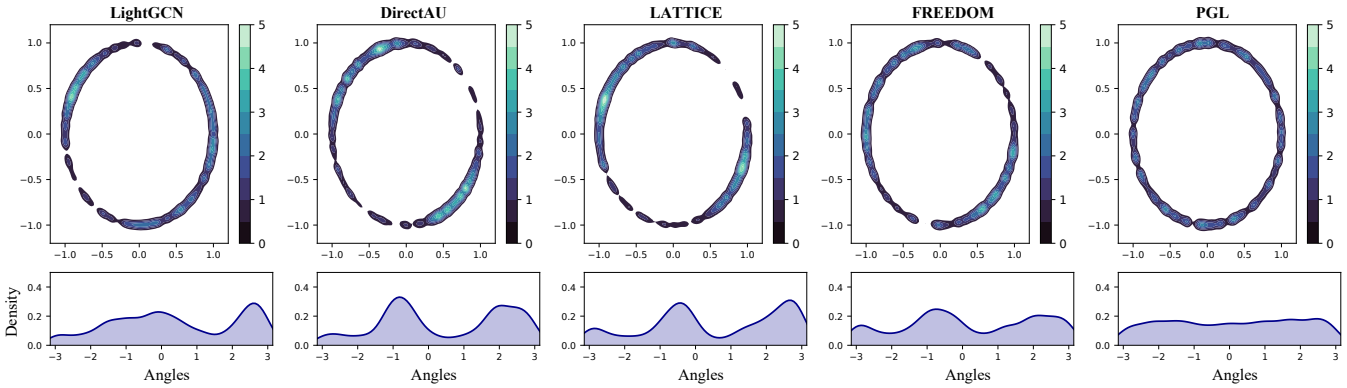

Figure 2: Distribution of user representations learned from the Amazon Clothing dataset. The upper half illustrates feature distributions on $\mathcal{S}^1$, while the lower half presents density estimates for different angles.

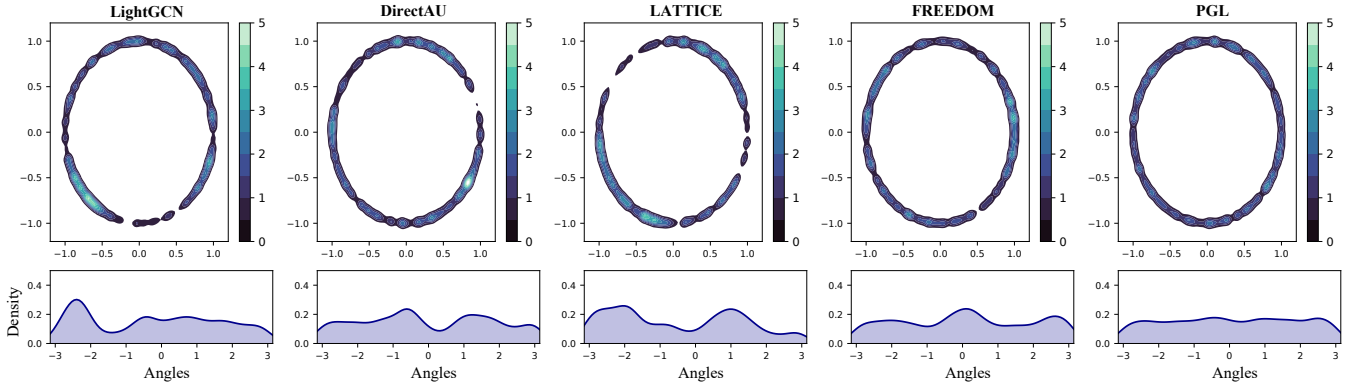

Figure 3: Distribution of item representations learned from the Amazon Clothing dataset. The upper half illustrates feature distributions on $\mathcal{S}^1$, while the lower half presents density estimates for different angles.

(Van der Maaten and Hinton 2008). Subsequently, we employ Gaussian kernel density estimation (Terrell and Scott 1992) to plot probability density distribution.

As shown in Figure 1, existing models obtain representations with limited uniformity. This leads to imprecise user preference modeling. Figures 2 and 3 more clearly demonstrate the non-uniformity of the features, where certain user/item representations are overly similar and tightly clustered together. It is shown as "bright spots" on the distribution map, and as a steep distribution on the probability density map. Though multimedia recommendation methods achieve higher uniformity than unimodal approaches by incorporating additional content information to enhance individual differences, the obtained representations still exhibit limited discriminability. Because these methods all fail to sufficiently mining the principal local structural features, resulting in suboptimal recommendation performance.

PGL

To better capture local structural features, we propose the principal graph learning framework for multimedia recommendation. It first extracts the principal subgraph from the complete user-item interaction graph, and then employs message passing on the principal subgraph. The framework of PGL is illustrated in Figure 4.

Embedding

As with many works (Zhou and Shen 2023; Yu et al. 2023c), we map the user unique hot ID representations to the user ID embeddings as the user raw representations. For the items, we map the content features using MLP to a low dimensional representation and concatenate them to obtain the item raw representations. Because the item content features are more informative than the item ID representations, it is more conducive to the model convergence.

Principal Graph Learning

To better mine local structural features, we propose the principal user-item graph learning. Specifically, during model training, we first extract the principal subgraph from the complete graph structure. We then perform message passing on this principal subgraph. This approach not only mines richer individual information, but also captures co-occurrence patterns through multi-hop message passing. During model inference, we employ the message passing mechanism on the original complete graph instead, avoiding the problem of insufficient global information utilization caused by subgraph fragmentation.

Formally, the message passing of x is represented as:

$$\tilde{x} = H(F(\tilde{\mathbf{A}}))x. \tag{10}$$

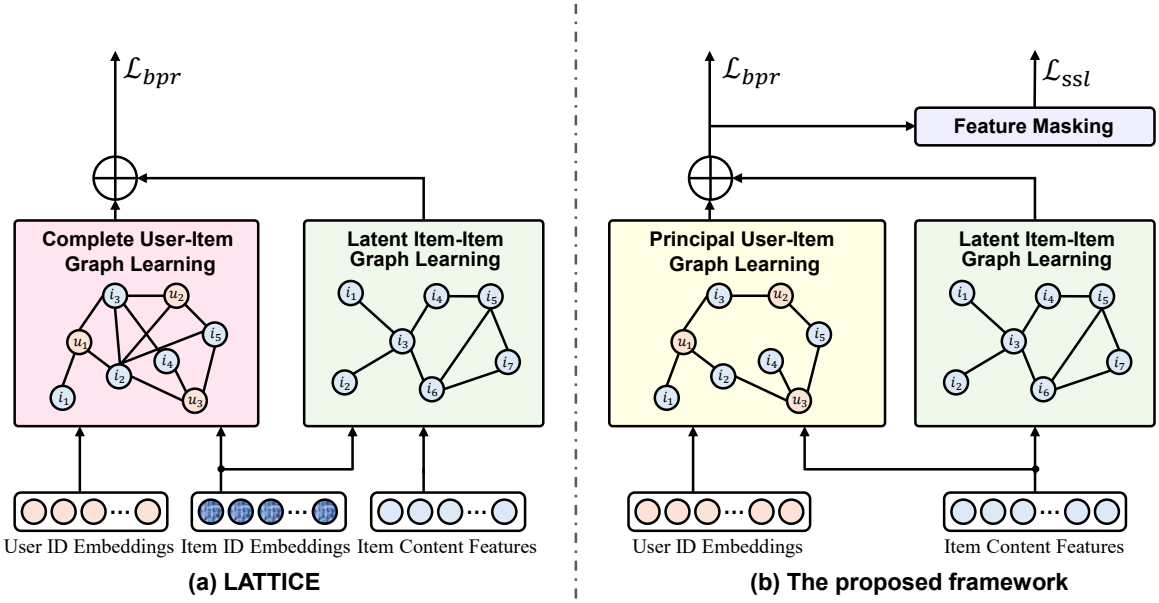

Figure 4: Comparison of (b) the proposed framework PGL and (a) exsiting framework LATTICE.

For subgraph extraction $F(\cdot)$, we design two operators: global-aware and local-aware subgraph extraction. The idea is to extract the most informative subgraph from the global/local perspective.

• **Global-Aware Extraction** Truncated reconstruction $\tilde{\mathbf{A}}$ identifies the principal subgraphs with the highest information content, guided by the global structural signal. Specifically, we first decompose $\tilde{\mathbf{A}}$ using Singular Value Decomposition (SVD) (Rangarajan 2001). However, performing the exact SVD on $\tilde{\mathbf{A}}$ is computationally expensive, making it impractical in real-world scenarios. Instead, we adopt the randomized SVD algorithm proposed by (Halko, Martinsson, and Tropp 2011). It approximates the range of the input matrix with a low-rank orthonormal matrix, and performs SVD on this smaller matrix. It is formally represented as:

$$\dot{\mathbf{U}}, \dot{\mathbf{\Lambda}}, \dot{\mathbf{U}}^\intercal = \text{ApproxSVD}(\tilde{\mathbf{A}}, d), \tag{11}$$

where $\dot{\mathbf{\Lambda}}$ is eigenvalues matrix, $\dot{\mathbf{U}}$ is corresponding eigenvector matrix, and d is the rank of decomposed matrix. It is equal to the dimension of user/item raw representations.

Next, we employ the truncated reconstruction to obtain principal subgraph $\dot{\mathbf{A}}$:

$$\begin{aligned} \ddot{\mathbf{U}}, \ddot{\mathbf{\Lambda}}, \ddot{\mathbf{U}}^\intercal &= \text{Truncation}(\dot{\mathbf{U}}, \dot{\mathbf{\Lambda}}, \dot{\mathbf{U}}^\intercal, \gamma), \\ \dot{\mathbf{A}} &= \ddot{\mathbf{U}}\ddot{\mathbf{\Lambda}}\ddot{\mathbf{U}}^\intercal, \\ \ddot{\mathbf{A}} &= \text{Sparsification}(\dot{\mathbf{A}}, \epsilon), \end{aligned} \tag{12}$$

where γ is the truncation ratio within the range $(0, 1)$. The truncation function retains the eigenvalues belonging to the range before or after γd, resulting in $\ddot{\mathbf{\Lambda}} = \text{Diag}(\lambda_1, \cdots, \lambda_{\gamma d}, 0, \cdots, 0, \lambda_{(1-\gamma)d}, \cdots, \lambda_d)$. The sparsification function sets elements with reconstruction values less than the threshold ϵ (typically $1e^{-3}$) to 0, ensuring the obtained principal graph $\dot{\mathbf{A}}$ remains sparse.

• **Local-Aware Extraction** The expressive principal subgraph can also be extracted from the complete graph by leveraging local exposure information. The extraction process is formalized as:

$$\ddot{\mathbf{A}} = \text{Extraction}(\tilde{\mathbf{A}}, p). \tag{13}$$

Here, the extraction function is a multinomial sampling procedure, where p denotes the percentage of original edges. Experiments finds that most of the time, $p = 0.3$ achieves an optimal result. The probability of sampling an edge connecting user u and item i is $1/\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_j|}$. This weighting scheme is motivated on the intuition that edges connected to nodes with lower exposure tend to be more reliable and informative (Wu et al. 2021).

Latent Graph Learning

Following (Zhang et al. 2021; Zhou and Shen 2023; Yu et al. 2023c), we also conduct message passing on the latent item-items to capture collaborative signals under modality information. Specifically, we first pre-construct latent item-item graphs based on similarity of item raw modality features $e_{i,m}$. It avoids the learning of latent graph structures in the classical LATTICE. Then, we use kNN sparsization (k=10) and Laplacian regularization to obtain the normalized matrix $\mathbf{S}_m$. Besides, each modality item-item matrices are weighted added to obtain the final item-item similarity matrix $\mathbf{S}$. Here, we still use the message passing mechanism of Equation 8.

Prediction

To make recommendations for user u, we predict the interaction scores between the user u and all candidate items, and choose K top-ranked items as recommendations to the user u. The interaction score is calculated as:

$$\begin{aligned} f_{\text{predict}}(u, i) &= \hat{\mathbf{e}}_u^\intercal \hat{\mathbf{e}}_i, \\ \hat{\mathbf{e}}_u = \mathbf{e}_u^{\text{principal}} + \mathbf{e}_u^{\text{latent}}, &\quad \hat{\mathbf{e}}_i = \mathbf{e}_i^{\text{principal}} + \mathbf{e}_i^{\text{latent}}, \end{aligned} \tag{14}$$

where $\mathbf{e}_i^{\text{principal}}$ and $\mathbf{e}_i^{\text{latent}}$ represent the output of principal graph learning and latent graph learning correspondingly. A

Datasets	Baby				Sports				Clothing			
Methods	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**
MF-BPR (UAI'09)	0.0357	0.0575	0.0192	0.0249	0.0432	0.0653	0.0241	0.0298	0.0187	0.0279	0.0103	0.0126
LightGCN (SIGIR'20)	0.0479	0.0754	0.0257	0.0328	0.0569	0.0864	0.0311	0.0387	0.0340	0.0526	0.0188	0.0236
DirectAU (KDD'22)	0.0460	0.0672	0.0263	0.0318	0.0630	0.0958	0.0351	0.0436	0.0468	0.0683	0.0257	0.0311
LayerGCN (ICDE'23)	0.0529	0.0820	0.0281	0.0355	0.0594	0.0916	0.0323	0.0406	0.0371	0.0566	0.0200	0.0247
VBPR (AAAI'16)	0.0423	0.0663	0.0223	0.0284	0.0558	0.0856	0.0307	0.0384	0.0281	0.0415	0.0158	0.0192
MMGCN (MM'19)	0.0378	0.0615	0.0200	0.0261	0.0370	0.0605	0.0193	0.0254	0.0218	0.0345	0.0110	0.0142
DualGNN (TMM'21)	0.0448	0.0716	0.0240	0.0309	0.0568	0.0859	0.0310	0.0385	0.0454	0.0683	0.0241	0.0299
LATTICE (MM'21)	0.0547	0.0850	0.0292	0.0370	0.0620	0.0953	0.0335	0.0421	0.0492	0.0733	0.0268	0.0330
SLMRec (TMM'22)	0.0529	0.0775	0.0290	0.0353	0.0663	0.0990	0.0365	0.0450	0.0452	0.0675	0.0247	0.0303
MICRO (TKDE'22)	0.0584	0.0929	0.0318	0.0407	0.0679	0.1050	0.0367	0.0463	0.0521	0.0772	0.0283	0.0347
BM3 (WWW'23)	0.0564	0.0883	0.0301	0.0383	0.0656	0.0980	0.0355	0.0438	0.0422	0.0621	0.0231	0.0281
MMSSL (WWW'23)	0.0613	0.0971	0.0326	0.0420	0.0673	0.1013	0.0380	0.0474	0.0531	0.0797	0.0291	0.0359
FREEDOM (MM'23)	0.0627	0.0992	0.0330	0.0424	0.0717	0.1089	0.0385	0.0481	0.0629	0.0941	0.0341	0.0420
MGCN (MM'23)	0.0620	0.0964	0.0339	0.0427	0.0729	0.1106	0.0397	0.0496	0.0641	0.0945	0.0347	0.0428
LGMRec (AAAI'24)	0.0644	0.1002	0.0349	0.0440	0.0720	0.1068	0.0390	0.0480	0.0555	0.0828	0.0302	0.0371
PGL w/ global (Ours)	0.0663	**0.1040**	0.0351	0.0448	0.0760	0.1144	0.0410	0.0509	0.0690	0.1014	0.0369	0.0451
PGL w/ local (Ours)	**0.0676**	0.1022	**0.0360**	**0.0449**	**0.0789**	**0.1174**	**0.0428**	**0.0528**	**0.0712**	**0.1034**	**0.0385**	**0.0467**

Table 1: Performance comparison of baselines and our method in terms of Recall@K (R@K), and NDCG@K (N@K). The best result is in boldface and the second best is underlined.

high score suggests that the user u is more likely to click the item i.

Optimization

During the phase of model training, we adopt the Bayesian Personalized Ranking (BPR) loss $\mathcal{L}_{\text{BPR}}$ as the basic optimization task, which assumes that users prefer historically interacted items over unclicked ones. And it is combined with auxiliary self-supervised tasks to jointly update the representations of users and items:

$$\mathcal{L} = \mathcal{L}_{\text{BPR}} + \lambda_{\text{SSL}}\mathcal{L}_{\text{SSL}}, \tag{15}$$

where λ_{SSL} is the hyperparameter to control the intensity of the self-supervised auxiliary task.

The self-supervised task is used to further enhance the distinguishability of representations. Specifically, it first masks certain node representations through feature masking, and then maximizes the consistency of features after two random masks through in batch InfoNCE:

$$\begin{aligned}
\hat{\mathbf{e}}'_u, \hat{\mathbf{e}}''_u &= \text{Mask}(\hat{\mathbf{e}}_u, \rho), \\
\hat{\mathbf{e}}'_i, \hat{\mathbf{e}}''_i &= \text{Mask}(\hat{\mathbf{e}}_i, \rho), \\
\mathcal{L}_{\text{SSL}} = \sum_{u\in\mathcal{U}} -\log &\frac{(\hat{\mathbf{e}}'_u \cdot \hat{\mathbf{e}}''_u/\tau)}{\sum_{v\in\mathcal{U}} \exp(\hat{\mathbf{e}}'_v \cdot \hat{\mathbf{e}}'_v/\tau)} \\
+ \sum_{i\in\mathcal{I}} -\log &\frac{\exp(\hat{\mathbf{e}}'_i \cdot \hat{\mathbf{e}}''_i/\tau)}{\sum_{j\in\mathcal{I}} \exp(\hat{\mathbf{e}}'_j \cdot \hat{\mathbf{e}}''_i/\tau)},
\end{aligned} \tag{16}$$

where ρ controls the feature masking ratio and τ is the temperature hyper-parameter of the softmax function.

Experiments

Datasets Following (Zhang et al. 2021), we conduct experiments on three categories of the widely used Amazon dataset: (a)Baby, (b) Sports and Outdoors, and (c) Clothing, Shoes, and Jewelry, which we refer to as Baby, Sports, and Clothing in brief. The statistics of these datasets are presented in the Appendix. We use the pre-extracted 4,096-dimensional visual features and 384-dimensional text features, which have been published in (Zhou 2023).

Compared Methods We compare PGL with several representative recommendation models. These recommendation models fall into two groups: General models only utilize user behavioral data for recommendation: MF-BPR (Koren, Bell, and Volinsky 2009), LightGCN (He et al. 2020), DirectAU (Wang et al. 2022), LayerGCN (Zhou et al. 2023b); Multimedia models utilize both user behavioral data and item content information for the recommendation: VBPR (He and McAuley 2016), MMGCN (Wei et al. 2019), DualGNN (Wang et al. 2021), LATTICE (Zhang et al. 2021), SLMRec (Tao et al. 2022), MICRO (Zhang et al. 2022), BM3 (Zhou et al. 2023c), MMSSL (Wei et al. 2023), FREEDOM (Zhou and Shen 2023), MGCN (Yu et al. 2023c), LGMRec (Guo et al. 2024b).

Evaluation Protocols To ensure a a fair comparison, we adhere to the standardized all-ranking protocol (Yu et al. 2023c). We calculate and present the average metrics, namely Recall@K (R@K) and NDCG@K (N@K). Our study encompasses over ten experiments, and the reported values represent their average outcomes. To determine the statistical significance of the improvements over the strongest baseline, we perform significance testing (Hsu and Lachenbruch 2014). The results indicate that our proposed approach outperforms the best baseline significantly (p-value < 0.05).

Dataset	Metric	VBPR	MMGCN	SLMRec	LATTICE	FREEDOM	MGCN	LGMRec	PGL
Baby	**Memory (GB)**	1.89	2.69	2.08	4.53	2.13	2.05	1.96	2.17
	Time (s/epoch)	0.83	4.78	2.69	4.12	1.54	2.68	3.01	2.30
Sports	**Memory (GB)**	2.71	3.91	3.04	19.93	3.34	3.32	3.18	3.41
	Time (s/epoch)	3.84	48.92	19.60	36.59	9.08	11.72	10.75	9.11
Clothing	**Memory (GB)**	3.02	4.24	3.40	28.22	4.15	4.08	3.79	4.32
	Time (s/epoch)	2.82	37.66	11.98	34.26	7.21	9.63	10.68	8.72

Table 2: Efficiency Comparison of Different Recommendation Models

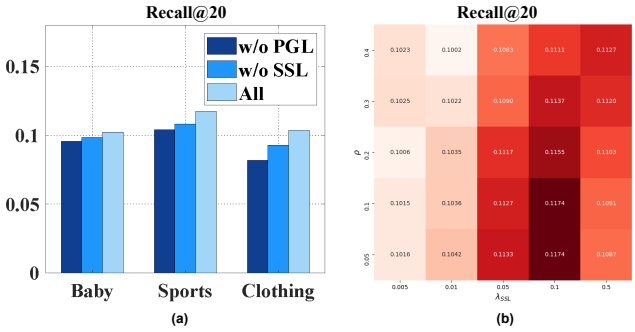

Figure 5: Performance Comparison between (a) different variants of PGL (b) different loss weights λ_{SSL} and masking ratios ρ.

Implementation Details Due to space limitations, implementation details are in the Appendix.

Overall Performance

• **Effectiveness:** According to Table 1, we have the following key observations: (1) **The superiority of PGL.** PGL outperforms all other recommendation methods. This suggests that capturing individual information can improve recommendation performance. By exploiting individual information, the model enhances the discriminability of node features. It allows to more accurately identify the specific items that users are genuinely interested in among multiple similar items. (2) **The flexibility of PGL.** PGL demonstrates effective recommendation performance, regardless of the specific principal subgraph extraction operator employed. Moreover, PGL with local-aware extraction performs better than PGL with global-aware extraction. This is because the local-aware extraction can not only mine individual information, but can also partially alleviate exposure bias. This indicates that PGL has high flexibility and scalability, and is expected to achieve even better performance through further refinement of the principal graph extraction operator.

• **Efficiency:** As depicted in Table 2, PGL has acceptable training time and cost. This is because PGL does not store separate item ID embeddings; rather, it maps content features to derive the item representations. Furthermore, the number of edges in the pre-extracted principal subgraph is substantially smaller than that of the complete user-item interaction graph, thereby lowering the computational cost of message passing. Overall, PGL has the potential to be applied in practical large-scale recommendation scenarios.

Ablation Study

To investigate the impact of the keys components in PGL, we conduct ablation studies on the following model variants: (1) *w/o PGL:* We remove the principal graph learning and use the original complete graph learning instead. (2) *w/o SSL:* We remove self-supervised auxiliary task. As shown in Figure 5(a), the results demonstrate that removing the principal graph learning significantly decreases the recommendation performance. This is because the principal graph learning effectively utilizes local structural features and captures individual information. Thus, it enables more accurate modeling of similar yet distinct user preferences by enhancing the discriminative power. Removing the self-supervised learning task also leads to a decline in performance. This indicates that the self-supervised learning task further boosts the discriminative power, ultimately improving the effectiveness of the recommendations.

Hyperparameter Analysis

In the self-supervised learning auxiliary task, there are two hyperparameters: ρ controls the feature masking ratio and λ_{SSL} controls the intensity of the self-supervised learning. We conduct a thorough hyperparameter analysis to understand their impact. As shown in Figure 5(b), we find that a setting of $\rho = 0.1$ and $\lambda_{\text{SSL}} = 0.1$ is suitable for Amazon-Sports dataset. However, the optimal hyperparameters vary across different datasets, indicating the necessity for careful tuning. Because excessive self-supervised learning may mislead user preference modeling, while too weak is insufficient to fully enhance the discriminability of representations.

Conclusion

In this paper, we analyze existing methods from spectral graph theory and point out their underutilization of individual information. Our experiments show that this results in overly similar user preference modeling, which ultimately limits recommendation performance. To address this issue, we propose a principal graph learning framework (PGL) with two novel pricipal graph extraction operators. PGL more effectively captures individual information by mining principal local structural features, thereby enabling comprehensive user preference modeling. In the future, we plan to design more effective operators to further improve the mining of local structural features.

Analysis of LATTICE

Combining Equation 9 and Definition 1, LATTICE models user preferences by exploiting the low-frequency signals, which reflect co-occurrence patterns on the complete user-item interaction graph. To further illustrate the effect of exploiting low-frequency signals, we take predicting whether user u will interact with item i as an example: [1]

$$\begin{aligned}\hat{y}_{ui} &= \Big(\mathbf{e}_{u,b} + \sum_{j\in\mathcal{N}_{u,b}} \beta_{u,j}\mathbf{e}_{j,b}\Big)^{\intercal} \\ &\Big(\mathbf{e}_{i,b} + \sum_{v\in\mathcal{N}_{i,b}} \beta_{v,i}\mathbf{e}_{v,b} + \sum_{m\in\mathcal{M}}\sum_{w\in\mathcal{N}_{i,m}} \beta_{w,i,m}\mathbf{e}_{w,b}\Big) \\ &= \mathbf{e}_{u,b}^{\intercal}\mathbf{e}_{i,b} + \sum_{j\in\mathcal{N}_{u,b}} \beta_{u,j}\mathbf{e}_{j,b}^{\intercal}\mathbf{e}_{i,b} \\ &+ \sum_{v\in\mathcal{N}_{i,b}} \beta_{u,v}\mathbf{e}_{u,b}^{\intercal}\mathbf{e}_{v,b} + \sum_{j\in\mathcal{N}_{u,b}}\sum_{v\in\mathcal{N}_{i,b}} \beta_{j,v}\mathbf{e}_{j,b}^{\intercal}\mathbf{e}_{v,b} \\ &+ \sum_{m\in\mathcal{M}}\sum_{w\in\mathcal{N}_{i,m}} \beta_{u,w,m}\mathbf{e}_{u,b}\mathbf{e}_{w,b} \\ &+ \sum_{m\in\mathcal{M}}\sum_{j\in\mathcal{N}_{u,b}}\sum_{w\in\mathcal{N}_{i,m}} \beta_{j,w,m}\mathbf{e}_{j,b}\mathbf{e}_{w,b},\end{aligned} \tag{17}$$

where $\mathbf{e}_{u,b}, \mathbf{e}_{v,b}, \mathbf{e}_{w,b}$ denote the user representations for users u, v, and w in the $\mathbf{E}_b$, respectively. Similarly, $\mathbf{e}_{i,b}$ and $\mathbf{e}_{j,b}$ denote the item representations for items i and j. The terms $\beta_{u,j}$, $\beta_{v,i}$, $\beta_{u,j}$, $\beta_{u,v}$, $\beta_{j,v}$ reflects the co-occurrence patterns mined from the user-item interaction graph, while $\beta_{w,i,m}$, $\beta_{u,w,m}$, $\beta_{j,w,m}$ reflect the co-occurrence patterns mined from the item-item modality similarity graph.

We observe that LATTICE exploits multiple different co-occurrence patterns, including user-item (u-i, u-w and j-v), item-item (i-j and w-j), and user-user (u-v) relationships. While this enables the capture of complex co-occurrence patterns, it also leads to excessive message transmission, weakening the representation of user-item (u-i) individual information. Additionally, the modeling of similar users based on analogous co-occurrence patterns (item-item or user-user relationships) may not fully account for nuanced behavioral differences. This limits the performance of current multimedia recommendation methods.

Therefore, it is essential to explore the principal local structures (high-frequency signals) that conveys similar but not completely identical user behavioral patterns. In other words, by performing message passing on the principal subgraph, the model can effectively avoid the issue of excessive message transmission while simultaneously mining co-occurrence patterns and individual information.

Datasets

We conduct experiments on three categories of the widely used Amazon dataset[2]: (a)Baby, (b) Sports and Outdoors, and (c) Clothing, Shoes, and Jewelry, which we refer to as Baby, Sports, and Clothing in brief. The statistics of these datasets are presented in Table.3.

Dataset	#User	#Item	#behavior	Density
Baby	19,445	7,050	160,792	0.117%
Sports	35,598	18,357	296,337	0.045%
Clothing	39,387	23,033	278,677	0.031%

Table 3: Statistics of the experimental datasets

Implementation Details

The proposed framework and all compared methods are implemented using the MMRec framework[3] (Zhou 2023), which is a unified open-source platform for developing and reproducing recommendation algorithms. To ensure a fair comparison, all methods are optimized using the Adam optimizer, and hyperparameters follow the settings reported in their original papers.

For general settings, the ID embeddings are initialized using the Xavier initialization method and set to a dimension of 64. The batch size is set to 2048. For the self-supervised task, the temperature parameter is set to 0.2, as this value is commonly considered a good choice. Regarding the parameters used in our method, the truncation ratio γ is set to 0.25, the sparsification threshold ϵ is set to 1e-3, and the sampling ratio p is set to 0.3. For the hyperparameters of the self-supervised learning task, a hyperparameter search is conducted. The feature masking ratio ρ is searched in the range of [0.05, 0.1, 0.2, 0.3, 0.4], and the weight of the self-supervised task λ_{SSL} is searched in the range of [0.005, 0.01, 0.05, 0.1, 0.5].

Furthermore, to avoid the overfitting issue, we employ an early stopping strategy. Following (Zhang et al. 2022), we use Recall@20 as the training-stopping indicator. If there is no improvement in Recall@20 for ten consecutive training epochs, the model training is halted. All experiments are performed using PyTorch on NVIDIA Tesla V100 GPUs.

Acknowledgments

This work was supported the National Nature Science Foundation of China under Grants (No.62325206, 72074038), the Key Research and Development Program of Jiangsu Province under Grant BE2023016-4, the Natural Science Foundation of Jiangsu Province (BK.20210595) and the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant (KYCX23_1026).

References

Chen, J.; Zhang, H.; He, X.; Nie, L.; Liu, W.; and Chua, T.-S. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 335–344.

[1] For the simplicity of the notation, we showcase our findings with a one-hop message passing mechanism. It should be noted that this conclusion can be naturally extended to various multimedia recommendation methods, as message passing mechanism are parameter-free linear transformations.

[2] Datasets are available at http://jmcauley.ucsd.edu/data/amazon/links.html

[3] https://github.com/enoche/MMRec

Deldjoo, Y.; Schedl, M.; Hidasi, B.; and Knees, P. 2018. Multimedia recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems*, 537–538.

Ding, Y.; Lai, Z.; Mok, P.; and Chua, T.-S. 2023. Computational technologies for fashion recommendation: A survey. *ACM Computing Surveys*, 56(5): 1–45.

Gao, C.; Zheng, Y.; Li, N.; Li, Y.; Qin, Y.; Piao, J.; Quan, Y.; Chang, J.; Jin, D.; He, X.; et al. 2023. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems*, 1(1): 1–51.

Guo, J.; Huang, K.; Yi, X.; Su, Z.; and Zhang, R. 2024a. Rethinking Spectral Graph Neural Networks with Spatially Adaptive Filtering. *arXiv preprint arXiv:2401.09071*.

Guo, Z.; Li, J.; Li, G.; Wang, C.; Shi, S.; and Ruan, B. 2024b. LGMRec: Local and Global Graph Learning for Multimodal Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 8, 8454–8462.

Halko, N.; Martinsson, P.-G.; and Tropp, J. A. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2): 217–288.

He, R.; and McAuley, J. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI conference on artificial intelligence*, 1.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. 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*, 639–648.

Hsu, H.; and Lachenbruch, P. A. 2014. Paired t test. *Wiley StatsRef: statistics reference online*.

Kaššák, O.; Kompan, M.; and Bieliková, M. 2016. Personalized hybrid recommendation for group of users: Top-N multimedia recommender. *Information Processing & Management*, 52(3): 459–477.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.

Mao, K.; Zhu, J.; Xiao, X.; Lu, B.; Wang, Z.; and He, X. 2021. UltraGCN: ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 1253–1262.

Marinó, G. C.; Petrini, A.; Malchiodi, D.; and Frasca, M. 2023. Deep neural networks compression: A comparative survey and choice recommendations. *Neurocomputing*, 520: 152–170.

Nica, B. 2018. A Brief Introduction to Spectral Graph Theory. *EMS Press*.

Ramakrishna, R.; Wai, H.-T.; and Scaglione, A. 2020. A user guide to low-pass graph signal processing and its applications: Tools and applications. *IEEE Signal Processing Magazine*, 37(6): 74–85.

Rangarajan, A. 2001. Learning matrix space image representations. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 153–168. Springer.

Sharma, K.; Lee, Y.-C.; Nambi, S.; Salian, A.; Shah, S.; Kim, S.-W.; and Kumar, S. 2024. A survey of graph neural networks for social recommender systems. *ACM Computing Surveys*, 56(10): 1–34.

Shen, J.; Wang, M.; Yan, S.; and Cui, P. 2013. Multimedia recommendation: technology and techniques. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 1131–1131.

Su, X.; and Khoshgoftaar, T. M. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009(1): 421425.

Tao, Z.; Liu, X.; Xia, Y.; Wang, X.; Yang, L.; Huang, X.; and Chua, T.-S. 2022. Self-supervised learning for multimedia recommendation. *IEEE Transactions on Multimedia*.

Terrell, G. R.; and Scott, D. W. 1992. Variable kernel density estimation. *The Annals of Statistics*, 1236–1265.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wang, C.; Yu, Y.; Ma, W.; Zhang, M.; Chen, C.; Liu, Y.; and Ma, S. 2022. Towards Representation Alignment and Uniformity in Collaborative Filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1816–1825.

Wang, Q.; Wei, Y.; Yin, J.; Wu, J.; Song, X.; and Nie, L. 2021. Dualgnn: Dual graph neural network for multimedia recommendation. *IEEE Transactions on Multimedia*.

Wang, S.; Wang, Y.; Tang, J.; Shu, K.; Ranganath, S.; and Liu, H. 2017. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th international conference on world wide web*, 391–400.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 165–174.

Wei, W.; Huang, C.; Xia, L.; and Zhang, C. 2023. Multimodal self-supervised learning for recommendation. In *Proceedings of the ACM Web Conference 2023*, 790–800.

Wei, Y.; Wang, X.; Nie, L.; He, X.; Hong, R.; and Chua, T.-S. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM international conference on multimedia*, 1437–1445.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 726–735.

Wu, S.; Sun, F.; Zhang, W.; Xie, X.; and Cui, B. 2022. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5): 1–37.

Xu, Y.; Zhu, L.; Cheng, Z.; Li, J.; Zhang, Z.; and Zhang, H. 2021. Multi-modal discrete collaborative filtering for efficient cold-start recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(1): 741–755.

Yu, J.; Xia, X.; Chen, T.; Cui, L.; Hung, N. Q. V.; and Yin, H. 2023a. XSimGCL: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 36(2): 913–926.

Yu, J.; Yin, H.; Xia, X.; Chen, T.; Li, J.; and Huang, Z. 2023b. Self-supervised learning for recommender systems: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(1): 335–355.

Yu, P.; Tan, Z.; Lu, G.; and Bao, B.-K. 2023c. Multi-view graph convolutional network for multimedia recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*, 6576–6585.

Zhang, F.; Yuan, N. J.; Lian, D.; Xie, X.; and Ma, W.-Y. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 353–362.

Zhang, J.; Zhu, Y.; Liu, Q.; Wu, S.; Wang, S.; and Wang, L. 2021. Mining latent structures for multimedia recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*, 3872–3880.

Zhang, J.; Zhu, Y.; Liu, Q.; Zhang, M.; Wu, S.; and Wang, L. 2022. Latent Structure Mining with Contrastive Modality Fusion for Multimedia Recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Zhou, H.; Zhou, X.; Zeng, Z.; Zhang, L.; and Shen, Z. 2023a. A Comprehensive Survey on Multimodal Recommender Systems: Taxonomy, Evaluation, and Future Directions. *arXiv preprint arXiv:2302.04473*.

Zhou, X. 2023. Mmrec: Simplifying multimodal recommendation. In *Proceedings of the 5th ACM International Conference on Multimedia in Asia Workshops*, 1–2.

Zhou, X.; Lin, D.; Liu, Y.; and Miao, C. 2023b. Layer-refined graph convolutional networks for recommendation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 1247–1259. IEEE.

Zhou, X.; and Shen, Z. 2023. A tale of two graphs: Freezing and denoising graph structures for multimodal recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*, 935–943.

Zhou, X.; Zhou, H.; Liu, Y.; Zeng, Z.; Miao, C.; Wang, P.; You, Y.; and Jiang, F. 2023c. Bootstrap latent representations for multi-modal recommendation. In *Proceedings of the ACM Web Conference 2023*, 845–854.