Mining Latent Structures for Multimedia Recommendation

Jinghao Zhang[1,3,*], Yanqiao Zhu[1,3,*], Qiang Liu[1,3,†], Shu Wu[1,3,4], Shuhui Wang[2], and Liang Wang[1,3]

[1]Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences
[2]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences
[3]School of Artificial Intelligence, University of Chinese Academy of Sciences
[4]Artificial Intelligence Research, Chinese Academy of Sciences
{jinghao.zhang,yanqiao.zhu}@cripac.ia.ac.cn, shuhui.wang@vipl.ict.ac.cn, {qiang.liu,shu.wu,wangliang}@nlpr.ia.ac.cn

ABSTRACT

Multimedia content is of predominance in the modern Web era. Investigating how users interact with multimodal items is a continuing concern within the rapid development of recommender systems. The majority of previous work focuses on modeling user-item interactions with multimodal features included as side information. However, this scheme is not well-designed for multimedia recommendation. Specifically, only *collaborative* item-item relationships are implicitly modeled through high-order item-user-item relations. Considering that items are associated with rich contents in multiple modalities, we argue that the latent *semantic* item-item structures underlying these multimodal contents could be beneficial for learning better item representations and further boosting recommendation. To this end, we propose a LATent sTructure mining method for multImodal reCommEndation, which we term LATTICE for brevity. To be specific, in the proposed LATTICE model, we devise a novel modality-aware structure learning layer, which learns item-item structures for each modality and aggregates multiple modalities to obtain latent item graphs. Based on the learned latent graphs, we perform graph convolutions to explicitly inject high-order item affinities into item representations. These enriched item representations can then be plugged into existing collaborative filtering methods to make more accurate recommendations. Extensive experiments on three real-world datasets demonstrate the superiority of our method over state-of-the-art multimedia recommendation methods and validate the efficacy of mining latent item-item relationships from multimodal features.

CCS CONCEPTS

• **Information systems → Recommender systems**; **Personalization**; **Multimedia and multimodal retrieval**.

KEYWORDS

Multimedia recommendation; graph structure learning

---

*The first two authors made equal contribution to this work.
†To whom correspondence should be addressed.

---

1 INTRODUCTION

With the rapid development of Internet, recommender systems have become an indispensable tool to help users find relevant information. Nowadays, users are easily accessible to large amounts of online information represented in multiple modalities, including images, texts, videos, etc. Recent years have witnessed growing research interests on multimedia recommendation, which aims to predict whether a user will interact with an item with multimodal contents. It has been successfully applied to many online applications, such as e-commerce, instant video platforms, and social media platforms.

Focusing on exploiting abundant user-item interactions, collaborative filtering (CF) serves as a foundation of personalized recommender systems, which encodes users and items into low-dimensional dense vectors and makes recommendations based on these embeddings [1, 15, 33]. Traditional work on multimedia recommendation, e.g., VBPR [13], DeepStyle [24], and ACF [3], extends the vanilla CF framework by incorporating multimodal contents as side information in addition to ID embeddings of items. However, as these methods only model direct user-item interactions, their expressiveness is confined.

Inspired by the recent surge of graph neural networks [16, 21, 35], Wang et al. [36] propose to model user-item relationships as bipartite graphs and inject high-order interactions into the embedding process to learn better representations. These graph-based recommender systems [14, 41, 43] achieve great success and obtain state-of-the-art performance. Recently, many attempts have been made to integrate multimodal contents into graph-based recommendation systems. MMGCN [39] constructs modality-specific user-item interaction graphs to model user preferences specific to each modality. Following MMGCN, GRCN [38] utilizes multimodal features to refine user-item interaction graphs by identifying false-positive feedbacks and prunes the corresponding noisy edges.

Despite their effectiveness, previous attempts fail to explicitly model item relationships, which have been proved to be important in recommender systems [31]. Specifically, the majority of previous work concentrates on modeling user-item interactions by constructing better interaction graphs or designing sophisticated user-item aggregation strategies, following the traditional CF paradigm. Therefore, only *collaborative* item relationships are
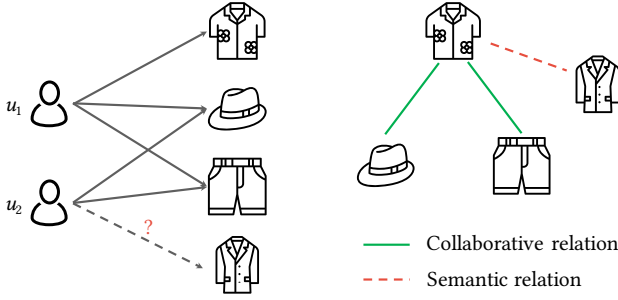
Figure 1: A toy example of recommendation with two types of item relations. In this paper, we argue that semantic structures mined from multimodal features are helpful for comprehensively discovering candidate items supplementary to collaborative signals in traditional work.

implicitly discovered through modeling high-order item-user-item co-occurrences, which potentially leads to a gap to the genuine item-item relations that carry *semantic* relationships. Taking Figure 1 as an example, existing methods will recommend the shirt (👕) for u_2 according to collaborative relationships, since shirts (👕), hats (🎩), and pants (🩳) all interacted with u_1. However, previous work may not be able to recommend coats (🧥) to u_2, which are visually similar to shirts. Considering that items are associated with rich multimodal features, the latent connections underlying multimodal contents would facilitate learning better item representations and assist the recommender models to comprehensively discover candidate items.

Towards this end, we propose a novel LATent sTructure mining scheme for multImodal reCommEndation, LATTICE for brevity. As shown in Figure 2, the proposed LATTICE consists of three key components. Firstly, we develop a novel modality-aware structure learning layer, which learns modality-aware item structures from multimodal features and aggregates modality-aware item graphs to construct latent multimodal item graphs. After that, we perform graph convolutions on the learned latent graphs to explicitly consider item relationships. The resulting item representations are thus infused with high-order item relationships, which will be added into the output item embeddings of CF models. Please kindly note that distinct from previous work that leverages raw features of multimodal contents as side information in addition to ID embeddings, in our work, multimodal features are only used to learn graph structures, and graph convolutions are employed on ID embeddings to directly model *item-item affinities*.

The LATTICE model enjoys two additional benefits. Firstly, previous work that leverages multimodal features to model user-item interactions faces the cold-start problem, when limited user-item interactions are provided. Our work, on the contrary, mines latent item graph structures from multimodal features. Even with limited interactions, items will get similar feedbacks from relevant neighbors through neighborhood aggregation. Secondly, unlike previous attempts which design sophisticated user-item aggregation strategies, LATTICE is agnostic to downstream CF methods. Therefore, it could be served as a play-and-plug module for existing recommender models.

In summary, the main contribution of this work is threefold.

- We highlight the importance of explicitly exploiting item relationships in multimedia recommendation, which are helpful for discovering comprehensive candidate items.
- We propose a novel framework for multimedia recommendation to mine latent relationships beneath multimodal features, which supplement the collaborative signals modeled by traditional CF methods.
- We perform extensive experiments on three public datasets to validate the effectiveness of our proposed model.

To foster reproducible research, our code is made publicly available at https://github.com/CRIPAC-DIG/LATTICE.

2 THE PROPOSED METHOD

In this section, we first formulate the problem and introduce our model in detail. As illustrated in Figure 2, there are three main components in our proposed framework: (1) a modality-aware graph structure learning layer that learns item graph structures from multimodal features and fuses multimodal graphs, (2) graph convolutional layers that learn the embeddings by injecting item-item affinities, and (3) downstream CF methods.

2.1 Preliminaries

Let $\mathcal{U}$, $\mathcal{I}$ denote the set of users and items, respectively. Each user $u \in \mathcal{U}$ is associated with a set of items $\mathcal{I}^u$ with positive feedbacks which indicate the preference score $y_{ui} = 1$ for $i \in \mathcal{I}^u$. $x_u, x_i \in \mathbb{R}^d$ is the input ID embedding of u and i, respectively, where d is the embedding dimension. Besides user-item interactions, multimodal features are offered as content information of items. We denote the modality features of item i as $e_i^m \in \mathbb{R}^{d_m}$, where d_m denotes the dimension of the features, $m \in \mathcal{M}$ is the modality, and $\mathcal{M}$ is the set of modalities. The purpose of multimodal recommendation is to accurately predict users' preferences by ranking items for each user according to predicted preference scores $\hat{y}_{ui}$. In this paper, we consider visual and textual modalities denoted by $\mathcal{M} = \{v, t\}$. Please kindly note that our method is not fixed to the two modalities and multiple modalities can be involved.

2.2 Modality-aware Latent Structure Learning

Multimodal features provide rich and meaningful content information of items, while existing methods only utilize multimodal features as side information for each item, ignoring the important *semantic* relationships of items underlying features. In this section, we introduce how to discover the underlying latent graph structure of item graphs in order to learn better item representations. To be specific, we first construct initial k-nearest-neighbor (kNN) modality-aware item graphs by utilizing raw multimodal features. After that, we learn the latent graph structures from transformed, high-level features based on the initial graph. Finally, we integrate the latent graphs from multiple modalities in an adaptive way.

2.2.1 Constructing initial kNN modality-aware graphs. We first construct initial kNN modality-aware graph S^m by using raw features for each modality m. Based on the hypothesis that similar items are more likely to interact than dissimilar items [27], we quantify the *semantic* relationship between two items by their similarity.
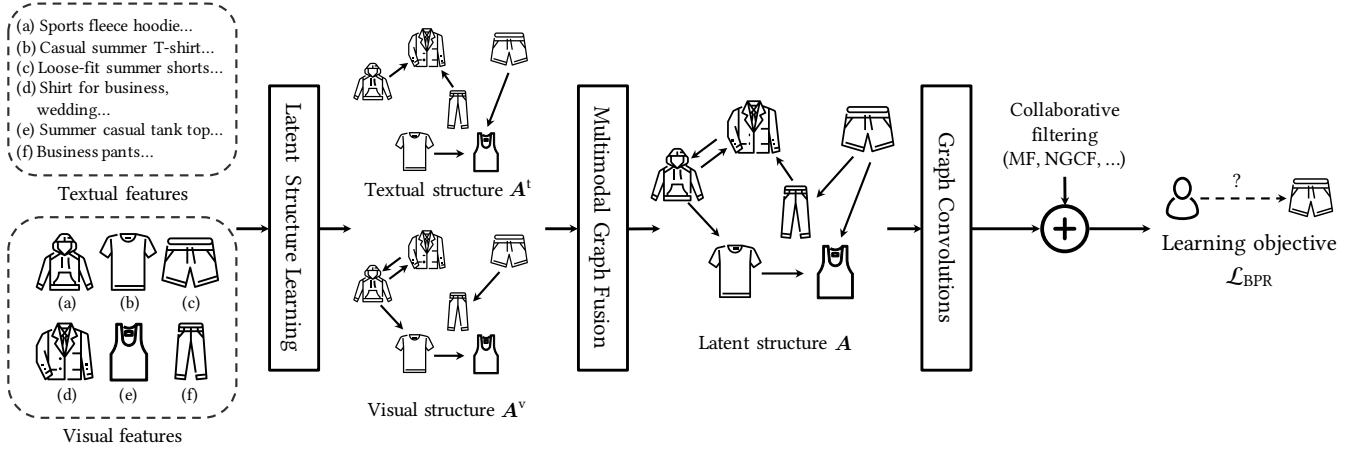
Figure 2: The overall framework of our proposed LATTICE model. We first learn modality-aware item graphs and aggregate multiple modalities in an adaptive manner. Based on the mined latent graphs, we conduct graph convolutions to inject high-order item relationships into item embeddings, which are then combined with downstream CF methods to make recommendations.

Common options for node similarity measurement include cosine similarity [37], kernel-based functions [22], and attention mechanisms [5]. Our method is agnostic to similarity measurements, and we opt to the simple and parameter-free cosine similarity in this paper. The similarity matrix $S^m \in \mathbb{R}^{N \times N}$ is computed by

$$S_{ij}^m = \frac{(e_i^m)^\top e_j^m}{\|e_i^m\|\|e_j^m\|}. \tag{1}$$

Typically, the graph adjacency matrix is supposed to be non-negative but S_{ij} ranges between $[-1, 1]$. Thus, we suppress its negative entries to zeros. Moreover, common graph structures are much sparser other than a fully-connected graph, which is computationally demanding and might introduce noisy, unimportant edges [5]. For this purpose, we conduct kNN sparsification [2] on the dense graph: for each item i, we only keep edges with the top-k confidence scores:

$$\widehat{S}_{ij}^m = \begin{cases} S_{ij}^m, & S_{ij}^m \in \text{top-}k(S_i^m), \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $\widehat{S}^m$ is the resulting sparsified, directed graph adjacency matrix. To alleviate the exploding or vanishing gradient problem [21], we normalize the adjacency matrix as:

$$\widetilde{S}^m = (D^m)^{-\frac{1}{2}} \widehat{S}^m (D^m)^{-\frac{1}{2}}, \tag{3}$$

where $D^m \in \mathbb{R}^{N \times N}$ is the diagonal degree matrix of $\widehat{S}^m$ and $D_{ii}^m = \sum_j \widehat{S}_{ij}^m$.

2.2.2 Learning latent structures. Although we have obtained the modality-aware initial graph structures $\widetilde{S}^m$ by utilizing raw multimodal features, they may not be ideal for the recommendation task. This is because the raw multimodal features are often noisy or even incomplete due to the inevitably error-prone data measurement or collection. Additionally, initial graphs are constructed from the original multimodal features, which may not reflect the genuine graph structures after feature extraction and transformation. To this end, we propose to dynamically learn the graph structures by

transformed, high-level multimodal features and combine learned structures with initial structures.

Firstly, we transform raw modality features into high-level features $\widetilde{e}_i^m$:

$$\widetilde{e}_i^m = W_m e_i^m + b_m, \tag{4}$$

where $W_m \in \mathbb{R}^{d' \times d_m}$ and $b_m \in \mathbb{R}^{d'}$ denote the trainable transformation matrix and bias vector, respectively. d' is the dimension of high-level feature vector $\widetilde{e}_i^m$. We then dynamically infer the graph structures utilizing $\widetilde{e}_i^m$, repeat the graph learning process described in Eqs. (1, 2, 3) and obtain the adjacency matrix $\widetilde{A}^m$.

Although the initial graph could be noisy, it typically still carries rich and useful information regarding item graph structures. Also, drastic change of adjacency matrix will lead to unstable training. To keep rich information of initial item graph and stabilize the training process, we add a skip connection that combines the learned graph with the initial graph:

$$A^m = \lambda \widetilde{S}^m + (1 - \lambda) \widetilde{A}^m, \tag{5}$$

where $\lambda \in (0, 1)$ is the coefficient of skip connection that controls the amount of information from the initial structure. The obtained A^m is the final graph adjacency matrix representing latent structures for modality m. It is worth mentioning that both $\widetilde{S}^m$ and $\widetilde{A}^m$ are sparsified and normalized matrices, thus the final adjacency matrix A^m is also sparsified and normalized, which is computationally efficient and stabilizes gradients.

2.2.3 Aggregating multimodal latent graphs. After we obtained modality-aware adjacency matrix A^m for each modality $m \in \mathcal{M}$, in this section, we explore how to fuse different modalities to compute the final latent structures. In multimedia recommendation, users usually focus on different modalities in different scenarios. For example, one may pay more attention to visual modality when selecting clothes, while focusing more on textual information when picking books, we thus introduce learnable weights to assign different importance scores to modality-specific graphs in an adaptive

way:

$$A = \sum_{m=0}^{|\mathcal{M}|} \alpha_m A^m, \tag{6}$$

where α_m is the importance score of modality m and $A \in \mathbb{R}^{N \times N}$ is the graph that represents multimodal item relationships. We apply the softmax function to keep the adjacency matrix A normalized, such that $\sum_{m=0}^{|\mathcal{M}|} \alpha_m = 1$.

2.3 Graph Convolutions

After obtained the latent structures, we perform graph convolution operations to learn better item representations by injecting item-item affinities into the embedding process. Graph convolutions can be treated as message propagating and aggregation. Through propagating the item representations from its neighbors, one item can aggregate the information within the first-order neighborhood. Furthermore, by stacking multiple graph convolutional layers, the high-order item-item relationships can be captured.

Following He et al. [14], Wu et al. [40], we employ simple message passing and aggregation without feature transformation and non-linear activations which is effective and computationally efficient. In the l-th layer, the message passing and aggregation could be formulated as:

$$\boldsymbol{h}_i^{(l)} = \sum_{j \in \mathcal{N}(i)} A_{ij} \boldsymbol{h}_j^{(l-1)}, \tag{7}$$

where $\mathcal{N}(i)$ is the neighbor items and $\boldsymbol{h}_i^{(l)} \in \mathbb{R}^d$ is the l-th layer item representation of item i. We set the input item representation $\boldsymbol{h}_i^{(0)}$ as its corresponding ID embedding vector $\boldsymbol{x}_i$. We utilize ID embeddings of items as input representations rather than multimodal features, since we employ graph convolutions in order to directly capture item-item affinities. After stacking L layers, $\boldsymbol{h}_i^{(L)}$ encodes the high-order item-item relationships that are constructed by multimodal information and thus can benefit the downstream CF methods.

2.4 Combining with Collaborative Filtering

Different from previous attempts which design sophisticated user-item aggregation strategies, LATTICE learns item representations from multimodal features and then combine them with downstream CF methods that model user-item interactions. It is flexible and could be served as a play-and-plug module for any CF methods.

We denote the output user and item embeddings from CF methods as $\widetilde{\boldsymbol{x}}_u, \widetilde{\boldsymbol{x}}_i \in \mathbb{R}^d$ and simply enhance item embeddings by adding normalized item embeddings $\boldsymbol{h}_i^{(L)}$ learned through item graph:

$$\widehat{\boldsymbol{x}}_i = \widetilde{\boldsymbol{x}}_i + \frac{\boldsymbol{h}_i^{(L)}}{\|\boldsymbol{h}_i^{(L)}\|_2}. \tag{8}$$

We then compute the user-item preference score by taking inner product of user embeddings and enhanced item embeddings:

$$\hat{y}_{ui} = \widetilde{\boldsymbol{x}}_u^\top \widehat{\boldsymbol{x}}_i. \tag{9}$$

We conduct experiments in Section 3.3 on different downstream CF methods. The play-and-plug paradigm separates the usage of multimodal features with user-item interactions, thus alleviating

Table 1: Statistics of the datasets

Dataset[1]	# Users	# Items	# Interactions	Density
Clothing	39,387	23,033	237,488	0.00026
Sports	35,598	18,357	256,308	0.00039
Baby	19,445	7,050	139,110	0.00101

[1] Datasets can be accessed at http://jmcauley.ucsd.edu/data/amazon/links. html.

the cold-start problem, where tailed items are only interacted with few users or even never interacted with users. We learn latent structures for items and the tailed items will get similar feedbacks from relevant neighbors through neighborhood aggregation. We conduct experiments in cold-start settings in Section 3.2 which proves the effectiveness of this play-and-plug paradigm.

2.5 Optimization

We adopt Bayesian Personalized Ranking (BPR) loss [30] to compute the pair-wise ranking, which encourages the prediction of an observed entry to be higher than its unobserved counterparts:

$$\mathcal{L}_{\text{BPR}} = - \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}_u} \sum_{j \notin \mathcal{I}_u} \ln \sigma \left(\hat{y}_{ui} - \hat{y}_{uj} \right), \tag{10}$$

where $\mathcal{I}^u$ indicates the observed items associated with user u and (u, i, j) denotes the pairwise training triples where $i \in \mathcal{I}^u$ is the positive item and $j \notin \mathcal{I}^u$ is the negative item sampled from unobserved interactions. $\sigma(\cdot)$ is the sigmoid function.

3 EXPERIMENTS

In this section, we conduct experiments on three widely used real-world datasets to answer the following research questions:

- **RQ1:** How does our model perform compared with the state-of-the-art multimedia recommendation methods and other CF methods in both warm-start and cold-start settings?
- **RQ2:** How effective are the item graph structures learned from multimodal features?
- **RQ3:** How sensitive is our model under the perturbation of several key hyper-parameters?

3.1 Experiments Settings

3.1.1 Datasets. We conduct experiments on three categories of widely used Amazon dataset introduced by McAuley et al. [26]: (a) Clothing, Shoes and Jewelry, (b) Sports and Outdoors, and (c) Baby, which we refer to as **Clothing**, **Sports**, and **Baby** in brief. The statistics of these three datasets are summarized in Table 1. The three datasets include both visual and textual modalities. We use the 4,096-dimensional visual features that have been extracted and published. For the textual modality, we extract textual embeddings by concatenating the title, descriptions, categories, and brand of each item and utilize sentence-transformers [29] to obtain 1,024-dimensional sentence embeddings.

3.1.2 Baselines. To evaluate the effectiveness of our proposed model, we compare it with several state-of-the-art recommendation models. These baselines fall into two groups: CF methods (i.e.

Model	Clothing			Sports			Baby		
	R@20	P@20	NDCG@20	R@20	P@20	NDCG@20	R@20	P@20	NDCG@20
MF	0.0191	0.0010	0.0088	0.0430	0.0023	0.0202	0.0440	0.0024	0.0200
NGCF	0.0387	0.0020	0.0168	0.0695	0.0037	0.0318	0.0591	0.0032	0.0261
LightGCN	0.0470	0.0024	0.0215	0.0782	0.0042	0.0369	0.0698	0.0037	0.0319
VBPR	0.0481	0.0024	0.0205	0.0582	0.0031	0.0265	0.0486	0.0026	0.0213
MMGCN	0.0501	0.0024	0.0221	0.0638	0.0034	0.0279	0.0640	0.0032	0.0284
GRCN	<u>0.0631</u>	<u>0.0032</u>	<u>0.0276</u>	<u>0.0833</u>	<u>0.0044</u>	<u>0.0377</u>	<u>0.0754</u>	<u>0.0040</u>	<u>0.0336</u>
LATTICE	**0.0710**	**0.0036**	**0.0316**	**0.0915**	**0.0048**	**0.0424**	**0.0829**	**0.0044**	**0.0368**
Improv.	12.5%	12.2%	14.6%	9.8%	8.7%	12.5%	9.9%	9.2%	9.5%

Table 2: Performance comparison of our LATTICE with different baselines in terms of Recall@20 (R@20), Precision@20 (P@20), and NDCG@20. The best performance is highlighted in bold and the second to best is highlighted by <u>underlines</u>. Improv. indicates relative improvements over the best baseline in percentage. All improvements are significant with p-value ≤ 0.05.

MF, NGCF, LightGCN) and deep content-aware recommendation models (i.e. VBPR, MMGCN, GRCN).

- **MF** [30] optimizes Matrix Factorization using the Bayesian personalized ranking (BPR) loss, which exploits the user-item direct interactions only as the target value of interaction function.
- **NGCF** [36] explicitly models user-item interactions by a bipartite graph. By leveraging graph convolutional operations, it allows the embeddings of users and items interact with each other to harvest the collaborative signals as well as high-order connectivity signals.
- **LightGCN** [14] argues the unnecessarily complicated design of GCNs (i.e. feature transformation and nonlinear activation) for recommendation systems and proposes a light model which only consists of two essential components: light graph convolution and layer combination.
- **VBPR** [13]: Based upon the BPR model, it integrates the visual features and ID embeddings of each item as its representation and feed them into Matrix Factorization framework. In our experiments, we concatenate multi-modal features as the content information to predict the interactions between users and items.
- **MMGCN** [39] is one of the state-of-the-art multimodal recommendation methods, which constructs modal-specific graphs and refines modal-specific representations for users and items. Tt aggregates all model-specific representations to obtain the representations of users or items for prediction.
- **GRCN** [38] is also one of the state-of-the-arts multimodal recommendation methods. It refines user-item interaction graph by identifying the false-positive feedback and prunes the corresponding noisy edges in the interaction graph.

3.1.3 Evaluation protocols. We conduct experiments in both warm-start and cold-start settings.

Warm-start settings. For each dataset, we select 80% of historical interactions of each user to constitute the training set, 10% for validation set, and the remaining 10% for testing set. For each observed user-item interaction, we treat it as a positive pair, and

then conduct the negative sampling strategy to pair them with one negative item that the user does not interact before.

Cold-start settings. We remove all user-item interaction pairs associated with a randomly selected 20% item set from the training set. We further divide the half of the items (10%) into the validation set and half (10%) into the testing set. In other words, these items are entirely unseen in the training set.

We adopt three widely-used metrics to evaluate the performance of preference ranking: Recall@k, NDCG@k, and Precision@k. By default, we set $k = 20$ and report the averaged metrics for all users in the testing set.

3.1.4 Implementation details. We implemente our method in PyTorch [28] and set the embedding dimension d fixed to 64 for all models to ensure fair comparison. We optimize all models with the Adam [20] optimizer, where the batch size is fixed at 1024. We use the Xavier initializer [10] to initialize the model parameters. The optimal hyper-parameters are determined via grid search on the validation set: the learning rate is tuned amongst {0.0001, 0.0005, 0.001, 0.005}, the coefficient of ℓ_2 normalization is searched in {0, 10^{-5}, 10^{-4}, 10^{-3}}, and the dropout ratio in {0.0, 0.1, $\cdots$, 0.8}. Besides, we stop training if Recall@20 on the validation set does not increase for 10 successive epochs to avoid overfitting.

3.2 Performance Comparison (RQ1)

We start by comparing the performance of all methods, and then explore how the item graph structures learned from multimodal features alleviate the cold-start problem. In this subsection, we combine our method with LightGCN as downstream CF method, and will also conduct experiments of different CF methods in Section 3.3.

3.2.1 Overall performance. Table 2 reports the performance comparison results, from which we can observe:

- Our method significantly outperforms both CF methods and content-aware methods, verifying the effectiveness of our method. Specifically, our method improves over the strongest baselines in terms of Recall@20 by 12.5%, 9.8%, and 9.9% in Clothing, Sports, and Baby, respectively. This indicates our

Model	Clothing			Sports			Baby		
	R@20	P@20	NDCG@20	R@20	P@20	NDCG@20	R@20	P@20	NDCG@20
MF	0.0191	0.0010	0.0088	0.0430	0.0023	0.0202	0.0440	0.0024	0.0200
MF+feats	0.0456	0.0023	0.0197	0.0674	0.0036	0.0304	0.0701	0.0037	0.0306
LATTICE/feats−MF	0.0519	0.0026	0.0224	0.0708	0.0038	0.0319	0.0729	0.0037	0.0326
LATTICE−MF	0.0577	0.0029	0.0246	0.0753	0.0040	0.0336	0.0767	0.0040	0.0339
Improv.	26.5%	25.9%	24.7%	11.7%	11.4%	10.7%	9.4%	9.4%	10.6%
NGCF	0.0387	0.0020	0.0168	0.0695	0.0037	0.0318	0.0591	0.0032	0.0261
NGCF+feats	0.0436	0.0022	0.0190	0.0748	0.0040	0.0344	0.0660	0.0035	0.0295
LATTICE/feats−NGCF	0.0480	0.0024	0.0212	0.0849	0.0043	0.0374	0.0713	0.0037	0.0307
LATTICE−NGCF	0.0488	0.0025	0.0216	0.0856	0.0045	0.0381	0.0727	0.0039	0.0313
Improv.	12.0%	11.9%	13.7%	14.5%	14.2%	10.9%	10.1%	9.4%	6.0%
LightGCN	0.0470	0.0024	0.0215	0.0782	0.0042	0.0369	0.0698	0.0037	0.0319
LightGCN+feats	0.0477	0.0024	0.0208	0.0754	0.0040	0.0350	0.0793	0.0042	0.0344
LATTICE/feats−LightGCN	0.0643	0.0033	0.0288	0.0832	0.0044	0.0386	0.0756	0.0040	0.0335
LATTICE−LightGCN	0.0710	0.0036	0.0316	0.0915	0.0048	0.0424	0.0836	0.0044	0.0373
Improv.	48.8%	48.4%	52.0%	21.3%	20.5%	21.3%	5.4%	5.2%	8.3%

Table 3: Performance of our proposed LATTICE on top of different downstream collaborative filtering (CF) methods. Improv. indicates relative improvements in percentage over the base CF model with multimodal features (CF+feats).

proposed method is well-designed for multimodal recommendation by discovering underlying item-item relationships from multimodal features.

- Compared with CF methods, content-aware methods yield better performance overall, which indicates that multimodal features provide rich content information about items, and can boost recommendation accuracies. GRCN outperforms other baselines in three datasets since it discovers and prunes false-positive edges in user-item interaction graphs. Despite the sophisticated designed mechanisms, GRCN is still suboptimal compared to LATTICE, which verifies the importance of explicitly capturing item-item relationships.

- Additionally, existing content-aware recommendation models are highly dependent on the representativeness of multimodal features and thus obtain fluctuating performances over different datasets. For Clothing dataset where visual features are very important in revealing item attributes [13, 24], VBPR, MMGCN, and GRCN outperform all CF methods, including the powerful LightGCN. For the other two datasets where multimodal features may not directly reveal item attributes, content-aware methods obtain relatively small improvements. The performances of VBPR and MMGCN are even inferior to CF method LightGCN. Different from existing content-aware methods, we discover the latent item relationships underlying multimodal features instead of directly using them as side information. The latent item relationships are less dependent on the representativeness of multimodal features, and thus we are able to obtain better performance.

3.2.2 Performance in cold-start settings. The cold-start problem remains a prominent challenge in recommendation systems [32]. Multimodal features of items provide rich content information, which can be exploited to alleviate the cold-start problem. We
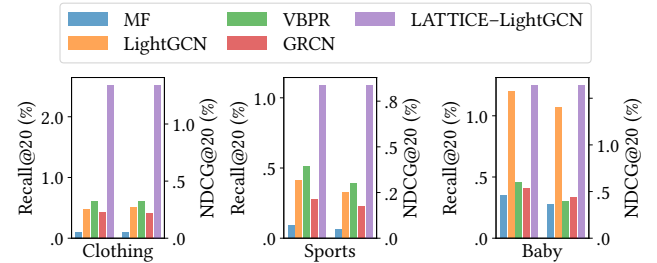
Figure 3: Performances of our method with different baselines in cold-start settings.

conduct cold start experiments and compare with representative baselines. Figure 3 reports the results of performance comparison, from which we can observe:

- LATTICE can alleviate cold-start problem and outperforms all baselines in three datasets. We learn item graphs from multimodal features, along which cold-start items will get similar feedbacks from relevant neighbors through neighborhood aggregation.

- CF methods MF and LightGCN obtain poor performance under cold-start settings in general, primarily because they only leverage users' feedbacks to predict the interactions between users and items. Although these methods may work well for items with sufficient feedbacks, they cannot help in cold-start settings, since no user-item interaction is available to update the representations of cold-start items.

- Content-aware model VBPR outperforms CF methods in general, which indicates the content information provided by multimodal features benefits recommendation for cold-start items. In particular, content information can help bridge the

gap between the existing items to cold-start items. However, some graph-based content-aware methods such as GRCN, although performs well in warm-start settings, obtains poor performance in cold-start settings. GRCN utilizes multimodal features on user-item interaction bipartite graphs, which is also heavily dependent on user-item interactions. For cold-start items, they never interacted with users and become isolated nodes in the user-item graphs, leading to inferior performance.
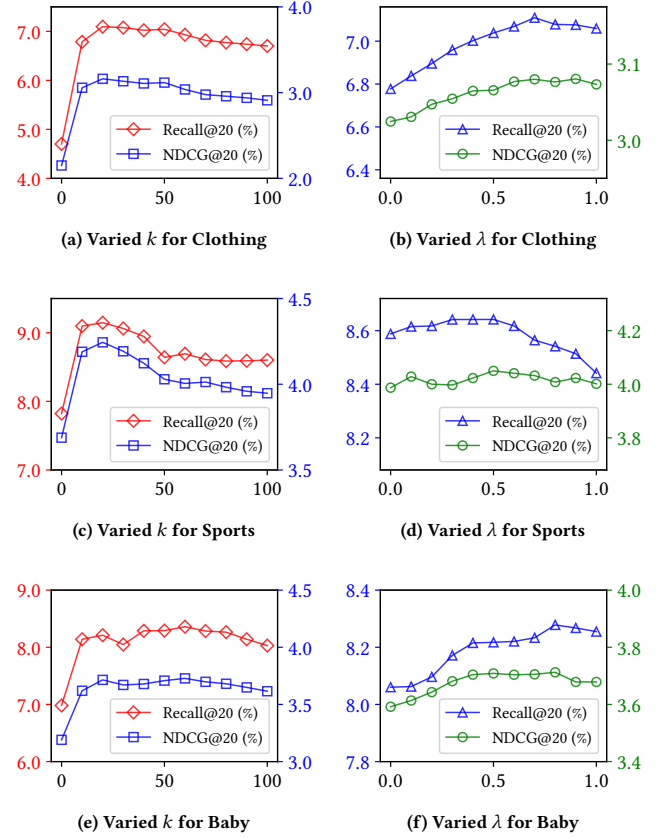
3.3 Ablation Studies (RQ2)

In this subsection, we combine LATTICE with three common-used CF methods, i.e. MF, NGCF, and LightGCN to validate the effectiveness of our proposed method. For each CF method, we have three other variants: the first one is combined with our original method, employs graph convolutions on ID embeddings as described in Section 2.3, named **LATTICE−CF**; the second is **LATTICE/feats−CF**, which employs graph convolutions on multimodal features instead of ID embeddings; the third is named as **CF+feats**, which does not consider latent item-item relationships and directly uses transformed multimodal features to replace the item representations learned from item graphs in Eq. (8). Table 3 summarizes the performance and the relative improvements gained by **LATTICE−CF** over **CF+feats**, from which we have the following observations:

- Our method significantly and consistently outperforms original CF methods and other two variants with all three CF methods, which verifies the effectiveness of discovering latent structures and the flexibility of our plug-in paradigm. Specifically, we obtain 17.6% average improvements over the **CF+feats** variants, which directly utilize multimodal features as side information of items.
- Based on the learned item graph structures, **LATTICE/feats − CF** employs graph convolutions on multimodal features. Our original method **LATTICE−CF** utilizes the same learned structures but employ graph convolutions on item ID embeddings, which aims to directly model *item affinities*. The improvements between two variants validate the effectiveness explicitly modeling item affinities. Multimodal features are used to bridge semantic relationships between items, which is important but not explicitly considered by existing methods.

3.4 Sensitivity Analysis (RQ3)

Since the graph structure learning layer plays a pivotal role in our method, in this subsection, we conduct sensitive analysis with different hyperparameters on graph structure learning layers. Firstly, we investigate performance of LATTICE-LightGCN with respect tor different k value of the k-NN sparsification operation since k value is important which determines the number of neighbors of each item, and controls the amount of information propagated from neighbors. Secondly, we discuss how the skip connection coefficient λ affects the performance which controls the amount of information the from initial graph topology.

3.4.1 Impact of varied k values. Figure 4 reports the results of performance comparison. $k = 0$ means no item relationships are

(a) Varied k for Clothing

(b) Varied λ for Clothing

(c) Varied k for Sports

(d) Varied λ for Sports

(e) Varied k for Baby

(f) Varied λ for Baby

Figure 4: Performances comparison over different hyperparameters settings.

included and the model is degenerated to original LightGCN. We have the following observations:

- Our method gains significant improvement between $k = 0$ and $k = 10$, which validates the rationality of item relationships mined from multimodal features. Even if only a small part of the neighbors are included, we can obtain better item representations by aggregating meaningful and important information, which boost the recommendation performance.
- Furthermore, the performance first imporves as k increases, which verifies the effectiveness of information aggregation along item-item graphs since more neighbors bring more meaningful information that helps to make more accurate recommendations.
- The trend, however, declines when k continues to increase, since there may exist many unimportant neighbors that will inevitably introduce noisy to the information propagation. This demonstrates the necessity of conducting kNN sparsification on the learned dense graph.

3.4.2 Impact of varied coefficients λ. Figure 4 reports the performance comparison. $\lambda = 0$ means only consider the graph structure learned by the transformed high-level multimodal features, and

$\lambda = 1$ means we only consider the structure generated by the raw multimodal features. We have the following observations:

- When we set $\lambda = 0$, the model obtain poor performance. It only learns graph structure from the transformed high-level features, completely updating the adjacency matrix every time, resulting in fluctuated training process.
- The performance first grows as λ becomes larger, validating the importance of initial structures constructed by raw multimodal features. However, it begins to deteriorate when λ continues to increase, since raw features are often noisy due to the inevitably error-prone data measurement or collection process.
- Overall, there are no apparent sharp rises and falls, indicating that our method is not that sensitive to the selection of λ. Notably, all models surpass the baselines (c.f. Table 2), proving the effectiveness of item graphs.

4 RELATED WORK

4.1 Multimodal Recommendation

Collaborative filtering (CF) has achieved great success in recommendation systems, which leverage users' feedbacks (such as clicks and purchases) to predict the preference of users and make recommendations. However, CF-based methods suffer from sparse data with limited user-item interactions and rarely accessed items. To address the problem of data sparsity, it is important to exploit other information besides user-item interactions. Multimodal recommendation systems consider massive multimedia content information of items, which have been successfully applied to many applications, such as e-commerce, instant video platforms, and social media platforms [7, 12, 26, 34].

For example, VBPR [13] extends matrix factorization by incorporating visual features extracted from product images to improve the performance. DVBPR [18] attempts to jointly train the image representation as well as the parameters in a recommender model. Sherlock [11] incorporates categorical information for recommendation based on visual features. DeepStyle [24] disentangles category information from visual representations for learning style features of items and sensing preferences of users. ACF [3] introduces an item-level and component-level attention model for inferring the underlying users' preferences encoded in the implicit user feedbacks. VECF [4] models users' various attentions on different image regions and reviews. MV-RNN [6] uses multimodal features for sequential recommendation in a recurrent framework. Recently, Graph Neural Networks (GNNs) have been introduced into recommendation systems [36, 41, 46] and especially multimodal recommendation systems [23, 38, 39]. MMGCN [39] constructs modal-specific graph and conduct graph convolutional operations, to capture the modal-specific user preference and distills the item representations simultaneously. In this way, the learned user representation can reflect the users' specific interests on items. Following MMGCN, GRCN [38] focuses on adaptively refining the structure of interaction graph to discover and prune potential false-positive edges.

The above methods directly utilize multimodal features as side information of each item. In our model, we step further by discovering fine-grained item-item relationships from multimodal features.

4.2 Deep Graph Structure Learning

GNNs have shown great power on analyzing graph-structured data and have been widely employed for graph analytical tasks across a variety of domains, including node classification [21, 50, 51], link prediction [45], information retrieval[44, 47], etc. However, most GNN methods are highly sensitive to the quality of graph structures and usually require a perfect graph structure that are hard to construct in real-world applications [8]. Since GNNs recursively aggregate information from neighborhoods of one node to compute its node embedding, such an iterative mechanism has cascading effects — small noise in a graph will be propagated to neighboring nodes, affecting the embeddings of many others. Additionally, there also exist many real-world applications where initial graph structures are not available. Recently, considerable literature has arisen around the central theme of Graph Structure Learning (GSL), which targets at jointly learning an optimized graph structure and corresponding representations. There are three categories of GSL methods: metric learning [5, 22, 37], probabilistic modeling [8, 25, 48], and direct optimization approaches [9, 17, 42].

For example, IDGL [5] casts the graph learning problem into a similarity metric learning problem and leverage adaptive graph regularization for controlling the quality of the learned graph; DGM [19] predicts a probabilistic graph, allowing a discrete graph to be sampled accordingly in order to be used in any graph convolutional operator. NeuralSparse [48] considers the graph sparsification task by removing task-irrelevant edges. It utilizes a deep neural network to learn k-neighbor subgraphs by selecting at most k neighbors for each node in the graph. We kindly refer to [49] for a recent overview of approaches for graph structure learning.

In personalized recommendation, although user-item interactions can be formulated bipartite graph naturally, item-item relations remain rarely explored. To model item relationships explicitly, we employ metric learning approaches to represent edge weights as a distance measure between two end nodes, which fits for multimedia recommendation since rich content information can be included to measure the semantic relationship between two items.

5 CONCLUSION

In this paper, we have proposed the latent structure mining method (LATTICE) for multimodal recommendation, which leverages graph structure learning to discover latent item relationships underlying multimodal features. In particular, we first devise a modality-aware graph structure learning layer that learns item graph structures from multimodal features and fuses multimodal graphs. Along the learned graph structures, one item can receive informative high-order affinities from its neighbors by graph convolutions. Finally, we combine our model with downstream CF methods to make recommendations. Empirical results on three public datasets demonstrate the effectiveness of our proposed model.

REFERENCES

[1] Charu C. Aggarwal. 2016. *Recommender Systems: The Textbook*. Springer.
[2] Jie Chen, Haw-ren Fang, and Yousef Saad. 2009. Fast Approximate kNN Graph Construction for High Dimensional Data via Recursive Lanczos Bisection. *JMLR* 10, 69 (2009), 1989–2012.
[3] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. In *SIGIR*. 335–344.
[4] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2019. Personalized Fashion Recommendation with Visual Explanations based on Multimodal Attention Network: Towards Visually Explainable Recommendation. In *SIGIR*. 765–774.
[5] Yu Chen, Lingfei Wu, and Mohammed Zaki. 2020. Iterative Deep Graph Learning for Graph Neural Networks: Better and Robust Node Embeddings. In *NeurIPS*. 19314–19326.
[6] Qiang Cui, Shu Wu, Qiang Liu, Wen Zhong, and Liang Wang. 2018. MV-RNN: A Multi-view Recurrent Neural Network for Sequential Recommendation. *TKDE* 32, 2 (2018), 317–331.
[7] Zeyu Cui, Feng Yu, Shu Wu, Qiang Liu, and Liang Wang. 2021. Disentangled Item Representation for Recommender Systems. *TIST* 12, 2 (Feb. 2021).
[8] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. 2019. Learning Discrete Structures for Graph Neural Networks. In *ICML*. 1972–1982.
[9] Xiang Gao, Wei Hu, and Zongming Guo. 2020. Exploring Structure-Adaptive Graph Learning for Robust Semi-Supervised Classification. In *ICME*. 1–6.
[10] Xavier Glorot and Yoshua Bengio. 2010. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *AISTATS*. 249–256.
[11] Ruining He, Chunbin Lin, Jianguo Wang, and Julian McAuley. 2016. Sherlock: Sparse Hierarchical Embeddings for Visually-aware One-class Collaborative Filtering. In *IJCAI*.
[12] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *WWW*. 507–517.
[13] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *AAAI*. 144–150.
[14] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. 639–648.
[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. 173–182.
[16] Fenyu Hu, Yanqiao Zhu, Shu Wu, Liang Wang, and Tieniu Tan. 2019. Hierarchical Graph Convolutional Networks for Semi-supervised Node Classification. In *IJCAI*. 4532–4539.
[17] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. 2020. Graph Structure Learning for Robust Graph Neural Networks. In *KDD*. 66–74.
[18] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian McAuley. 2017. Visually-Aware Fashion Recommendation and Design with Generative Image Models. In *ICDM*. 207–216.
[19] Anees Kazi, Luca Cosmo, Nassir Navab, and Michael Bronstein. 2020. Differentiable Graph Module (DGM) for Graph Convolutional Networks. *arXiv.org* (Feb. 2020). arXiv:2002.04999v3 [cs.LG]
[20] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.
[21] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.
[22] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. 2018. Adaptive Graph Convolutional Neural Networks. In *AAAI*. 3546–3553.
[23] Xingchen Li, Xiang Wang, Xiangnan He, Long Chen, Jun Xiao, and Tat-Seng Chua. 2020. Hierarchical Fashion Graph Network for Personalized Outfit Recommendation. In *SIGIR*. 159–168.
[24] Qiang Liu, Shu Wu, and Liang Wang. 2017. DeepStyle: Learning User Preferences for Visual Recommendation. In *SIGIR*. 841–844.
[25] Dongsheng Luo, Wei Cheng, Wenchao Yu, Bo Zong, Jingchao Ni, Haifeng Chen, and Xiang Zhang. 2021. Learning to Drop: Robust Graph Neural Network via Topological Denoising. In *WSDM*. 779–787.
[26] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *SIGIR*. 43–52.
[27] Miller McPherson, Lynn Smith-Lovin, and James M. Cook. 2003. Birds of a Feather: Homophily in Social Networks. *Annu. Rev. Sociol.* 27, 1 (2003), 415–444.
[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*. 8024–8035.
[29] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP*. 3980–3990.
[30] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*. 452–461.
[31] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *WWW*. 285–295.
[32] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and Metrics for Cold-Start Recommendations. In *SIGIR*. 253–260.
[33] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Adv. Artif. Intell.* (2009), 421425:1–421425.
[34] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. 2015. Learning Visual Clothing Style with Heterogeneous Dyadic Co-occurrences. In *ICCV*. 4642–4650.
[35] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.
[36] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*. 165–174.
[37] Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, and Jian Pei. 2020. AM-GCN: Adaptive Multi-channel Graph Convolutional Networks. In *KDD*. 1243–1253.
[38] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-Refined Convolutional Network for Multimedia Recommendation with Implicit Feedback. In *ACM Multimedia*. 3451–3459.
[39] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal Graph Convolution Network for Personalized Recommendation of Micro-video. In *ACM Multimedia*. 1437–1445.
[40] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr., Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019. Simplifying Graph Convolutional Networks. In *ICML*. 6861–6871.
[41] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based Recommendation with Graph Neural Networks. In *AAAI*. 346–353.
[42] Liang Yang, Zesheng Kang, Xiaochun Cao, Jin Di, Bo Yang, and Yuanfang Guo. 2019. Topology Optimization based Graph Convolutional Network. In *IJCAI*. 4054–4061.
[43] Feng Yu, Yanqiao Zhu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2020. TAGNN: Target Attentive Graph Neural Networks for Session-based Recommendation. In *SIGIR*. 1921–1924.
[44] Xueli Yu, Weizhi Xu, Zeyu Cui, Shu Wu, and Liang Wang. 2021. Graph-based Hierarchical Relevance Matching Signals for Ad-hoc Retrieval. In *WWW*.
[45] Muhan Zhang and Yixin Chen. 2018. Link Prediction Based on Graph Neural Networks. In *NeurIPS*.
[46] Mengqi Zhang, Shu Wu, Meng Gao, Xin Jiang, Ke Xu, and Liang Wang. 2020. Personalized Graph Neural Networks with Attention Mechanism for Session-Aware Recommendation. *TKDE* (2020).
[47] Yufeng Zhang, Jinghao Zhang, Zeyu Cui, Shu Wu, and Liang Wang. 2021. A Graph-based Relevance Matching Model for Ad-hoc Retrieval. In *AAAI*.
[48] Cheng Zheng, Bo Zong, Wei Cheng, Dongjin Song, Jingchao Ni, Wenchao Yu, Haifeng Chen, and Wei Wang. 2020. Robust Graph Representation Learning via Neural Sparsification. In *ICML*. 11458–11468.
[49] Yanqiao Zhu, Weizhi Xu, Jinghao Zhang, Qiang Liu, Shu Wu, and Liang Wang. 2021. Deep Graph Structure Learning for Robust Representations: A Survey. *arXiv.org* (March 2021). arXiv:2103.03036v1 [cs.LG]
[50] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep Graph Contrastive Representation Learning. In *GRL+@ICML*.
[51] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph Contrastive Learning with Adaptive Augmentation. In *WWW*. 2069–2080.