

# Graph-Refined Convolutional Network for Multimedia Recommendation with Implicit Feedback

Yinwei Wei  
Shandong University  
weiyinwei@hotmail.com

Xiang Wang<sup>§</sup>  
National University of Singapore  
xiangwang@u.nus.edu

Liqiang Nie<sup>§</sup>  
Shandong University  
nieliqiang@gmail.com

Xiangnan He  
University of Science and Technology  
of China  
xiangnanhe@gmail.com

Tat-Seng Chua  
National University of Singapore  
chuats@comp.nus.edu.sg

## ABSTRACT

Reorganizing implicit feedback of users as a user-item interaction graph facilitates the applications of graph convolutional networks (GCNs) in recommendation tasks. In the interaction graph, edges between user and item nodes function as the main element of GCNs to perform information propagation and generate informative representations. Nevertheless, an underlying challenge lies in the quality of interaction graph, since observed interactions with less-interested items occur in implicit feedback (say, a user views micro-videos accidentally). This means that the neighborhoods involved with such false-positive edges will be influenced negatively and the signal on user preference can be severely contaminated. However, existing GCN-based recommender models leave such challenge under-explored, resulting in suboptimal representations and performance.

In this work, we focus on adaptively refining the structure of interaction graph to discover and prune potential false-positive edges. Towards this end, we devise a new GCN-based recommender model, *Graph-Refined Convolutional Network* (GRCN), which adjusts the structure of interaction graph adaptively based on status of model training, instead of remaining the fixed structure. In particular, a graph refining layer is designed to identify the noisy edges with the high confidence of being false-positive interactions, and consequently prune them in a soft manner. We then apply a graph convolutional layer on the refined graph to distill informative signals on user preference. Through extensive experiments on three datasets for micro-video recommendation, we validate the rationality and effectiveness of our GRCN. Further in-depth analysis presents how the refined graph benefits the GCN-based recommender model.

<sup>§</sup>Xiang Wang and Liqiang Nie are the corresponding authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

MM '20, October 12–16, 2020, Seattle, WA, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7988-5/20/10...\$15.00

<https://doi.org/10.1145/3394171.3413556>

## CCS CONCEPTS

• Information systems → Recommender systems.

## KEYWORDS

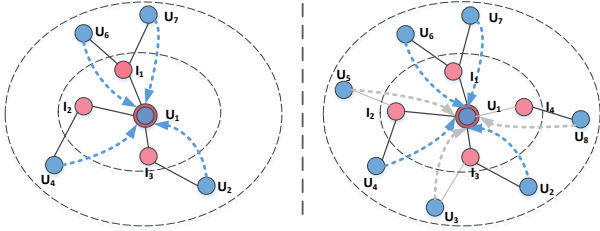
Graph Neural Network, Multimedia Recommendation, Implicit Feedback

## 1 INTRODUCTION

With the high prevalence of the Internet, people have access to large amounts of online multimedia content, such as movies, news, and music. In multimedia content sharing platforms (e.g., Instagram, YouTube, and Tiktok), multimedia recommendation has been a core service to help users identify items of interest. At the core of the recommendation is exploiting multimedia contents of items and historical behaviors of users (e.g., views, clicks) to capture user preference and consequently suggest a ranking list of items.

Learning informative representations of users and items has become a central theme in multimedia recommender systems. Early works like VBPR [11] and ACF [4] integrate multimedia contents (e.g., visual features) and ID embeddings of items together in the traditional collaborative filtering (CF) framework. However, these models limit to explore underlying relationships among users and items, since only direct user-item interactions are taken into consideration. More recently, inspired by the success of graph convolutional networks (GCNs) [15, 18, 20, 24, 32, 39, 41], some efforts [7, 12, 35, 36, 40, 43] have been made to organize user behaviors as a bipartite user-item graph and integrate multi-hop neighbors into representations. Such GCN-based recommender models benefit from powerful representation ability of GCN and have achieved the state-of-the-art performance.

Despite their remarkable performance, we argue that the fixed interaction graphs built upon implicit feedback are highly likely to contain noisy behaviors of users. For example, a user might click some videos shared by her/his friends or even accidentally, while she has no interest in these videos. Such false-positive behaviors appear in the interaction graph as edges between user and item nodes, which are treated equally with the true-positive interactions. When performing information propagation of GCNs, the neighborhoods around these false-positive edges will be influenced negatively and the signals on user preference can be severely contaminated. This is consistent to the vulnerability of



**Figure 1: Illustration of information flows (denoted as dashed curve) caused by true-positive and false-positive interactions in the GCN-based recommender models.**

GCNs against structure perturbations [3, 6]. Therefore, we further argue that the performance of GCN-based recommendation can be significantly degraded by adding a few edges of false-positive interactions in the graph.

**Running Example.** Figure 1 illustrates how the false-positive feedback disturbs the information propagation of the GCN-based recommender model. In the left subfigure, the clean graph involves user-item connections, each of which indicates that a user is truly interested in the item (*i.e.*, solid black lines like  $(U_4, I_2)$ ); whereas, the graph in the right side additionally includes user interactions with less interested items (*i.e.*, solid grey lines like  $(U_5, I_2)$ ). Along with the graph structures, GNN-based recommenders conduct the information propagation mechanism [9, 20, 31] to distill collaborative signal and generate collaborative embeddings of users and items. However, due to the structure difference, not only signal pertinent to user preference (*i.e.*, dashed blue curves like  $(U_4, U_1)$ ), but also noisy information (*i.e.*, dashed grey curves like  $(U_5, U_1)$ ) are aggregated into  $U_1$ 's collaborative embeddings. Further stacking more graph convolutional layers will introduce more noisy signal from multi-hop neighbors. As a result, the representations can be contaminated, and the performance of GNN-based recommenders can be severely degraded.

**Present Work.** To tackle this challenge, we aim to identify and prune the edges that are potential false-positive interactions. Towards this end, we develop a new model, *Graph-Refined Convolutional Network* (GRCN), which exploits the rich content of items and historical behaviors of users to adaptively refine the structure of interaction graph. It consists of three components: graph refining, graph convolutional, and prediction layers. Specifically, the graph refining layer hires the neighbor routing mechanism [27] to refine a prototypical network, highlighting the user preference towards a item prototype *w.r.t.* content in individual modalities. Intuitively, for a given user, an affinity between the target item and her/his prototype reflects the confidence of the target item being true positive in each modality. Then a pruning operation is adopted to prune the edges according to their affinity scores, to corrupt the propagation of their noisy signal. Built upon the refined interaction graph, we apply a graph convolutional layer to obtain the high-quality collaborative embeddings of users and items. Finally, in the prediction layer, we predict how likely a user adopts an item by calculating the similarity between their representations. To demonstrate our proposed method, we conduct extensive experiments on three public datasets. Empirical results

validate that our proposed model outperforms the state-of-the-art baselines like MMGCN [40], DisenGCN [27], and GAT [31]. Moreover, the visualization of the learned user and item embeddings offers a reasonable explanations on why the graph refining operations boosts the GCN-based recommendation method. In a nutshell, our contributions are summarized as follows:

- (1) We explore the influence of implicit feedback to the GCN-based recommendation model. To the best of our knowledge, this is the first attempt to solve the implicit feedback problem against the high-order connectivity.
- (2) We develop a new method GRCN, which adaptively refines the structure of user-item interaction graph to harness the applications of GNNs in recommendation tasks.
- (3) Extensive experiments in three real-world datasets validate the rationality of our assumptions and the effectiveness of our method. Our codes are available in <https://github.com/weiyinwei/GRCN>.

## 2 METHODOLOGY

### 2.1 Preliminary

Suppose there are numbers of historical interaction records (*i.e.* implicit feedback) between users and items. We collect a set  $\mathcal{U}$  of  $N$  users and a set  $\mathcal{I}$  of  $M$  items from the records. Beyond the interaction signal, the multimodal features of items are extracted from their content involving the visual, acoustic, and textual modalities, which are denoted as  $v$ ,  $a$ , and  $t$ , respectively. For a item  $i \in \mathcal{I}$ , we denote its feature vector as  $\mathbf{i}_m \in \mathcal{R}^{M \times D_m}$ , where  $m \in \mathcal{M} = \{v, a, t\}$  is the indicator of multiple modalities and  $D_m$  is the dimension of the vector.

To conduct the graph convolutional operations, we construct a user-item interaction graph  $\mathcal{G} = \{\mathbf{E}, \mathbf{A}\}$ , which follows the GCN-based recommendation [36, 40]. In particular,  $\mathbf{E} \in \mathcal{R}^{D \times (N+M)}$  denotes the trainable embedding matrix of nodes (*i.e.* users and items), where  $D$  represents the dimension of the embedding. And,  $\mathbf{A} \in \mathcal{R}^{N \times M}$  is the symmetric matrix reflecting the connections of user and item pairs. Given a user  $u \in \mathcal{U}$  and a item  $i \in \mathcal{I}$ , we denote  $\mathbf{A}_{u,i} = 1$  if  $u$  has interacted with  $i$ ; otherwise,  $\mathbf{A}_{u,i} = 0$ .

### 2.2 Model Framework

In this section, we detail our proposed model. As illustrated in Figure 2, the model consists of three components: 1) the graph refining layer that adjusts the graph structure by identifying and pruning the noisy edges in interaction graph; 2) the graph convolutional layer which performs the graph convolutional operations on the refined graph to enrich the embeddings of items and users; and 3) the prediction layer that infers the interaction of each user and item pair.

**2.2.1 Graph Refining Layer.** To refine the structure of constructed interaction graph, we work under the reasonable assumption that the content of item belonging to false-positive interaction is far from the user preference. Therefore, we introduce the prototypical network to learn user preference to the content information, and then prune the noisy edges according to the confidence of edges being the false-positive interactions.

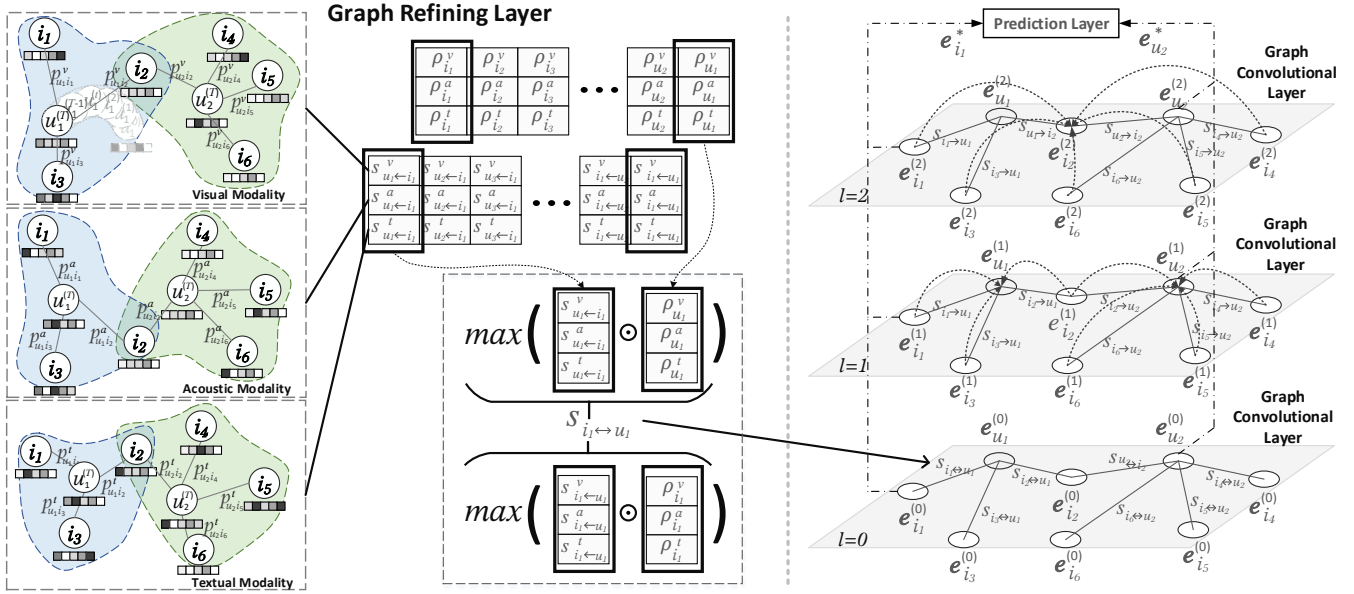


Figure 2: Schematic illustration of our proposed model. It consists of three components, namely graph refining layer, graph convolutional layer, and prediction layer.

**Prototypical Network.** Intuitively, each user preference could be learned from the content of items which directly connect to the user node in the user-item graph. However, since there are some noisy edges in the graph, it is hard to immediately model the user preference with the neighbor nodes. Inspired by the idea of prototype learning [34], we regard the user preference as her/his prototype in a metric space and harness a prototypical network to approach it.

For this goal, the content signal of item is projected into a metric space to distill the informative features related to the user preference, as

$$\bar{\mathbf{i}}_m = \text{leaky\_relu}(\mathbf{W}_m \mathbf{i}_m + \mathbf{b}_m) \quad (1)$$

where  $\text{leaky\_relu}(\cdot)$ ,  $\mathbf{W}_m \in \mathbb{R}^{D' \times D_m}$  and  $\mathbf{b}_m \in \mathbb{R}^{D' \times 1}$  denote the activation function [28], trainable weight matrix and bias vector, respectively. And  $D'$  is the dimension of distilled feature vector  $\bar{\mathbf{i}}_m$ .

Then, we introduce the neighbor routing mechanism [27] into prototypical network, to approach the prototype w.r.t. representation of user preference. Given a user, with the iterative routing operations, her/his representation is adjusted by jointly analyzing her/his similarities to its neighbors. To facilitate the description, we elaborate on the process in the single modality and do the same operations on the others.

In the initial iteration, we define a trainable vector  $\mathbf{u}_{(0)}$  to represent the preference of user  $u \in \mathcal{U}$ . And, we conduct the inner product between user preference and item features to calculate their similarity, formally,

$$p_{u,i} = \frac{\exp(\bar{\mathbf{i}}^T \mathbf{u}_{(0)})}{\sum_{j \in \mathcal{N}(u)} \exp(\bar{\mathbf{j}}^T \mathbf{u}_{(0)})}, \quad (2)$$

where  $p_{u,i}$  denotes the similarity between  $u$  and  $i$ . A higher value suggests that the content signal more informative to the user

preference modeling. In addition,  $\mathcal{N}(u)$  is used to represent the set of neighbors of node  $u$  in the user-item graph.

Following this, we tune the representation of user preference in the metric space via combining the weighted sum of its neighbors' feature vectors. It is formulated as,

$$\mathbf{u}_{(1)} = \mathbf{u}_{(0)} + \sum_{i \in \mathcal{N}(u)} p_{u,i} \bar{\mathbf{i}}, \quad (3)$$

where  $\mathbf{u}_{(1)}$  is the user representation after one iteration operation. Moreover, we normalize it to avoid its scale of increasing with iterative operations.

With the iteration  $t = 2, \dots, T$ , based on the output of previous iteration, the user representation is adjusted towards the prototype of her/his preference, which is recursively formulated as:

$$\begin{cases} \mathbf{u}_{(t)} = \mathbf{u}_{(t-1)} + \sum_{i \in \mathcal{N}(u)} p_{u,i} \bar{\mathbf{i}}, \\ p_{u,i} = \frac{\exp(\bar{\mathbf{i}}^T \mathbf{u}_{(t-1)})}{\sum_{j \in \mathcal{N}(u)} \exp(\bar{\mathbf{j}}^T \mathbf{u}_{(t-1)})}. \end{cases} \quad (4)$$

Finally, it outputs user preference to the item content, as  $\bar{\mathbf{u}} = \mathbf{u}_{(T)}$ . In what follows, we use  $\bar{\mathbf{u}}$  to denote the user preference to the content information.

**Pruning Operations.** To identify noisy edges, we score the affinity between user preference and item content to measure the confidence of the corresponding edge being true-positive interaction in each modality. Then, we integrate the scores of each edge in multiple modalities to yield the weight and assign it to the edge, which implements the pruning operations in a soft manner.

For each modality, with the obtained user preference and distilled item features, we calculate the relative distances between them in

two directions. It is formulated as,

$$\begin{cases} \bar{s}_{u \leftarrow i}^m = \frac{\exp(\bar{\mathbf{u}}_m^T \bar{\mathbf{i}}_m)}{\sum_{j \in \mathcal{N}(u)} \exp(\bar{\mathbf{u}}_m^T \bar{\mathbf{j}}_m)}, \\ \bar{s}_{i \leftarrow u}^m = \frac{\exp(\bar{\mathbf{i}}_m^T \bar{\mathbf{u}}_m)}{\sum_{v \in \mathcal{N}(i)} \exp(\bar{\mathbf{i}}_m^T \bar{\mathbf{v}}_m)}, \end{cases} \quad (5)$$

where  $\bar{s}_{u \leftarrow i}^m$  and  $\bar{s}_{i \leftarrow u}^m$  are the scores reflecting the affinities between  $\bar{\mathbf{u}}_m$  and  $\bar{\mathbf{i}}_m$  in  $m$ -th modality.

To integrate the multimodal scores, we define a base vector for each user or item, as follows:

$$\boldsymbol{\rho} = [\rho^v, \rho^a, \rho^t], \quad (6)$$

where  $\boldsymbol{\rho}$  denotes the base vector. Elements of the user's base vector are used to measure her/his relative preferences to the different modalities. For the item's base vector, each element represents the importance of content signal in the corresponding modality to the item representation.

Incorporated base vectors, the weights for the edges are computed by fusing the multimodal scores, as

$$\begin{cases} s_{u \leftarrow i} = \max(\rho_u^v \bar{s}_{u \leftarrow i}^v, \rho_u^a \bar{s}_{u \leftarrow i}^a, \rho_u^t \bar{s}_{u \leftarrow i}^t), \\ s_{i \leftarrow u} = \max(\rho_i^v \bar{s}_{i \leftarrow u}^v, \rho_i^a \bar{s}_{i \leftarrow u}^a, \rho_i^t \bar{s}_{i \leftarrow u}^t). \end{cases} \quad (7)$$

where  $\max(\cdot)$  denotes maximization operation selecting the max value. Besides, the combination operation is also able to implement in different forms, such as mean and maximization operations without base values.

In summary, with the base vector and obtained affinity scores, we achieve the weight for each edge to softly prune the noisy edge.

**2.2.2 Graph Convolutional Layer.** Following the mainstream of GCN-based models [2, 36], we treat the graph convolutional operations as the message passing and aggregation. Using the graph convolutional operations, we could model the collaborative signal conveyed by user-item interaction graph. Further, by running the stacked graph convolutional layers, the high-order connectivity information is captured and aggregated. Towards the implicit feedback, the obtained weights for the edges are used to control the passed message. In particular, it corrupts the propagation of noise signal from false-positive interaction.

Formally, in the  $l$ -th layer, the message passing and aggregation could be formulated as,

$$\begin{cases} \mathbf{e}_u^{(l)} = \sum_{i \in \mathcal{N}(u)} s_{u \leftarrow i} \mathbf{e}_i^{(l-1)}, \\ \mathbf{e}_i^{(l)} = \sum_{u \in \mathcal{N}(i)} s_{i \leftarrow u} \mathbf{e}_u^{(l-1)}. \end{cases} \quad (8)$$

where  $\mathbf{e} \in \mathbb{R}^{D \times 1}$  denotes the corresponding ID embedding vector. With this operation, we collect the collaborative signal from  $l$ -hop neighbors.

Stacking  $L$  layers, we obtain the embedding at each layer and integrate them:

$$\mathbf{e}_u = \sum_{l=0}^L \mathbf{e}_u^{(l)}, \quad \mathbf{e}_i = \sum_{l=0}^L \mathbf{e}_i^{(l)}. \quad (9)$$

Whereinto,  $\mathbf{e}_u^{(0)}$  and  $\mathbf{e}_i^{(0)}$  denote the initial ID embeddings from the embedding matrix  $\mathbf{E}$ , respectively. The enriched embeddings (i.e.  $\mathbf{e}_u$  and  $\mathbf{e}_i$ ) are constituted by combing the embeddings from 0-th layer

**Table 1: Summary of the datasets. The dimensions of visual, acoustic, and textual modalities are denoted by V, A, and T, respectively.**

Dataset	#Interactions	#Items	#Users	Sparsity	V	A	T
MovieLens	1,239,508	5,986	55,485	99.63%	2,048	128	100
Tiktok	726,065	76,085	36,656	99.97%	128	128	128
Kwai	298,492	86,483	7,010	99.95%	2,048	-	-

to  $L$ -th layer. It encodes and injects the high-order connectivity information into the embedding of each node to enhance the representativeness.

**2.2.3 Prediction Layer.** To gain the representation of each user or item, we follow the idea that users have varying preferences in different modalities [40]. Specifically, we concatenate the multimodal features and the enriched ID embedding as a whole vector, formally,

$$\begin{cases} \mathbf{e}_u^* = \mathbf{e}_u \parallel \bar{\mathbf{u}}_v \parallel \bar{\mathbf{u}}_a \parallel \bar{\mathbf{u}}_t, \\ \mathbf{e}_i^* = \mathbf{e}_i \parallel \bar{\mathbf{i}}_v \parallel \bar{\mathbf{i}}_a \parallel \bar{\mathbf{i}}_t, \end{cases} \quad (10)$$

where the symbol  $\parallel$  means the concatenation operation.

Beyond the collaborative signals, the representation contains the user preference to the item content, which contributes to inferring the interaction between users and items.

Finally, we conduct the inner product between user and item representations, as

$$y_{u,i} = \mathbf{e}_u^{*T} \mathbf{e}_i^*, \quad (11)$$

where the output  $y_{u,i}$  is used to estimate the user's preference towards the target item. A higher score suggests that the user prefers the item more and vice versa.

## 2.3 Optimization

To learn the parameters of the proposed model, we adopt Bayesian Personalized Ranking (BPR) [33] to conduct the pair-wise ranking. As such, we construct a triplet of one user  $u$ , one observed item  $i$ , and one unobserved item  $j$ , formally as,

$$\mathcal{T} = \{(u, i, j) \mid \mathbf{A}_{u,i} = 1, \mathbf{A}_{u,j} = 0\}, \quad (12)$$

where  $\mathcal{T}$  is a triplet set for training.

Therefore, the objective function can be defined as,

$$\mathcal{L} = \sum_{(u,i,j) \in \mathcal{T}} -\ln \phi(y_{u,i} - y_{u,j}) + \lambda \|\theta\|_2, \quad (13)$$

where  $\phi(\cdot)$ ,  $\lambda$ , and  $\theta$  represent the *sigmoid* function, regularization weight and parameters of models, respectively.

## 3 EXPERIMENTS

Through conducting extensive experiments on three public datasets, we evaluate our proposed model and answer the following research questions:

- **RQ1** How does our proposed model perform compared with state-of-the-art GCN-based recommendation models?
- **RQ2** How does each design (i.e. prototypical network and pruning operations) affect the performance of our model?
- **RQ3** How does the representation benefit from the refined graph?

**Table 2: Performance comparison between our model and the baselines over the three datasets.**

Model	Movielens			Tiktok			Kwai		
	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG
GraphSAGE	0.0496	0.1984	0.2136	0.0128	0.0631	0.0606	0.008	0.0286	0.0467
MMGCN	<b>0.0581</b>	<b>0.2345</b>	<b>0.2517</b>	0.0144	0.0808	0.0674	0.0120	0.0398	0.0681
NGCF	0.0547	0.2196	0.2342	0.0135	0.0780	0.0661	0.0118	0.0402	0.0699
DisenGCN	0.0555	0.2222	0.2401	0.0145	0.0760	0.0639	0.0127	0.0403	0.0683
GAT	0.0569	0.2307	0.2434	<b>0.0166</b>	<b>0.0891</b>	<b>0.0802</b>	<b>0.0151</b>	<b>0.0441</b>	<b>0.0744</b>
Ours	<b>0.0639*</b>	<b>0.2569*</b>	<b>0.2754*</b>	<b>0.0195*</b>	<b>0.1048*</b>	<b>0.0938*</b>	<b>0.0168*</b>	<b>0.0492*</b>	<b>0.0864*</b>
%Improv.	<b>9.98%</b>	<b>9.55%</b>	<b>9.42%</b>	<b>17.47%</b>	<b>17.62%</b>	<b>16.96%</b>	<b>11.26%</b>	<b>11.56%</b>	<b>15.66%</b>

Before answering the above three questions, we describe the datasets, evaluation protocols, baselines, and parameter settings in the experiments.

### 3.1 Experiments Settings

**3.1.1 Dataset.** As the micro-video contains rich multimedia information — frames, sound tracks, and descriptions [25, 26, 30], we performed the micro-video personalized recommendation to evaluate our proposed method. Following MMGCN, we conducted extensive experiments on three publicly accessible datasets, including Movielens, Tiktok, and Kwai. The statistics of datasets are summarized in Table 1.

- **Movielens.** The dataset is widely used in the personalized recommendation<sup>1</sup>. To evaluate the multimedia recommendation, researchers extracted keyframes and soundtracks from the video trailers, as well as collected the video descriptions [40]. With some pre-trained deep learning models [1, 10, 14], the visual, acoustic, and textual features are captured from the keyframes, audio tracks, and descriptions, respectively. In our experiments, we treat all ratings as the implicit feedback of the corresponding user and item pairs.
- **Tiktok.** This dataset is released by Tiktok<sup>2</sup> which is a popular micro-video sharing platform. Beyond the interaction information, the visual, acoustic, and textual features are extracted from the micro-videos and provided.
- **Kwai.** As a micro-video service provider, Kwai<sup>3</sup> released a large-scale micro-video dataset. The dataset contains users, micro-videos, and the users’ behavior records with the timestamps. To evaluate the proposed method from implicit feedback, we collected some click records associated with the corresponding users and micro-videos in a certain period. Different from the above datasets, the audio and textual features are not given.

For each dataset, we used the ratio 8 : 1 : 1 to randomly split the historical interactions of each user and constituted the training set, validation set, and testing. For the training set, we conducted a negative sampling strategy to create the triples for parameter optimization. The validation set and testing set are used to tune the hyper-parameters and evaluate the performance in the experiments, respectively.

**3.1.2 Evaluation Protocols.** For each user in the validation and the testing sets, we treated all micro-videos she/he did not consume before as the negative samples. During the validation and testing phases, we used the trained model to score the interactions of user and micro-video pairs and ranked them in a descending order. Moreover, following the widely-used evaluation metrics [23, 37, 38, 44], we adopted precision@K, recall@K, and Normalized Discounted Cumulative Gain (NDCG@K) to evaluate the performance of methods. By default, we set  $K = 10$  and reported the average values of the three metrics for all users in the test set.

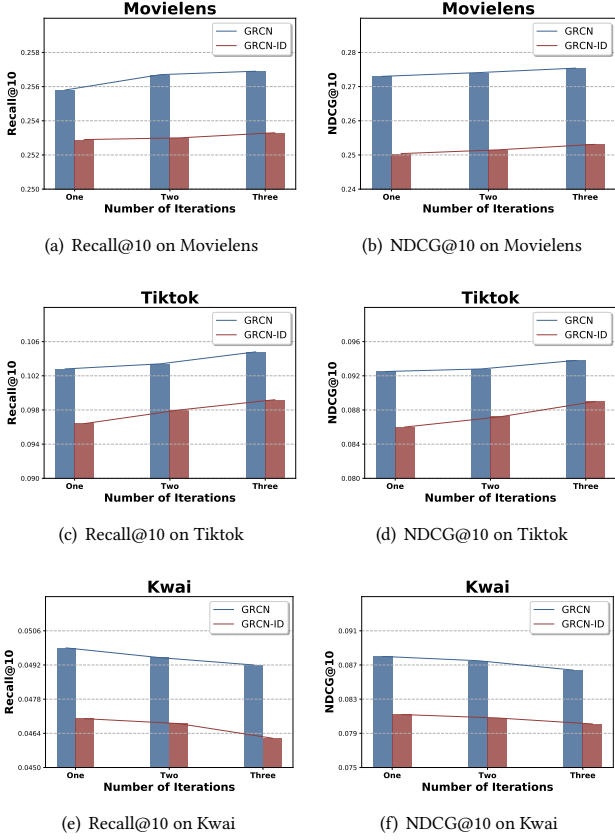
**3.1.3 Baselines.** To evaluate the boosting of our proposed model towards the GCN-based recommendation, we compared it with several state-of-the-art GCN-based models for recommendation with implicit feedback. We briefly divided them into two groups: message-nonadaptive methods (*i.e.* GraphSAGE and MMGCN) and message-adaptive methods (*i.e.* NGCF, GAT, and DisenGCN). For a fair comparison, we employed two graph convolutional layers in our proposed model and baselines.

- **GraphSAGE** [9] We applied GraphSAGE on the user-item graph from implicit feedback to predict the interaction between user and item. With the trainable aggregation functions, the model is able to pass the message along the graph structure and collect them to update the representation of each node.
- **MMGCN** [40] The model learns the model-specific user preference to the content information via the directly information interchange between user and item in each mormaitality. Based on the assumption that the user prefers the observed item over the unobserved one, it is trained from implicit feedback.
- **NGCF** [36] As the state-of-the-arts GCN-based model of personalized recommendation, NGCF explicitly models and injects the collaborative signal into user and item embeddings. It measures the distance between user and item embeddings to control the passed message.
- **DisenGCN** [27] The method could disentangle the representation of each node into several factors. By dynamically identifying the correlation of factor between nodes, it assigns the different weights to edges and aggregates the message to improve the representation of each node.
- **GAT** [31] This method is able to automatically learn and specify different weights to the neighbors of each node. With the learned weights, it denoises the information from the neighbors to improve the personalized recommendation.

<sup>1</sup><https://movielens.org/>.

<sup>2</sup><https://www.tiktok.com/>.

<sup>3</sup><https://www.kwai.com/>.



**Figure 3: Performance in terms of Recall@10 and NDCG@10 w.r.t. different numbers of iterations performing prototypical network on Movielens, Tiktok, and Kwai.**

**3.1.4 Parameter Settings.** The Pytorch<sup>4</sup> and torch-geometric packages<sup>5</sup> are utilized to implement our proposed model. We used the Xavier [8] and Adam [19] methods to initialize and optimize parameters, respectively. In addition, the learning rate is searched in  $\{0.0001, 0.001, 0.01, 0.1, 1\}$  and regularization weight is tuned in  $\{0.00001, 0.0001, 0.001, 0.01, 0.1\}$ . Besides, we stopped the training if recall@10 on the validation data does not increase for 20 successive epochs. For the baselines, we followed the designs in their articles to achieve the best performance. Further, we did the same options and fixed the dimension of the ID embedding vector to 64 for all models to ensure a fair comparison.

## 3.2 Performance Comparison (RQ1)

Table 2 reports the performance comparison results. We have the following observations:

- Without any doubt, our proposed model consistently achieves the best performance on three datasets. In particular, the method improves over the strongest baselines in terms of Recall@10 by 9.55%, 17.62%, and 11.56% in Movielens, Tiktok, and Kwai, respectively. With the similar graph convolutional operations, the

improvements could be attributed to the graph refining. It verifies our suggestion that identifying and pruning the noisy edges in the interaction graph benefits the GCN-based recommendation model.

- Comparing with GraphSAGE, message-adaptive methods, including NGCF, GAT, DisenGCN, and GRCN, yield better results. It implies that the message during the passing and aggregation process conveys some meaningless or harmful signal for the node representation. In other words, the local structure information in the graph constructed by implicit feedback cannot directly reflect the user preference pattern.
- It is worthwhile pointing that MMGCN slightly outperforms other baselines in several cases. We believe one possible reason is that MMGCN sufficiently leverages the multimodal information extracted from items to represent the user preference. Although the method ignores the perturbations in the structure of graph caused by implicit feedback, it implicitly distills the information during its cross-modalities combination layers. The results on Kwai, which only contains the visual modality, could be used to verify this point.
- Obviously, adaptively adjusting the passed message in the graph convolutional operations contributes to the user and item modeling in recommendation with implicit feedback. However, compared with GRCN, other message-adaptive models are suboptimal. We argue the message yielded by these models is corrupted by the graph structure. Specifically, the uncertain message totally depending on initial nodes' ID embeddings is alternately propagated from one node to the other over each edge. Rather, our proposed method measures the affinity between user preference and item content to control the message passing before iteratively conducting the graph convolution layers.

## 3.3 Ablation Study (RQ2)

In this section, we evaluated the designs of our proposed model, especially the graph refining layer. The graph refining layer is devised to model the user preference from content information and refine the interaction graph for optimizing the GCN-based recommendation method. As such, we conducted experiments to evaluate two components in the refining layer, including the prototypical network and pruning operations.

**3.3.1 Effects of Prototypical Network.** To evaluate the effect of prototypical network on user preference learning, we performed the experiments under different numbers of routing iterations. Meanwhile, we compared the proposed model with the variant, marked as GRCN-ID, which discarded the multimodal user preference and item features in the prediction layer and directly predicted the interaction between user and item with only their ID embeddings. As illustrated in Fig 3, we observe that:

- With the increasing of iterations, the values in terms of Recall@10 and NDCG@10 are varying. It is indicated that the performance of our proposed method is affected by user preference modeling.
- On Movielens and Tiktok, the values are increasing when we iteratively perform the prototypical network. We believe that the user representation is tuned towards her/his preference to the content information, which benefits the correlation computing and graph refining.

<sup>4</sup><https://pytorch.org/>.

<sup>5</sup><https://pytorch-geometric.readthedocs.io/>.



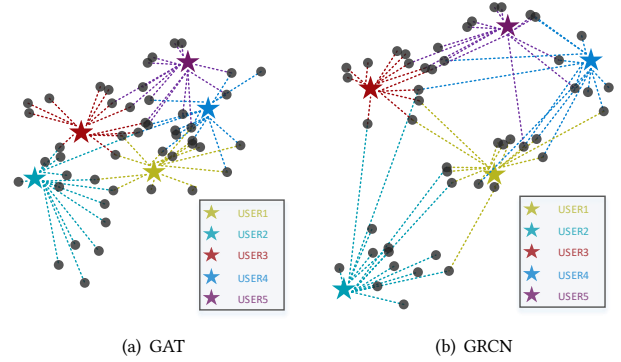
**Table 3: Effect of pruning operations on Movielens and Tiktok. (Visual, Acoustic, and Textual denote running GRCN on the visual, acoustic, and textual modality, respectively.)**

Model	Movielens			Tiktok		
	Precision	Recall	NDCG	Precision	Recall	NDCG
Visual	0.0633	0.2545	0.2714	0.0175	0.0906	0.0822
Acoustic	0.0621	0.2540	0.2701	0.0144	0.0788	0.0694
Textual	0.0611	0.2531	0.2648	0.0142	0.0770	0.0675
GRCN <sub>max</sub>	0.0621	0.2542	0.2701	0.0175	0.0941	0.0838
GRCN <sub>mean</sub>	0.0617	0.2477	0.2660	0.0159	0.0854	0.0751
GRCN <sub>hard</sub>	0.0639	0.2547	0.2750	0.0180	0.0962	0.0868
<b>GRCN</b>	<b>0.0639</b>	<b>0.2569</b>	<b>0.2754</b>	<b>0.0195*</b>	<b>0.1048*</b>	<b>0.0938*</b>

- For Kwai, the curves of results *w.r.t* Recall@10 and NDCG@10 tend to decline with the iterative operations. We suggest the phenomenon is caused by the smooth representation of users. In this dataset, the number of average interaction records of a user is much more than the others. Hence, the iterative operations make the representation smooth, which is hard to capture the discriminative features for users.
- Obviously, GRCN outperforms GRCN-ID, which benefits from the user (item) representation enriched by corresponding user preference (item features). Although GRCN-ID is suboptimal on three datasets, it is still better than the abovementioned GCN-based baselines. Jointly analyzing the performance of baselines shown in Tabel 2, it qualitatively verifies the refined graph structure contributes to the GCN-based recommendation model.

**3.3.2 Effects of Pruning Operations.** To explore the pruning operations, we compared the performance of our proposed model with three different implementations. Specifically, we adopted the mean and maximization operations without the base value, which are named as GRCN<sub>mean</sub> and GRCN<sub>max</sub>, to fuse the multimodal affinity scores, respectively. Besides, we compared our model with the hard pruning strategy which is labeled as GRCN<sub>hard</sub> and implemented with ReLU function [29] to completely interrupt the message passed from the false-positive interaction. In addition, we conducted GRCN in each modality as a comparison. From the results in Table 3, we have following observations:

- In most cases, we observe that the results of three implements are better than that of models within the single modality. It demonstrates that incorporating the information from multiple modalities facilitates the pruning operations, since users have various opinions about the different modalities of micro-videos.
- Observing the results of GRCN<sub>mean</sub> and GRCN<sub>max</sub>, we find that the later is superior to the former. Especially, the results of GRCN<sub>mean</sub> significantly underperform the model which merely considers the visual modality. It probably implies that the maximization operation is consistent with the relationship among different modalities and able to model it.
- Both our proposed model and its variant GRCN<sub>hard</sub> outperform the other two implements without the base value. It shows that incorporating the base value is capable of boosting performance, which justifies our purpose regarding the base values.



**Figure 4: Visualization of the learned t-SNE transformed representations derived from GAT and GRCN. The star marks denote the user randomly selected from Tiktok. The link between star and circle means the interaction between them.**

- As expected, compared with other variants, our proposed model yields the best results. Beyond the multimodal information and base value incorporating, it also makes sense because of the soft pruning operation. Different from the hard pruning, pruning operation in a soft manner not only weakens the noise caused by false-positive interaction but enhances the message from true-positive ones. It contributes to refine the graph structure for graph convolutional operations.

### 3.4 Visualization

In this section, we provided several users randomly selected from Tiktok associated with their consumed items. Using the t-Distributed Stochastic Neighbor Embedding (t-SNE) in 2-dimension, we visualized their ID embeddings, which are learned from GAT and GRCN, respectively. Besides, we used same color to denote edges from a user node to the nodes of items she/he interacted with. As illustrated in Figure 4, we have two key observations:

- From the distribution of nodes in two figures, we find that the nodes representing the items exhibit discernible clustering around the user nodes in Figure 4(b). It means that GRCN discriminately represent the users, although there are several items consumed by the same users. We attribute this to the graph refining operations assigning different weights for edges according to affinities between corresponding user preference and item content, which eliminates the noises from false-positive interactions.
- By observing Figure 4(a), it is shown that the distribution of users is closer. Theoretically, GAT should be able to model the correlation between user and item and distinguish them with the help of attention mechanism. Unexpectedly, it pulls the nodes of users who consumed the same items closer. The reason I suggest is that GAT heavily affected by the initialization of ID embeddings, which is consistent with the finding [21]. On the contrary, our proposed method is stable and efficient to capture the correlation between users and items.

## 4 RELATED WORK

In this section, we review the existing work related to our research, including the recommendation with implicit feedback and GCN-based personalized recommendation.

### 4.1 Recommendation with Implicit Feedback

Comparing with the work focusing on explicit feedback [5, 17, 22], profiling the user from implicit feedback is more practical and challenge. Therefore, researchers shift to explore the user-item interaction from implicit feedback data.

To address the challenge of implicit feedback, the core is how to distinguish the negative instances from the positive ones. As such, Hu *et al.* [16] treated the all user’s unobserved item as negative instances and indicated the numerical value of implicit feedback as confidence. Besides, Rendle *et al.* [33] proposed Bayesian Personalized Ranking (BPR) method, which sampled negative instances from the user’s unobserved items to construct the triple of  $\langle \text{user}, \text{positive item}, \text{negative item} \rangle$  for pair-wise ranking. Comparing with the method assigning a uniform weight to each user’s unobserved item, He *et al.* [13] proposed to weight them based on item popularity and designed a model to efficiently optimize with variably-weighted item. Recently, Yang *et al.* [42] treated the items belonging to user’s high-order neighbors as positive instances and others as negative ones, which enriches the training set to optimize the parameters of proposed graph and matrix factorization (MF) combination model.

In terms of the multimedia personal recommendation, He *et al.* [11] extended BPR method and proposed Visual Bayesian Personalized Ranking (VBPR), which incorporated the visual information to improve the performance. Following BPR method, they used all user’s unobserved items as negative instances and performed the pair-wise ranking operation. Beyond exploring the positive and negative instances from implicit feedback, Chen *et al.* [4] designed Attentive Collaborative Filtering (ACF) model to capture item- and component-level implicit feedback in the multimedia recommendation.

Different from the existing studies, we propose to model the user preference to the item content and measure the similarity between them to discover the false-positive feedback from the historical records.

### 4.2 GCN-based Personalized Recommendation

Due to the effectiveness in representation learning, graph convolutional network are widely exploited to model the interactions between users and items for personalized recommendation. For instance, Berg *et al.* [2] formulated the recommendation task as a link prediction problem on graphs and utilized the graph convolutional operation to predict links between user and item. Based on differentiable message passing on the bipartite graph, they devised a graph auto-encoder framework. Nevertheless, the method is designed for the recommender system with explicit feedback data (*i.e.* ratings).

Towards the implicit feedback, Ying *et al.* [43] constructed a bipartite interaction graph according to users’ browsing records and developed a large-scale recommendation engine for image recommendation. On the constructed graph, the method jointly

conducts the graph convolutional operations and random walks to represent the users and items, which supercharges the efficiency on web-scale personalized recommendation. Similarly, Wang *et al.* [36] constructed the user-item graph, whose edges corresponded to implicit feedback. With their proposed neural graph collaborative filtering (NGCF) method, the collaborative signal conveyed by the edges and high-order connectivity explicit modelled and injected into each user and item embedding. More recently, the GCN-based model has been introduced into multimedia recommendation in implicit feedback settings. Wei *et al.* [40] constructed the modal-specific bipartite graph with implicit data to model the user preference in multiple modalities. They developed a multimedia recommendation framework, dubbed multimodal graph convolutional network (MMGCN), which represented the user preference in each modality with her/his directly and indirectly connected neighbors.

However, these GCN-based recommendation models ignore the effect of implicit feedback. Moreover, with iteratively performing the graph convolutional operations, the disruption of node representation, which is caused by distorting of graph structure, becomes worse. Against this issue, we propose to refine the user-item graph to eliminate its effect in the graph convolutional operations.

## 5 CONCLUSION AND FUTURE WORK

In this paper, we propose to solve the problem of implicit feedback towards the GCN-based recommendation method. Therefore, we develop a novel model, named Structure-Refined Graph Convolutional Networks, which yields a refined user-item interaction graph for graph convolutional operations. It identifies the false-positive feedback and prunes the corresponding noisy edge in the interaction graph. Empirical results on three public benchmarks demonstrate the efficiency of our proposed model.

To the best of our knowledge, this work is the first attempt to explore the disadvantage of the GCN-based recommendation caused by implicit feedback. Despite the state-of-the-art performance our proposed model achieved, we believe there is a long distance to solve the implicit feedback problem thoroughly. We attribute the issues caused by implicit feedback to the gap between user preference and behaviors. Beyond the user preference, the motivation why people prefer some items (*i.e.* user intent) is critical to estimate the user behaviors, but inefficiently explored. As such, in future work, we expect to study how to learn and leverage the user intents, in order to provide a high-quality personalized recommender system.

**Acknowledgments** This work is supported by the National Natural Science Foundation of China, No.:61802231, No.:U1936203, and No. U19A2079; the Innovation Teams in Colleges and Universities in Jinan, No.:2018GXRC014; and the Shandong Provincial Natural Science Foundation, No.:ZR2019JQ23. Also, this research is supported by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.



## REFERENCES

- [1] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of International Conference on Learning Representations*. 1–16.
- [2] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1–9.
- [3] Heng Chang, Yu Rong, Tingyang Xu, Wenbing Huang, Honglei Zhang, Peng Cui, Wenwu Zhu, and Junzhou Huang. 2020. A Restricted Black-box Adversarial Framework Towards Attacking Graph Embedding Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 1–6.
- [4] 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 *Proceedings of the International ACM SIGIR conference on Research and Development in Information Retrieval*. 335–344.
- [5] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan Kankanhalli. 2018. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *Proceedings of the World Wide Web conference*. 639–648.
- [6] Peng Cui, Wenwu Zhu, Dingyuan Zhu, Ziwei Zhang. 2019. Robust Graph Convolutional Networks Against Adversarial Attacks. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1399–1407.
- [7] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *Proceedings of the International Conference on World Wide Web*. 417–426.
- [8] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and statistics*. 249–256.
- [9] Will Hamilton, Zitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of International Conference on Neural Information Processing Systems*. 1024–1034.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.
- [11] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 144–150.
- [12] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 639–648.
- [13] Xiangnan He, Hanwang Zhang, Min Yen Kan, and Tat Seng Chua. 2016. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. In *Proceedings of the International ACM SIGIR conference on Research and Development in Information Retrieval*. 549–558.
- [14] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. 2017. CNN architectures for large-scale audio classification. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*. 131–135.
- [15] Jun Hu, Shengsheng Qian, Quan Fang, and Changsheng Xu. 2019. Hierarchical Graph Semantic Pooling Network for Multi-modal Community Question Answer Matching. In *Proceedings of the ACM International Conference on Multimedia*. 1157–1165.
- [16] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *IEEE International Conference on Data Mining*. 263–272.
- [17] Yupeng Hu, Chong Yang, Peng Zhan, Jia Zhao, Yujun Li, and Xueqing Li. 2019. Efficient continuous KNN join processing for real-time recommendation. *Personal and Ubiquitous Computing* (2019), 1–11.
- [18] Ziling Huang, Zheng Wang, Wei Hu, Chia-Wen Lin, and Shin'ichi Satoh. 2019. DoT-GNN: Domain-Transferred Graph Neural Network for Group Re-identification. In *Proceedings of the ACM International Conference on Multimedia*. 1888–1896.
- [19] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*. 1–16.
- [20] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of International Conference on Learning Representations*. 1–14.
- [21] Boris Knyazev, Graham W Taylor, and Mohamed Amer. 2019. Understanding Attention and Generalization in Graph Neural Networks. In *Advances in Neural Information Processing Systems*. 4204–4214.
- [22] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*. 305–314.
- [23] Fan Liu, Zhiyong Cheng, Changchang Sun, Yinglong Wang, Liqiang Nie, and Mohan Kankanhalli. 2018. User Diverse Preference Modeling by Multimodal Attentive Metric Learning. In *Proceedings of the ACM International Conference on Multimedia*. 1526–1534.
- [24] Jiawei Liu, Zheng-Jun Zha, Richang Hong, Meng Wang, and Yongdong Zhang. 2019. Deep Adversarial Graph Attention Convolution Network for Text-Based Person Search. In *Proceedings of the ACM International Conference on Multimedia*. 665–673.
- [25] Meng Liu, Liqiang Nie, Meng Wang, and Baoquan Chen. 2017. Towards Micro-Video Understanding by Joint Sequential-Sparse Modeling. In *Proceedings of the ACM International Conference on Multimedia*. 970–978.
- [26] Meng Liu, Liqiang Nie, Xiang Wang, Qi Tian, and Baoquan Chen. 2018. Online data organizer: micro-video categorization by structure-guided multimodal dictionary learning. *IEEE Transactions on Image Processing* 28, 3 (2018), 1235–1247.
- [27] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled Graph Convolutional Networks. In *Proceedings of International Conference on Machine Learning*. 4212–4221.
- [28] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the international conference on machine learning*. 3–9.
- [29] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the international conference on machine learning*. 807–814.
- [30] Liqiang Nie, Xiang Wang, Jianglong Zhang, Xiangnan He, Hanwang Zhang, Richang Hong, and Qi Tian. 2017. Enhancing Micro-video Understanding by Harnessing External Sounds. In *Proceedings of ACM Multimedia Conference on Multimedia Conference*. 1192–1200.
- [31] Arantxa Casanova, Adriana Romero, Pietro Liò, Yoshua Bengio, Petar Velićević, Guillem Cucurull. 2018. Graph Attention Networks. In *Proceedings of International Conference on Learning Representations*. 1–12.
- [32] Xufeng Qian, Yueting Zhuang, Yimeng Li, Shaoning Xiao, Shiliang Pu, and Jun Xiao. 2019. Video relation detection with spatio-temporal graph. In *Proceedings of the ACM International Conference on Multimedia*. 84–93.
- [33] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the conference on Uncertainty in Artificial Intelligence*. 452–461.
- [34] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*. 4077–4087.
- [35] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1–9.
- [36] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the International ACM SIGIR conference on Research and Development in Information Retrieval*. 165–174.
- [37] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1001–1010.
- [38] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, and Tat-Seng Chua. 2020. Reinforced Negative Sampling over Knowledge Graph for Recommendation. In *Proceedings of The Web Conference*. 99–109.
- [39] Yinwei Wei, Zhiyong Cheng, Xuzheng Yu, Zhou Zhao, Lei Zhu, and Liqiang Nie. 2019. Personalized Hashtag Recommendation for Micro-videos. In *Proceedings of the ACM International Conference on Multimedia*. 1446–1454.
- [40] 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 *Proceedings of ACM Multimedia Conference on Multimedia Conference*. 1437–1445.
- [41] Jiaxin Wu, Sheng-Hua Zhong, and Yan Liu. 2019. MvsGCN: A Novel Graph Convolutional Network for Multi-video Summarization. In *Proceedings of the ACM International Conference on Multimedia*. 827–835.
- [42] Jheng-Hong Yang, Chih-Ming Chen, Chuan-Ju Wang, and Ming-Feng Tsai. 2018. HOP-rec: high-order proximity for implicit recommendation. In *Proceedings of the ACM Conference on Recommender Systems*. 140–144.
- [43] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2019. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 974–983.
- [44] Yang Zhang, Fuli Feng, Chenxu Wang, Xiangnan He, Meng Wang, Yan Li, and Yongdong Zhang. 2020. How to Retrain Recommender System? A Sequential Meta-Learning Method. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1479–1488.