

Concept-Aware Denoising Graph Neural Network for Micro-Video Recommendation

Yiyu Liu¹, Qian Liu¹, Yu Tian¹, Changping Wang², Yanan Niu², Yang Song², Chenliang Li^{1†}

¹Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber

Science and Engineering, Wuhan University, Wuhan, 430072, China

{liuyiyu, littleliu}@whu.edu.cn; s.braylon1002@gmail.com; clee@whu.edu.cn

²Kuaishou Technology Co., Ltd., Beijing, 10010, China

{wangchangping, niuyan, yangsong}@kuaishou.com

ABSTRACT

Recently, micro-video sharing platforms such as Kuaishou and Tiktok have become a major source of information for people's lives. Thanks to the large traffic volume, short video lifespan and streaming fashion of these services, it has become more and more pressing to improve the existing recommender systems to accommodate these challenges in a cost-effective way. In this paper, we propose a novel concept-aware denoising graph neural network (named CONDE) for micro-video recommendation. CONDE consists of a three-phase graph convolution process to derive user and micro-video representations: *warm-up propagation, graph denoising and preference refinement*. A heterogeneous tripartite graph is constructed by connecting user nodes with video nodes, and video nodes with associated concept nodes, extracted from captions and comments of the videos. *To address the noisy information in the graph, we introduce a user-oriented graph denoising phase to extract a subgraph which can better reflect the user's preference*. Despite the main focus of micro-video recommendation in this paper, we also show that our method can be generalized to other types of tasks. Therefore, we also conduct empirical studies on a well-known public E-commerce dataset. The experimental results suggest that the proposed CONDE achieves significantly better recommendation performance than the existing state-of-the-art solutions.

CCS CONCEPTS

• Information systems → Recommender systems.

KEYWORDS

Micro-video Recommendation, Graph Neural Network, Graph Denoising

[†]Chenliang Li is the corresponding author. Work done when Yiyu, Qian and Yu were interns at Kuaishou.

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.

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8446-9/21/11...\$15.00

<https://doi.org/10.1145/3459637.3482417>

ACM Reference Format:

Yiyu Liu, Qian Liu, Yu Tian, Changping Wang, Yanan Niu, Yang Song, Chenliang Li. 2021. Concept-Aware Denoising Graph Neural Network for Micro-Video Recommendation. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21)*, November 1–5, 2021, Virtual Event, QLD, Australia. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3459637.3482417>

1 INTRODUCTION

Information overload has become an increasingly crucial challenge for today's world. In the past few years, micro-video sharing platforms like Kuaishou have harnessed a huge user base on a global scale. Users on these platforms can share micro-videos ranging from a few seconds to minutes with their friends and the public. Due to the significant cognitive load reduction empowered by visual communication, micro-videos have become the new time killer. As of early 2020, there are more than 300M daily active users (DAUs) and over 20B videos on the Kuaishou platform.

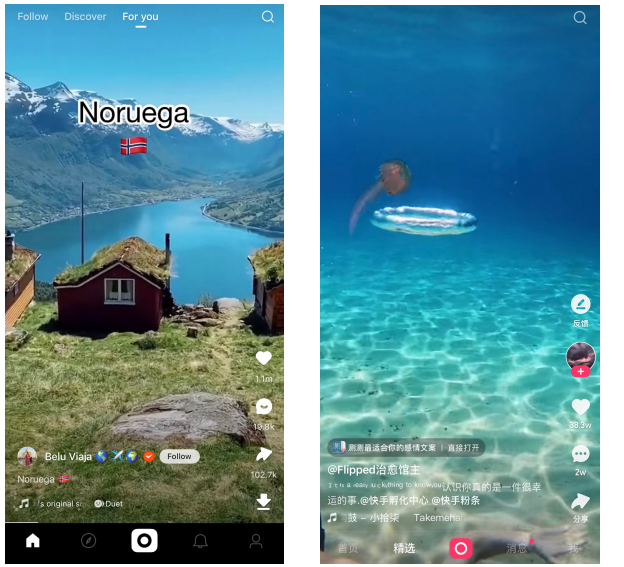
Faced with a massive upsurge of micro-video data traffic, the key techniques to alleviate the information overload problem is recommender systems, which aim to precisely rank items in terms of the user's preference. A myriad of recommendation algorithms have been proposed in the past, including sequential based recommendation [17, 18], graph-based recommendation [16, 34] and a range of CTR prediction models that are practical in industrial settings [7, 9, 22, 23, 33, 36].

Nevertheless, the following three unique characteristics in micro-video sharing platforms impede the existing recommendation techniques to deliver good performance:

C1: The existing multi-modal models that extract visual content are infeasible for micro-video sharing platforms due to the continuous and large traffic volume nature of the latter. On the other hand, exploiting the rich story expressed in a micro-video will help understand the user's preference to its maximum.

C2: Although micro-video platforms provide users with "like" and "comment" buttons to interact with the video, the vast majority of user actions are still just swiping up to the next video, leaving very sparse user feedback. Without an explicit user interaction, it is hard to tell whether the user really likes what she/he watches. Although we can utilize watching time to speculate the user's preference (i.e., analogous to click behavior), there would still be many false positives which hinder the effective preference learning for recommendation.

C3: The average lifecycle of a micro-video is extremely short. In our data analysis over user behaviors from a large scale micro-video



(a) The homepage of Kwai. (b) The homepage of Kuaishou.

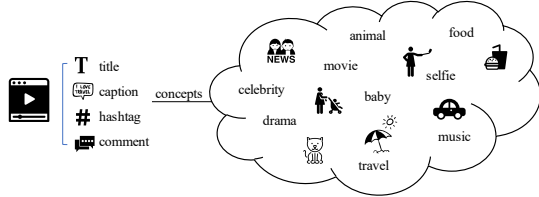


Figure 1: A snapshot of the popular micro-video platform kuaishou and its international version Kwai. The text information within a micro-video contains rich semantics, which can be extracted as millions of concepts in our work.

sharing platform, it is observed that the number of user interactions over a micro-video reduces sharply just two days later after its announcement¹. Moreover, most of the user behaviors fall on the micro-videos uploaded by a few Internet celebrities. Specifically, the micro-videos that have less than 10 user clicks comprise about 85% in our dataset. This number goes up to 96% when we use 50 as the cutoff threshold. These long-tail ones are rarely recommended. The feedback loop underlying the recommender system further aggravates this recommendation bias, which is called "Matthew Effect" [3].

All these characteristics together give rise to sparser user interactions and poor performance for micro-video recommendation, especially for long-tail ones. To this end, in this paper, we propose a novel **concept-aware denoising** graph neural network (named CONDE) to address the above challenges. In CONDE, we aim to build a marriage between textual information and graph neural network to support video content extraction and user preference learning. As illustrated in Figure 1, the caption and comments associated with a micro-video serve as a good proxy for the video content. Hence, to account for the rich semantics in a micro-video, we resort to extracting concepts mentioned in the captions and the comments

¹Due to the privacy policy of the company, we cannot release the lifecycle distribution here.

of micro-videos. Here, the concepts are defined as the named entities and semantic key phrases (e.g., "heartwarming", "classic HK movie"). By connecting the micro-videos to the associated concepts and users to their consumed micro-videos, we can form a tripartite heterogeneous graph, in which the concepts work as a backbone with rich semantic information.

It is intuitive that not all concepts of a micro-video can reflect all different users' preferences equally. Moreover, as aforementioned, there would be many false positives for user-video edges in the graph. Hence, we further introduce a personalized denoising procedure for each user when we derive her/his representation via graph convolution. Specifically, CONDE utilizes a three-phase graph convolution process to derive user and item representations. In the *warm-up propagation* phase, we firstly apply convolution by propagating semantic concept information to the related micro-videos and users through the aggregation order: concept \rightarrow micro-video \rightarrow user. Afterwards, we further perform another layer of convolution over the user-video edges to exploit the collaborative signals. With the user and item representations calculated by the warm-up phase, *graph denoising* phase aims to extract a subgraph for a given user where the noisy micro-videos, concepts and other users in the neighborhood are removed in a breadth-first search fashion. It is expected that the resultant subgraph could reflect the user's preference more precisely. At last, in *preference refinement* phase, we again perform the same convolution as in the warm-up phase over the subgraph to derive the user representation.

The extensive experiments over a large scale internal micro-video dataset and a traditional E-Commerce dataset in different languages demonstrate that the proposed CONDE achieves significantly superior recommendation performance to the existing state-of-the-art technical alternatives. In summary, the main contributions of this paper are as follows:

- We fully exploit textual information to support video content extraction. The rich semantics in micro-videos demonstrate great potential in representing the user's preference.
- We propose a denoising phase for graph neural networks to help recommendation systems, especially those in a streaming fashion, get rid of noisy information and catch user's highly dynamic preference more precisely.
- To the best of our knowledge, this is the first attempt in recommendation systems to study the heterogeneous graph neural network with a denoising purpose.

2 RELATED WORK

Since our work is related to micro-video recommendation, graph-based recommendation and denoising for graph and recommendation, we therefore mainly focus on reviewing existing methods in these three lines.

2.1 Micro-Video Recommendation

Different from item purchase in E-Commerce sites, micro-videos contain much more storylines. To enable better semantic learning for micro-videos, MMGCN [32] incorporates multi-modal information into a collaborative filtering framework. It captures modal-specific user preferences by separately constructing a user-item

bipartite graph for each modality. ALPINE [11] also designs a graph-based sequential network for recommendation tasks, which can better model a user’s diverse and dynamic interest. THACIL [2] proposes a hierarchical attention mechanism for modeling both short-term and long-term behaviors. Though encouraging performance is achieved by these efforts, they all require visual feature extraction. However, this heavy treatment could be too expensive to be adopted for a real-world application.

2.2 Graph-based Recommendation

Recent years have witnessed the superior performance of graph neural networks in many fields that require network embedding and structure modeling. In practice, GNN methods have more expressive capacity than traditional feature-based methods. The key idea of GNN is to recursively aggregate information from local neighborhoods. Graph Convolution Networks [10] is the first work that introduces the convolution operation over a network structure, which gathers the information from source nodes’ one-hop neighbors via the neighborhood aggregation, and achieves message passing by stacking multiple GCN layers. Graph Attention Network (GAT) [27] introduces the attention mechanism into GNNs, which enables the model to learn the importance of different neighbors during the information aggregation. These solutions are mainly devised for homogeneous graphs where all nodes or edges are of the same type. To deal with heterogeneous graph that contains different types of nodes and edges, HAN [31] extends the original GAT model with a semantic attention layer to learn the importance of different meta-paths. The major idea underlying these works has been widely adopted for different kinds of recommendation tasks.

Specifically, a heterogeneous graph is generally built by considering users, items and relevant entities (e.g., side information) in a knowledge graph as nodes, and users’ behaviors like click, purchase, add-to-cart and other semantic relations as edges. Based on the real recommendation scenario, [1, 4, 5, 35] extend the recommendation framework as a heterogeneous graph modeling mainly by aggregating the neighborhood information via meta-paths. GraphSAGE[6] extends the standard graph convolution network to the inductive setting, and utilize batch training and neighborhood sampling to support recommendation on large-scale graphs. AGNN [19] designs an attribute graph based on user-item interactions and utilizes VAE to learn the distribution of attributes for cold-start users/items. PinSAGE [34] combines random walks and graph convolution layers to aggregate and propagate neighborhood information. MG-BERT [12] utilizes the transformer architecture on the item homogeneous graph to pretrain the item representation, including graph structure reconstruction and masked node feature reconstruction. With rich semantic information in a knowledge graph, many efforts are devoted to explore auxiliary connections between users and items, therefore making recommendations more accurate and explainable. RippleNet [28] is a path-based method that first assigns entities in the KG with initial embeddings and then samples ripple sets from the KG based on the user’s historically clicking items. It uses attention networks to simulate user preferences on sampled ripple sets to represent a user. Some of its extensions [21, 24, 30] focus on using the embedding propagation mechanism on the item Knowledge

Graph. DKN [29] is another representative work that incorporates knowledge graph representation into news recommendations.

2.3 Graph Denoising

Performing data denoising in a task-dependent fashion is a promising strategy to alleviate the adverse impact of noisy information. For example, kicking out task-irrelevant edges have been validated to enhance the performance of node classification [13, 37]. Very recently, [20] chose to remove the irrelevant historical records for better sequential recommendation. [26] perform knowledge pruning iteratively by removing irrelevant triples covered by the knowledge graph for better news recommendation.

3 METHOD

In this section, we present a concept-aware denoising graph neural network for micro-video recommendation. Specifically, the proposed CONDE aims to derive the representations for users and micro-videos by propagating the conceptual semantics to both relevant users and micro-videos. Illustrated in Figure 2, CONDE consists of three phases: *warm-up propagation*, *graph denoising* and *preference refinement*. In the following, we describe each phase following their usage order in CONDE.

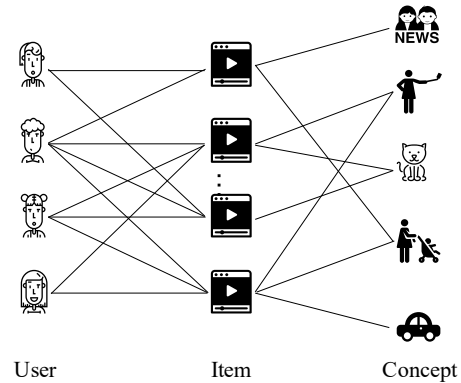


Figure 2: The tripartite heterogeneous graph which is composed of three disjoint vertex sets: users, items, and concepts, such that no two graph vertices within the same set are adjacent.

3.1 Warm-Up Propagation

Firstly, after concept extraction for micro-videos, we form a tripartite heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} and \mathcal{E} are the node set and the edge set respectively. A node $o \in \mathcal{V}$ could be a user u , a micro-video m or a concept c . Moreover, there are two kinds of edges in \mathcal{E} , which connect a user with a micro-video when the user clicks the latter, and a micro-video with a concept when the concept is extracted from the micro-video. Figure 2 illustrates an example of this tripartite heterogeneous graph. Since we plan to inject the conceptual information into the user and item representations, at the beginning, we perform a series of graph convolution operations in terms of concepts in \mathcal{V} .

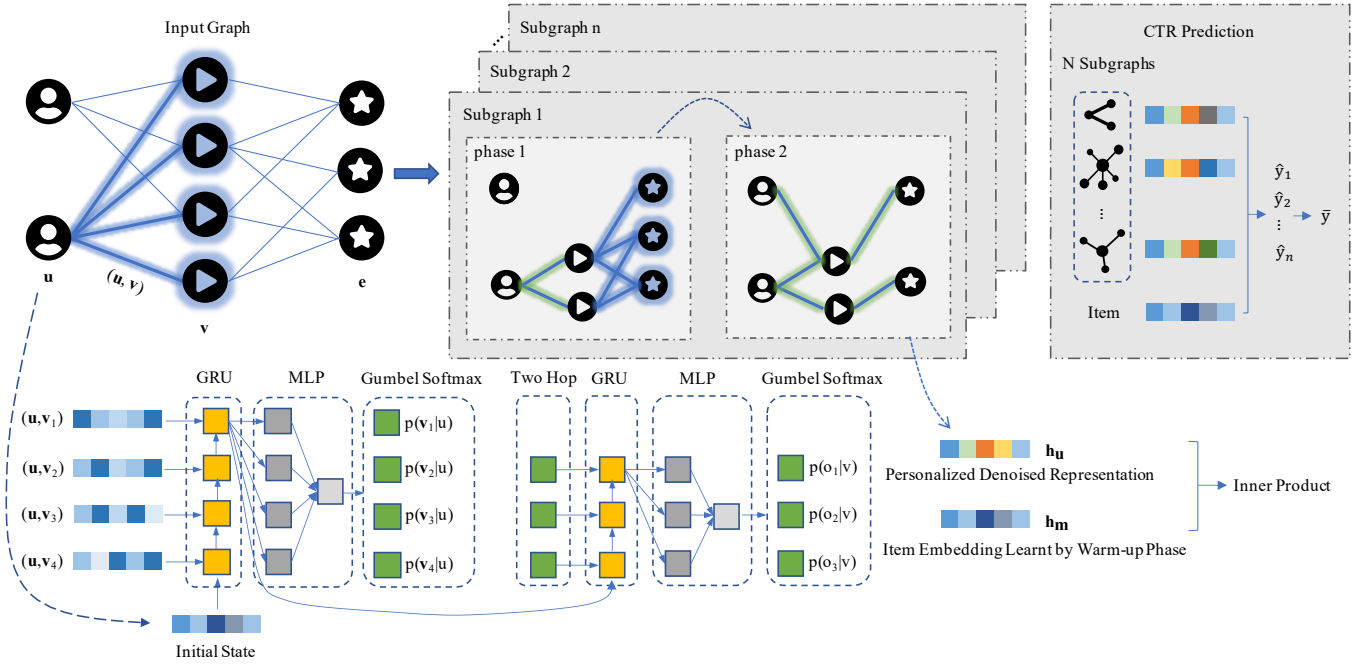


Figure 3: The network architecture of our proposed CONDE.

Specifically, given a micro-video m with a set of concept neighbors C_m , the hidden feature vector \mathbf{h}_m of micro-video m can be derived by aggregating the embeddings of its concept neighbors: $\mathbf{h}_m = \text{AGG}(\mathbf{e}_m, \{\mathbf{e}_c, \forall c \in C_m\})$, where $\mathbf{e}_m, \mathbf{e}_c$ are the embedding vector for micro-video m and its concept neighbor c respectively. Given the simplicity and effectiveness, we choose graph attention network[27] to instantiate $\text{AGG}(\cdot)$ function as follows:

$$\mathbf{h}_m = \sigma \left(\sum_{c \in C_m} \alpha_c \mathbf{W} \mathbf{e}_c \right) \quad (1)$$

$$\alpha_c = \frac{\exp(\mathbf{a}^\top [\mathbf{e}_m \| \mathbf{e}_c])}{\sum_{i \in C_m} \exp(\mathbf{a}^\top [\mathbf{e}_m \| \mathbf{e}_i])} \quad (2)$$

where α_c is the attention weight indicating the importance of concept c , σ is the LeakyReLU activation and $\|$ is the vector concatenation operation. It is expected that \mathbf{h}_m could encode semantic information regarding the content of micro-video m .

Following similar operation, we then derive the hidden preference vector \mathbf{h}_u of user u by using micro-video neighbors \mathcal{M}_u of the latter: $\mathbf{h}_u = \text{AGG}(\mathbf{e}_u, \{\mathbf{h}_j, \forall j \in \mathcal{M}_u\})$, where \mathbf{e}_u is the embedding vector of user u , and \mathbf{h}_j is the hidden feature vector of micro-video neighbor j derived in Equation 1. It is clear now that we propagate the semantic information first from concepts to micro-videos, and then from micro-videos to users.

So far, the whole propagation process is purely concept driven. Note that not all micro-videos are well expressed by their concept neighbors. Plus, the collaborative signals are not exploited during this process. Hence, we then apply another convolution operation for micro-videos by aggregating their user neighbors back. Specifically, the hidden feature vector \mathbf{h}_m of micro-video m is updated with its user neighbor set \mathcal{U}_m : $\mathbf{h}_m = \text{AGG}(\mathbf{h}_m, \{\mathbf{h}_u, \forall u \in \mathcal{U}_m\})$. In this

sense, we can propagate the conceptual semantics across the user-video bipartite part further, enriching the feature representation of micro-videos, especially the long-tail ones.

3.2 Graph Denoising

It is intuitive that different users would be interested in different aspects of the micro-video content. Plus, there are inevitably some noisy concepts and false clicks to distract the effective representation learning for users. Although we have an attention mechanism in Equation 2, the irrelevant and noisy information still complicates the learning process. Here, we introduce a user-oriented denoising process to filter the micro-videos and the concepts in a breadth-first search fashion.

Specifically, we adopt a gated recurrent unit (GRU) as the compositor to firstly identify the relevance between user u and each of her micro-video neighbors as follows:

$$\mathbf{f}_{u,m} = \text{GRU}(\mathbf{h}_u, \mathbf{h}_m), \forall m \in \mathcal{M}_u \quad (3)$$

where \mathbf{h}_u works as the initial state of GRU, and $\mathbf{f}_{u,m}$ encodes the relevant information. Then, we perform the neighbor denoising as a sampling without replacement to retain only n micro-video neighbors: $\mathcal{M}_u^* = \text{Den}(\{\mathbf{f}_{u,m}, \forall m \in \mathcal{M}_u\})$ and $|\mathcal{M}_u^*| = n$. These n micro-videos are expected to convey the user's preference more precisely. In detail, we can utilize a fully-connected layer with a softmax function to derive the likelihood of retaining each micro-video neighbor m as follows:

$$s_{u,m} = \frac{\exp(\mathbf{w}^\top \mathbf{f}_{u,m})}{\sum_{i \in \mathcal{M}_u} \exp(\mathbf{w}^\top \mathbf{f}_{u,i})} \quad (4)$$

where \mathbf{w} is the parameter vector for graph denoising. Note that instead of attention mechanism, the denoising process produces

discrete selections, which are not differentiable for model learning. Hence, we choose Gumbel-Softmax[8] to instantiate $Den(\cdot)$ for differentiable discrete sample generation:

$$\pi_{u,m} = \frac{\exp((\log(s_{u,m}) + \epsilon_m)/\tau)}{\sum_{i \in \mathcal{M}_u} \exp((\log(s_{u,m}) + \epsilon_m)/\tau)} \quad (5)$$

where $\epsilon_* = -\log(-\log(x))$ and x is i.i.d sampled from Uniform(0, 1). The temperature parameter τ controls sharpness of the likelihood distribution. When τ is small, Equation 5 produces a multi-modal distribution. On the contrary, when τ is large, the resultant distribution is nearly equivalent to a uniform one.

Afterwards, we continue the above denoising process for the user's two-hop neighbors via each micro-video m in \mathcal{M}_u^* . Note that we only consider the concept of neighbors of m : $\mathcal{N}_{u,m} = \mathcal{C}_m$. We have tried mixing user and concept neighbors in this second phase, which turns out to be inferior to using concept nodes only. We speculate that putting different types of nodes in GRU simultaneously could complicate the model learning, since heterogeneous information is difficult to be fused together.

Then, we still utilize the same compositor in the first-hop denoising to calculate the relevance information:

$$\mathbf{f}_{u,m,v} = GRU(\mathbf{f}_{u,m}, \mathbf{h}_v), \forall v \in \mathcal{N}_{u,m} \quad (6)$$

where \mathbf{h}_v is the representation of two-hop neighbor v . Here, the concept embedding \mathbf{e}_c is used (i.e., $\mathbf{h}_v = \mathbf{e}_c$). Also, when we consider the neighboring users together, when v is user u' , the hidden preference vector $\mathbf{h}_{u'}$ is used instead (i.e., $\mathbf{h}_v = \mathbf{h}_{u'}$). Then, we apply the similar denoising process for $\mathcal{N}_{u,m}$ and obtain n two-hop neighbors for user u via micro-video m : $\mathcal{N}_{u,m}^* = Den(\{\mathbf{f}_{u,m,i}, \forall i \in \mathcal{N}_{u,m}\})$ and $|\mathcal{N}_{u,m}^*| = n$.

We can see that the whole denoising process is performed sequentially by starting with each user. Since the GRU-based compositor is initialized with the user preference vector of the warm-up phase, the user's neighborhood in \mathcal{G} is reshaped to express the user's preference more precisely in a subgraph \mathcal{G}_u^* , where the nodes are $\{u, \mathcal{M}_u^*, \mathcal{N}_{u,m}^*, \forall m \in \mathcal{M}_u^*\}$, and the corresponding edges between them in \mathcal{G} are remained².

3.3 Preference Refinement

Now, it is straightforward to refine the hidden preference representation for each user based on the corresponding \mathcal{G}_u^* . Specifically, for each micro-video m in \mathcal{M}_u^* , we firstly refine the hidden feature vector \mathbf{h}_m^* for m : $\mathbf{h}_m^* = AGG(\mathbf{e}_m, \{\mathbf{h}_v, \forall v \in \mathcal{N}_{u,m}^*\})$. Here, as to \mathbf{h}_v , we adopt the same setting as in Equation 6. Then, we refine the hidden preference vector \mathbf{h}_u^* : $\mathbf{h}_u^* = AGG(\mathbf{e}_u, \{\mathbf{h}_m^*, \forall m \in \mathcal{M}_u^*\})$. At last, the ranking score for a micro-video m w.r.t. user u is calculated as follows:

$$\hat{y}_{u,m} = \text{sigmoid}(\mathbf{h}_u^{*\top} \mathbf{h}_m) \quad (7)$$

where \mathbf{h}_m is the hidden feature vector derived in the warm-up phase for micro-video m .

²Note that we can continue the denoising process for three-hop neighbors and more. The computation cost is very high hence we leave it for further work.

3.4 Model Optimization

Besides the embedding vectors and parameters for graph convolution, the key of CONDE is to perform graph denoising in a personalized way to enhance the user preference learning. Following the work in [37], for each user u , we can generate k subgraphs $\{\mathcal{G}_{u,1}^*, \dots, \mathcal{G}_{u,k}^*\}$ by repeating the denoising process k times. After that, we calculate the cross-entropy loss with each subgraph $\mathcal{G}_{u,i}^*$ as follows:

$$\mathcal{L}_{u,m,i} = y_{u,m} \log(\hat{y}_{u,m}) + (1 - y_{u,m}) \log(1 - \hat{y}_{u,m}) \quad (8)$$

where $y_{u,m}$ is the ground truth of micro-video m clicked by user u , $\hat{y}_{u,m}$ is the corresponding ranking score calculated with subgraph $\mathcal{G}_{u,i}^*$. Finally, the total loss function is as follows:

$$\mathcal{L} = \sum_u \sum_m \sum_i \mathcal{L}_{u,m,i} + \lambda \|\Theta\|^2 \quad (9)$$

where $\|\Theta\|^2$ represents $L2$ regularization over model parameters and λ is the corresponding coefficient.

Discussion. After model training, we follow the order of *warm-up* \rightarrow *graph denoising* \rightarrow *preference refinement* for each user and generate the corresponding subgraph \mathcal{G}_u^* and user preference vector \mathbf{h}_u^* . We then maintain these preference vectors in memory for online service. There is no need to re-run the graph denoising and preference refinement each time, since we believe that the learnt model is effective in capturing the user's preference. The marginal change in the subgraph structure would introduce ignorable variations. Actually, all micro-video hidden feature vectors could be reused in the warm-up propagation phase for each user, by calculating \mathbf{h}_m beforehand.

4 EXPERIMENTS

We evaluate our proposed CONDE on two real-world datasets, hoping to answer the following three research questions: **RQ1** How does CONDE perform compared with state-of-the-art graph-based methods? **RQ2** How do different components affect CONDE's performance? **RQ3** Can CONDE provide reasonable explanations about user preferences towards items?

In what follows, we first present the experimental settings in detail, (i.e., the datasets, baselines, evaluation protocols, and the parameter settings), then compare our model's performance with state-of-the-art methods, followed by answering the above three questions.

4.1 Dataset Description

Micro-Video dataset is collected from a popular large scale Chinese micro-video sharing platform. It serves to record and share videos taken by a user and matches content to targeted consumers with the help of a recommender system. The recommendation model serves in a streaming setting, where users can swipe up on the touch screen to watch the next video one by one.

To be as consistent as possible with the real-world online training setting, we construct a dataset by randomly sampling 100K users and their watched micro-videos over a period of two days. After filtering the users and micro-videos with no click behavior, the dataset consists of 82,193 users and 784,863 clicked micro-videos,

Table 1: Statistics of the datasets.

Dataset	#User	#Item	#Concept	#Record
Micro-Video	82,193	784,863	415,796	9,455,804
Amazon Electronics	192,403	63,001	185,894	1,612,381

which is referred to as *Micro-Video* dataset in the rest of this paper. The user behaviors in first-day are used for training. We then shuffle and split the second-day’s data equally for validation and testing respectively.

Amazon Electronics [15] is a widely used benchmark whose task is to predict whether a user will write a review for a target item given her historical reviews (in English). Initially we wanted to find a public micro-video recommendation dataset in English but failed. So we have to turn to data from other fields. Surprisingly, CONDE works out quite well on other types of tasks and thus proving its robustness. Here, we select the first 70% of historical interactions as the training set. The last 10% interactions are used as the testing set. The remaining 20% is used as the validation set for hyper-parameter selection. We refer to this benchmark as the *Amazon* dataset.

4.2 Data Preprocessing

As for the Micro-Video dataset, we build a concept inventory by merging several proprietary knowledge bases and internal video tagging systems. Specifically, the resultant concept inventory contains about 2.32M named entities and semantic key phrases, covering a wide spectrum of topics in Micro-Video. The average length of a concept is 4. The concepts are extracted from textual information of a micro-video including comment, caption and hashtag by performing a longest match. As the user generated content contains a lot of Internet buzzwords and slangs, we filter the corpus first by removing stop words, punctuation and emoji. After concept extraction, we calculate the TF-IDF score for each concept and micro-video pair, and retain the informative ones with a threshold filtering. In total, we obtained 415,796 unique concepts. For the Amazon dataset, we utilize the TextBlob toolkit³ to extract the noun phrases as the concepts from the reviews written for each item. Similarly, a TF-IDF weighting scheme is adopted to filter noisy concepts. The detailed statistics of the two datasets are reported in Table 1.

4.3 Baselines.

- **GraphSage** [6] is a state-of-the-art method of GCN that aggregates neighborhood information into center nodes. In particular, it considers the structure information as well as the distribution of node features in the neighborhood. For a fair comparison, we form two homogeneous graphs based on user-item interaction.
- **PinSage** [34] is designed to employ GraphSAGE on user-item graph, which performs efficient, localized convolutions by sampling the neighborhood around a node and dynamically constructing a computation graph. In this work, we

employ two graph convolution layers as suggested in the user-item interaction graph.

- **GAT** [27] introduces a multi-head attention between each node and its neighbors to calculate aggregate coefficients.
- **HAN** [31] is a heterogeneous graph neural network with attention mechanism based on GAT, which deals with heterogeneous nodes via different meta paths to aggregate neighborhood information. We modify the model for top-N recommendation. In detail, we make use of four types of meta-paths: user-item-user, user-item-concept-item-user, item-concept-item, and item-user-item. Therefore we construct four homogeneous graphs from the original graph for model learning.
- **NIA-GCN** [25] models the interactions between neighbors with element-wise products by a bi-linear neighborhood aggregator. The Euclidean distance between users and items with their positive neighbors are added in loss function. The experimental setting is the same as the original NIA-GCN paper. This is the up-to-date state-of-the-art solution in the paradigm of GNN.

For those network embedding models (*i.e.*, GraphSage, GAT and HAN), we also utilize Equation 7 to calculate the ranking score for a fair comparison.

4.4 Experimental Settings

With the extracted concepts, a heterogeneous concept-aware graph is built by connecting the items to the associated concepts and users to their consumed items. Recall that we aim to propagate semantic concept information via graph convolution to represent users and micro-videos for preference learning. In the warm-up propagation phase, we derive coarse-level representations for micro-videos and users following the aggregation order of concept \rightarrow micro-video \rightarrow user. Note that the concept-aware graph is not perfect but contains lots of noisy information. At last, we again perform the same convolution as in the warm-up propagation phase over the subgraph to refine the user representation for recommendation.

Evaluation Metrics. For performance evaluation metrics, we choose AUC (Area Under Curve), HIT@5, NDCG@5, MAP@5 which are widely adopted in many related works [20, 26]. Note that we use UAUC here, which means AUC score for each user, and we calculate the mean of all users’ AUC scores as the final result. Meanwhile, We treat all unobserved interactions as the negative instances when reporting performance. It is worthwhile to mention that the positive and negative instances are highly imbalanced. As for model training, the ratio of true and negative samples in real production scenarios is approximate to 1:1. Hence, we also choose the same number of unobserved items for each user in evaluation.

Hyperparameter Settings. Following PinSage [34], we perform mini-batch training and edge sampling for better training efficiency. According to the average number of concepts that an item contains, we randomly sample a subgraph for training by randomly picking p neighbors in every epoch. We set $p = 40$ for the Micro-Video dataset and $p = 100$ for the Amazon dataset after grid search. The proposed CONDE will select a fixed number of neighbors out of this subgraph

³<https://github.com/sloria/TextBlob>

Table 2: Performance comparison between methods on the Micro-Video and Amazon dataset. The * indicates the best performance of the baselines. The best results are highlighted in boldface.

Method	Micro-Video				Amazon			
	AUC	NDCG@5	HIT@5	MAP@5	AUC	NDCG@5	HIT@5	MAP@5
PinSage	0.7254	0.8922	0.2772	0.2610	0.6957	0.7688	0.7185	0.6109
GraphSage	0.7139	0.8847	0.2750	0.2580	0.6961	0.7703	0.7204	0.6125
NIA-GCN	0.7203	0.8657	0.2717	0.2517	0.6903	0.7647	0.7174	0.6049
GAT	0.7345*	0.9102*	0.2811*	0.2668*	0.7009*	0.7734*	0.7210*	0.6162*
HAN	0.7251	0.8988	0.2782	0.2630	0.6984	0.7724	0.7204	0.6144
CONDE	0.7952	0.9140	0.2890	0.2747	0.7064	0.7792	0.7266	0.6226

(i.e., $n < p$). For fair comparison, after grid search, we fix the embedding size and all hidden sizes to be 128 and 50 for the Micro-Video and the Amazon dataset respectively across different methods in comparison. As for CONDE, we firstly apply Word2Vec [14] to pre-train the word embeddings over the textual information of items in the Micro-Video dataset. Here, the textual information associated with a micro-video is merged as a single document. For the Amazon dataset, we use pre-trained GloVe word embeddings⁴. The concept embeddings are fixed as the averaged embedding of their constituent words, and are not further fine-tuned during the model training. Recall that the temperature parameter τ in Equation 5 could affect the sample probabilities significantly. The higher the temperature, the smoother the sampling probability is. Starting from a higher temperature, the model can gather information from more neighbors in the first few minibatches. In our experiments, the temperature starts at a high value and gradually anneal to a small one as follows:

$$\tau = \tau_0 \times \exp(-\eta x)$$

where τ_0 is the initial temperature to start with, η is the hyperparameter, and x is the number of minibatches. We set τ_0 to be 10, and η to be 0.0002 and 0.001 for the Micro-Video and the Amazon dataset respectively.

4.5 RQ1: Performance Comparison

Table 2 reports the overall performance compared with baselines. For each model, we run five times and report the average results. From the results, we make the following observations.

- CONDE consistently yields the best performance on all datasets. This result demonstrates the superiority of our concept-aware denoising framework for recommendation.
- It is no surprise that methods based on a bipartite graph like PinSage [34] and NIA-GCN [25] consistently yields poor performance on all evaluations, which indicates that learning user and item representations simply on a bipartite graph is not adequate.
- The results of GraphSage [6] show that homogeneous graphs with directly connected items and users can not make up for the negative impact of noisy information. Attention

mechanism contributes a lot to GAT [27] performance improvement and HAN [31] are more powerful in information utilization and aggregation with item-concept connection when constructing item-item, user-user graph homogeneous graph.

- However, according to our experiments, none of them ever achieves the best result, which verifies the advantages of the denoising optimization process of our proposed model.

Long-Tail Performance. As to Micro-Video dataset, we divide the entire item space into hot and long-tail items by click frequency. We set 50 as a threshold and find that long-tail items take up a percentage of 96.0% in the entire item space. However, the majority of user-item interactions in the test set are dominated by hot items, leaving only 34.48% records for long-tail ones to share. We split the test set into two parts according to click frequency= 50 and do the experiments on them separately. From Table 3 we can observe that the performance of each model for hot items is always much higher than that of long-tail items, which indicates the distributions of hot and long-tail items are inconsistent. What’s more, the improvements of CONDE over the baselines are remarkable. Therefore we can draw a conclusion that CONDE shows its strong robustness on both hot and long-tail items, and proves its ability to learn better features in the entire space for different datasets and applications.

Table 3: Performance comparison between the methods with CONDE and other baselines on Micro-Video dataset. “Hot” represents hot items in the test set, “Long-tail” represents long-tail items in the test set.

Method	AUC		NDCG@2		HIT@2		MAP@2	
	Hot	Long-tail	Hot	Long-tail	Hot	Long-tail	Hot	Long-tail
PinSage	0.7524	0.6231	0.9176	0.7135	0.2470	0.3332	0.2358	0.2929
GraphSage	0.7662	0.6436	0.9234	0.7298	0.2473	0.3378	0.2369	0.2985
GAT	0.7710	0.6526	0.9409	0.7134	0.2505	0.3406	0.2413	0.3018
HAN	0.7585	0.6419	0.9320	0.7107	0.2487	0.3366	0.2387	0.2970
NIA-GCN	0.7606	0.6354	0.9118	0.6694	0.2443	0.3307	0.2332	0.2883
CONDE	0.8331	0.6924	0.9445	0.7587	0.2552	0.3471	0.2552	0.3122

4.6 RQ2: Model Analysis

Ablation Study. We conduct ablation studies on the Micro-Video dataset for each proposed component to justify its effectiveness. For each setting, we run 5 times and report the average results.

⁴<https://github.com/stanfordnlp/GloVe>

Note that our graph is tripartite, which is composed of three disjoint vertex sets: users, items, and concepts, such that no two graph vertices within the same set are adjacent. Therefore we perform the denoising algorithm in a two-phase strategy. Starting from a user, GRU first picks some candidate micro-videos out of its neighbors in the first phase. Then for every remaining micro-video, the same GRU picks some candidate concepts out of its neighbors in the second phase. The whole process can be split into two phases, hence we can do some ablation studies as follows:

For better comparison, we split the denoising process into two phases and remove either phase with a random module, which randomly selects some micro-videos instead of denoising, to check if denoising operation really makes some difference.

Table 4: Ablation study on the Micro-Video dataset.

Method	Micro-Video			
	AUC	NDCG@5	HIT@5	MAP@5
random phase 1	0.7870	0.9106	0.2877	0.2729
random phase 2	0.7854	0.9090	0.2866	0.2714
random phase 1 & 2	0.7780	0.9087	0.2856	0.2704
denoising phase 1	0.7780	0.9042	0.2849	0.2689
denoising phase 2	0.7874	0.9106	0.2871	0.2718
denoising phase 1 & 2	0.7952	0.9140	0.2890	0.2747

From the results shown in Table 4, we make the following observations: 1) Cut out either denoising phase is detrimental to the final result. 2) The second phase of denoising plays a more important role. 3) The best performance appears when two phases work together, indicating that two phases should work together to enhance representation learning. This is reasonable since the user’s representation is refined in terms of the relevant micro-videos whose representations are also dependent on the relevant concepts. This pipeline convolution nature for user representation learning requires us to denoising the irrelevant information in each step.

Number of Remaining Neighbors. Given the graph to denoise, we choose a fixed number of nodes for information aggregation. To concentrate on crucial points, we mainly discuss the concept denoising in this paper. Note that we have also tested a different number of one-hop neighbors for each user, where we decide to pick 6 micro-videos for every user out of her 10 neighbors. Here, we test the number of remaining concepts amongst {6, 10, 20, 30} out of 40. The number of subgraphs is fixed to be four in this case.

From Table 5 we can draw a conclusion that if we leave too many neighbors, noisy information will still exist and harm the model training. Yet if we leave too few neighbors, the model is likely to be over-denoised and miss much important information. In general, this parameter is not very sensitive for model performance. We could suggest using a relatively small value to enhance computation efficiency.

Number of Subgraphs Apparently, the number of subgraphs is also an important parameter to CONDE. Similarly, we test the number of subgraphs amongst {2, 4, 6}, and Table 6 indicates that adding the number of subgraphs improves the performance further. However the improvement of evaluation metrics can not keep the same

Table 5: Performance over different k values.

Remaining Neighbors	Micro-Video			
	AUC	NDCG@5	HIT@5	MAP@5
n=6	0.7942	0.9082	0.2883	0.2733
n=10	0.7952	0.9140	0.2890	0.2747
n=20	0.7913	0.9112	0.2875	0.2723
n=30	0.7924	0.9131	0.2885	0.2739

pace with the resultant increasing training time. As the training cost is proportional to the number of subgraphs and the model can not obtain further remarkable performance gain by adding more subgraphs, we use $k = 4$ in this paper.

Table 6: Performance over different G values.

Number of Subgraphs	Micro-Video			
	AUC	NDCG@5	HIT@5	MAP@5
G=2	0.7898	0.9071	0.2866	0.2710
G=4	0.7952	0.9140	0.2890	0.2747
G=6	0.7941	0.9143	0.2887	0.2741

Impact of Different Temperatures. We also test CONDE’s sensitivity to different temperatures. Specifically speaking, we set different starting temperatures and annealing ratios, and the model performs best when $\tau = 10$ and $\eta = 2e - 4$ on Micro-Video dataset as shown in Table 7.

Table 7: Performance over different temperature settings.

Different Starting Temperature	Micro-Video			
	AUC	NDCG@5	HIT@5	MAP@5
10 ($\eta = 2e - 4$)	0.7952	0.9140	0.2890	0.2747
100 ($\eta = 5e - 4$)	0.7875	0.9103	0.2870	0.2718
1000 ($\eta = 1e - 3$)	0.7859	0.9098	0.2870	0.2717

4.7 RQ3: Explainability Analysis

We further investigate whether CONDE can filter out noise and retain more meaningful information. To better visualize the results, we pick some users and their corresponding tripartite subgraphs, with items and concepts connected to each other. As ID is not explicit to describe an item, we use each micro-video’s concept neighbors instead, to facilitate better understanding of its content.

In order to demonstrate the explainability of CONDE, we first choose two user pairs, each of which share at least one same relevant item after performing denoising, according to CONDE’s result. As demonstrated in Figure 4, the concepts of the item are split out by orange and blue lines are the relevant ones after 2-phase denoising for the two users respectively. Here we use the Micro-Video dataset and translate every concept in English for easy understanding. We can observe that although two users have the same relevant item,

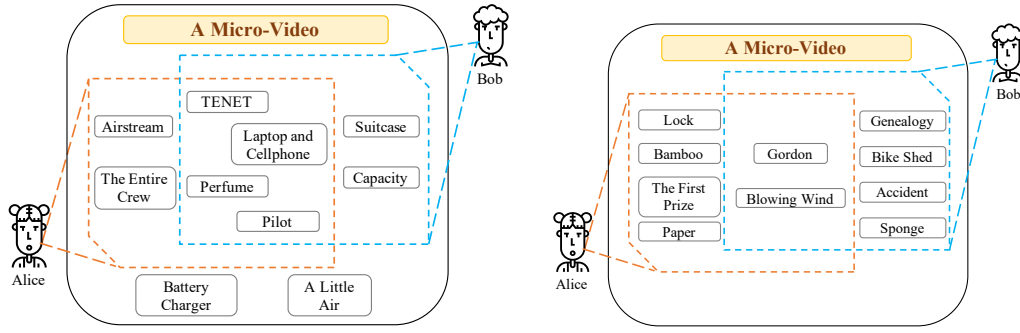


Figure 4: Examples of two users focusing on different points of the same item.

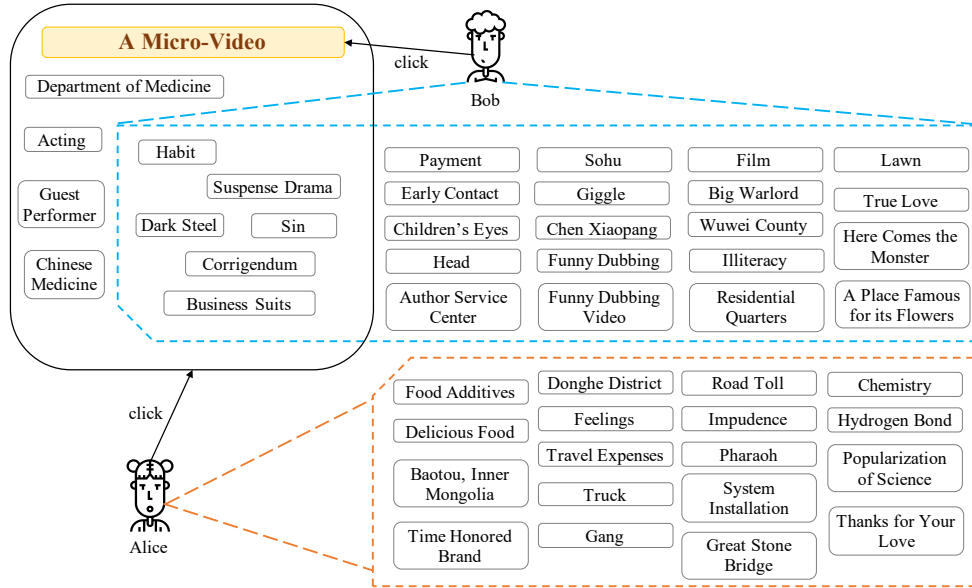


Figure 5: Two users interacted with the same item.

they focus on different points of interest. Figure 4 further indicates that a micro-video itself can include various points of interest.

What's more, we find another pair of users who interacted with the same item, and this item is deleted for one user but kept for the other by CONDE. We compare the related item and all remaining concepts of these two users in Figure 5. The result shows that there is an obvious divergence between their preferences. Overall, these results suggest that CONDE is able to discover informative yet relevant concepts out of noisy ones, besides enhancing the recommendation performance, but also support recommendation explainability.

5 CONCLUSION

In this paper, we propose a novel **concept-aware denoising** graph neural network (named CONDE) to address information redundancy challenges. To make full use of the textual information in micro-video recommendation scenarios, we extract concepts from captions and the comments associated with micro-videos. In this way, we can

form a tripartite heterogeneous graph by connecting user, micro-video, and concept nodes. As a user will interact with hundreds of micro-videos and a micro-video will receive lots of comments everyday, neither all micro-videos nor all concepts of a micro-video can reflect users' preference precisely. Hence, we propose a personalized denoising methodology to derive user and item representations via graph neural network. The extensive experiments over a large micro-video dataset and a traditional E-Commerce dataset in two different languages demonstrate the effectiveness of CONDE. By abundant ablation studies with different experiment settings, the proposed CONDE has proven its excellent potential to reflect the user's preference.

ACKNOWLEDGMENTS

This work was funded by Kuaishou and National Natural Science Foundation of China (No. 61872278). Chenliang Li is the corresponding author.

REFERENCES

- [1] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Representation learning for attributed multiplex heterogeneous network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1358–1368.
- [2] Xusong Chen, Dong Liu, Zheng-Jun Zha, Wengang Zhou, Zhiwei Xiong, and Yan Li. 2018. Temporal Hierarchical Attention at Category- and Item-Level for Micro-Video Click-Through Prediction. In *Proceedings of the 26th ACM International Conference on Multimedia*. 1146a–1153.
- [3] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. ESAM: Discriminative Domain Adaptation with Non-Displayed Items to Improve Long-Tail Performance. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 579a–588.
- [4] Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019. Metapath-guided heterogeneous graph neural network for intent recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2478–2486.
- [5] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. In *Proceedings of The Web Conference 2020*. 2331–2341.
- [6] Will Hamilton, Zhitaoying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of Advances in neural information processing systems*. 1024–1034.
- [7] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.
- [8] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *Proceedings of the 5th International Conference on Learning Representations*.
- [9] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 43–50.
- [10] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*.
- [11] Yongqi Li, Meng Liu, Jianhua Yin, Chaoran Cui, Xin-Shun Xu, and Liqiang Nie. 2019. Routing micro-videos via a temporal graph-guided recommendation system. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1464–1472.
- [12] Yong Liu, Susen Yang, Chenyi Lei, Guoxin Wang, Haihong Tang, Juyong Zhang, Aixin Sun, and Chunyan Miao. 2020. Pre-training Graph Transformer with Multimodal Side Information for Recommendation. *arXiv e-prints* (2020), arXiv:2010.2010.
- [13] 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 *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 779–787.
- [14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
- [15] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 188–197.
- [16] Aditya Pal, Chantat Eksombatchai, Yitong Zhou, Bo Zhao, Charles Rosenberg, and Jure Leskovec. 2020. PinnerSage: Multi-Modal User Embedding Framework for Recommendations at Pinterest. *arXiv preprint arXiv:2007.03634* (2020).
- [17] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2671–2679.
- [18] Pi Qi, Xiaoqiang Zhu, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, and Kun Gai. 2020. Search-based User Interest Modeling with Lifelong Sequential Behavior Data for Click-Through Rate Prediction. *arXiv preprint arXiv:2006.05639* (2020).
- [19] Tieyun Qian, Yile Liang, Qing Li, and Hui Xiong. 2020. Attribute Graph Neural Networks for Strict Cold Start Recommendation. In *Proceedings of the IEEE Transactions on Knowledge and Data Engineering*. 1–1.
- [20] Yuqi Qin, Pengfei Wang, and Chenliang Li. 2021. The World is Binary: Contrastive Learning for Denoising Next Basket Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 859–868.
- [21] Yanru Qu, Ting Bai, Weinan Zhang, Jianyun Nie, and Jian Tang. 2019. An End-to-End Neighborhood-Based Interaction Model for Knowledge-Enhanced Recommendation. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*. Article 8, 9 pages.
- [22] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *Proceedings of the 16th International Conference on Data Mining*. 1149–1154.
- [23] Steffen Rendle. 2010. Factorization machines. In *Proceedings of the 10th International Conference on Data Mining*. 995–1000.
- [24] Xiao Sha, Zhu Sun, and Jie Zhang. 2019. Attentive Knowledge Graph Embedding for Personalized Recommendation. *arXiv preprint arXiv:1910.08288* (2019).
- [25] Jianing Sun, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, Xiuqiang He, Chen Ma, and Mark Coates. 2020. Neighbor Interaction Aware Graph Convolution Networks for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1289a–1298.
- [26] Yu Tian, Yuhao Yang, Xudong Ren, Pengfei Wang, Fangzhao Wu, Qian Wang, and Chenliang Li. 2021. Joint Knowledge Pruning and Recurrent Graph Convolution for News Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 51–60.
- [27] Petar Velić, J. Kovač, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Li, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proceedings of the International Conference on Learning Representations*.
- [28] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippletNet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 417–426.
- [29] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *Proceedings of the 2018 World Wide Web Conference*. 1835a–1844.
- [30] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 950–958.
- [31] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *Proceedings of The World Wide Web Conference*. 2022–2032.
- [32] 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 the 27th ACM International Conference on Multimedia*. 1437–1445.
- [33] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: learning the weight of feature interactions via attention networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 3119–3125.
- [34] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 974–983.
- [35] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 793–803.
- [36] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In *Proceedings of the European Conference on Information Retrieval*. 45–57.
- [37] 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 *International Conference on Machine Learning*. 11458–11468.