X-2ch: Quad-Channel Collaborative Graph Network over Knowledge-aware Edges

Kachun Lo
Tohoku Universiy
Sendai, Japan
argent.lo.hk@gmail.com

Tsukasa Ishigaki
Tohoku Universiy
Sendai, Japan
isgk@tohoku.ac.jp

ABSTRACT

Carrying abundant side information, knowledge graph (KG) has shown its great potential in enriching the sparsity of collaborative filtering (CF) for recommendation. Although graph neural networks (GNNs) have been successfully employed to learn user preferences from KG and CF signals simultaneously, most models suffer from inferior performance due to their deficient designs, i.e., 1) formulating no distinction between users, items and KG entities, 2) confounding KG signals with CF signals and 3) completely neglecting the effects of edges, which is vital for graph information propagation.

In this paper, we propose a quad-channel graph model (**X-2ch**) to tackle these problems. First, rather than lodging KG entities on graph as nodes, X-2ch distills KG information and embeds them as edge attributes in a bi-directional manner to model the natural user-item interaction process. Second, X-2ch introduces a novel quad-channel learning scheme, including a collaborative user-item update and a CF-KG attentive propagation, to holistically capture the interconnectivity of users and items while preserving their distinct properties. Experiments on two real-world benchmarks show substantial improvement over the state-of-the-art baselines.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommender Systems, Knowledge Graph, Collaborative Filtering, Personalized Recommendation

ACM Reference Format:
Kachun Lo and Tsukasa Ishigaki. 2021. X-2ch: Quad-Channel Collaborative Graph Network over Knowledge-aware Edges. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463003

1 INTRODUCTION

Recommender systems (RSs) has been ubiquitously employed in modern online businesses, e.g. e-commerce and social networks. Among the literature, collaborative filtering (CF) is the most widely-used technique. Through mining similar user patterns, CF provides recommendations based on the assumption that similar users (e.g. watched alike movies) would be interested in resemblant items. Generally, CF models represent users and items as embedding vectors and formulate their interaction graph (as shown in Fig.1.a) through some specialized operations [4, 12]. However, since the user-item interactions lack explicit contextual information and rely solely on latent relations, CF models suffer from cold-start and sparsity problems. Recently, study of integrating knowledge graph (KG) into CF frameworks has caught huge attention of researchers. Through proper modeling, the rich *side information*, e.g. user/item attributes, provided by KG (illustrated in Fig.1.b) can help RSs to learn and understand users' preferences better towards personalized recommendations.

To effectively fuse the heterogeneous information of KG signals and CF signals, graph neural networks (GNNs) naturally becomes the primary method due to its flexibility and capability of capturing long-range (indirect) user-item relations. Existing graph-based methods can be categorized into 2 types: 1) path-based methods [1, 16, 22], which compute proximity between users and items by mining patterns of paths within KG entities, and 2) propagation-based methods [15, 18, 21], which merge CF graph (user-item interactions) and KG graph as a unified graph and "aggregate" information for each node from its neighbors. Although these graph-based KG-enhanced models have shown improvement over classic CF models, they still suffer from the following **limitations**:

(1) Formulating no distinction between users, items and KG entities. Most models embed these distinct subjects as identical "nodes" on graph, regardless the intrinsic differences between them. Moreover, as KG being used, distinctly modeling of users and items becomes a necessity, since their differences are amplified in the context of KG, where items have limited KG attributes (e.g. written by 1 author) and are passively chosen by users, while users normally have complex interests in many different KG entities (e.g. loving 10 movie directors) and are initiative.
(2) Confounding KG signals with CF signals when propagating information on graph. These two signals convey very different yet complementary information [3, 6]. KG delivers explicit contexts and attributes of items, which are especially helpful for cold-start problem. For instance, if a user has only watched *"Titanic"*, then a KG-based system can safely suggest all other movies directed also by *"James Cameron"*. On the other hand, CF aims at finding "latent patterns" among users, which is more capable of giving diverse recommendations (e.g. not always from a same director) and capturing subtle users' preferences (e.g. preferring specific plot while indifferent to the movie's cast). Thus, properly dealing with the 2 signals may lead to substantial improvement.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

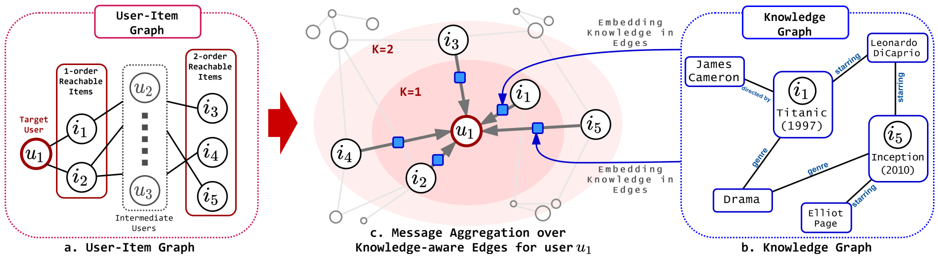

Figure 1: An Example of the quad-channel aggregation for a specific user u_1.

(3) Neglecting the effects of edges. Although "attributed edge" has recently been studied for some applications, e.g. few-shot learning and node importance [7, 8, 11, 14], current KG-based recommendations only employ undirected and plain edges with the sole purpose of linking nodes. The exceptional flexibility of graph model is greatly underutilized when the edges are completely neglected.

Given these observations, we propose **X-2ch**, a quad-channel CF graph model propagating information over knowledge-aware edges. The two most significant ideas behind X-2ch are: the knowledge-aware edges and a quad-channel aggregation mechanism. Specifically, rather than lodging KG entities on graph as nodes, X-2ch distills KG information and embeds them as edge attributes to model the natural user-item interaction process. Moreover, the "loaded edges" are bi-directional, carrying different information according to the corresponding graph flows. Furthermore, the quad-channel aggregation propagates information to user and item nodes separately via 2 tracks, CF-channel and KG-channel aggregation.

To sum up, the main contributions of our work are as follows:

- We propose a novel graph-based model for recommendation, X-2ch, which is capable of learning representative user and item embeddings by distributing information over knowledge-aware edges through a quad-channel mechanism.
- To the best of our knowledge, this is the first work to explicitly embed KG attributes in edges in a bi-directional style for recommendation.
- Comprehensive experiments, conducted on two real-world benchmarks, demonstrate the superiority of X-2ch to state-of-the-art baselines.

2 METHODOLOGY

2.1 Notations and Overall Structure

X-2ch is designed for **recommendation task** of the ubiquitous implicit feedback [10], where user-item interactions are binary (e.g. clicks or likes). Given a user $u \in U$ and an item $i \in I$, X-2ch outputs the probability $\hat{y}_{ui}$ measuring u's preference to i.

Specifically, X-2ch utilizes two graphs as inputs. **User-Item Bipartite Graph (UIG)**, with abundant CF signals, consists of user-item interactions $<u, y_{ui}, i>$, where $u \in U, i \in I$ and $y_{ui} \in \{0, 1\}$ is the indicator of the historical interaction. **Knowledge Graph (KG)**, rich of item attributes information, comprises knowledge triplets $<h, r, t>$, where the head entity h denotes an item, the relation r is an attribute type and the tail entity t is the attribute value.

Algorithm 1: X-2ch Embeddings Generation Algorithm

Input : UIG $<u, y_{ui}, i>$ and KG $<h, r, t>$.
Output: Users and Item embeddings
for $l = 1, \cdots, L$ **do**
  `// User Node Update`
  for $u = 1, \cdots, |U|$ **do**
    $e_u^{cf(l+1)} \Leftarrow \mathrm{CFAgg}(e_u^{cf(l)}, e_i^{cf(l)})$; `// CF-channel`
    $e_u^{kg(l+1)} \Leftarrow \mathrm{KGAgg}(e_i^{cf(l)}, e_i^{kg(l)}, e^E)$; `// KG-channel`
  end
  `// Item Node Update`
  for $i = 1, \cdots, |I|$ **do**
    $e_i^{cf(l+1)} \Leftarrow \mathrm{CFAgg}(e_i^{cf(l)}, e_u^{cf(l)})$; `// CF-channel`
    $e_i^{kg(l+1)} \Leftarrow \mathrm{KGAgg}(e_u^{cf(l)}, e_u^{kg(l)}, e^E)$; `// KG-channel`
  end
end
`// Final User and Item Embeddings`
$e_u \Leftarrow e_u^{cf(L)} || e_u^{kg(L)}$;
$e_i \Leftarrow e_i^{cf(L)} || e_i^{kg(L)}$;

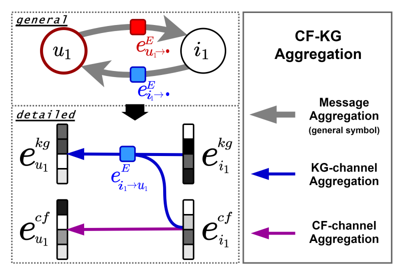

Figure 2: Example of the CF and KG-channel aggregation.

For example, $<Titanic, genre, Drama>$ indicates that the movie *Titanic* is of the *genre* "*Drama*". In Figure 1 (a. and b.), examples from the perspective of a user u_1 are shown.

The **overall structure of X-2ch** (described in Algorithm 1) is in accordance with the L-layer GNNs framework, where states of nodes get repeatedly updated from (l) to $(l + 1)$ layer by aggregating messages from neighbors. Precisely, given the UIG, X-2ch first construct k-order heterogeneous neighbor sets for users and items to comprehend high-order interactions [5]. Next, knowledge attributes are distilled from the KG and embedded in the edges to link neighbors and source nodes. Finally, a quad-channel message aggregation over knowledge-aware edges is conducted to update states of nodes to the next layer. The **quad-channel aggregation** implies that, on each layer (l), user and item embeddings, $e_u^{(l)}$ and $e_i^{(l)}$, are updated independently and consist of a CF part $e^{cf(l)}$ and a KG part $e^{kg(l)}$.

2.2 K-order Heterogeneous Neighbor Set

To learn complex user-item relationship, we construct K-order heterogeneous neighbor sets from UIG for users and items separately. A UIG is represented by an incidence matrix $\mathbf{M} \in \{0,1\}^{|U|\times|I|}$, where $\mathbf{M}_{ui} = 1$ indicates positive interaction between u and i. To capture the natural relation between users and items, the neighbor set of users in X-2ch only consist of items, and vice versa.

For user u, her 1-order reachable item set is $N_u^1 = \{i|\mathbf{M}_{ui} \neq 0\}$ and the k-order reachable item set is $N_u^k = \{i|(\mathbf{M}^k)_{ui} \neq 0\}$, where $\mathbf{M}^k = \mathbf{M}(\mathbf{M}^\top \mathbf{M})^{k-1}$, for k>1.

For item i, its 1-order reachable user set is $N_i^1 = \{u|\mathbf{M}_{ui} \neq 0\}$ and the k-order reachable user set is $N_i^k = \{u|(\mathbf{M}^k)_{ui} \neq 0\}$.

Different from most graph-base CF models, X-2ch only collects heterogeneous neighbors, which we believe, through the proposed aggregation strategy in next section, can better imitate the natural interactive processes between the two intrinsically different subjects, users (active; having complicated interests) and items (passive; with limited attributes).

2.3 CF-channel Attentive Aggregation

X-2ch propagates information among nodes through 2 specific channels, CF-channel and KG-channel aggregation, to maintain their individual properties. Through CF(or KG)-channel aggregation, the corresponding embedding of the node gets updated to the next layer, e.g. $e^{cf(l)} \rightarrow e^{cf(l+1)}$.

For user u, we propose a feature-wise attention mechanism to collect message from its k-order reachable items:

$$e_u^{cf(l+1)} = \sum_{k=1}^{K} \frac{1}{w_k \cdot |N_u^k|} \sum_{i \in N_u^k} W_{cf}^{(l)} (\rho_{ui} \odot e_i^{cf(l)}), \tag{1}$$

where $\odot$ is the element-wise product operation, w_k, usually set to $\frac{1}{k^2}$, is the factor to account for neighbors of different orders and $W_{cf}^l \in \mathbb{R}^{d^{(l+1)}\times d^{(l)}}$ is the matrix projecting features from layer l to *(l+1)* and $d(l)$ is the dimension of layer l. The ρ_{ui} is the **feature-wise attention**, which utilizes the similarity between source node and its neighbor to refine the inflowing message. The $(\rho_{ui})_j$, the j-th feature of ρ_{ui}, is computed by:

$$(\rho_{ui})_j = \frac{exp((e_u^{cf(l)} \odot e_i^{cf(l)})_j)}{\sum_{g=1}^{d(l)} exp((e_u^{cf(l)} \odot e_i^{cf(l)})_g)}, \quad \text{for } j = 1, ..., d^{(l)}. \tag{2}$$

For item i, the CF-channel aggregation is computed in the same manner:

$$e_i^{cf(l+1)} = \sum_{k=1}^{K} \frac{1}{w_k \cdot |N_i^k|} \sum_{u \in N_i^k} (\rho_{ui} \odot e_u^{cf(l)}) W_{cf}^{(l)}, \tag{3}$$

.

2.4 KG-channel Attentive Aggregation

In the KG-channel aggregation, X-2ch formulates the message flows to be propagated with the guidance of the knowledge-aware edges.

2.4.1 Knowledge-aware Edges. The purpose of the knowledge-aware edges is, by integrating knowledge attributes, acting as filters to refine knowledge signals propagating through KG-channel.

Given an item i, we denote its related attributes set in KG as A_i, then for each attributes $a \in A_i$, we have $R_a \in \mathbb{R}^{d^A \times d^A}$ and $e_a \in \mathbb{R}^{d^A}$, which is the corresponding relation matrix and attribute embedding (d^A is the dimension of attributes). For example, given an item *Titanic* (one of its attribute a is {*genre.drama*}), we can thus retrieve R_{genre} and e_{drama}. Correspondingly, for a user u, the set of attributes that she prefers would consist of attributes of her neighbor items: $A_u = \{a|a \in A_j, j \in N_u\}$.

Then the **knowledge-aware edges** from a node j (user or item) on the l-th layer is:

$$e_{j\rightarrow\cdot}^{E(l)} = W_E^l \left(\frac{1}{|A_j|} \sum_{a \in A_j} R_a e_a\right), \tag{4}$$

where $j \rightarrow \cdot$ means its an out-going edge from node j and $W_E^l \in \mathbb{R}^{d^l \times d^A}$ is the matrix projecting attributes to the node embedding space. Note that the edge attributes are **bi-directional**, indicating the KG signal $[u \rightarrow i]$, $e_{u\rightarrow\cdot}^E$, is different from $[i \rightarrow u]$, $e_{i\rightarrow\cdot}^E$.

2.4.2 Knowledge Aggregation. After the knowledge is embedded in edges, X-2ch aggregates neighborhood message to the next layer, $(e_i^{cf(l)} + e_i^{kg(l)}) \rightarrow e_u^{kg(l+1)}$, by aggregating KG embeddings with support of CF signals and attention on edge attributes.

For user u, the KG-channel aggregation is:

$$e_u^{kg(l+1)} = \sum_{k=1}^{K} \frac{1}{w_k \cdot |N_u^k|} \sum_{i \in N_u^k} \rho_{i\rightarrow u} \odot (W_{kg}^{(l)} e_i^{cf(l)} + e_i^{kg(l)}), \tag{5}$$

where $W_{kg}^{(l)} \in \mathbb{R}^{d^l \times d^l}$ is the matrix to adjust CF embedding before integrating with KG embedding and $\rho_{i\rightarrow u}$ is the **edge attention** from i to u:

$$(\rho_{i\rightarrow u})_j = \frac{exp((e_u^{kg(l)} \odot e_{i\rightarrow\cdot}^{E(l)})_j)}{\sum_{g=1}^{d(l)} exp((e_u^{kg(l)} \odot e_{i\rightarrow\cdot}^{E(l)})_g)}, \quad \text{for } j = 1, ..., d^{(l)}. \tag{6}$$

For item i, accordingly, the KG-channel aggregation is:

$$e_i^{kg(l+1)} = \sum_{k=1}^{K} \frac{1}{w_k \cdot |N_i^k|} \sum_{u \in N_i^k} \rho_{u\rightarrow i} \odot (W_{kg}^{(l)} e_u^{cf(l)} + e_u^{kg(l)}), \tag{7}$$

2.5 Discussion

First, the "bi-directional" edges, by distributing different messages, can help complement the knowledge features e^{kg} for both users (interested in complex attributes) and items (limited attributes). Especially for items that are "cold" in KG, the complementary messages from user nodes are beneficial. For example, if i is only described as "genre.horror" in KG and most of i's viewers N_i^k actually prefer "genre.comedy", then i is highly likely to be a horror comedy movie, which is omitted in KG.

Second, we design to integrate KG and CF embeddings via attentions on knowledge attributes for KG aggregation. Not only it's beneficial for learning complex relations, but, more realistically, this can help alleviate the "cold knowledge" problem [3, 6]. In fact, KG also suffer sparsity issue and most existing works simply remove items that are "cold" in KG to prevent drop in performance [3, 6, 18, 20]. We argue that this can be avoided by properly propagating knowledge information with collaboration of CF signals.

Table 1: **Datasets Statistics**

		Last-FM	Amazon-Book
UIG	Users / Item	23,566 / 48,123	70,679 / 24,915
	Interactions	3,034,796	847,733
KG	Entities / Relations	58,266 / 9	88,572 / 39
	Triplets	464,567	2,557,746

2.6 Optimization

To represent users and items, X-2ch concatenate the CF-KG embeddings of the last layer L, $e_u = e_u^{cf(L)} || e_u^{kg(L)}$ and $e_i = e_i^{cf(L)} || e_i^{kg(L)}$, where the $\cdot||\cdot$ is the concatenation operation. The estimated score is measured as the similarity between the target user and candidate item: $\hat{y}_{ui} = e_u^\top e_i$. Since X-2ch is designed for the ubiquitous implicit feedback settings, where data are binary (clicks or likes), we leverage the Bayesian Pairwise Ranking loss function (BPR) [13] to complete the loss function:

$$\mathcal{L} = -\sum_{(u,i^+,i^-)\in\mathcal{P}} -\ln\sigma\left(\hat{y}_{ui^+} - \hat{y}_{ui^-}\right) + \lambda\|\Theta\|_2^2,$$

where $\sigma(\cdot)$ is the sigmoid function, $\mathcal{P}$ denotes the pair-wire training set, i^+ is the positive item interacted by user u and i^- is a randomly sampled negative example. λ is the regularization parameter and Θ indicates all trainable parameters.

3 EXPERIMENTS

3.1 Experiment Settings

Datasets. A description of the datasets are in Table 1. We adopt a *72-8-20* split [2, 5, 18, 19], where 72%, 8% and 20% of a user's historical items are randomly sampled for training, validation and testing, respectively. **Last-FM** is collected and managed by last.fm website, where music tracks are items to recommend. **Amazon-Book** comprises user reviews to books on amazon.com website. As in [5, 15, 18], we keep nodes with at least ten data and binarize the ratings to {0, 1}. To construct the KGs for both datasets, we follow [16, 18] by matching items into Freebase KG database and collecting up to two-hop neighbor attributes for each item.

Baseline Models and Settings. The embedding size is set to 64 (for X-2ch, meaning $d^{cf(L)} = d^{kg(L)} = 32$.), except RippleNet is set as 16 due to its heavy computation [16, 18].

- **CKE** [22] is a classic KG-enhanced model, which merges knowledge from TranR [9] to augment the basic CF model.
- **RippleNet** [16] is a path-based model. Users embeddings are represented by item attributes from pre-defined paths.
- **KGCN** [17] is a propagating-based model. It enhances the GNNs to learn contextual and structural information from KG and UIG simultaneously.
- **KGAT** [18] is a state-of-the-art model, which models users, items and KG entities identically and propagate on a unified graph constructed from UIG and KG.
- **CKAN** [20] is a state-of-the-art model, which associates users and items with their neighbor knowledge entities and performs propagation on the KG.

Evaluation Metrics. Performance is evaluated upon the outputted top-20 recommended items by 2 widely-used metrics: Recall (*Rec@20*) and Normalized Discounted Cumulative Gain (*NDCG@20*).

Table 2: **Experimental results. Best results are in bold and ‡ denotes the best baselines. L in X-2ch(L) indicates total number of layers.**

@20	Last-FM		Amazon-Book	
	Rec.	NDCG	Rec.	NDCG
CKE	0.0736	0.1184	0.1343	0.0885
RippleNet	0.0791	0.1238	0.1336	0.0910
KGCN	0.0804	0.1281	0.1329	0.0900
KGAT	0.0870	0.1325	0.1489‡	0.1006‡
CKAN	0.0875‡	0.1337‡	0.1423	0.0998
X-2ch(1)	0.0897 (+2.55%)	0.1385 (+3.20%)	0.1536 (+3.16%)	0.1049 (+4.35%)
X-2ch(2)	**0.0906** (+3.54%)	**0.1392** (+3.72%)	**0.1570** (+5.44%)	**0.1075** (+6.83%)
X-2ch(3)	0.0886 (+1.29%)	0.1366 (+1.78%)	0.1525 (+2.45%)	0.1040 (+3.40%)

3.2 Results Analysis

Experimental results are summerized in Table 2.

Study of X-2ch of different numbers of layers can be found at the bottom of Table 2. Though X-2ch(2) is the best variant, X-2ch(1) already shows significant improvements over the best baselines, supporting the efficacy of the quad-channel design in simultaneously learning from CF and KG. X-2ch(3) underperforms its variants of less layers, implying that stacking multiple X-2ch layers might require stronger regularization or it might lead to overfitting.

Baselines comparisons. On both datasets, X-2ch outperforms CKE, which validates the effectiveness of utilizing high-order neighbor information. X-2ch also shows better results than RippleNet. While X-2ch dynamically and individually updating users and items, RippleNet forces users to be only represented by pre-sampled *RippleSets*, which hinders the learning of complex interactions. The superiority of X-2ch over KGCN justifies the necessity of explicitly modeling CF signals. Though KGAT is the best baseline on Amazon-Book, the way it formulates no distinction between users, items and KG entities on a unified graph seems to restrict the ability to discern the differences between these subjects. Finally, CKAN, while naturally combining CF and KG signals via the initial neighbors sampling on early stage, fails to further explore the CF-KG relatedness during message propagation. Overall, X-2ch outperforms all baselines thanks to the quad-channel aggregation strategy which also propagates message over knowledge-aware edges.

4 CONCLUSION AND FUTURE WORK

In the work, based on the inspection of current limitations, we proposed a novel graph recommendation model, `X-2ch`, which not only passes information via quad-channel attentive aggregations, but also innovatively embeds knowledge bi-directionally in edges to filter information flows. The effectiveness is supported by experiments. As future work, we plan to further explore effects of each components of X-2ch and to include other knowledge, e.g. text or demographic data, in edges so as to adapt the RS for other targets, e.g. diversity or explainability.

ACKNOWLEDGMENTS

The work is supported by the JSPS KAKENHI under Grant No.: JP20K01983, JP18H00904 and JP17H01001.

REFERENCES

[1] Chong Chen, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Jointly non-sampling learning for knowledge graph enhanced recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 189–198.

[2] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative memory network for recommendation systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 515–524.

[3] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[4] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

[5] Shuyi Ji, Yifan Feng, Rongrong Ji, Xibin Zhao, Wanwan Tang, and Yue Gao. 2020. Dual channel hypergraph collaborative filtering. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020–2029.

[6] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388* (2020).

[7] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. 2019. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11–20.

[8] Lu Lin and Hongning Wang. 2020. Graph Attention Networks over Edge Content-Based Channels. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1819–1827.

[9] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[10] Douglas W Oard, Jinmook Kim, et al. 1998. Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*, Vol. 83. WoUongong.

[11] Namyong Park, Andrey Kan, Xin Luna Dong, Tong Zhao, and Christos Faloutsos. 2020. MultiImport: Inferring Node Importance in a Knowledge Graph from Multiple Input Signals. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 503–512.

[12] Steffen Rendle. 2010. Factorization Machines. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*. 995–1000.

[13] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[14] Rui Sun, Xuezhi Cao, Yan Zhao, Junchen Wan, Kun Zhou, Fuzheng Zhang, Zhongyuan Wang, and Kai Zheng. 2020. Multi-modal Knowledge Graphs for Recommender Systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1405–1414.

[15] Chang-You Tai, Meng-Ru Wu, Yun-Wei Chu, Shao-Yu Chu, and Lun-Wei Ku. 2020. MVIN: Learning Multiview Items for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 99–108.

[16] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: 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.

[17] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge graph convolutional networks for recommender systems. In *The world wide web conference*. 3307–3313.

[18] 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.

[19] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. *arXiv preprint arXiv:1905.08108* (2019).

[20] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu. 2020. CKAN: Collaborative Knowledge-aware Attentive Network for Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 219–228.

[21] Deqing Yang, Zengchun Song, Lvxin Xue, and Yanghua Xiao. 2020. A Knowledge-Enhanced Recommendation Model with Attribute-Level Co-Attention. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1909–1912.

[22] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.