Multi-level Cross-view Contrastive Learning for Knowledge-aware Recommender System

Ding Zou[1,6], Wei Wei[1,6] ⊠, Xian-Ling Mao[2], Ziyang Wang[1,6], Minghui Qiu[3], Feida Zhu[4], Xin Cao[5]
[1] Cognitive Computing and Intelligent Information Processing (CCIIP) Laboratory, School of Computer Science and Technology, Huazhong University of Science and Technology, China
[2] School of Computer Science and Technology, Beijing Institute of Technology, China
[3] Alibaba Group, China
[4] Singapore Management University, Singapore
[5] School of Computer Science and Engineering, The University of New South Wales, Australia
[6] Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL), China
[1] {m202173662, weiw, ziyang1997 }@hust.edu.cn [2] maoxl@bit.edu.cn [3] minghuiqiu@gmail.com
[4] fdzhu@smu.edu.sg [5] xin.cao@unsw.edu.au

ABSTRACT

Knowledge graph (KG) plays an increasingly important role in recommender systems. Recently, graph neural networks (GNNs) based model has gradually become the theme of knowledge-aware recommendation (KGR). However, there is a natural deficiency for GNN-based KGR models, that is, the sparse supervised signal problem, which may make their actual performance drop to some extent. Inspired by the recent success of contrastive learning in mining supervised signals from data itself, in this paper, we focus on exploring the contrastive learning in KG-aware recommendation and propose a novel multi-level cross-view contrastive learning mechanism, named MCCLK. Different from traditional contrastive learning methods which generate two graph views by uniform data augmentation schemes such as corruption or dropping, we comprehensively consider three different graph views for KG-aware recommendation, including global-level structural view, local-level collaborative and semantic views. Specifically, we consider the user-item graph as a collaborative view, the item-entity graph as a semantic view, and the user-item-entity graph as a structural view. MCCLK hence performs contrastive learning across three views on both local and global levels, mining comprehensive graph feature and structure information in a self-supervised manner. Besides, in semantic view, a k-Nearest-Neighbor (kNN) item-item semantic graph construction module is proposed, to capture the important item-item semantic relation which is usually ignored by previous work. Extensive experiments conducted on three benchmark datasets show the superior performance of our proposed method over the state-of-the-arts. The implementations are available at: https://github.com/CCIIPLab/MCCLK.

⊠: Corresponding Author.

SIGIR '22, July 11–15, 2022, Madrid, Spain.

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3532025

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Graph Neural Network, Contrastive Learning, Knowledge Graph, Recommender System, Multi-view Graph Learning

ACM Reference Format:
Ding Zou[1,6], Wei Wei[1,6] ⊠, Xian-Ling Mao[2], Ziyang Wang[1,6], Minghui Qiu[3], Feida Zhu[4], Xin Cao[5] . 2022. Multi-level Cross-view Contrastive Learning for Knowledge-aware Recommender System. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3477495.3532025

1 INTRODUCTION

Recommender system is crucial for users to discover items of interest in practice. Conventional recommendation approaches (*e.g.,* collaborative filtering (CF) [15, 21, 25, 42]) rely on the availability of historical user behavior data (*e.g.,* user-item interactions [46, 57]) to capture collaborative signals for recommendation. However, they severely suffer from the cold-start problem, since they often treat each interaction as an independent instance while neglecting their relations, such as NFM [15], xDeepFM [21]. A widely-adopted solution is to incorporate various kinds of side information, such as knowledge graph (KG) [26], which contains rich facts and connections about items, to learn high-quality user and item representations for recommendation (*aka.* knowledge-aware recommendation, KGR).

Indeed, there already exists much research effort [37, 53, 55] devoted to KGR, the core of which is how to effectively leverage the graph of item side (*heterogeneous*) information into the latent user/item representation learning. Most of early studies [17, 35, 37, 53] on KGR focus on employing different *knowledge graph embedding* (KGE) models (*e.g.,* TransE [2], TransH [47]), to pre-train entity embeddings for item representation learning. However, these methods perform poorly, since they treat each item-entity relation independently for learning. Thus, the learning process is incapable of distilling sufficient collaborative signals for item representations.

Sequentially, many connection-based approaches are proposed to model multiple patterns of connections among user, item, and entity for recommendation, which can be further categorized into two classes, namely, *path*-based [16, 31, 45] and *graph neural networks (GNN)* based [16, 31, 45]. The *former* mainly focuses on enriching user-item interactions via capturing the long-range structure of KG, such as the selection of prominent paths over KG [33] *or* representing the interactions with multi-hop paths from users to items [16, 45]. However, these methods heavily rely on manually designed meta-paths, and are thus hard to optimize in reality. The *later* is widely-adopted as an informative aggregation paradigm to integrate multi-hop neighbors into node representations [30, 40, 41, 43], due to its powerful capability in effectively generating local permutation-invariant aggregation on the neighbors of a node for representation. Despite effectiveness, current GNN-based models greatly suffer from sparse supervision signal problem, owing to the extreme sparsity of interactions [1, 49] and even terrible side effects, *e.g.,* degeneration problem [9], *i.e.,* degenerating node embeddings distribution into a narrow cone, even leading to the indiscrimination of generated node representations.

However, alleviating the *sparse supervision signal* problem faces a significant challenge, that is, the inadequacy of training labels, as labels are usually scarce in real recommendation applications. Recently, contrastive learning, one of the classical Self-supervised learning (SSL) methods, is proposed to pave a way to enable training models without explicit labels [24], as its powerful capability in learning discriminative embeddings from unlabeled sample data, via maximizing the distance between negative samples while minimizing the distance between positive samples. To this end, in this paper we mainly focus on designing an end-to-end knowledge-aware model within a contrastive learning paradigm, which requires us to sufficiently leverage the limited user-item interactions and additional KG facts (*e.g.,* item-entity affiliations) for recommendation.

Actually, it is still non-trivial to design a proper contrastive learning framework, for that characteristics of both contrastive learning and knowledge-aware recommendation are needed to be carefully considered for balance, which requires us to address the following fundamental issues [44]: (1) *How to design a proper contrastive mechanism?* Due to heterogeneity, the designed model is naturally required to simultaneously handle multiple types of nodes (*e.g.,* user, item, and entity) and relations (*e.g.,* user-item, item-entity and *etc*). (2) *How to construct proper views for contrastive learning?* A straightforward way is that, we can augment (or corrupt) the input user-item-entity graph as a graph view, and contrast it with the original one, analogous to [5, 13, 20]. However, it is far from enough to solely consider global-level view (*i.e.,* user-item-entity graph) for KGR, because it is incapable of fully leveraging the rich collaborative information (*i.e.,* item-user-item co-occurrence) and semantic information (*i.e.,* item-entities-item co-occurrence). Transparently, only utilizing one graph view (*e.g.,* user-item-entity graph) at a coarse-grained level makes it difficult in fully exploiting the rich collaborative and semantic information for recommendation.

In this paper, we emphasize that the designed model should explore more graph views for learning in a more fine-grained manner. Besides, since the considered distinct graph views may be in different levels, it's not feasible to simply contrast them at the same level, and thus a multi-level cross-view contrastive mechanism is

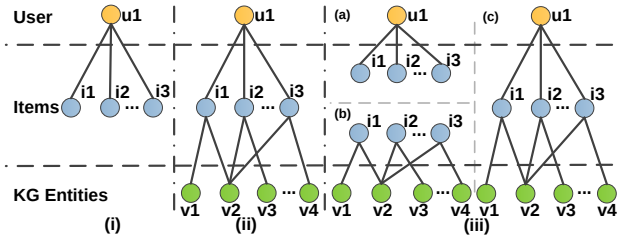


Figure 1: A toy example of our selected multi views. (i) Traditional CF-based recommendation learns from collaborative view. (ii) Previous KGR methods learn from the structural view. (iii) MCCLK learns from three selected views, including local-level collaborative view (a) and semantic view (b), global-level structural view (c).

inevitably important for the model designing. Therefore, this paper proposes a novel model based on the self-supervised learning paradigm, named **M**ulti-level **C**ross-view **C**ontrastive **L**earning for **K**nowledge-aware Recommender System (MCCLK), to fully leverage the rich collaborative and semantic information over KG and user-item interactions for KGR. Specifically, we first comprehensively consider **three complementary graph views**. As shown in Figure 1, we consider the user-item graph as collaborative view and item-entity graph as semantic view, both of which are local-level views. Besides, to preserve complete structural information (*i.e.,* long-range user-item-entity connections), user-item-entity graph is considered as a structural view in global level. Then a novel **multi-level cross-view contrastive learning mechanism** is proposed to collaboratively supervise the three graph views, which performs local-level contrastive learning between collaborative view and semantic view, global-level contrastive learning between global-level and local-level views. In particular, in the less explored semantic view, an effective k-Nearest-Neighbor (kNN) item-item semantic graph construction module is proposed, equipped with a relation-aware GNN, for explicitly considering item-item semantic similarity from knowledge information. Moreover, adaptive GNN-based graph encoders are adopted for each graph view, stressing different parts of graph information according to the views' features. Empirically, MCCLK outperforms the state-of-the-art models on three benchmark datasets.

Our contributions of this work can be summarized as follows:

- **General Aspects:** We emphasize the importance of incorporating self-supervised learning into knowledge-aware recommendation, which takes node self-discrimination as a self-supervised task to offer auxiliary signal for graph representation learning.
- **Novel Methodologies:** We propose a novel model MCCLK, which builds a multi-level cross-view contrastive framework for knowledge-aware recommendation. MCCLK considers three views from user-item-entity graph, including global-level structural view, local-level collaborative and semantic views. MCCLK then performs local-level and global-level contrastive learning to enhance representation learning from

multi-faced aspects. Moreover, in semantic view, a kNN item-item semantic graph construction module is proposed to explore item-item semantic relation.

- **Multifaceted Experiments:** We conduct extensive experiments on three benchmark datasets. The results demonstrate the advantages of our MCCLK in better representation learning, which shows the effectiveness of our multi-level cross-view contrastive learning framework and specially tailored graph aggregating mechanisms.

2 RELATED WORK

2.1 Knowledge-aware Recommendation

2.1.1 Embedding-based methods. Embedding-based methods [3, 17, 35, 37, 39, 53, 55] use knowledge graph embeddings(KGE) [2, 22, 47] to preprocess a KG, then incorporate the learned entity embeddings and relation embeddings into the recommendation. Collaborative Knowledge base Embedding(CKE) [53] combines CF module with structural, textual, and visual knowledge embeddings of items in a unified Bayesian framework. KTUP [3] utilizes TransH [47] on user-item interactions and KG triplets to jointly learn user preference and perform KG completion. RippleNet [36] explores users' potential interests by propagating users' historical clicked items along links (relations) in KG. Embedding-based methods show high flexibility in utilizing KG, but the KGE algorithms focus more on modeling rigorous semantic relatedness (*e.g.,* TransE [2] assumes head + relation = tail), which are more suitable for link prediction rather than recommendation.

2.1.2 Path-based methods. Path-based methods [16, 31, 45, 51, 52, 56] explore various patterns of connections among items in KG to provide additional guidance for the recommendation. For example, regarding KG as a Heterogeneous Information Network (HIN), Personalized Entity Recommendation (PER) [52] and meta-graph based recommendation [16] extract the meta-path/meta-graph latent features and exploit the connectivity between users and items along different types of relation paths/graphs. KPRN [45] further automatically extracts paths between users and items, and utilizes RNNs to model these paths. Path-based methods make use of KG in a more natural way, but they rely heavily on manually designed meta paths which can be hard to tune in reality. In addition, defining effective meta-paths requires domain knowledge, which is usually labor-intensive especially for complicated knowledge graphs.

2.1.3 GNN-based methods. GNN-based methods [16, 45, 51, 52, 56] are founded on the information aggregation mechanism of graph neural networks (GNNs) [7, 11, 19, 50]. Typically it integrate multi-hop neighbors into node representations to capture node feature and graph structure, which hence could model long-range connectivity. KGCN [40] and KGNN-LS [38] firstly utilize graph convolutional network (GCN) to obtain item embeddings by aggregating items' neighborhood information iteratively. Later, KGAT [41] combines user-item graph with knowledge graph as a heterogeneous graph, then utilizes GCN to recursively perform aggregation on it. More recently, KGIN [43] models user-item interactions at an intent level, which reveals user intents behind the KG interactions and combines KG interactions to perform GNN on the user-item-entity graph. However, all these approaches adopts the paradigm of supervised learning for model training, relying on their original sparse interactions. In contrast, our work explores self-supervised learning in knowledge-aware recommendation, exploiting supervised signals from data itself to improve node representation learning.

2.2 Contrastive Learning

Contrastive Learning methods [34, 44, 49] learn node representations by contrasting positive pairs against negative pairs. DGI [34] first adopts Infomax [23] in graph representation learning, and focuses on contrasting the local node embeddings with global graph embeddings. Then GMI [27] proposes to contrast center node with its local nodes from node features and topological structure. Similarly, MVGRL [12] learns node- and graph-level node representations from two structural graph views including first-order neighbors and a graph diffusion, and contrasts encoded embeddings between two graph views. More recently, HeCo [44] proposes to learn node representations from network schema view and meta-path view, and performs contrastive learning between them. And in traditional collaborative filtering (CF) based recommendation domain, SGL [49] conducts contrastive learning between original graph and corrupted graph on user-item interactions. However, little effort has been done towards investigating the great potential of contrastive learning on knowledge-aware recommendation.

3 PROBLEM FORMULATION

In this section, we first introduce two types of necessary structural data, *i.e.,* user-item interactions and knowledge graph, and then present the problem statement of our knowledge-aware recommendation problem.

Interaction Data. In a typical recommendation scenario, let $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ and $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ denote the sets of M users and N items, respectively. The user-item interaction matrix $\mathbf{Y} \in \mathbf{R}^{M \times N}$ is defined according to users' implicit feedbacks, where $y_{uv} = 1$ indicates that user u engaged with item v, such as behaviors like clicking or purchasing; otherwise $y_{uv} = 0$.

Knowledge Graph. In addition to the historical interactions, the real-world facts (*e.g.,* item attributes, or external commonsense knowledge) associated with items are stored in a KG, in the form of a heterogeneous graph [8, 31, 48]. Let $\mathcal{G} = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$ be the knowledge graph, where h, r, t are on behalf of head, relation, tail of a knowledge triple correspondingly; $\mathcal{E}$ and $\mathcal{R}$ refer to the sets of entities and relations in $\mathcal{G}$. For example, the triple (Batman Begins, film.film.star, Christian Bale) means that Christian Bale is an actor of the movie Batman Begins. In many recommendation scenarios, an item $v \in \mathcal{V}$ corresponds to one entity $e \in \mathcal{E}$. For example, in movie recommendation, the item "Iron Man" also appears in the knowledge graph as an entity with the same name. So we establish a set of item-entity alignments $\mathcal{A} = \{(v, e) | v \in \mathcal{V}, e \in \mathcal{E}\}$, where (v, e) indicates that item v can be aligned with an entity e in the KG. With the alignments between items and KG entities, KG is able to profile items and offer complementary information to the interaction data.

Problem Statement. Given the user-item interaction matrix Y and the knowledge graph $\mathcal{G}$, our task of knowledge-aware recommendation is to learn a function that can predict how likely a user would adopt an item.

4 METHODOLOGY

We now present the proposed MCCLK. MCCLK aims to incorporate self-supervised learning into knowledge-aware recommendation for improving user/item representation learning. Figure 2 displays the working flow of MCCLK, which comprises three main components: 1) **Multi Views Generation.** It generates three different graph views, including global-level structural view, local-level collaborative and semantic view. For exploring the rarely noticed semantic view, an item-item semantic graph is constructed with a proposed relation-aware GNN. 2) **Local-level contrastive learning.** It first encodes collaborative and semantic views with LightGCN, and then performs cross-view contrastive learning between two views for learning comprehensive node embeddings in the local level. 3) **Global-level contrastive learning.** It first encodes structural view with path-aware GNN, and then performs cross-view contrastive learning between global- and local-level views for learning discriminative node embeddings in the global level. We next present the three components in detail.

4.1 Multi Views Generation

Different from previous methods only considering global user-item-entity graph, we propose to learn in a more comprehensive and fine-granularity way, by jointly considering local- and global-level view. We first divide the user-item-entity graph into a user-item graph and an item-entity graph, according to their different types of item-item relationship. For the user-item graph, we treat it as **collaborative view**, aiming to mine the collaborative relationship between items, *i.e.,* item-user-item co-occurrences. For the item-entity graph, it is viewed as **semantic view**, towards exploring the semantic similarity between items, *i.e.,* item-entity-item co-occurrences. For the original user-item-entity graph, it is deemed to **structural view**, aiming to preserve the complete path information, *i.e.,* user-item-entity long-range connectivity.

Although much research effort has been devoted to collaborative and structural views, they usually inadequately explore the semantic view, leaving the crucial item-item semantic similarity untouched. Towards explicitly considering the item-item semantic relationship, we propose to construct a ***k*-Nearest-Neighbor item-item semantic graph** S with a relation-aware aggregation mechanism which preserves both neighbor entities and relations information. Each entry S_{ij} in S denotes the semantic similarity between item i and item j. In particular, $S_{ij} = 0$ means there is no link between them.

Specifically, we first recursively learn item representations for K' times from the knowledge graph $\mathcal{G}$, with the proposed relation-aware aggregating mechanism as follows:

$$\begin{aligned}\mathbf{e}_i^{(k+1)} &= \tfrac{1}{|\mathcal{N}_i|} \sum_{(r,v)\in\mathcal{N}_i} \mathbf{e}_r \odot \mathbf{e}_v^{(k)}, \\ \mathbf{e}_v^{(k+1)} &= \tfrac{1}{|\mathcal{N}_v|} \left(\sum_{(r,v)\in\mathcal{N}_v} \mathbf{e}_r \odot \mathbf{e}_v^{(k)} + \sum_{(r,i)\in\mathcal{N}_v} \mathbf{e}_r \odot \mathbf{e}_i^{(k)} \right),\end{aligned} \tag{1}$$

where $\mathbf{e}_i^{(k)}$ and $\mathbf{e}_v^{(k)}$ $(\forall k \in K')$ separately denote the representations of item i and entity v, which memorize the relational signals propagated from their $(k-1)$-hop neighbors. For each triplet (i, r, v), a relational message $\mathbf{e}_r \odot \mathbf{e}_v^{(k)}$ is designed for implying different meanings of triplets, via modeling the relation r through the projection or rotation operator [32].

As such, both neighbor entities and relations in KG are encoded into item representation. Thereafter, inspired by [54], the item-item similarity graph is built based on a cosine similarity, which is calculated as follows:

$$S_{ij} = \frac{\left(\mathbf{e}_i^{(K')}\right)^\top \mathbf{e}_j^{(K')}}{\left\|\mathbf{e}_i^{(K')}\right\| \left\|\mathbf{e}_j^{(K')}\right\|}. \tag{2}$$

Sequentially, a kNN sparsification [4] is conducted on the fully-connected item-item graph, decreasing computationally demanding, feasible noisy, and unimportant edges [6].

$$\widehat{S}_{ij} = \begin{cases} S_{ij}, & S_{ij} \in \text{ top-k}\,(S_i), \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $\widehat{S}_{ij}$is a sparsified and directed graph adjacency matrix. To alleviate the exploding or vanishing gradient problem [19], the adjacency matrix is normalized as follows:

$$\widetilde{S} = (D)^{-\frac{1}{2}}\, \widehat{S}\, (D)^{-\frac{1}{2}}, \tag{4}$$

where $D \in \mathbb{R}^{N\times N}$is the diagonal degree matrix of $\widetilde{S}$ and $D_{i,i} = \sum_j \widehat{S}_{ij}$. Hence the item-item semantic graph S and its normalized sparsified adjacency matrix $\widetilde{S}$ are finally obtained.

By doing so, each graph view is now acquired, that is: user-item interaction graph Y for collaborative view, item-item semantic graph S for semantic view, and the whole user-item-entity graph for structural view. The following local- and global-level contrastive learning are performed across such three views, which will be illustrated in detail.

4.2 Local-level Contrastive Learning

Based on the obtained complementary collaborative and semantic views in the local level, we move on to explore two graph views with proper graph encoder, and perform contrastive learning between them to supervise each other. Specifically, an effective Light-GCN [14] is performed in two views to learn a comprehensive item representation. Then with two view-specific embeddings encoded, the local-level contrastive learning is proposed, encouraging the two views to collaboratively improve representations.

4.2.1 Collaborative View Encoder. The collaborative view stresses collaborative signals between items, *i.e.,* item-user-item co-occurrences. As a result, collaborative information could be captured in the collaborative view, by modeling long-range connectivity from user-item interactions. Inspired by precious collaborative filter (CF) based work [14, 42], a Light-GCN is adopted here, which recursively performs aggregation for K times. Light-GCN contains simple message passing and aggregation mechanism without feature transformation and non-linear activation, which is effective and computationally efficient. In the k-th layer, the aggregation proceeding can be

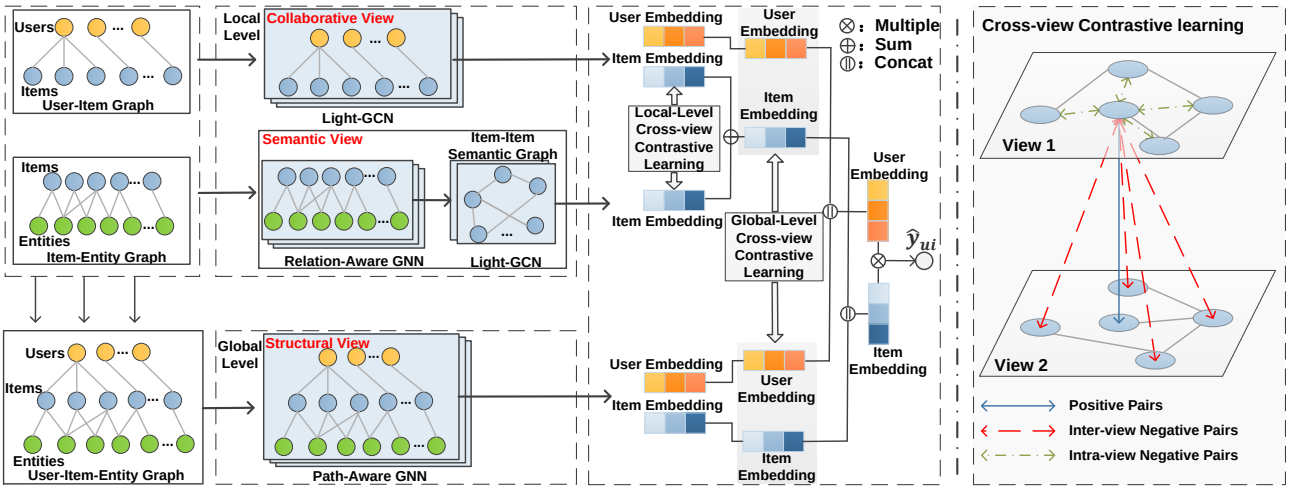


Figure 2: Illustration of the proposed MCCLK model. The left subfigure shows model framework of MCCLK; and the right subfigure presents the details of cross-view contrastive learning mechanism. Best viewed in color.

formulated as follows:

$$\begin{aligned} \mathbf{e}_u^{(k+1)} &= \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}i|}} \mathbf{e}_i^{(k)}, \\ \mathbf{e}_i^{(k+1)} &= \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}i|}} \mathbf{e}_u^{(k)}, \end{aligned} \tag{5}$$

where $\mathbf{e}_i^{(k)}$ and $\mathbf{e}_u^{(k)}$ represent embeddings of user u and item i at the k-th layer, $\mathcal{N}_u$, $\mathcal{N}_i$ represent neighbors of user u and item i respectively. Then we sum representations at different layers up as the local collaborative representations $\mathbf{z}_i^c$ and $\mathbf{z}_u^c$, as follows:

$$\mathbf{z}_u^c = \mathbf{e}_u^{(0)} + \cdots + \mathbf{e}_u^{(K)}, \quad \mathbf{z}_i^c = \mathbf{e}_i^{(0)} + \cdots + \mathbf{e}_i^{(K)}. \tag{6}$$

4.2.2 Semantic View Encoder. The semantic view focuses on semantic similarity between items, which has been confirmed to be important but ignored by previous work. Having explicitly constructed the item-item semantic graph from the item-entity affiliations, a Light-GCN is adopted on it with L times aggregation operation, to learn better item representations by injecting item-item affinities into the embedding. In the l-th layer ($\forall l \in L$), the message passing and aggregation process could be formulated as:

$$\mathbf{e}_i^{(l+1)} = \sum_{j \in \mathcal{N}(i)} \widetilde{S} \mathbf{e}_j^{(l)}, \tag{7}$$

where $\mathcal{N}(i)$ is the neighbor items, $\widetilde{S}$ is the normalized sparsified graph adjacency matrix in Equation 4, and $\mathbf{e}_i^{(l)}$ is the l-th layer representation of item i. Here the input item representation $\mathbf{e}_j^{(0)}$ is set as its corresponding ID embedding vector, rather than the aggregated features, since the Light-GCN is employed in order to directly capture item-item affinities. Then we sum item representations at different layers up to get the local semantic representations $\mathbf{z}_i^s$:

$$\mathbf{z}_i^s = \mathbf{e}_i^{(0)} + \cdots + \mathbf{e}_i^{(L)}. \tag{8}$$

4.2.3 Local-level Cross-view Contrastive Optimization. With the view-specific embeddings $\mathbf{z}_i^s$ and $\mathbf{z}_i^c$ for item i from the collaborative and semantic views, a local-level cross-view contrastive learning is performed, for supervising two views to learn discriminative representations. Aiming to map them into the space where contrastive loss is calculated, embeddings are first feed into a MLP with one hidden layer:

$$\begin{aligned} \mathbf{z}_i^c\_\mathrm{p} &= W^{(2)} \sigma\left(W^{(1)} \mathbf{z}_i^c + b^{(1)}\right) + b^{(2)}, \\ \mathbf{z}_i^s\_\mathrm{p} &= W^{(2)} \sigma\left(W^{(1)} \mathbf{z}_i^s + b^{(1)}\right) + b^{(2)}, \end{aligned} \tag{9}$$

where $W^{(\cdot)} \in \mathbb{R}^{d \times d}$ and $b^{(\cdot)} \in \mathbb{R}^{d \times 1}$ are trainable parameters, σ is ELU non-linear function. Then we define the positive and negative samples here, inspired by works in other areas [58, 59], for any node in one view, the same node embedding learned by the other view forms the positive sample; and in two views, nodes embeddings other than it are naturally regarded as negative samples.

With the defined positive and negative samples, we have the following contrastive loss:

$$\mathcal{L}^{local} = -\log \frac{e^{s\left(\mathbf{z}_{i\_\mathrm{p}}^s, \mathbf{z}_{i\_\mathrm{p}}^c\right)/\tau}}{\underbrace{e^{s\left(\mathbf{z}_{i\_\mathrm{p}}^s, \mathbf{z}_{i\_\mathrm{p}}^c\right)/\tau}}_{\text{positive pair}} + \underbrace{\sum_{k \neq i} e^{s\left(\mathbf{z}_{i\_\mathrm{p}}^s, \mathbf{z}_{k\_\mathrm{p}}^s\right)/\tau}}_{\text{intra-view negative pairs}} + \underbrace{\sum_{k \neq i} e^{s\left(\mathbf{z}_{i\_\mathrm{p}}^s, \mathbf{z}_{k\_\mathrm{p}}^c\right)/\tau}}_{\text{inter-view negative pairs}}}, \tag{10}$$

where $\mathrm{s}(\cdot)$ denotes the cosine similarity calculating, and τ denotes a temperature parameter. It's worth mentioning that negative samples come from two sources, which are intra-view and inter-view nodes, corresponding to the second and the third term in the denominator in Equation 10. In this way, the local-level cross-view contrastive learning is successfully achieved.

4.3 Global-level Contrastive Learning

Although user/item feature information has been revealed from local-level views, the complete graph structural information hasn't been explored, that is, the long-range connectivity unifying both user-item and item-entity graphs, *i.e.,* user-item-entity connections. Hence the global-level contrastive learning is introduced, which first explores the structural view with a path-aware encoder, and

then performs contrastive learning between the global-level and local-level views to supervise each other level. To be more specific, inspired by [43], we design a path-aware GNN to automatically encode path information into node embeddings. Then with the encoded embeddings from global-level view and local-level view, the global-level contrastive learning is performed, for supervising two-level views to learn comprehensive representations.

4.3.1 Structural View Encoder. Aiming to encode the structural information under structural view (*i.e.,* the variety of paths), inspired by [43], a path-aware GNN is proposed here, which aggregates neighboring information for L' times meanwhile preserving the path information, *i.e.,* long-range connectivity such as user-interact-item-relation-entity.

In particular, in l-th layer ($\forall l \in L'$) the aggregation process can be formulated as:

$$\begin{aligned} \mathbf{e}_u^{(l+1)} &= \tfrac{1}{|\mathcal{N}_u|} \sum_{i \in \mathcal{N}_u} \mathbf{e}_i^{(l)}, \\ \mathbf{e}_i^{(l+1)} &= \tfrac{1}{|\mathcal{N}_i|} \sum_{(r,v) \in \mathcal{N}_i} \beta(i,r,v) \mathbf{e}_r \odot \mathbf{e}_v^{(l)}, \end{aligned} \tag{11}$$

where $\mathbf{e}_i^{(l)}$ and $\mathbf{e}_v^{(l)}$ separately denote the representations of item i and entity v, which memorize the relational signals propagated from their $(l-1)$-hop neighbors and hence store the holistic semantics of multi-hop paths. And aiming to weight each relation and entity, the attention weights $\beta(i,r,v)$ is calculated as follows:

$$\begin{aligned} \beta(i,r,v) &= \text{softmax}\left((\mathbf{e}_i || \mathbf{e}_r)^T \cdot (\mathbf{e}_v || \mathbf{e}_r)\right) \\ &= \frac{\exp\left((\mathbf{e}_i || \mathbf{e}_r)^T \cdot (\mathbf{e}_v || \mathbf{e}_r)\right)}{\sum_{(v',r) \in \hat{\mathbf{N}}(i)} \exp\left((\mathbf{e}_i || \mathbf{e}_r)^T \cdot (\mathbf{e}_{v'} || \mathbf{e}_r)\right)}, \end{aligned} \tag{12}$$

where $||$ denotes concat operation, $\hat{\mathbf{N}}(i)$ denotes the set of neighboring entities $N(i)$ and item i itself. Then we sum all layers' representations up to have the global representations $\mathbf{z}_u^g$ and $\mathbf{z}_i^g$:

$$\mathbf{z}_u^g = \mathbf{e}_u^{(0)} + \cdots + \mathbf{e}_u^{(L')}, \quad \mathbf{z}_i^g = \mathbf{e}_i^{(0)} + \cdots + \mathbf{e}_i^{(L')}. \tag{13}$$

4.3.2 Global-level Cross-view Contrastive Optimization. Obtaining the node representations under global- and local-level views, they are first mapped into the space where the contrastive loss is calculated, the same as local-level contrastive loss calculating:

$$\begin{aligned} \mathbf{z}_{i\_\text{p}}^g &= W^{(2)} \sigma\left(W^{(1)} \mathbf{z}_i^g + b^{(1)}\right) + b^{(2)}, \\ \mathbf{z}_{i\_\text{p}}^l &= W^{(2)} \sigma\left(W^{(1)} (\mathbf{z}_i^c + \mathbf{z}_i^s) + b^{(1)}\right) + b^{(2)}. \end{aligned} \tag{14}$$

With the same positive and negative sampling strategy to local-level contrastive learning, we have the following contrastive loss:

$$\begin{aligned} \mathcal{L}_i^g &= -\log \frac{e^{s\left(\mathbf{z}_{i\_\text{p}}^g, \mathbf{z}_{i\_\text{p}}^l\right)/\tau}}{\underbrace{e^{s\left(\mathbf{z}_{i\_\text{p}}^g, \mathbf{z}_{i\_\text{p}}^l\right)/\tau}}_{\text{positive pair}} + \underbrace{\sum_{k \neq i} e^{s\left(\mathbf{z}_{i\_\text{p}}^g, \mathbf{z}_{k\_\text{p}}^g\right)/\tau}}_{\text{intra-view negative pairs}} + \underbrace{\sum_{k \neq i} e^{s\left(\mathbf{z}_{i\_\text{p}}^g, \mathbf{z}_{k\_\text{p}}^l\right)/\tau}}_{\text{inter-view negative pairs}}}, \\ \mathcal{L}_i^l &= -\log \frac{e^{s\left(\mathbf{z}_{i\_\text{p}}^l, \mathbf{z}_{i\_\text{p}}^g\right)/\tau}}{\underbrace{e^{s\left(\mathbf{z}_{i\_\text{p}}^l, \mathbf{z}_{i\_\text{p}}^g\right)/\tau}}_{\text{positive pair}} + \underbrace{\sum_{k \neq i} e^{s\left(\mathbf{z}_{i\_\text{p}}^l, \mathbf{z}_{k\_\text{p}}^l\right)/\tau}}_{\text{intra-view negative pairs}} + \underbrace{\sum_{k \neq i} e^{s\left(\mathbf{z}_{i\_\text{p}}^l, \mathbf{z}_{k\_\text{p}}^g\right)/\tau}}_{\text{inter-view negative pairs}}}, \end{aligned} \tag{15}$$

where $\mathcal{L}_i^g$ and $\mathcal{L}_i^l$ denote the contrastive learning loss calculated from the global view and local view. And the contrastive loss $\mathcal{L}_u^g / \mathcal{L}_u^l$ calculating from user embedding is similar as $\mathcal{L}_i^g / \mathcal{L}_i^l$, where only item embeddings are exchanged into user embeddings in the formula. Then the overall objective is given as follows:

$$\mathcal{L}^{global} = \frac{1}{2N} \sum_{i=1}^{N} (\mathcal{L}_i^g + \mathcal{L}_i^l) + \frac{1}{2M} \sum_{i=1}^{M} (\mathcal{L}_u^g + \mathcal{L}_u^l). \tag{16}$$

4.4 Model Prediction

After performing multi-layer aggregation in three views and optimizing through multi-level cross-view contrastive learning, we obtain multiple representations for user u, namely z_u^c and z_u^g; analogous to item i, z_i^c, z_i^s and z_i^g. By summing and concatenating the above representations, we have the final user/item representations and predict their matching score through inner product, as follows:

$$\begin{aligned} \mathbf{z}_u^* &= \mathbf{z}_u^g || \mathbf{z}_u^c, \\ \mathbf{z}_i^* &= \mathbf{z}_i^g || (\mathbf{z}_i^c + \mathbf{z}_i^s), \\ \hat{\mathbf{y}}(u,i) &= \mathbf{z}_u^{*\top} \mathbf{z}_i^*. \end{aligned} \tag{17}$$

4.5 Multi-task Training

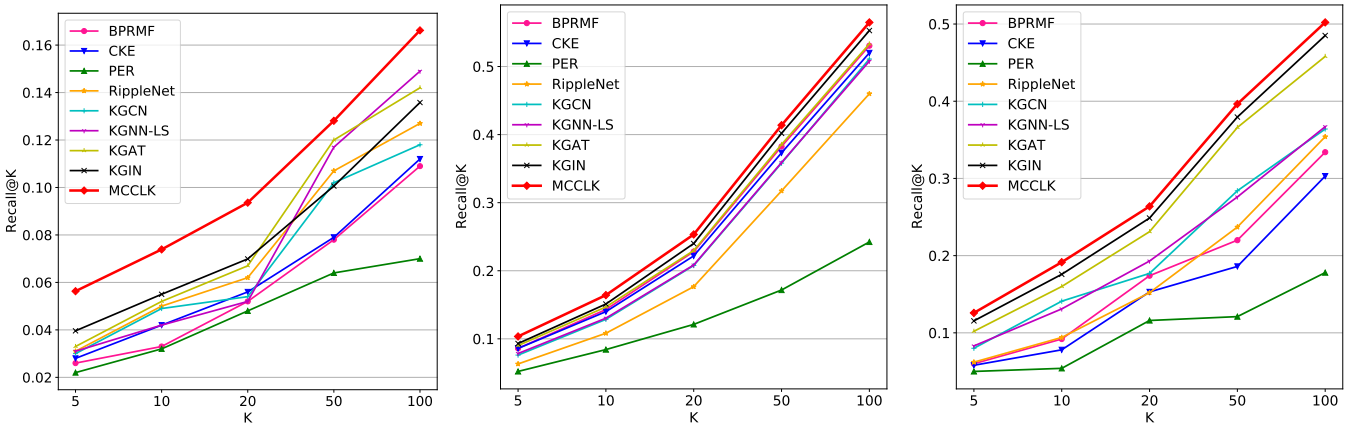
To combine the recommendation task with the self-supervised task, we optimize the whole model with a multi-task training strategy. For the KG-aware recommendation task, a pairwise BPR loss [29] is adopted to reconstruct the historical data, which encourages the prediction scores of a user's historical items to be higher than the unobserved items.

$$\mathcal{L}_{\text{BPR}} = \sum_{(u,i,j) \in O} -\ln \sigma\left(\hat{\mathbf{y}}_{ui} - \hat{\mathbf{y}}_{uj}\right), \tag{18}$$

where $\boldsymbol{O} = \left\{(u,i,j) \mid (u,i) \in \boldsymbol{O}^+, (u,j) \in \boldsymbol{O}^-\right\}$ is the training dataset consisting of the observed interactions $\boldsymbol{O}^+$ and unobserved counterparts $\boldsymbol{O}^-$; σ is the sigmoid function. By combining the global- and local-level contrastive loss with BPR loss, we minimize the following objective function to learn the model parameter:

$$\mathcal{L}_{MCCLK} = \mathcal{L}_{\text{BPR}} + \beta(\alpha \mathcal{L}^{local} + (1-\alpha)\mathcal{L}^{global}) + \lambda \|\Theta\|_2^2, \tag{19}$$

where Θ is the model parameter set, α is a hyper parameter to determine the local-global contrastive loss ratio, β and λ are two hyper parameters to control the contrastive loss and L_2 regularization term, respectively.



(a) Book-Crossing (b) MovieLens-1M (c) Last.FM

Figure 3: The result of Recall@K in top-K recommendation.

		Book-Crossing	MovieLens-1M	Last.FM
User-item Interaction	# users	17,860	6,036	1,872
	# items	14,967	2,445	3,846
	# interactions	139,746	753,772	42,346
Knowledge Graph	# entities	77,903	182,011	9,366
	# relations	25	12	60
	# triplets	151,500	1,241,996	15,518
Hyper-parameter Settings	# α	0.2	0.2	0.2
	# β	0.1	0.1	0.1
	# K	2	2	3
	# K'	2	2	2
	# L	1	1	2
	# L'	2	2	2

Table 1: Statistics and hyper-parameter settings for the three datasets.(α: local-level contrastive loss weight, β: contrastive loss weight, K: local collaborative aggregation depth, K': aggregation depth of item-item semantic graph construction, L: local semantic aggregation depth, L': global structural aggregation depth.)

5 EXPERIMENT

Aiming to answer the following research questions, we conduct extensive experiments on three public datasets:

- **RQ1:** How does MCCLK perform, compared to present models?
- **RQ2:** Are the main components (*e.g.,* local-level contrastive learning, global-level contrastive learning) really working well?
- **RQ3:** How do different hyper-parameter settings (*e.g.,* aggregation layer in structural view, local-level contrastive loss weight α *etc*) affect MCCLK?
- **RQ4:** Is the self-supervised task really improving the representation learning?

5.1 Experiment Settings

5.1.1 Dataset Description. We use three benchmark datasets to evaluate the effectiveness of MCCLK: Book-Crossing, MovieLens-1M, and Last.FM. The three datasets of different domains are publicly accessible and vary in size and sparsity, making our experiments more convincing.

- `Book-Crossing`[1]: It is collected from the book-crossing community, which consists of trenchant ratings (ranging from 0 to 10) about various books.
- `MovieLens-1M`[2]: It's a benchmark dataset for movie recommendations, which contains approximately 1 million explicit ratings (ranging from 1 to 5) on a total of 2,445 items from 6,036 users.
- `Last.FM`[3]: It is a music listening dataset collected from Last.FM online music systems with around 2 thousand users.

Since the interactions in MovieLens-1M, Book-Crossing, and Last.FM are explicit feedback, we follow RippleNet [36] and transform them into the implicit feedback in which 1 indicates the positive samples (the threshold of the rating to be viewed as positive is 4 for MovieLens-1M, but no threshold is set for Last.FM and Book-Crossing due to their sparsity). As to negative samples, for each user, we randomly sample from his unwatched items with the size equal to his positive ones.

As for the sub-KG construction, we follow RippleNet [36] and use Microsoft Satori[4] to construct it for MovieLens-1M, Book-Crossing, and Last.FM datasets. Each sub knowledge graph that follows the triple format is a subset of the whole KG with a confidence level greater than 0.9. Given the sub-KG, we gather Satori IDs of all valid movies/books/musicians through matching their names with the tail of triples. Then we match the item IDs with the head of all triples and select all well-matched triples from the sub-KG. The basic statistics of the three datasets are presented in Table 1.

[1] http://www2.informatik.uni-freiburg.de/~cziegler/BX/
[2] https://grouplens.org/datasets/movielens/1m/
[3] https://grouplens.org/datasets/hetrec-2011/
[4] https://searchengineland.com/library/bing/bing-satori

Model	Book-Crossing		MovieLens-1M		Last.FM	
	AUC	*F1*	*AUC*	*F1*	*AUC*	*F1*
BPRMF	0.6583(−10.42%)	0.6117(−6.60%)	0.8920(−4.31%)	0.7921(−7.10%)	0.7563(−12.00%)	0.7010(−9.98%)
CKE	0.6759(−8.66%)	0.6235(−5.42%)	0.9065(−2.86%)	0.8024(−6.07%)	0.7471(−12.92%)	0.6740(−12.68%)
RippleNet	0.7211(−4.14%)	0.6472(−3.05%)	0.9190(−1.61%)	0.8422(−2.09%)	0.7762(−10.01%)	0.7025(−9.83%)
PER	0.6048(−15.77%)	0.5726(−10.51%)	0.7124(−22.27%)	0.6670(−19.61%)	0.6414(−23.49%)	0.6033(−19.75%)
KGCN	0.6841(−7.84%)	0.6313(−4.64%)	0.9090(−2.61%)	0.8366(−2.65%)	0.8027(−7.36%)	0.7086(−9.22%)
KGNN-LS	0.6762(−8.63%)	0.6314(−4.63%)	0.9140(−2.11%)	0.8410(−2.21%)	0.8052(−7.11%)	0.7224(−7.84%)
KGAT	0.7314(−3.11%)	0.6544(−2.33%)	0.9140(−2.11%)	0.8440(−1.91%)	0.8293(−4.70%)	0.7424(−5.84%)
KGIN	0.7273(−3.52%)	0.6614(−1.63%)	0.9190(−1.61%)	0.8441(−1.90%)	0.8486(−2.77%)	0.7602(−4.06%)
MCCLK	**0.7625***	**0.6777***	**0.9351***	**0.8631***	**0.8763***	**0.8008***

Table 2: The result of *AUC* and *F1* in CTR prediction. The best results are in boldface and the second best results are underlined. * denotes statistically significant improvement by unpaired two-sample *t*-test with $p < 0.001$.

5.1.2 Baselines. To demonstrate the effectiveness of our proposed MCCLK, we compare MCCLK with four types of recommender system methods: CF-based methods (BPRMF), embedding-based method (CKE, RippleNet), path-based method (PER), and GNN-based methods(KGCN, KGNN-LS, KGAT, KGIN) as follows:

- `BPRMF` [29]: It's a typical CF-based method that uses pairwise matrix factorization for implicit feedback optimized by the BPR loss.
- `CKE` [53]: It's a embedding-based method that combines structural, textual, and visual knowledge in one framework.
- `RippleNet` [36]: It's a classical embedding-based method which propagates users' preferences on the KG.
- `PER` [52]: It's a typical path-based method which extracts meta-path-based features to represent the connectivity between users and items.
- `KGCN` [40]: It's a GNN-based method which iteratively integrates neighboring information to enrich item embeddings.
- `KGNN-LS` [38]: It is a GNN-based model which enriches item embeddings with GNN and label smoothness regularization.
- `KGAT` [41]: It's a GNN-based method which iteratively integrates neighbors on user-item-entity graph with an attention mechanism to get user/item representations.
- `KGIN` [43]: It's a state-of-the-art GNN-based method, which disentangles user-item interactions at the granularity of user intents, and performs GNN on the proposed user-intent-item-entity graph.

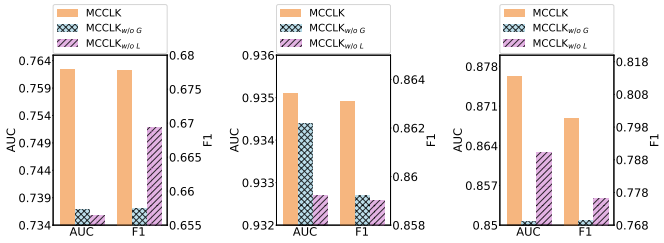
5.1.3 Evaluation Metrics. We evaluate our method in two experimental scenarios: (1) In click-through rate (CTR) prediction, we apply the trained model to predict each interaction in the test set. We adopt two widely used metrics [36, 40] *AUC* and *F1* to evaluate CTR prediction. (2) In top-K recommendation, we use the trained model to select K items with the highest predicted click probability for each user in the test set, and we choose Recall@K to evaluate the recommended sets.

5.1.4 Parameter Settings. We implement our MCCLK and all baselines in Pytorch and carefully tune the key parameters. For a fair comparison, we fix the embedding size to 64 for all models, and the embedding parameters are initialized with the Xavier method [10]. We optimize our method with Adam [18] and set the batch size to 2048. A grid search is conducted to confirm the optimal settings, we tune the learning rate η among$\{0.0001, 0.0003, 0.001, 0.003\}$ and λ of $L2$ regularization term among $\{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$. Other hyper-parameter settings are provided in Table 1. The best settings for hyper-parameters in all comparison methods are researched by either empirical study or following the original papers.

5.2 Performance Comparison (RQ1)

We report the empirical results of all methods in Table 2 and Figure 3. The improvements and statistical significance test are performed between MCCLK and the strongest baselines (highlighted with underline). Analyzing such performance comparison, we have the following observations:

- **Our proposed MCCLK achieves the best results.** MCCLK consistently outperforms all baselines across three datasets in terms of all measures. More specifically, it achieves significant improvements over the strongest baselines *w.r.t.* AUC by 3.11%, 1.61%, and 2.77% in Book, Movie, and Music respectively, which demonstrates the effectiveness of MCCLK. We attribute such improvements to the following aspects: (1) By contrasting collaborative and semantic views at the local level, MCCLK is able to capture collaborative and semantic feature information better; (2) The global-level contrastive mechanism preserves both structure and feature information from two-level self-discrimination, hence capturing more comprehensive information for MCCLK than methods only modeling global structure.
- **Incorporating KG benefits recommender system.** Comparing CKE with BPRMF, leaving KG untapped limits the performance of MF. By simply incorporating KG embeddings into MF, CKE performs better than MF. Such findings are consistent with prior studies [3], indicating the importance of side information like KG.
- **The way of exploiting KG information determines the model performance.** Path-based method PER performs even worse than BPRMF, because the optimal user-defined meta-paths are hard to be defined in reality. This fact stresses the importance of capturing structural path information from the whole graph.



(a) Book (b) Movie (c) Music

Figure 4: Effect of ablation study.

	Book		Movie		Music	
	Auc	F1	Auc	F1	Auc	F1
L=1	**0.7602**	**0.6777**	**0.9350**	**0.8631**	0.8711	0.7858
L=2	0.7601	0.6768	0.9347	0.8628	**0.8742**	**0.7945**
L=3	0.7591	0.6733	0.9345	0.8627	0.8726	0.7891
L=4	0.7583	0.6749	0.9343	0.8627	0.8720	0.7846

Table 3: Impact of aggregation depth in semantic view.

- **GNN has a strong power of graph learning.** Most of the GNN-based methods perform better than embedding bassed and path-based ones, suggesting the importance of modeling long-range connectivity for graph representation learning. The truth inspires us that learning local/global graph information with a proper aggregation mechanism could improve the model performance.

5.3 Ablation Studies (RQ2)

As shown in table 4, here we examine the contributions of main components in our model to the final performance by comparing MCCLK with the following two variants:

- MCCLK$_{w/o\ G}$: In this variant, the global-level contrastive learning module is removed, nodes are encoded from two local level views.
- MCCLK$_{w/o\ L}$: This variant removes the local-level contrastive learning module, and only remains the structural view learning of the user-bundle-item graph.

The results of two variants and MCCLK are reported in Figure 4, from which we have the following observations: 1) Removing the global-level contrastive learning significantly degrades the model's performance, which suggests its importance of exploring graph structural information for KG-aware recommendation. 2) In most cases, MCCLK$_{w/o\ L}$ is the least competitive model, especially in massive datasets (*i.e.,* book and movie), which demonstrates the superiority of learning discriminative information between collaborative and semantic views in the local level.

5.4 Sensitivity Analysis (RQ3)

5.4.1 Impact of aggregation depth in semantic view. To study the influence of item-item semantic graph aggregation depth, we vary L in range of {1, 2, 3, 4} and demonstrate the performance comparison on book, movie, and music datasets in Table 3. MCCLK performs best when $L = 1, 1, 2$, on Book, Movie, and Music respectively. We can convince that: one- or two-hops are enough for aggregating neighbor information in the item-item semantic graph, which conveys the effectiveness of item-item semantic graph construction.

	Book		Movie		Music	
	Auc	F1	Auc	F1	Auc	F1
L'=1	0.7602	0.6776	0.9350	0.8628	0.8711	0.7858
L'=2	**0.7625**	**0.6777**	**0.9351**	**0.8631**	**0.8763**	**0.8008**
L'=3	0.7550	0.6719	0.9334	0.8589	0.8713	0.7899
L'=4	0.7569	0.6680	0.9320	0.8574	0.8706	0.7841

Table 4: Impact of aggregation depth in structural view.

	Book		Movie		Music	
	Auc	F1	Auc	F1	Auc	F1
β=1	0.7520	0.6649	0.9337	0.8593	0.8735	0.7938
β=0.1	**0.7625**	**0.6713**	**0.9351**	**0.8622**	**0.8758**	**0.7972**
β=0.01	0.7608	0.6689	0.9346	0.8610	0.8721	0.7913
β=0.001	0.7607	0.6675	0.9343	0.8604	0.8714	0.7856

Table 5: Impact of contrastive loss weight β.

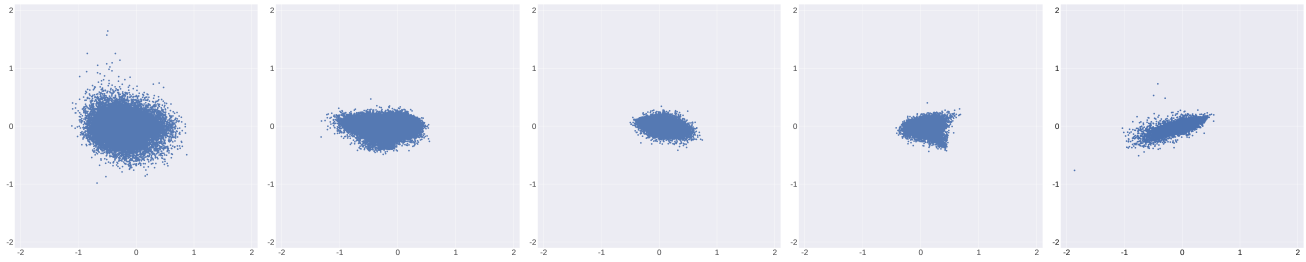
5.4.2 Impact of aggregation depth in structural view. To analyze the influence of aggregation depth in structural view, we vary L' in range of {1, 2, 3, 4}, and illustrate the performance changing curves on book, movie, and music datasets in table 4. We find that: $L' = 2$ are proper distance for collecting global structural signals from the longer-range connectivity (*i.e.,* u-r-i-r-e, i-r-e-r-i, *etc*), further stacking more layers only introduces more noise.

5.4.3 Impact of local-level contrastive loss weight α. The trade-off parameter α controls the influence of local-level contrastive loss in final contrastive loss. To study the influence of α, we vary α in {0, 0.2, 0.4, 0.6, 0.8, 1.0 }. According to the results shown in Figure 6, we have the following observations: (1) The worst performance usually occurs when $\alpha = 1$, which emphasizes the importance of global-level contrastive loss. (2) The worse performance of scenarios where $\alpha = 0, 1$ demonstrates the effectiveness of both two-level contrastive loss, and $\alpha = 0.2$ balances local- and global-level contrastive loss on model optimization.

5.4.4 Impact of contrastive loss weight β. The parameter β determines the importance of the contrastive loss during the multi-task training. Towards studying the influence of contrastive loss weight β, we vary β in {1, 0.1, 0.01, 0.001} From the results shown in Table 5, we can observe that: $\beta = 0.1$ brings the best model performance, the main reason is that changing the contrastive loss to a fairly equal level to recommendation task loss could boast the model performance.

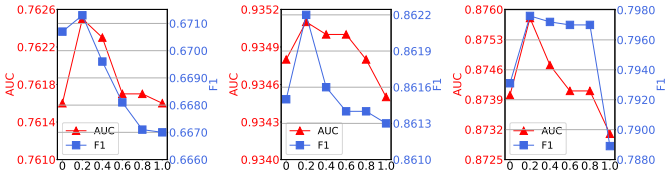
5.5 Visualization (RQ4)

To evaluate whether the contrastive mechanism affects the representation learning performance, following previous contrastive learning work [28], we adopt SVD decomposition to project the



(a) MCCLK (b) MCCLK$_{w/o\ G}$ (c) MCCLK$_{w/o\ L}$ (d) KGIN (e) RippleNet

Figure 5: Visualization of model representation learning ability on Book-Crossing.



(a) Book (b) Movie (c) Music

Figure 6: Impact of local-level contrastive loss weight α.

learned item embeddings into 2D and give out the regularized singular. As shown in Figure 5, we compare the visualized results of MCCLK, MCCLK$_{w/o\ G}$, MCCLK$_{w/o\ L}$, KGIN, and RippleNet on Book-Crossing, from which we can have the following observations:

- The node embeddings of KGIN and RippleNet are mixed to some degree and fall into a narrow cone. In contrast, the node embeddings of MCCLK have a more diverse distribution and hence are able to represent different node feature information, which demonstrates our superiority in better representation learning and alleviating the representation degeneration problem.
- By comparing MCCLK with its variants, we observe that removing the local-level or global-level contrastive loss makes the learned embeddings more indistinguishable, which convinces the effectiveness and robustness of representation learning are coming from the multi-level cross-view contrastive learning mechanism.

6 CONCLUSION

In this work, we focus on exploring contrastive learning on KG-aware recommendation, improving the quality of user/item representation learning in a self-supervised manner. We propose a novel framework, MCCLK, which achieves better user/item representation learning from two dimensions: (1) MCCLK considers user/item representation learning from three views, including global-level structural view, local-level collaborative and semantic view, and explicitly construct a kNN item-item semantic graph to mine rarely noticed item-item semantic similarity in semantic view. (2) MCCLK performs multi-level cross-view contrastive learning among three views, exploring both feature and structural information, and further learning discriminative representations.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant No.61602197, Grant No.L1924068, Grant No.61772076, in part by CCF-AFSG Research Fund under Grant No.RF20210005, in part by the fund of Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL), and in part by the National Research Foundation (NRF) of Singapore under its AI Singapore Programme (AISG Award No: AISG-GC-2019-003). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore. The authors would also like to thank the anonymous reviewers for their comments on improving the quality of this paper.

REFERENCES

[1] Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle. 2017. A generic coordinate descent framework for learning from implicit feedback. In *WWW*. 1341–1350.
[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*. 1–9.
[3] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *WWW*. 151–161.
[4] Jie Chen, Haw-ren Fang, and Yousef Saad. 2009. Fast Approximate kNN Graph Construction for High Dimensional Data via Recursive Lanczos Bisection. *Journal of Machine Learning Research* (2009).
[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. 1597–1607.
[6] Yu Chen, Lingfei Wu, and Mohammed Zaki. 2020. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. *Advances in Neural Information Processing Systems* (2020).
[7] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2020. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982* (2020).
[8] Bin Gao, Tie-Yan Liu, Wei Wei, Taifeng Wang, and Hang Li. 2011. Semi-supervised ranking on very large graphs with rich metadata. In *SIGKDD*. 96–104.
[9] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009* (2019).
[10] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*. 249–256.
[11] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*. 1025–1035.
[12] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *ICML*. PMLR, 4116–4126.
[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*. 9729–9738.
[14] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[16] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *SIGKDD*. 1531–1540.

[17] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *SIGIR*. 505–514.

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[20] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019).

[21] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *SIGKDD*. 1754–1763.

[22] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*.

[23] Ralph Linsker. 1988. Self-organization in a perceptual network. *Computer* (1988), 105–117.

[24] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *TKDE* (2021).

[25] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. 2014. Exploiting geographical neighborhood characteristics for location recommendation. In *CIKM*. 739–748.

[26] Weiran Pan, Wei Wei, and Xian-Ling Mao. 2021. Context-aware Entity Typing in Knowledge Graphs. In *Findings of EMNLP*. 1–8.

[27] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph representation learning via graphical mutual information maximization. In *WWW*. 259–270.

[28] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2021. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. *arXiv preprint arXiv:2110.05730* (2021).

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* (2012).

[30] Xiao Sha, Zhu Sun, and Jie Zhang. 2019. Attentive knowledge graph embedding for personalized recommendation. *arXiv preprint arXiv:1910.08288* (2019).

[31] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. 2018. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2018), 357–370.

[32] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197* (2019).

[33] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent knowledge graph embedding for effective recommendation. In *RecSys*. 297–305.

[34] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. *ICLR (Poster)* (2019), 4.

[35] Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, and Qi Liu. 2018. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In *WSDM*. 592–600.

[36] 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 *CIKM*. 417–426.

[37] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *WWW*. 1835–1844.

[38] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *SIGKDD*. 968–977.

[39] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-task feature learning for knowledge graph enhanced recommendation. In *WWW*. 2000–2010.

[40] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge graph convolutional networks for recommender systems. In *WWW*. 3307–3313.

[41] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *SIGKDD*. 950–958.

[42] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*. 165–174.

[43] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning Intents behind Interactions with Knowledge Graph for Recommendation. In *WWW*. 878–887.

[44] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. 2021. Self-supervised Heterogeneous Graph Neural Network with Co-contrastive Learning. *arXiv preprint arXiv:2105.09111* (2021).

[45] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *AAAI*, Vol. 33. 5329–5336.

[46] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. 2020. Global context enhanced graph neural networks for session-based recommendation. In *SIGIR*. 169–178.

[47] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*.

[48] Wei Wei, Bin Gao, Tie-Yan Liu, Taifeng Wang, Guohui Li, and Hang Li. 2015. A ranking approach on large-scale graph with multidimensional heterogeneous information. *IEEE transactions on cybernetics* 46, 4 (2015), 930–944.

[49] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. 726–735.

[50] 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 *SIGKDD*. 974–983.

[51] Xiao Yu, Xiang Ren, Quanquan Gu, Yizhou Sun, and Jiawei Han. 2013. Collaborative filtering with entity similarity regularization in heterogeneous information networks. *IJCAI HINA* (2013).

[52] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *WSDM*. 283–292.

[53] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *SIGKDD*. 353–362.

[54] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. 2021. Mining Latent Structures for Multimedia Recommendation. *arXiv preprint arXiv:2104.09036* (2021).

[55] Yongfeng Zhang, Qingyao Ai, Xu Chen, and Pengfei Wang. 2018. Learning over knowledge-base embeddings for recommendation. *arXiv preprint arXiv:1803.06540* (2018).

[56] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Meta-graph based recommendation fusion over heterogeneous information networks. In *SIGKDD*. 635–644.

[57] Sen Zhao, Wei Wei, Zou Ding, and Xian-Ling Mao. 2022. Multi-view Intent Disentangle Graph Networks for Bundle Recommendation. In *AAAI*. 1–7.

[58] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131* (2020).

[59] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *WWW*. 2069–2080.