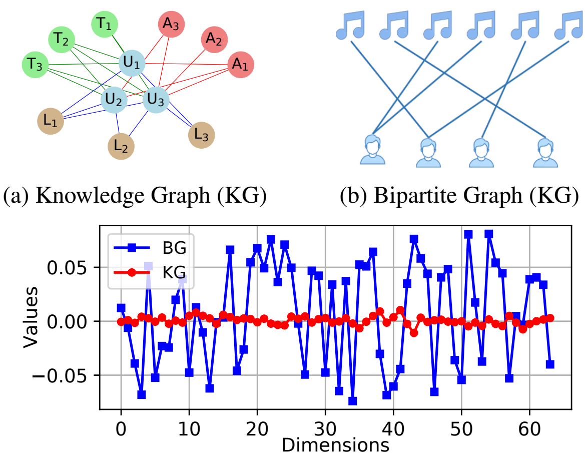
Multi-graph Fusion Cross-model Contrastive Learning for Recommendation

Shengjun Ma[1,2], Yuhai Zhao[1,2*], Fenglong Ma[3], Baoyin Liu[1,2], Zhengkui Wang[4], Wen Shan[5]

[1]School of Computer Science and Engineering, Northeastern University, China
[2]Key Laboratory of Intelligent Computing in Medical Image of Ministry of Education, Northeastern University, China
[3]College of Information Sciences and Technology, Pennsylvania State University, USA
[4]InfoComm Technology Cluster, Singapore Institute of Technology, Singapore
[5]Singapore University of Social Sciences, Singapore
masj1@mails.neu.edu.cn, zhaoyuhai@mail.neu.edu.cn, liubaoyiny@gmail.com
fenglong@psu.edu, zhengkui.wang@singaporetech.edu.sg, viviensw@suss.edu.sg

Abstract

Knowledge Graph (KG)-supported Graph Neural Network models are becoming crucial in recommendation systems due to their ability to mitigate the data sparsity challenge. However, these models remain suboptimal because they overlook the representation differences between the inherent user-item Bipartite Graph (BG) and the external head-relation-tail KG, leading to semantic misalignment. Moreover, they indiscriminately incorporate various types of relations from the KG, which may introduce noise information into the model, ultimately degrading recommendation performance. To address these challenges, we propose an end-to-end model named Multi-graph Fusion Cross-model Contrastive Learning (MFCCL). To uncover users' interest in items and explore the associations between items, we first construct a user-interest graph by integrating information from both the BG and KG, and an item-association graph derived from the BG. We devise a multi-graph representation learning module that incorporates rich semantics into user and item representations in parallel. Simultaneously, a classical collaborative filtering module is introduced to fully leverage user-item collaborative signals. Additionally, we design a novel free data-augmentation cross-model contrastive learning to facilitate the exchange of complementary information between different models. Empirical evaluations on three widely used benchmarks demonstrate that our MFCCL method achieved significant improvements over the baselines.



(a) Knowledge Graph (KG) (b) Bipartite Graph (KG)

(c) Embedding Difference

Figure 1: Representation Difference. As shown in (a), KG relations are diverse and abundant, BG interactions are sparse and singular in (b). In (c), item embeddings from the KG (red) and BG (blue) show significant representation differences, based on CKE results on the Last-FM dataset.

Introduction

Recommendation systems provide suggestions for items that are most relevant to a particular user, helping people filter content in the era of data explosion and readily available computing power networks (Zhang et al. 2025). They are used in a variety of areas such as e-commerce, social media, and many more (Zheng et al. 2023). To accurately predict users' preferences, Collaborative Filtering (CF) learns the likelihood of new interactions based on either user similarity or item similarity, and is extensively researched based on user-item historical interaction (Wu et al. 2023). CF has proven effective in predicting user preferences when there is abundant interaction data (Wu et al. 2023).

Side information, such as item attributes and user profiles, can essentially enhance CF (Wang et al. 2019a). Knowledge Graph (KG) is regarded as valuable auxiliary information and has shown great potential in improving recommendation performance. Most existing KG-aware methods adopt the classic two-stage approach, which first generates embeddings of the items through a KG-specific encoder and then treats them as prior information to supplement item representation (Cao et al. 2019). For example, CKE (Zhang et al. 2016) leverages TransR (Lin et al. 2015) to encode a KG as part of item representations. However, these methods overlooked the differences in representation between the inherent Bipartite Graph (BG) and the external KG, which can lead to semantic misalignment, as shown in Figure 1. Recently, some end-to-end models based on Graph Neural Network (GNN) (Wang et al. 2023) have been developed to effectively integrate high-order neighbors into represen-

*Corresponding author

tations, which combines the BG and the KG to form a larger Collaborative Knowledge Graph (CKG) and complex modeling of users and items (Wang et al. 2019b). For example, KGIN (Wang et al. 2021) explored user intents and relational path-aware aggregation. However, the diverse and complex relations in the KG may introduce noise, which not only expands the parameter in representation learning but also leads to optimization challenges during training.

Contrastive Learning (CL) has been increasingly utilized due to its remarkable ability to leverage unsupervised signals to alleviate the data sparsity issue (Zhang et al. 2023). Besides the classic CL methods like SGL (Wu et al. 2021) and XSimGCL (Yu et al. 2024), some KG-supported CL-based recommendation models, such as KGCL (Yang et al. 2022) and KGRec (Yang et al. 2023), have been proposed, achieving strong performance. KGCL employs graph contrastive learning on the KG to mitigate issues related to noise and long-tail distributions. By unifying generative and contrastive learning, KGRec aims to distill meaningful relational patterns from the KG. However, CL-based methods that rely on data augmentation often introduce noise, leading to sub-optimal model accuracy (Wang et al. 2025b).

To solve the above problems, we explore a multi-graph representation fusion that alleviates the representation gap between the KG and the BG, and design a cross-model CL that avoids the noise introduction through free data-augmentation, and propose a Multi-graph Fusion Cross-model Contrastive Learning model (MFCCL). To minimize CKG training costs and KG relational complexity, we first construct a User-interest Graph (UiG) based on the BG and the KG, and an Item-association Graph (IaG) based on the BG. The UiG connects users from the BG with item attributes from the KG, revealing users' preferences for specific features, while the IaG captures item co-occurrence relationships, uncovering intrinsic item-to-item associations. Secondly, we devise a multi-graph representation learning module that individually embeds informative signals into the learned representations of users and items, leveraging the BG and the UiG for users, the KG and the IaG for items. This module ensures that each graph contributes distinct insights, promoting consistency in user and item representations: users acquire attribute-level semantics from the UiG, while items do so from the KG. In parallel, a classical CF module is introduced to fully leverage user-item collaborative signals. Finally, to reduce the effect of noise, a novel free data-augmentation cross-model CL is designed to encourage the exchange of complementary knowledge between models. Experimental results on three widely used benchmarks demonstrate that our MFCCL significantly outperforms SOTA methods. We performed further analyses to validate the effectiveness of the proposed MFCCL. The contributions of this work are summarized as follows:

- We develop a multi-graph representation fusion that mitigates the representation gap between the BG and the KG, and introduce a free data-augmentation cross-model CL approach to enhance representation consistency.
- We propose a new end-to-end model, MFCCL, which achieves enhanced representation learning and effective contrastive training within a GNN framework.
- We conduct extensive experiments on three cutting-edge datasets to demonstrate the effectiveness of MFCCL and compare the performance to baselines.

Preliminaries

Bipartite Graph (BG): We typically have historical user-item interactions (e.g., views, purchases, and clicks) in a recommendation scenario. Let $\mathcal{U}$ be a set of users and $\mathcal{I}$ be a set of items and let $\mathcal{O}^+ = \{(u,i)|u \in \mathcal{U}, i \in \mathcal{I}\}$ be a set of implicit feedback, where each (u,i) pair indicates that user u has interacted with item i before. Generally, we also view interaction records as a $\mathcal{G}_{bg} = \{(u,i)|u \in \mathcal{U}, i \in \mathcal{I}\}$.

Knowledge Graph (KG): A KG is made up of two main components: entities and relations. Any object can be an entity, and a relation defines the relationship between the entities. Let $\mathcal{V}$ be a set of entities, and $\mathcal{R}$ be a set of relations. Commonly, a KG is organized in the form of subject-relation-object triple facts. Let $\mathcal{G}_{kg} = \{(h,r,t)|h,t \in \mathcal{V}, r \in \mathcal{R}\}$, where each triplet (h,r,t) describes that there is a relation r from head entity h to tail entity t.

Connecting BG and KG: Given both the $\mathcal{G}_{bg}$ and the $\mathcal{G}_{kg}$, we build a set of item-entity connection $\mathcal{C} = \{(i,v)|i \in \mathcal{I}, v \in \mathcal{V}\}$, where (i,v) indicates that the item i in the $\mathcal{G}_{bg}$ can be aligned with the entity v in the $\mathcal{G}_{kg}$. As mentioned, the set of items is typically a subset of the set of entities, i.e., $\mathcal{I} \subset \mathcal{V}$. With the alignment between the BG items and the KG entities, items can be profiled by the KG.

Task Formulation: We now formulate the task to be addressed in this work. Given the $\mathcal{G}_{bg}$ and the $\mathcal{G}_{kg}$, our task is to learn a function that can forecast how likely a user would interact with an item.

Methodology

We present the proposed MFCCL in this section. The overall workflow of MFCCL is shown in Figure 2. It consists of four key components: (1) multi-graph generation, (2) multi-graph representation fusion, (3) collaborative filtering, and (4) cross-model contrastive learning.

Multi-graph Generation

To reduce training costs on the CKG and minimize the impact of complex relations within the KG, we explicitly construct two auxiliary graphs: IaG and UiG.

Item-association Graph: The association between items not only reflects the relationship between them but also reveals users' intrinsic preference (Wang et al. 2025a). Although previous GNN-based CF methods can implicitly capture the association between items using multiple convolutional layers, the large number of high-order neighbors introduces a significant amount of noise during the multi-layer propagation process (Wang et al. 2019d). Hence, we explicitly build an IaG to capture relationships between items and users' purchasing habits. First, a item co-occurrence matrix $\mathcal{A}_{co}$ is calculated based on the $\mathcal{G}_{bg}$ via the equation.

$$\mathcal{A}_{co} = \mathcal{G}_{bg}^{\top}\mathcal{G}_{bg} \quad (1)$$

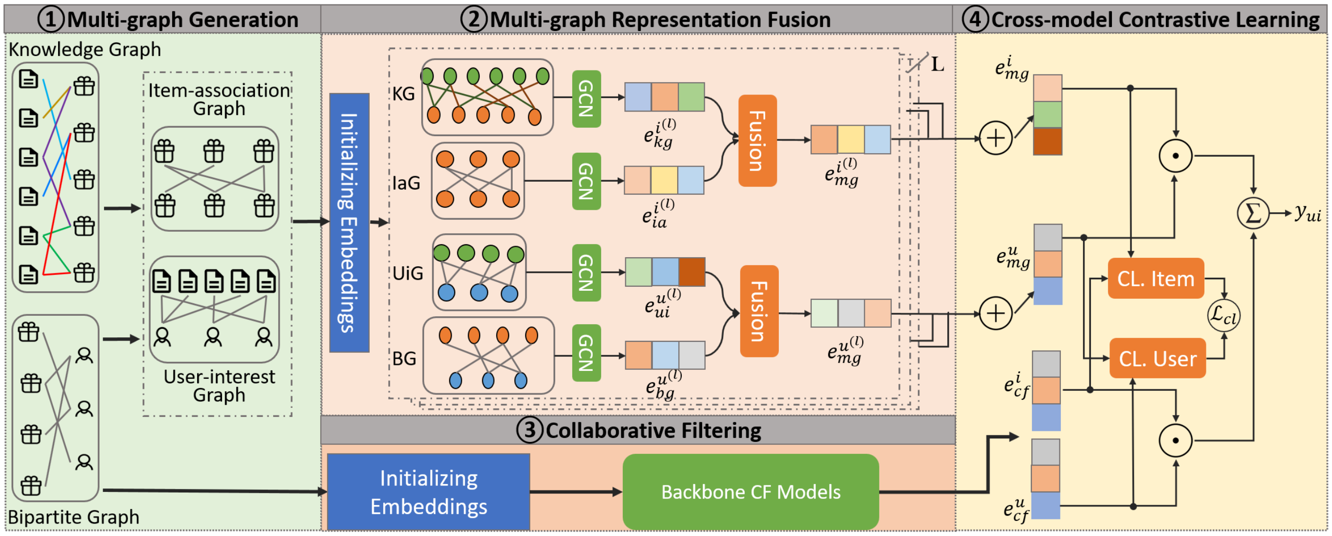


Figure 2: Overall Framework of MFCCL.

Then, the similarity score between each pair of items in the matrix $\mathcal{A}_{co}$ is computed, as shown in the following formula.

$$w_{i,j} = \frac{\mathcal{A}_{co}^{i,j}}{Deg(i) - \mathcal{A}_{co}^{i,i}} \sqrt{\frac{Deg(i)}{Deg(j)}} \quad (2)$$

where $w_{i,j}$ represents the similarity score between items i and j, $Deg(i)$ represents the degree of item i in the $\mathcal{G}_{bg}$, and $\mathcal{A}_{co}^{i,j}$ represents the value at the i-th row and the j-th column in the $\mathcal{A}_{co}$. Here, based on the similarity scores, k most similar items are selected for each item. These k selected items are then used as the neighbors of the original item in the graph $\mathcal{G}_{ia}$. The adjacency matrix of the $\mathcal{G}_{ia}$ is defined as;

$$\mathcal{A}_{ia}^{i,j} = \begin{cases} 1, & \text{if } w_{i,j} \in \text{top-k}(w_{i,:}) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

User-interest Graph: In practice, a user tends to interact with an item when its attributes align with the user's preferences. Therefore, the alignment between a user's preferences and an item's attributes significantly influences the probability of interaction (Wu et al. 2018). Hence, we explicitly build a UiG to capture the user's interest in the item's attributes. We initialize a matrix $\mathcal{A}_{ua}$, where each element $\mathcal{A}_{ua}^{i,j}$ measures the interest of user i in attribute j of items. In this context, the users are drawn from the BG, and the attributes of items are derived from the KG. The idea is that if an attribute occurs in most items clicked by a user, then it is likely that the user is interested in the attribute. To mitigate the impact of random behaviors and maintain a strong interest, we prune infrequent elements. The adjacency matrix of a $\mathcal{G}_{ui}$ defined as (where c is the pruning threshold);

$$\mathcal{A}_{ui}^{i,j} = \begin{cases} 1, & \text{if } \mathcal{A}_{ua}^{i,j} > c \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Multi-graph Representation Fusion

We employ a two-tower architecture (Yang et al. 2020) for its scalability and ability to capture user/item semantics, learning representations from multiple graphs.

Item Representation Learning: Besides the items' influential attributes, another reason a user interacts with an item is the association between items, as mentioned by the beer-diaper phenomenon. We perform representation learning and fusion for items in three steps.

(1) Representation learning based on the KG: In our experiments, we observed that the relations in the KG do not consistently improve the performance of representation learning. To enhance model efficiency while preserving relational semantics, we introduce a virtual relation mechanism that replaces real relations with a fixed set of virtual relations *(they are called virtual relations to indicate that they are generated)*. The virtual relations are initialized in the same vector space as entities to enable joint optimization during training. Specifically, the representation learning of item i based on the $\mathcal{G}_{kg}$ is given by the following equation;

$$e_{kg}^{i^{(l)}} = \sum_{0 \le p \le P} \beta(p,i) e_p \odot \frac{1}{\left|\mathcal{N}_{kg}^{i}\right|} \sum_{t \in \mathcal{N}_{kg}^{i}} e_t^{(l)} \quad (5)$$

where $e_t^{(l)} \in \mathbb{R}^d$ is the embedding of the attribute t at layer l; $\mathcal{N}_{kg}^{i}$ represents the set of relations and attributes associated with item i in the $\mathcal{G}_{kg}$, i.e. $\mathcal{N}_{kg}^{i} = \{(r,t)|(i,r,t) \in \mathcal{G}_{kg}\}$; $\odot$ represents the dot product of embeddings; $e_p \in \mathbb{R}^d$ is the embedding of the virtual relation p and P is a hyperparameter representing the number of the virtual relations; $e_{kg}^{i^{(l)}}$ is the representation embedding of the item i at layer l based on the $\mathcal{G}_{kg}$; and $\beta(p,i)$ is attention score between the virtual relation p and item i, and defined as follows;

$$\beta(p,i) = \frac{exp(e_p^{\top} e_{kg}^{i^l})}{\sum_{0 \le p \le P} exp(e_p^{\top} e_{kg}^{i^l})} \quad (6)$$

(2) Representation learning based on the IaG: The association between items not only reveals their similarity but also

provides essential signals for CF. We adopt the lightweight aggregation approach from LightGCN (He et al. 2020) to learn item representations from the $\mathcal{G}_{ia}$ by the equation;

$$e_{ia}^{i^{(l)}} = \frac{1}{\sqrt{\left|\mathcal{N}_{ia}^{i}\right|\left|\mathcal{N}_{ia}^{j}\right|}} \sum_{j \in \mathcal{N}_{ia}^{i}} e_{ia}^{j^{(l)}} \tag{7}$$

where $e_{ia}^{i^{(l)}}$ represents the embedding of item i based on the $\mathcal{G}_{ia}$ at layer l; $\mathcal{N}_{ia}^{i}$ represents the set of first-order neighbors of item i. As a result, the item embeddings incorporate shared features from similar items.

(3) Fusion of item representations: Although item representations can be obtained from the $\mathcal{G}_{kg}$ and the $\mathcal{G}_{ia}$, respectively, each representation only captures partial information about the item, leading to limitations in the final embedding. The two embeddings are fused through an attention mechanism, formulated as;

$$e_{mg}^{i^{(l+1)}} = att(e_{kg}^{i^{(l)}})e_{kg}^{i^{(l)}} + att(e_{ia}^{i^{(l)}})e_{ia}^{i^{(l)}} \tag{8}$$

where $e_{mg}^{i^{(l+1)}}$ is the embedding of the item i at layer $l+1$ in the multi-graph representation phase; and $att(\cdot)$ represents the attention score and can be calculated as;

$$att(e_{*}^{i^{(l)}}) = softmax(\mathcal{W}_{*}\left[e_{*}^{i^{(l)}}, e_{mg}^{i^{(l)}}\right]) \tag{9}$$

where $*$ represents kg or ia, $\mathcal{W}_{*}$ are the trainable weight matrices to distill information from the $\mathcal{G}_{kg}$ and the $\mathcal{G}_{ia}$.

User Representation Learning: Besides the user's long-term historical purchasing habit, another reason a user interacts with an item is a few of the item's attributes that are well-suited to the user's short-term preference. We perform representation learning and fusion for users in three steps.

(1) Representation learning oriented to habit: Capturing user long-term habits from the $\mathcal{G}_{bg}$ to learn user representations is a highly efficient and widely used technique. Borrowing a simple and efficient aggregation from LightGCN, the aggregation process is shown in the following equation;

$$e_{bg}^{u^{(l)}} = \frac{1}{\sqrt{\left|\mathcal{N}_{bg}^{u}\right|\left|\mathcal{N}_{bg}^{i}\right|}} \sum_{i \in \mathcal{N}_{bg}^{u}} e_{bg}^{i^{(l)}} \tag{10}$$

where $e_{bg}^{i^{(l)}} \in \mathbb{R}^{d}$ is the representation of the item i of the neighbors of the user u at layer l; $\mathcal{N}_{bg}^{*}$ represents the set of the first-order neighbors of the user u or the item i in the $\mathcal{G}_{bg}$; and $e_{bg}^{u^{(l)}}$ is the user u representation oriented to habit.

(2) Representation learning oriented to preference: For short-term preferences of users towards item attributes, we explicitly build the $\mathcal{G}_{ui}$ that connects users to item attributes, aiming to capture users' preferences for item attributes effectively. The aggregation of the preference representation is defined in the following equation.

$$e_{ui}^{u^{(l)}} = \frac{1}{\sqrt{\left|\mathcal{N}_{ui}^{u}\right|\left|\mathcal{N}_{ui}^{i}\right|}} \sum_{i \in \mathcal{N}_{ui}^{u}} e_{ui}^{i^{(l)}} \tag{11}$$

where $e_{ui}^{i^{(l)}} \in \mathbb{R}^{d}$ is the representation of the item i (item means attribute) of the neighbors of the user u at layer l; $\mathcal{N}_{ui}^{*}$ represents the set of the first-order neighbors of the user u or the item i in the $\mathcal{G}_{ui}$, and $e_{ui}^{u^{(l)}}$ is the user u representation oriented to preference.

(3) Fusion of user representations: As item representations, we calculate the fusion weights for the two embeddings of a user based on the attention mechanism and then perform a weighted fusion as the following equation;

$$e_{mg}^{u^{(l+1)}} = att(e_{bg}^{u^{(l)}})e_{bg}^{u^{(l)}} + att(e_{ui}^{u^{(l)}})e_{ui}^{u^{(l)}} \tag{12}$$

where $e_{mg}^{u^{(l+1)}}$ is the embedding of the user u at layer $l+1$ in the multi-graph representation phase; and $att(\cdot)$ represents the attention score and can be calculated as;

$$att(e_{*}^{u^{(l)}}) = softmax(\mathcal{W}_{*}\left[e_{*}^{u^{(l)}}, e_{mg}^{u^{(l)}}\right]) \tag{13}$$

where $*$ represents bg or ui, $\mathcal{W}_{*}$ are the trainable weight matrices to distill information from the $\mathcal{G}_{bg}$ and the $\mathcal{G}_{ui}$.

Collaborative Filtering

The BG-based CF representation learning has become a widely accepted technique for effectively capturing collaborative signals. We leverage the core of LightGCN as the CF model, and the user aggregation is defined as;

$$e_{cf}^{u^{(l+1)}} = \frac{1}{\sqrt{\left|\mathcal{N}_{bg}^{u}\right|\left|\mathcal{N}_{bg}^{i}\right|}} \sum_{i \in \mathcal{N}_{bg}^{u}} e_{cf}^{i^{(l)}} \tag{14}$$

where $e_{cf}^{u^{(l)}} \in \mathbb{R}^{d}$ and $e_{cf}^{i^{(l)}} \in \mathbb{R}^{d}$ represent the embedding of the user u and the item i at layer l, respectively. The aggregation process of items is similar to that of users. Notably, in this CF representation learning, the ID embedding of users and items based on the BG is initialized independently of the ID embeddings used in the multi-graph fusion representation learning. To distinguish, we denote the embedding of items and users in multi-graph representation learning as e_{mg}^{*} and in CF as e_{cf}^{*}, respectively.

Cross-model Contrastive Learning

Instead of generating augmented views, we construct contrastive pairs using two types of representations: one from the multi-graph representation module and the other from the CF module. For each item or user, the embeddings corresponding to the same ID from the two modules are treated as positive pairs, while those with different IDs are considered negative pairs. During the multi-layer propagation process, different layers capture different information because of neighbors at different hops. Therefore, we first sum the information from all layers to obtain the final representation embedding. The aggregation processes for the multi-graph representation and the CF representation are calculated as;

$$e_{*}^{u} = \sum_{0 \le l \le L} e_{*}^{u^{(l)}}, \quad e_{*}^{i} = \sum_{0 \le l \le L} e_{*}^{i^{(l)}} \tag{15}$$

where, $*$ represents mg or cf. We then utilize the InfoNCE (Gutmann and Hyvärinen 2010) to maximize the similarity between positive pairs while minimizing the similarity between negative pairs.

$$\mathcal{L}_{cl} = InfoNCE(i) + InfoNCE(u) \tag{16}$$

$$InfoNCE(*) = \sum_{*\in\mathcal{B}} -log \frac{exp(\frac{e_{cf}^{*\top} e_{mg}^{*}}{\tau})}{\sum_{j\in\mathcal{B}} exp(\frac{e_{cf}^{*\top} e_{mg}^{j}}{\tau})} \tag{17}$$

where $InfoNCE(\cdot)$ is the loss function; $*$ represents u or i; τ is the temperature parameter, $\mathcal{B}$ is the batch size, and $\mathcal{L}_{cl}$ refers to the cross-model contrastive loss.

Model Optimization

Without loss of generality, we use the BPR loss (Rendle et al. 2009) to measure the recommendation. The BPR optimizes item rankings for users by maximizing the probability that interacted items are ranked higher than those that were not interacted with, reflecting user preferences. The following formula represents the calculation.

$$\mathcal{L}_{bpr} = \sum_{u,i,j\in\mathcal{O}} -log\sigma(y_{ui} - y_{uj}) \tag{18}$$

where, $\mathcal{O} = \{(u,i,j)|(u,i) \in \mathcal{O}^+, (u,j) \in \mathcal{O}^-\}$ represents the training data, with $\mathcal{O}^+$ denotes positive pairs and $\mathcal{O}^-$ denotes negative pairs. $\sigma(\cdot)$ represents the sigmoid function, and y_{ui} and y_{uj} are predictions defined as follow;

$$y_{ui} = a * e_{mg}^{u}{}^{\top} e_{mg}^{i} + (1-a) * e_{cf}^{u}{}^{\top} e_{cf}^{i} \tag{19}$$

where a is a parameter that harmonizes the two representations. The model is trained by jointly optimizing BPR and cross-model contrastive loss.

$$\mathcal{L} = \mathcal{L}_{bpr} + \lambda_1 \mathcal{L}_{cl} + \lambda_2 \parallel \Theta \parallel_2^2 \tag{20}$$

where $\Theta = \{e_u^{(0)}, e_i^{(0)}, e_p^{(0)}, W_{kg}, W_{ia}, W_{bg}, W_{ui} | u \in \mathcal{U}, i \in \mathcal{I}, 1 \leq p \leq P\}$ is the set of model parameters; λ_1 and λ_2 are two hyperparameters to control the independence loss and L_2 regularization term, respectively.

Model Analysis

The UiG and IaG are constructed offline, incurring no runtime cost during training or inference. The main computational cost of MFCCL arises from the aggregation schemes in the multi-graph fusion and CF modules. The per-epoch time complexity is $O(L(3|U|+(P+2)|I|)d)$, where L, $|U|$, $|I|$, P and d denote the number of layers, users, items, virtual relations and embedding size, respectively. This complexity is comparable to other SOTA methods like KGCL and KGRec under the same settings.

Experiments

We design experiments to answer the following questions:

- **RQ1**: How does MFCCL perform, compared to the SOTA KG-supported recommendation models?
- **RQ2**: How do different graph components of multi-graph representation affect MFCCL?
- **RQ3**: How does the cross-model contrastive learning module in MFCCL affect performance?

Experimental Settings

Datasets Description: We evaluated our MFCCL on three public benchmarks: Alibaba-iFashion, Yelp2018, and Last-FM. We follow the data processing protocols used in previous works(Yang et al. 2023). The statistics of these datasets are summarized in Table 1.

		Alibaba-iFashion	Last-FM	Yelp2018
Bipartite Graph	#Users	114,737	23,566	45,919
	#Items	30,040	48,123	45,538
	#Interactions	1,781,093	3,034,796	1,185,068
	#Density	$5.2e^{-4}$	$2.7e^{-3}$	$5.7e^{-4}$
Knowledge Graph	#Entities	59,156	58,266	90,961
	#Relations	51	9	42
	#Triplets	279,155	464,567	1,853,704

Table 1: The Statistics of Datasets.

Evaluation Metrics: In the evaluation phase, we employ the all-ranking strategy. For each user, all items that they have not interacted with before are considered negative, while the relevant items in the testing set are treated as positive. All these items are ranked based on the predictions from the recommendation model. To evaluate top-K recommendations, we adopt the metrics: recall@K and ndcg@K, with K set to 20 by default, and we present the mean of multiple independent experiments.

Alternative Baselines: We compared MFCCL with the methods, **traditional CF models** (BPR (Rendle et al. 2009), NFM (He and Chua 2017)), **GNN-based CF methods** (GC-MC (Van Den Berg, Thomas, and Welling 2017), Light-GCN (He et al. 2020)), **GCL-based methods** (SGL (Wu et al. 2021), SimGCL (Yu et al. 2022), XSimGCL (Yu et al. 2024)), **KG-supported methods** (CKE (Zhang et al. 2016), KTUP (Cao et al. 2019), KGCN (Wang et al. 2019b), KGAT (Wang et al. 2019c), KGIN (Wang et al. 2021)), and **methods KG-supported and CL-based** (KGCL (Yang et al. 2022), KGRec (Yang et al. 2023)).

Parameter Settings: We implemented the proposed MFCCL using PyTorch. For a fair comparison, the embedding size is fixed to 64 for all models. We optimize all models with the Adam optimizer, where the batch size is fixed at 1024. We use the Xavier initializer to initialize model parameters. We apply a grid search for hyper-parameters: the learning rate is tuned in $\{1e^{-3}, 5e^{-4}, 1e^{-4}\}$, the number of GNN layers L is tuned in $\{2, 3\}$ for GNN-based models, and the edges dropout ratio is tuned in $\{0.3, 0.5\}$. We set k to 10 in the Equation 3, the pruning threshold c to 2 in the Equation 4. The number of virtual relations P in the Equation 5 is tuned in $\{1, \cdots, 2^3\}$. The temperature value in the InfoNCE is in the range of $\{0.1, \cdots, 0.5\}$. The harmonizing parameter for two representations a in the Equation 19 is tuned in $\{0.1, \cdots, 0.8\}$. The coefficient of L_2 regularization is set to 0.0001, and the coefficient for cross-model contrastive learning is equal to $\{1e^{-3}, \cdots, 1e^{-5}\}$ in the objective function.

Performance Comparison (RQ1)

The overall performance comparison results are presented in Table 2, where the best evaluation results within the baseline methods are indicated with an underline, and the best results among all methods are highlighted in bold.

Model	Alibaba-iFashion		Last-FM		Yelp2018	
	recall	ndcg	recall	ndcg	recall	ndcg
NFM	0.1095	0.0670	0.0673	0.0538	0.0418	0.0258
BPR	0.0822	0.0501	0.0690	0.0585	0.0568	0.0462
GC-MC	0.0845	0.0502	0.0709	0.0631	0.0688	0.0453
LightGCN	0.1091	0.0682	0.0699	0.0617	0.0686	0.0446
SGL	0.1126	0.0538	0.0816	0.0725	0.0675	0.0555
SimGCL	0.1151	0.0567	0.0802	0.0718	0.0721	0.0601
XSimGCL	<u>0.1196</u>	0.0586	0.0884	0.0791	0.0723	**<u>0.0604</u>**
CKE	0.0835	0.0512	0.0845	0.0718	0.0653	0.0423
KTUP	0.0976	0.0634	0.0865	0.0671	0.0672	0.0411
KGCN	0.0983	0.0633	0.0879	0.0694	0.0532	0.0338
KGAT	0.1030	0.0627	0.0870	0.0743	0.0705	0.0463
KGIN	0.1147	0.0716	<u>0.0978</u>	<u>0.0848</u>	0.0705	0.0451
KGCL	0.1146	0.0719	0.0905	0.0769	<u>0.0756</u>	0.0441
KGRec	0.1188	<u>0.0743</u>	0.0943	0.0810	0.0745	0.0478
MFCCL	**0.1288**	**0.0832**	**0.1025**	**0.0978**	**0.0774**	0.0501

Table 2: Overall Performance Comparison.

MFCCL achieves consistently good performance on nearly all metrics across all datasets. In particular, it outperforms the well-established baselines (underlined) in Recall@20 by 4.04%, 4.81%, and 2.38% on Alibaba-iFashion, Last-FM, and Yelp2018, respectively, demonstrating its effectiveness. These gains are attributed to two key components: (1) the multi-graph representation module, which enables users to capture both long-term habits and short-term preferences, and items to model intrinsic attributes and inter-item relationships; and (2) the cross-model CL module, which effectively combines multi-graph semantic information with structural signal from CF, enhancing the robustness of representations against noise. **Traditional factorization models**, such as NFM and BPR, performed poorly due to their reliance on linear and quadratic interactions, which are insufficient to capture complex high-order dependencies. **GNN-based CF models**, including GC-MC and LightGCN, show noticeable improvements over factorization methods, but still suffer from the sparsity of the BG, limiting their effectiveness. **Classical CL-based models**, such as SGL and XSimGCL, achieved strong performance by applying simple augmentations on graph structure or embedding dimensions based on LightGCN, leading to significant gains over their base model. **KG-supported models**, such as the two-stage KTUP and the end-to-end KGIN, delivered competitive results, showing that incorporating KGs can substantially enhance recommendation performance. Finally, **KG-supported CL-based models**, including KGCL and KGRec, demonstrated stable performance, offering more advantages over methods based solely on KG or CL.

Multi-graph Representation Effects (RQ2)

Since multi-graph representation learning is the core of MFCCL, we conducted comprehensive ablation studies to evaluate its effectiveness. As shown in Table 3, we compared several model variants against the MFCCL, reporting relative performance drops. Specifically, **MFCCL-uig** removes the UiG and **MFCCL-ubg** excludes the BG for user representation; **MFCCL-iag** omits the IaG and **MFCCL-ikg** removes the KG for item representation; and **MFCCL-vr** replaces the proposed virtual relations with the original relations in the KG.

The ablation results in Table 3 show that removing any graph component, whether on the user side (UiG or BG) or item side (IaG or KG), leads to consistent performance drops (1.26%–8.90% relative reduction), confirming that each graph captures complementary information. On the user side, the BG proves most critical, serving as the backbone of user representation, while the UiG offers supplementary behavioral signals. On the item side, the KG and IaG show comparable impact, indicating that both structured knowledge and co-occurrence patterns contribute distinct yet equally important insights for item representation. Our virtual relations significantly boost item representation, improving ndcg@20 by 7.21% (Alibaba-iFashion) and 8.28% (Last-FM) over original KG relations.

Cross-model Contrastive Learning Effects (RQ3)

We conduct ablation studies to assess the impact of the proposed cross-model contrastive learning (CMC) module. Specifically, we compare the MFCCL with a variant (**MFCCL-cmc**) that removes the CMC module, and analyze how varying its loss coefficient affects overall performance.

The ablation study demonstrates the critical importance of the CMC module in MFCCL. Experimental results show in Table 3 that removing CMC leads to significant performance drops (13.85%-22.29% across metrics), confirming its essential role in effectively aligning multi-graph semantics with CF structures for users and items. This alignment is crucial for maintaining the model's recommendation accuracy and robustness. To further evaluate the effect of the CMC module, we study both its loss magnitude and training dynamics (Figure 3). On the left, setting the contrastive loss coefficient to 0.0001 yields the best Recall@20 across all datasets. On the right, we observe that MFCCL converges faster than the variant on the Last-FM dataset, reaching early stopping at epoch 100. Moreover, as training, MFCCL shows a more rapid and consistent increase in Recall, highlighting the CMC module's effectiveness.

Hyperparameter Sensitivity Analysis

We examine the robustness of MFCCL of key three hyperparameters: the number of the virtual relations P, the harmonization parameter α, and temperature parameter τ.

A moderate P enhances performance by simplifying KG and facilitating feature fusion, while an excessive P leads to performance degradation, as shown in Figure 4 (a). This confirms the effectiveness of the virtual relations we designed, while also highlighting the intrinsic challenges in

	Alibaba-iFashion				Last-FM			
	recall@20	recall@40	ndcg@20	ndcg@40	recall@20	recall@40	ndcg@20	ndcg@40
MFCCL-vr	0.1232$_{4.35\%}$	0.1765$_{2.16\%}$	0.0772$_{7.21\%}$	0.0952$_{3.35\%}$	0.1010$_{1.46\%}$	0.1327$_{1.56\%}$	0.0897$_{8.28\%}$	0.0999$_{1.87\%}$
MFCCL-uig	0.1276$_{0.93\%}$	0.1781$_{1.28\%}$	0.0807$_{3.00\%}$	0.0972$_{1.32\%}$	0.1019$_{0.59\%}$	0.1345$_{0.22\%}$	0.0904$_{7.57\%}$	0.1009$_{0.88\%}$
MFCCL-ubg	0.1264$_{1.86\%}$	0.1779$_{1.39\%}$	0.0816$_{1.92\%}$	0.0968$_{1.73\%}$	0.1011$_{1.37\%}$	0.1327$_{1.56\%}$	0.0891$_{8.90\%}$	0.0994$_{2.36\%}$
MFCCL-iag	0.1260$_{2.17\%}$	0.1777$_{1.50\%}$	0.0814$_{2.16\%}$	0.0967$_{1.83\%}$	0.1012$_{1.27\%}$	0.1328$_{1.48\%}$	0.0892$_{8.79\%}$	0.0994$_{2.36\%}$
MFCCL-ikg	0.1262$_{2.02\%}$	0.1778$_{1.44\%}$	0.0815$_{2.04\%}$	0.0969$_{1.63\%}$	0.1012$_{1.27\%}$	0.1331$_{1.26\%}$	0.0892$_{8.79\%}$	0.0995$_{2.26\%}$
MFCCL-cmc	0.1094$_{15.06\%}$	0.1515$_{16.02\%}$	0.0673$_{19.11\%}$	0.0766$_{22.23\%}$	0.0883$_{13.85\%}$	0.1155$_{14.32\%}$	0.0760$_{22.29\%}$	0.0897$_{11.89\%}$
MFCCL	**0.1288**	**0.1804**	**0.0832**	**0.0985**	**0.1025**	**0.1348**	**0.0978**	**0.1018**

Table 3: Ablation Performance Comparison on MFCCL's Graph Components.

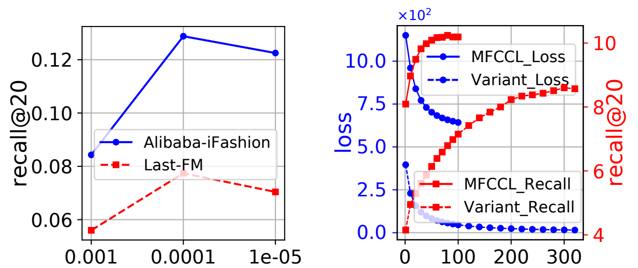
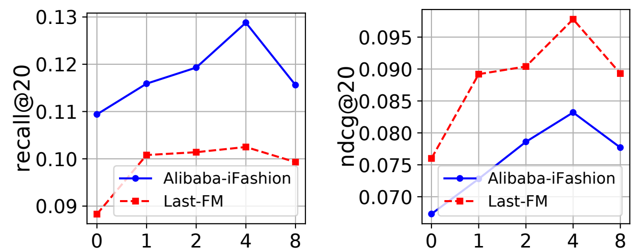
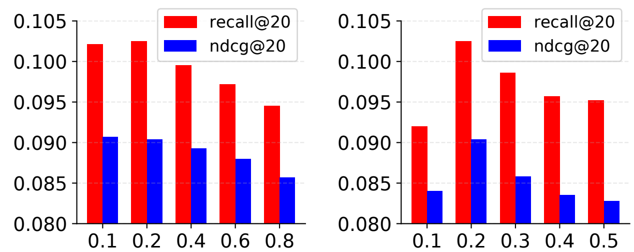


Figure 3: Ablation Performance of the CMC Module (Left: λ_1 weight for the CL, Right: training dynamics across epochs on Last-FM).



(a) Number of the virtual relations P.



(b) Harmonization parameter α. (c) Temperature parameter τ.

Figure 4: Hyperparameter Sensitivity Analysis.

KG relation modeling. A systematic comparison of virtual relation generation strategies is crucial for the next step. On the Last-FM, $\alpha = 0.2$ in Figure 4 (b) yields the best performance, suggesting an optimal balance between the multi-graph semantic information and the CF structural signal. Meanwhile, $\tau = 0.2$ in Figure 4 (c) provides the optimal temperature setting, as smaller values lead to over-concentration on hard negatives, while larger values dilute the contrastive signal. These results align with prior CL studies.

Related Work

Our proposed method is closely related to four lines of research: collaborative filtering (CF), graph neural networks (GNN), contrastive learning (CL), and Knowledge Graph (KG)-enhanced recommendation. Traditional CF methods (Wu et al. 2023; Baltrunas, Ludwig, and Ricci 2011; Rendle 2010) capture user-item interactions via latent embeddings, while recent GNN-based models (Wang et al. 2019d; He et al. 2020; Mao et al. 2021) model high-order connectivity on user-item graphs through simplified yet effective architectures. CL has been applied to enhance graph-based recommenders (Wu et al. 2021; Yu et al. 2022; Wei et al. 2022) by generating contrastive views via data augmentation or selective edge/node dropping. KG provide auxiliary semantics to improve recommendation accuracy (Zhang et al. 2016; Yang et al. 2023, 2022), where both embedding propagation and contrastive denoising strategies are adopted.

Conclusion

In this paper, we tackled the representation incompatibility issue in KG-supported GNN-based recommendation systems by proposing a novel framework, MFCCL. MFCCL explicitly constructs a UiG and an IaG to leverage both the BG and the KG, and integrates their representations via a multi-graph fusion module. It further employs a cross-model CL mechanism to align user and item representations from different sources, enabling complementary learning. Extensive experiments on three cutting-edge datasets show that MFCCL consistently outperforms SOTA methods. Ablation studies confirm the importance of multi-graph representation learning for modeling preferences and associations, and highlight the effectiveness of cross-model CL in improving robustness and alignment. In future work, we will leverage computing power networks to incorporate LLMs for generating richer side information. This approach aims to alleviate data sparsity and enhance representation quality through advanced NLP techniques.

Acknowledgments

This research acknowledges the generous support provided by the National Natural Science Foundation of China under Grant No. 62432003, and the Singapore Institute of Technology Ignition Grant under Grant No. R-IE3-A405-0005.

References

Baltrunas, L.; Ludwig, B.; and Ricci, F. 2011. Matrix factorization techniques for context aware recommendation. In Mobasher, B.; Burke, R. D.; Jannach, D.; and Adomavicius, G., eds., *ACM Conference on Recommender Systems*, 301–304. ACM.

Cao, Y.; Wang, X.; He, X.; Hu, Z.; and Chua, T. 2019. Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences. In *Proceedings of the ACM on Web Conference 2019*, 151–161. ACM.

Gutmann, M.; and Hyvärinen, A. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, 297–304. JMLR.org.

He, X.; and Chua, T. 2017. Neural Factorization Machines for Sparse Predictive Analytics. In Kando, N.; Sakai, T.; Joho, H.; Li, H.; de Vries, A. P.; and White, R. W., eds., *Proceedings of the 40th International SIGIR Conference on Research and Development in Information Retrieval*, 355–364. ACM.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 639–648. ACM.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2181–2187. AAAI Press.

Mao, K.; Zhu, J.; Xiao, X.; Lu, B.; Wang, Z.; and He, X. 2021. UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation. In *CIKM '21*, 1253–1262. ACM.

Rendle, S. 2010. Factorization Machines. In *The 10th IEEE International Conference on Data Mining*, 995–1000. IEEE Computer Society.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Conference on Uncertainty in Artificial Intelligence*. AUAI Press.

Van Den Berg, R.; Thomas, N. K.; and Welling, M. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2(8): 9.

Wang, H.; Zhao, M.; Xie, X.; Li, W.; and Guo, M. 2019a. Knowledge Graph Convolutional Networks for Recommender Systems. In *The World Wide Web Conference*, 3307–3313. ACM.

Wang, H.; Zhao, M.; Xie, X.; Li, W.; and Guo, M. 2019b. Knowledge Graph Convolutional Networks for Recommender Systems. In *The World Wide Web Conference*, 3307–3313. ACM.

Wang, Q.; Lin, D.; Lin, W.; and Wu, Z. 2025a. FROG: Effective Friend Recommendation in Online Games via Modality-aware User Preferences. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2822–2826. ACM.

Wang, X.; He, X.; Cao, Y.; Liu, M.; and Chua, T.-S. 2019c. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD*, 950–958.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T. 2019d. Neural Graph Collaborative Filtering. In *SIGIR 2019*, 165–174. ACM.

Wang, X.; Huang, T.; Wang, D.; Yuan, Y.; Liu, Z.; He, X.; and Chua, T.-S. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the web conference 2021*, 878–887.

Wang, Y.; Zhao, Y.; Wang, D. Z.; and Li, L. 2023. GALOPA: Graph Transport Learning with Optimal Plan Alignment. *Advances in Neural Information Processing Systems*, 36: 9117–9130.

Wang, Y.; Zhao, Y.; Wang, Z.; Li, L.; Wang, J.; Li, F.; Huang, M.; Pan, S.; and Wang, X. 2025b. Equivalence is All: A Unified View for Self-supervised Graph Learning. In *Forty-second International Conference on Machine Learning*.

Wei, C.; Liang, J.; Liu, D.; and Wang, F. 2022. Contrastive Graph Structure Learning via Information Bottleneck for Recommendation. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems*.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021. Self-supervised Graph Learning for Recommendation. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, 726–735. ACM.

Wu, L.; He, X.; Wang, X.; Zhang, K.; and Wang, M. 2023. A Survey on Accuracy-Oriented Neural Recommendation: From Collaborative Filtering to Information-Rich Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5): 4425–4445.

Wu, P.; Tu, Y.; Yang, Z.; Jatowt, A.; and Odagaki, M. 2018. Deep Modeling of the Evolution of User Preferences and Item Attributes in Dynamic Social Networks. In *Proceedings of the ACM on Web Conference 2018*, 115–116. ACM.

Yang, J.; Yi, X.; Cheng, D. Z.; Hong, L.; Li, Y.; Wang, S. X.; Xu, T.; and Chi, E. H. 2020. Mixed Negative Sampling for Learning Two-tower Neural Networks in Recommendations. In *Companion of The 2020 Web Conference*, 441–447. ACM / IW3C2.

Yang, Y.; Huang, C.; Xia, L.; and Huang, C. 2023. Knowledge Graph Self-Supervised Rationalization for Recommendation. In *KDD 2023*, 3046–3056. ACM.

Yang, Y.; Huang, C.; Xia, L.; and Li, C. 2022. Knowledge graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on* *research and development in information retrieval*, 1434–1443.

Yu, J.; Xia, X.; Chen, T.; Cui, L.; Hung, N. Q. V.; and Yin, H. 2024. XSimGCL: Towards Extremely Simple Graph Contrastive Learning for Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 36(2): 913–926.

Yu, J.; Yin, H.; Xia, X.; Chen, T.; Cui, L.; and Nguyen, Q. V. H. 2022. Are Graph Augmentations Necessary?: Simple Graph Contrastive Learning for Recommendation. In *SIGIR '22*, 1294–1303. ACM.

Zhang, C.; Huang, C.; Tian, Y.; Wen, Q.; Ouyang, Z.; Li, Y.; Ye, Y.; and Zhang, C. 2023. When Sparsity Meets Contrastive Models: Less Graph Data Can Bring Better Class-Balanced Representations. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 41133–41150. PMLR.

Zhang, F.; Yuan, N. J.; Lian, D.; Xie, X.; and Ma, W. 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. ACM.

Zhang, X.; Xu, B.; Li, C.; He, B.; Lin, H.; Ma, C.; and Ma, F. 2025. A Survey on Side Information-Driven Session-Based Recommendation: From a Data-Centric Perspective. *IEEE Transactions on Knowledge and Data Engineering*, 37(8): 4411–4431.

Zheng, J.; Gu, H.; Song, C.; Lin, D.; Yi, L.; and Chen, C. 2023. Dual Interests-Aligned Graph Auto-Encoders for Cross-domain Recommendation in WeChat. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 4988–4994.