

Mixed Information Flow for Cross-Domain Sequential Recommendations

MUYANG MA, PENGJIE REN, ZHUMIN CHEN, ZHAOCHUN REN, and
LIFAN ZHAO, Shandong University
PEIYU LIU, Shandong Normal University
JUN MA, Shandong University
MAARTEN DE RIJKE, University of Amsterdam

Cross-domain sequential recommendation is the task of predict the next item that the user is most likely to interact with based on past sequential behavior from multiple domains. One of the key challenges in **cross-domain sequential recommendation** is to grasp and transfer the flow of information from multiple domains so as to promote recommendations in all domains. Previous studies have investigated the flow of *behavioral* information by exploring the connection between items from different domains. The flow of knowledge (i.e., the connection between knowledge from different domains) has so far been neglected. In this article, we propose a **mixed information flow network** for **cross-domain sequential recommendation** to consider both the flow of behavioral information and the flow of knowledge by incorporating a **behavior transfer unit** and a **knowledge transfer unit**. The proposed **mixed information flow network** is able to decide when cross-domain information should be used and, if so, which cross-domain information should be used to enrich the sequence representation according to users' current preferences. Extensive experiments conducted on four e-commerce datasets demonstrate that the proposed **mixed information flow network** is able to improve recommendation performance in different domains by modeling mixed information flow. In this article, we focus on the application of **mixed information flow networks** to a scenario with two domains, but the method can easily be extended to multiple domains.

CCS Concepts: • **Information systems** → **Recommender systems**;

Additional Key Words and Phrases: Cross-domain recommendation, sequential recommendation, knowledge base, graph transfer

This research was partially supported by the Key Research and Development Program of China under Grant (2018YFC0831000), the Natural Science Foundation of China (62072279, 61972234, 61902219, 62102234), the Key Scientific and Technological Innovation Program of Shandong Province (2019JZZY010129), the Natural Science Foundation of Shandong Province (ZR202102230192), the Tencent WeChat Rhino-Bird Focused Research Program (JR-WXG-2021411), the Fundamental Research Funds of Shandong University, the National Key R&D Program of China (2020YFB1406700), the Meituan incorporation, and the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organization for Scientific Research, <https://hybrid-intelligence-centre.nl>. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

Authors' addresses: M. Ma, P. Ren (corresponding author), Z. Chen (corresponding author), Z. Ren, L. Zhao, and J. Ma, Shandong University, Qingdao 266237, China; emails: muyang0331@gmail.com, jay.ren@outlook.com, {chenzhumin, zhaochun.ren}@sdu.edu.cn, mogician233@outlook.com, majun@sdu.edu.cn; P. Liu, Shandong Normal University, Jinan 250014, China; email: liupy@sdu.edu.cn; M. de Rijke, University of Amsterdam, Amsterdam 1012 WX, The Netherlands; email: m.derijke@uva.nl.

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

© 2022 Association for Computing Machinery.

1556-4681/2022/01-ART64 \$15.00

<https://doi.org/10.1145/3487331>

ACM Reference format:

Muyang Ma, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Lifan Zhao, Peiyu Liu, Jun Ma, and Maarten de Rijke. 2022. Mixed Information Flow for Cross-Domain Sequential Recommendations. *ACM Trans. Knowl. Discov. Data.* 16, 4, Article 64 (January 2022), 32 pages.
<https://doi.org/10.1145/3487331>

1 INTRODUCTION

Sequential recommendation (SR) aims at predicting the next item that the user is most likely to interact with based on her/his past sequential behavior (e.g., clicks on items) [57]. Recently, **cross-domain sequential recommendation (CDSR)** has emerged as a way to promote recommendation performance by leveraging and combining information from different domains [47]. Users usually have related preferences in different domains, such as finding a movie with a certain style or looking for a book written by a well-known author, as illustrated in Figure 1. One of the key challenges in CDSR is to capture and transfer useful information about related preferences across different domains.

Zhuang et al. [92] and Ma et al. [47] have shown that behavioral information across domains is helpful for improving recommendation performance. However, behavioral information by itself can only support the use of cross-domain connections in a limited manner. Behavioral information may be insufficient for a model to capture fine-grained connections between item attributes or features. As an example, consider Figure 1 and assume that there is a user who has read *Harry Potter* (the book) and watched *Captain America* (the movie). If there is no external knowledge to indicate that both items belong to the category of “fantasy,” it is difficult for the model to capture this connection based solely on the user’s behavior from both domains. When supported by external knowledge, the model can capture this characteristic from the knowledge information flow, and this information can be used to model user preferences across different domains. We hypothesize that enabling a flow of knowledge across different domains is able to alleviate this issue. As a result, for a user who has read the book *The Great Gatsby*, we then can recommend her/him a movie having the same name or movies featuring the same category of “tragic love,” such as *Atonement*, *Waterloo Bridge*, and so on, when she/he logs on to the movie recommendation system.

There is a growing body of work aimed at improving recommendation performance by using knowledge [43, 59]. Of particular relevance to us is work that has proposed to incorporate knowledge and combine it with behavioral information for SR [see e.g., 34, 35, 82]. However, this work targets a single domain recommendation scenario. The situation is dramatically different in cross-domain scenarios where it is necessary to distinguish information from different domains and effectively link them. We need to select behavioral and knowledge related to users’ current preference, and determine when and what to use in order to learn a better sequence representation.

To address the issue of using behavioral information and knowledge across domains, we propose a **mixed information flow network (MIFN)** to consider mixed information flow across domains, i.e., the flow of behavioral information as well as the flow of knowledge. The former is based on user’s behavior, which captures the temporal connection between the items they have interacted with, while the latter takes cross-domain knowledge as a bridge to connect different domains to obtain better cross-domain sequence representations. First, we employ a **behavior transfer unit (BTU)** to grasp useful information from the flow of behavioral information, which can extract information related to the user’s preference and then transfer it to another domain at the level of user behavior level. Then, we propose a **knowledge transfer unit (KTU)** that is guided by the user’s preference to model the connection between items from different domains; we introduce a **cross-domain graph convolutional mechanism** to distinguish items in the

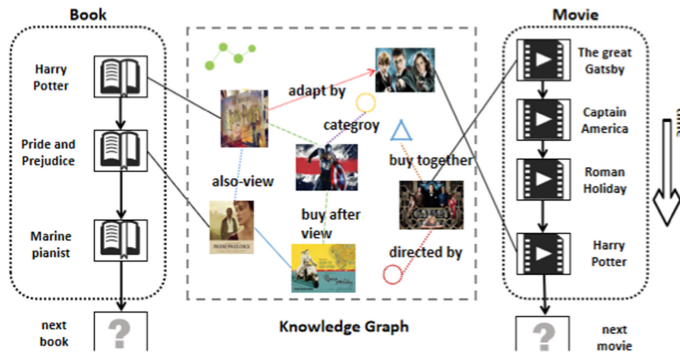


Fig. 1. Illustration of a connection between a user's behavioral information across domains via a **knowledge graph**. Lines in different colors represent different connections.

knowledge graph (KG) and grasp useful information for fusion at the knowledge level. Finally, we generate recommendations based on the fusion of the two types of information. During learning, **mixed information flow network** (MIFN) is jointly trained on multiple domains in an end-to-end back-propagation training paradigm. Experiments on the Amazon datasets show that MIFN outperforms state-of-the-art methods in terms of MRR and Recall.

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

- We propose a mixed information flow framework, MIFN, for CDSR, which consists of a **behavior transfer unit** and a **knowledge transfer unit** to simultaneously model the flow of behavioral information and of knowledge across domains.
- We devise a **cross-domain graph convolutional mechanism** to disseminate item information in the KG, which leads to the better up-to-date item representation.
- We conduct experiments to demonstrate that MIFN is able to improve recommendation performance in different domains by modeling mixed information flow.

2 RELATED WORK

In this section, we briefly introduce related work from the following categories: (1) SR, (2) cross-domain recommendation, and (3) knowledge-aware recommendation.

2.1 Sequential Recommendations

Early work on **recommender systems** (RSs) typically uses **collaborative filtering** (CF) to generate recommendations [32] according to users' preferences reflected in similar items such as **K-Nearest neighbors** (KNN) or **matrix factorization** (MF) algorithms. Such methods do not consider sequential aspects. More recently, however, SR and next-basket recommendation have witnessed rapid developments.

Before the widespread application of deep learning, **Markov chains** (MCs) [10, 24, 62, 93] and **Markov decision processes** (MDPs) [65, 83] were used to predict users' next actions given information about their past behavior [77, 86]. All these methods take into account sequential characteristics. However, there are considerable challenges with the size of the state space when considering arbitrary sequences [56].

Recurrent neural networks (RNNs) have been introduced to SR to handle variable-length sequential data. Hidasi et al. [27] are the first to leverage **recurrent neural networks** (RNNs) for SR. They utilize session-parallel mini-batch training and employ ranking-based loss functions to

train the model. Then, Tan et al. [69] propose two techniques to improve the performance, i.e., data augmentation and a method to account for shifts in the input data distribution. Li et al. [40] incorporate an attention mechanism into the encoder to capture the users' main preference in the current sequence. Ren et al. [58] point out the repeat consumption in SR, where the same item is re-consumed repeatedly over time. Quadrana et al. [57] propose a hierarchical RNN model that can relay and evolve latent hidden states of the RNNs across user sequences. Donkers et al. [16] explicitly model user information in a gated architecture with extra input layers for **gated recurrent unit** (GRU). Chen et al. [12] argue that most work considers user preferences to be static and relies on post-processing the list of recommendations to promote diversity. They propose an intent-aware diversified network to improve accuracy and diversity by introducing self-attention mechanism based on GRU networks. Memory enhanced RNN has been well studied for SR recently. Chen et al. [13] introduce a memory mechanism to SR and design a memory-augmented neural network integrated with the insights of CF. Wang et al. [75] propose two parallel memory modules: one to model a user's own information in the current sequence and the other to exploit collaborative information in neighborhood sequences. Wu et al. [82] argue that prior work on conventional sequential methods neglects complex transitions between items. They model the sequence as graph-structured data and then represent it as the composition of global preference and the current preference of that sequence using an attention network. Zhang et al. [88] propose a feature-level deeper self-attention network to capture transition patterns between features of items by integrating various heterogeneous features. Sun et al. [68] argue that previous work often assumes a rigidly ordered sequence, which is not always practical. They employ deep bidirectional self-attention to model a user's behavioral sequences.

In addition to sequential information, auxiliary information is also vital for SR. Hidasi et al. [28] investigate how to add item property information such as text and images to an RNNs framework and introduce a number of parallel RNN (p-RNN) architectures. Liu et al. [44] incorporate contextual information into SR and propose a context-aware RNN model to capture external situations and lengths of time intervals. Bogina and Kuflik [9] explore a user's dwell time based on an existing RNN-based framework by boosting items above a predefined dwell time threshold. Ma et al. [48] propose a cross-attention memory network for multi-modal tweets via both textual and visual information. Li et al. [39] study how to enlist the semantic signals covered by user reviews for the task of CF. They propose a neural review-driven model by considering users' intrinsic preference and sequential patterns. To investigate the influence of temporal sentiments on user preference, Zheng et al. [91] propose to generate preferences by guiding user behavior through sequential sentiments. They design a dual-channel fusion mechanism to match and guide sequential user behavior, and to assist in preference generation. Ren et al. [60] model the effect of context information on SR and train the model in an adversarial manner by proposing multiple context-specific discriminators to evaluate the generated sub-sequence from the perspectives of different contexts.

Although these studies have made great progress, none of them has considered how to combine knowledge information under cross-domain situations.

2.2 Cross-Domain Recommendations

Cross-domain recommendation has emerged as a potential solution to the cold-start and data-sparse problem [6, 55] in RS. It aims at mitigating the lack of data by exploiting user preference and item attributes in domains distinct but related to the target domain [20].

Traditional methods for cross-domain recommendation can be grouped into two main categories [18]. One category of methods *aggregates information* across different domains. According to different aggregating strategies, such methods can be further divided into three groups [20]. The

first group is *merging user preference* (e.g., ratings, transaction behavior, and browsing logs) from different domains to obtain better a preference representation so as to improve the recommendation performance in the target domain. The merge operation is performed by merging a multi-domain rating matrix [5, 63], leveraging users' social influence [1, 19], linking users' preference by a multi-domain graph [15, 70] or user behavioral information features [46, 50]. The second group is *mediating user modeling data* in the source domain to explore the connection between users or items so as to make recommendations in the target domain especially for cold start users. For example, Tiroshi and Kuflik [71] and Shapira et al. [66] propose to find similar neighbors and transfer user-user similarity to the target domain. The third group is *combining single-domain recommendations* (e.g., rating matrices, probability distributions), in which recommendations are generated independently for each domain and later aggregated for the final recommendation. In contrast to the second group, this type of aggregation strategy aims at modeling the weights assigned to recommendations coming from different domains. For example, Givon and Lavrenko [22] focus on book recommendations accomplished by a CF method and model-based recommendations, relying on the similarity of a book and the user's model, as well as the book content and tags. And the final recommendations are combined in a weighted manner. The other category of cross-domain recommendation aims to *transfer information* from the source domain to the target domain by means of shared latent features or rating patterns. Hu et al. [33] propose tensor-based factorization to share latent features between different domains by using the same parameters in both factorization models. Li et al. [38] propose a code-book-transfer by co-clustering the source domain rating matrix and exploit it in the target domain to transfer rating patterns across different domains. Similarly, Mirbakhsh and Ling [53] focus on extending clustering-based MF in a single domain into multiple domains through overlapping users.

In order to model more complex connections across different domains, a variety of deep learning methods have been proposed for cross-domain recommendation [23, 79, 80]. Elkahky et al. [17] propose a multi-view deep learning recommendation system by using rich auxiliary features to represent users from different domains. Then, Lian et al. [42] propose a multi-view neural framework of a dual network for user and item, each network models CF information (user and item embeddings) and content information (user preference for item features), which ties CF and content-based filtering together. Hu et al. [31] propose a model using a cross-stitch network [54] to learn complex user behavioral information based on neural CF [26]. Wang et al. [79] propose to combine user behavioral information in information domains and user-user connection in social domains to do recommendation. Wang et al. [80] embed item-level information and cluster-level correlative information from different domains into a unified framework. Gao et al. [21] transfer item embeddings across domains without sharing user-relevant data. Li and Tuzhilin [41] develop a latent orthogonal mapping method to extract user preference over multiple domains while preserving connection between users across different latent spaces based on the mechanism of dual learning. Krishnan et al. [37] propose to guide neural CF with domain-invariant components shared across the dense and sparse domains, improving user and item representations learned in the sparse domains. They leverage contextual invariances across domains to develop these shared modules. Zhao et al. [89] propose to model user preference transfer at the aspect-level derived from reviews, which does not require overlapping users or items in all domains. Bi et al. [7] utilize cross-domain mechanism to promote recommendations for cold start users in insurance domain. They design a meta-path based method over complex insurance products to learn better item representations and learn the mapping function between domains through the overlapping users. Bi et al. [8] are the first to combine a cross-domain mechanism and a heterogeneous information network (including userid, agents, insurance products, and insurance product properties based on data from the insurance application PingAn Jinguanjia) to give personalized recommendations for cold start users in the

insurance domain. Despite the fact that the methods listed above have been proven to be effective, they cannot be directly applied to SRs.

Recently, cross-domain recommendation has been introduced to SRs as well. Zhuang et al. [92] propose a novelty seeking model based on sequences in multi-domains to model an individual's propensity by transferring novelty seeking traits learned from a source domain for improving the accuracy of recommendations in the target domain. Yang et al. [85] propose a **long short-term memory** (LSTM) with sequence completion to model the temporal information in a cross-domain scenario. They propose two methods for sequence completion. One uses measures the similarity between item features and temporal features, the other uses an improved LSTM network. Ma et al. [47] study CDSR in a shared-account scenario. They propose a novel gating mechanism to extract and share user-specific information between domains.

Although some studies have begun to explore CDSR, they only focus on user behavioral information to conduct information transfer, and neglect exploring extra knowledge to promote sequence representation across domains.

2.3 Knowledge-Aware Recommendations

Side-information, e.g., user reviews and product descriptions, is usually provided together with user-item ratings [3]. Alexandridis et al. [4] propose to extend CF-MF algorithms by fusing the side information into the user/item representations. They propose to use the paragraph vector model to encode user reviews of variable length into feature representations of fixed length. Then, they fuse the resulting feature vectors with the rating scores in a hybrid probabilistic MF algorithm based on maximum a-posteriori estimation. Hu [29] argues that traditional CF approaches only exploit user-item relations (e.g., clicks, likes, and views), which suffer from the data sparsity issue. They design a personalized neural embedding model to fuse side information for effective estimation of user preferences. Furthermore, Hu and Dai [30] integrate item reviews into MF-based Bayesian personalized ranking, which can generate a ranked list of items for individual users.

Considerable efforts have also been made to utilize some structural side-information [74, 81], especially **knowledge graphs**, to enhance the performance of recommendations. Zhao et al. [90] propose a graph-based method to iteratively update user and item distributions in a heterogeneous user-item graph and incorporate them as features into the MF for item recommendations. Zhang et al. [87] combine CF with structural knowledge, textual knowledge, and visual knowledge in a unified framework. Ai et al. [2] apply TransE on the graph including users, items, and their connections, which casts the recommendation task as a plausibility prediction task. As **graph convolutional networks** (GCNs) have been shown to be effective on many tasks [36], there have been a number of publications that propose variants of **graph convolutional networks** (GCNs) for recommendation by considering different types of information. Wang et al. [73] simulate users' hierarchical preferences over knowledge entities by extending users' potential preferences along links in a KG. Wang et al. [74] consider the connections among items based on higher-order entity features in KGs. Wang et al. [78] explicitly model the high-order connections in KGs by employing an attention mechanism to discriminate the importance of the neighbors. Ma et al. [49] propose a joint framework to integrate the induction of explainable rules from KGs with the construction of a rule-guided recommendation model. Xian et al. [84] perform explicit reasoning with knowledge so that the recommendations are supported by an interpretable inference procedure via a policy-guided reinforcement learning approach.

Not surprisingly, KGs have also been considered in SRs. Huang et al. [35] are the first to integrate KGs into SR; they utilize RNNs to capture user sequential preference and knowledge memory networks to capture attribute-level user preference. Song et al. [67] model users' social influence with a graph-attention neural network, which dynamically infers the influencers based on the

users' current preference. Huang et al. [34] introduce a taxonomy-aware memory-based multi-hop reasoning architecture by incorporating taxonomy data as structural knowledge to enhance the reasoning capacity. Wang et al. [76] formalize the SR problem as a **Markov decision process** (MDP) and propose a knowledge guided **reinforcement learning** (RL) model named KERL for fusing KG information into an RL framework, resulting in three main technical contributions concerning state representation, reward function, and learning algorithm. Chen and Wong [11] propose a heterogeneous graph neural network to learn user and item representations by integrating knowledge from user social networks. Meng et al. [52] incorporate the user micro-behavior (such as adding to cart or reading reviews) and item knowledge (item attributes) into a multi-task learning framework. Wang et al. [72] argue that the temporal evolution of the effects caused by different item relations are usually neglected. To model the dynamic meaning of an item in different sequences, they propose a method named Chorus to take both item relations and the corresponding temporal dynamics into consideration.

However, no previous work has considered KGs for SR in a cross-domain scenario, which brings new challenges, e.g., how to find useful and accurate cross-domain knowledge to improve information transfer across domains to promote the performance in both domains.

3 METHOD

In this section, we first give a formulation of the CDSR task. Then, we give an overview of our model MIFN. Finally, we describe each component of MIFN in detail. Table 1 summarizes the main symbols and notation used in this article.

3.1 Task Formulation

Cross-domain sequential recommendation (CDSR) aims at predicting the next item the user is mostly likely to interact with in multiple domains simultaneously, by mining users' previous sequential behavior over a period of time. In this work, we take two domains (i.e., domain A and B) as an example, e.g., watching movies, reading books. Let $\mathbb{A} = \{A_1, A_2, A_3, \dots, A_n\}$ denote the item set for domain A , which consists of n unique items. Similarly, let $\mathbb{B} = \{B_1, B_2, B_3, \dots, B_m\}$ denote the item set for domain B , which consist of m unique items. A *hybrid interaction sequence* from the two domains A and B has the form $S = [A_1, B_1, B_2, \dots, A_i, \dots, B_j, \dots]$, where $A_i \in \mathbb{A}$ ($1 \leq i \leq n$) and $B_j \in \mathbb{B}$ ($1 \leq j \leq m$) are the indices of consumed items in domain A and B , respectively.

We also associate each S with a **knowledge graph** (KG), which is defined over an entity set E and a relation set R , containing a set of KG triples. A triple $\langle e_1, r, e_2 \rangle$ represents a relation $r \in R$ between two entities e_1 and e_2 from E . In the cross-domain scenario, entities in the KG come from different domains, hence we represent them as E_A and E_B . For example, the triple $\langle e_{A_1}, \text{Is_the_same_category}, e_{B_3} \rangle$ means that entity e_{A_1} from domain A has the same category as entity e_{B_3} from domain B . Since we aim at linking recommended items to KG entities, an item set can be considered as a subset of KG entity set, i.e., $\mathbb{A} \subseteq E_A$ and $\mathbb{B} \subseteq E_B$. When extracting KG information for each sequence S , we also refer to the “items” in S as “item entities”. We will explain the details of the KG construction method in Section 3.2.

Based on these preliminaries, we are ready to define the CDSR task. Formally, given S and $\langle E, R \rangle$, we formulate CDSR as a task of evaluating the recommendation probabilities for all candidates in both domains, respectively, as shown in the following equation:

$$\begin{aligned} P(A_{i+1}|S, \langle E, R \rangle) &\sim f_A(S, \langle E, R \rangle) \\ P(B_{j+1}|S, \langle E, R \rangle) &\sim f_B(S, \langle E, R \rangle), \end{aligned} \tag{1}$$

where $P(A_{i+1}|S, \langle E, R \rangle)$ denotes the probability of recommending the next item A_{i+1} in domain A given the hybrid interaction sequence S and KG $\langle E, R \rangle$. $f_A(S, \langle E, R \rangle)$ is the model or function used

Table 1. Summary of the Main Symbols and Notation Used in the Article

Symbol	Gloss
\mathbb{A}	Item set for domain A .
\mathbb{B}	Item set for domain B .
n	The number of item set for domain A , i.e., $n = \mathbb{A} $.
m	The number of item set for domain B , i.e., $m = \mathbb{B} $.
A_i	The interacted item at time step i from domain A .
B_j	The interacted item at time step j from domain B .
S	Hybrid interaction sequence, i.e., $S = [A_1, B_1, B_2, \dots, A_i, \dots, B_j, \dots]$.
\mathbb{S}	Set of all hybrid interaction sequences in the training set.
S_A	A sub-sequence of S which only contains items from domain A .
S_B	A sub-sequence of S which only contains items from domain B .
$e_{A_k} \in E_A$	e_{A_k} represents any entity in the KG from domain A ; E_A is the set of all entities from domain A .
$e_{B_k} \in E_B$	e_{B_k} represents any entity in the KG from domain B ; E_B is the set of all entities from domain B .
$e_k \in E$	e_k represents any entity in the KG; E is the set of all entities; note that $E = E_A \cup E_B$.
R	Relation set in the KG.
$h_{A_i} \in H_A$	h_{A_i} represents the item representation of item A_i ; H_A is the set of all item representations for S_A .
$h_{B_j} \in H_B$	h_{B_j} represents the item representation of item B_j ; H_B is the set of all item representations for S_B .
$h_{(A \rightarrow B)_i}$	Transferred behavioral information flow from domain A to domain B at time step i .
$N_i(k)$	Neighbor entity set of entity e_k from the same domain as e_k .
$N_c(k)$	Neighbor entity set of entity e_k from the complementary domain.

to estimate $P(A_{i+1}|S, \langle E, R \rangle)$. Similar definitions apply to $P(B_{j+1}|S, \langle E, R \rangle)$ and $f_B(S, \langle E, R \rangle)$. It is important to note that when recommending items (such as B_x) in the B domain, the model can only see items before the interaction with B_x from both domains. As to modeling hybrid sequences, we follow the definition of previous work on CDSR [47].

3.2 Knowledge Graph Construction

Compared with other auxiliary information, KG information has the following advantages. First, it introduces more semantic relationships between items, which can help to discover user preferences at a deeper level. Second, the KG can connect the user's historical records and recommendation results, thereby improving the user's satisfaction and acceptance of the recommendation results, and enhancing the user's trust in the **recommender systems** (RSs).

In this work, we extract data (entities and relations) from the Amazon product collection¹ as the complete KG, which is collected from massive user logs. Besides, we also crawl some relations between the entities from Wikipedia.² The entities include “movies,” “books,” “kitchenware,” and “food,” each of which corresponds to one domain. The relations include “Also_buy,” “Also_view,” “Buy_together,” “Buy_after_viewing,” “Adapted_from,” and “Is_the_same_category.” For

¹<https://jmcauley.ucsd.edu/data/amazon>.

²<https://en.wikipedia.org/>.

example, (steak (the food), Buy_together, saucepan (the kitchenware)) means that the user also buys the saucepan while buying the steak. However, the complete KG contains a large number of related entities, and complicated relations among these entities, which will raise memory and computational efficiency issues. Therefore, we propose to extract a KG from the complete KG for each hybrid interaction sequence S . We require that, given any pair of items from both domains, respectively, we can find at least one path in the KG that connects them.

ALGORITHM 1: KG Construction Algorithm for Each Hybrid Interaction Sequence.

Input:

Hybrid interaction sequence, S ;
 Complete KG triples, $\langle E, R, E \rangle$;
 Maximum hop count, H ;
 Number of entities in the KG, N ;

Output:

Multiple relational adjacency matrix, A ;

```

1: Divide the hybrid interaction sequence  $S$  into  $S_A$  and  $S_B$ ;
2: set  $\vartheta_1^0 = S_A, \vartheta_2^0 = S_B, T_1^0 = \emptyset, T_2^0 = \emptyset, connected = false$ ;
3: for  $k \in range(H)$  do
4:   Extract related entities  $\vartheta_1^k = \{t | (h, r, t) \in \langle E, R, E \rangle \text{ and } h \in \vartheta_1^{k-1}\}$  and triples  $T_1^k = \{(h, r, t) | (h, r, t) \in \langle E, R, E \rangle \text{ and } h \in \vartheta_1^{k-1}\}$ ;
5:   Extract related entities  $\vartheta_2^k = \{t | (h, r, t) \in \langle E, R, E \rangle \text{ and } h \in \vartheta_2^{k-1}\}$  and triples  $T_2^k = \{(h, r, t) | (h, r, t) \in \langle E, R, E \rangle \text{ and } h \in \vartheta_2^{k-1}\}$ ;
6:    $connected = Is\_Connect(\vartheta_1^k, \vartheta_2^k)$ 
7:   if  $connected = true$  or  $k = H-1$  then
8:      $\theta = Select\_triples(T_1^k, T_2^k, S_A, S_B)$ ;
9:      $A = Construct\_Adjacency\_matrix(\theta)$ ;
10:    break;
11:  end if
12: end for
13: return adjacency matrix  $A$ ;
```

The KG construction algorithm is shown in Algorithm 1. H represents the maximum hop count and N represents the number of entities in the final KG. We extract all triples and entities that are related to items involved in the hybrid interaction sequence S within H hops. We stop extracting more hops when there is a path for any given pair of items from both domains. Finally, we construct the relational adjacency matrix for the extracted KG. This process can be run in parallel. For about 50,000 sequences, it takes around 3 hours using 10 CPU cores, and each sequence takes 0.2 seconds on average. In this work, we focus on modeling the behavioral and knowledge flow of information across domains. The focus is on how to model this flow rather than how to build the knowledge itself, although a better KG construction algorithm is expected to improve the results a lot (as shown in Section 5.3).

Specifically, for the input S , we first divide it into two sub-sequences S_A and S_B according to the domain to which the items belongs (see line 1). We initialize the related entities ϑ_1^0 or ϑ_2^0 with the item entities, and initialize the related triples T_1^0 and T_2^0 with the empty set (see line 2). At each hop, we extract all related entities in the current KG (i.e., ϑ_1^k and ϑ_2^k), where $\vartheta_1^k = \{t | (h, r, t) \in \langle E, R, E \rangle \text{ and } h \in \vartheta_1^{k-1}\}$, and $\vartheta_2^k = \{t | (h, r, t) \in \langle E, R, E \rangle \text{ and } h \in \vartheta_2^{k-1}\}$, $k = 1, 2, \dots, H$, where $\vartheta_1^0 = S_A$ and $\vartheta_2^0 = S_B$. At the same time, we record the related triples as $T_1^k = \{(h, r, t) | (h, r, t) \in \langle E, R, E \rangle \text{ and } h \in \vartheta_1^{k-1}\}$ and $T_2^k = \{(h, r, t) | (h, r, t) \in \langle E, R, E \rangle \text{ and } h \in \vartheta_2^{k-1}\}$.

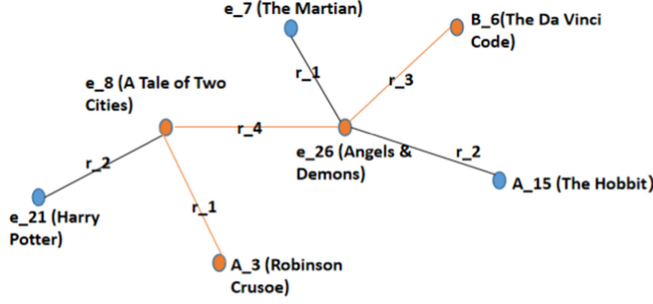


Fig. 2. An example of path finding during KG construction.

$h \in \mathcal{V}_1^{k-1}$ and $T_2^k = \{(h, r, t) | (h, r, t) \in \langle E, R, E \rangle \text{ and } h \in \mathcal{V}_2^{k-1}\}$ (see line 4 and line 5). If there is a path for any given pair of items from both domains (e.g., $A_3 \rightarrow e_8 \rightarrow e_{26} \rightarrow e_9 \rightarrow B_6$) or the number of hops reaches the maximum number of hops H , we stop extracting other triples from the complete KG and construct the relational adjacency matrix (see line 7 to line 12). Otherwise, we continue to extract other related triples. When constructing the adjacency matrix, we limit the number of entities in the KG to N . Therefore, we need to select some triples if the number of entities in all related triples is larger than N (see line 8). To do so, we first gather all the entities that connect a pair of item entities from two domains, e.g., e_8, e_{26} in the path $A_3 \rightarrow e_8 \rightarrow e_{26} \rightarrow B_6$. As shown in Figure 2, A_3 is from domain A while B_6 is from domain B . During the KG construction phase, we aim at finding paths that can connect A_3 and B_6 , i.e., the path marked in orange ($A_3 \rightarrow e_8 \rightarrow e_{26} \rightarrow B_6$).

Then, we select the entities according to their smallest distance w.r.t. any item from the two domains until the number of all entities meets N . After that, we construct the relational adjacency matrix A based on the selected entities and their relations (see line 10).

3.3 MIFN

In the following subsections, we will demonstrate the details of MIFN. Generally, MIFN models $P(A_{i+1}|S, \langle E, R \rangle)$ and $P(B_{j+1}|S, \langle E, R \rangle)$ (see Equation (1)) by taking two recommendation modes into consideration, as shown in the following equation:

$$\begin{aligned}
 P(A_{i+1}|S, \langle E, R \rangle) &= \\
 &P(M_{SA}|S, \langle E, R \rangle)P(A_{i+1}|M_{SA}, S, \langle E, R \rangle) + P(M_{GA}|S, \langle E, R \rangle)P(A_{i+1}|M_{GA}, S, \langle E, R \rangle) \\
 P(B_{j+1}|S, \langle E, R \rangle) &= \\
 &P(M_{SB}|S, \langle E, R \rangle)P(B_{j+1}|M_{SB}, S, \langle E, R \rangle) + P(M_{GB}|S, \langle E, R \rangle)P(B_{j+1}|M_{GB}, S, \langle E, R \rangle).
 \end{aligned} \tag{2}$$

Here, M_S and M_G denote *sequence mode* and *graph mode*, which make recommendations at the user behavior level and the knowledge level, respectively. $P(M_{SA}|S, \langle E, R \rangle)$ and $P(M_{GA}|S, \langle E, R \rangle)$ represent the probabilities under the *sequence mode* and the *graph mode* in domain A , respectively, $P(A_{i+1}|M_{SA}, S, \langle E, R \rangle)$ and $P(A_{i+1}|M_{GA}, S, \langle E, R \rangle)$ refer to the probabilities of recommending the next item A_{i+1} under the *sequence mode* and *graph mode* given a hybrid interaction sequence S and the KG triples $\langle E, R \rangle$. The same definitions apply to domain B .

As shown on the left side of Figure 3, MIFN consists of four main components: a sequence encoder, a **behavior transfer unit** (BTU), a **knowledge transfer unit** (KTU), and a mixed recommendation decoder. The sequence encoder encodes the interacted item sequence into a sequence of item representations. The BTU takes the representations from the source domain as input, extracts behavioral information flow, and transfers it to the target domain. The KTU aims at grasping useful knowledge from the KG and propagates it to both domains. The mixed recommendation

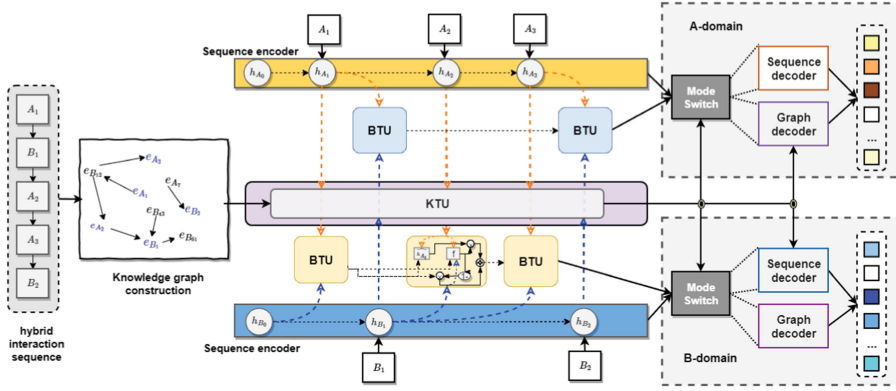


Fig. 3. An overview of the **mixed information flow network** (MIFN). The BTU extracts behavioral information flow from source domain, and transfers it to the target domain. The KTU aims at grasping useful knowledge from the KG and propagates it to both domains. Section 3.3 contains a walkthrough of the model.

decoder contains two decoders w.r.t. *graph mode* and *sequence mode*, respectively. The *graph recommendation decoder* evaluates the probability for all candidate items from the KG, corresponding to Equation (10). The *sequence recommendation decoder* evaluates the probability of clicking items, corresponding to Equation (11).

3.4 Sequence Encoder

As with existing studies [27, 28], we use an RNN to encode the sub-sequences S_A and S_B . Here, we employ a GRU as the recurrent unit. The initial state of the GRUs is set to zero vectors, i.e., $h_0 = 0$. After that, we can obtain $H_A = \{h_{A_1}, h_{A_2}, \dots, h_{A_i}, \dots, h_{A_n}\}$ for domain A, and $H_B = \{h_{B_1}, h_{B_2}, \dots, h_{B_j}, \dots, h_{B_m}\}$ for domain B. Each h_{A_i} or h_{B_j} is the item representation of an item A_i in sequence S_A or B_j in S_B .

3.5 Behavior Transfer Unit

The **behavior transfer unit** (BTU) takes the representations from the source domain as input, extracts behavioral information, and transfers it to the target domain. It aims to output the transferred behavioral representation. The outputs H_A and H_B from the sequence encoder are representations of user behavior in single domains. It has been shown that there is connection between H_A and H_B [47]. For example, a user who has read the book *Harry Potter* (e.g., “*Harry Potter and the Philosopher’s Stone*” or “*Harry Potter and the Chamber of Secrets*” and so on) has also watched the movie “*Pirates of the Caribbean*” within the same time period. Based on behavioral information from both domains, it is easier for the model to infer that the user might like some magic and fantasy movies and books.

To achieve this, we employ the BTU to model the flow of behavioral information from domain A to domain B, i.e., $h_{A_i \rightarrow B}$, as follows:

$$\begin{aligned} f_{A_i} &= \sigma(W_{f_A} \cdot h_{A_i} + W_{f_B} \cdot h_{B_j} + W_f \cdot h_{A_{i-1} \rightarrow B} + b_f) \\ \widehat{h_{A_i}} &= \tanh(W_h \cdot h_{A_i} + U_h \cdot h_{A_{i-1} \rightarrow B} + b_h) \\ h_{A_i \rightarrow B} &= f_{A_i} \odot \widehat{h_{A_i}} + (1 - f_{A_i}) \odot h_{A_{i-1} \rightarrow B}, \end{aligned} \quad (3)$$

where h_{A_i} and h_{B_j} are the representations of domain A and B at timestamp i and j , respectively. f_{A_i} measures the degree of connection between these two representations h_{A_i} and h_{B_j} from both

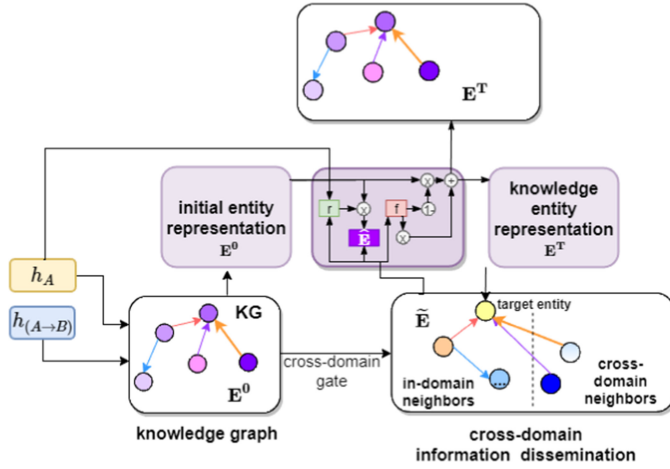


Fig. 4. Knowledge transfer unit (KTU).

domains, which employs the gate mechanism to control how much information is to be transferred from domain A to domain B . \widehat{h}_{A_i} is the updated representation of the current input. W_A and W_B are the parameters, b_f is the bias term, \odot indicates element-wise multiplication. $h_{A_i \rightarrow B}$ can be seen as a combination of \widehat{h}_{A_i} and $h_{A_{i-1} \rightarrow B}$ balanced by f_{A_i} . Note that the BTU can be applied bidirectionally from “domain A to domain B ” and “domain B to domain A ”. Here, we take the “domain A to domain B ” direction and achieve recommendations in domain B as an example.

$h_{A_i \rightarrow B}$ is the information extracted from domain A , which is ready to be transferred to domain B . Since it still belongs to domain A , we employ an RNN structure to transfer it to domain B as follows:

$$h_{(A \rightarrow B)_i} = \text{GRU}(h_{(A \rightarrow B)_{i-1}}, h_{A_i \rightarrow B}). \quad (4)$$

After that, we can obtain the transferred behavioral representation $h_{(A \rightarrow B)_i}$ in domain B at time step i .

3.6 Knowledge Transfer Unit

The BTU only models the flow of behavioral information. We hypothesize that this may not be enough for the model to be able to encode the connection between items from the two domains in some cases. For example, if there is no knowledge indicating that both “*Pirates of the Caribbean*” and “*Harry Potter*” belong to magic and fantasy, it is difficult for the model to capture the connection between them solely based on behavioral information. To better transfer the information of items from both domains, we propose the **knowledge transfer unit (KTU)**; it takes the item representation from sequence encoder and transferred behavioral representations from BTU as input and grasps useful knowledge from KG and propagates it to both domains, as shown in the Figure 4.

For each hybrid interaction sequence S , we get the item representations $\{h_{A_1}, h_{A_2}, \dots, h_{A_i}, \dots\}$ from the sequence encoder (Section 3.4) and the transferred behavioral representations $\{h_{(A \rightarrow B)_1}, h_{(A \rightarrow B)_2}, \dots, h_{(A \rightarrow B)_i}, \dots\}$ from the BTU (Section 3.5). We use the item representation of the last time step to denote the sequence representation h_A and the transferred behavioral representation $h_{(A \rightarrow B)}$. We also obtain the relational adjacency matrix A of the KG, which consists of N entities

and the corresponding relations (Section 3.2). Here, we represent these N entities as E , the corresponding relations are represented as R .

We initialize all entities in the KG and we can get the initialized entity representations h_{E^0} . That is, for each entity $e_k \in E$, the initialized entity representation is $h_{e_k^0} \in h_{E^0}$. Then, we learn a transferred entity representation $h_{e_k^T} \in h_{E^T}$ for each entity e_k by leveraging the relations in the KG, as shown in the following equation:

$$\begin{aligned} r &= \sigma \left(W_r \cdot h_{e_k^0} + U_r \cdot \widetilde{h_{e_k^L}} + b_r \right) \\ f &= \sigma \left(W_f \cdot h_{e_k^0} + U_f \cdot \widetilde{h_{e_k^L}} + V_f \cdot h_A + b_f \right) \\ \widetilde{h_{e_k}} &= \tanh \left(W_h \cdot \widetilde{h_{e_k^L}} + U_h \odot \left(r \odot h_{e_k^0} \right) \right) \\ h_{e_k^T} &= (1 - f) \odot h_{e_k^0} + f \odot \widetilde{h_{e_k}}. \end{aligned} \quad (5)$$

The explanations for the main parts of Equation (5) are as follows:

- (i) **Gated functions.** r and f are the update gate function and the forget gate function, which aim at regulating how much of the update information should be propagated. W_r , U_r , W_f , U_f , and V_f are the parameters; b_r and b_f are the bias term.
- (ii) **Candidate knowledge transfer representation.** $\widetilde{h_{e_k}}$ is the candidate knowledge transfer representation, which is calculated based on the cross-domain disseminated entity representation $\widetilde{h_{e_k^L}}$ at the L th hop layer (we will explain this later in Equation (6)) and the updated entity representation $r \odot h_{e_k^0}$. W_h and U_h are the parameters, \odot indicates element-wise multiplication.
- (iii) **Transferred entity representation.** The transferred entity representation $h_{e_k^T}$ is a combination of the initialized entity representation $h_{e_k^0}$ and the candidate knowledge transfer representation $\widetilde{h_{e_k}}$ balanced by the forget gate f , where the information among entities has been disseminated through L hop layers.

Graph convolutional techniques are commonly used to disseminate information among entities based on their relations [36, 73, 74]. However, in the cross-domain scenario that we consider, the information disseminated by entities from different domains is different. Hence, we propose a **cross-domain graph convolutional mechanism** that can distinguish entities from different domains and adopt different modeling methods to disseminate their information so as to get better entity representation. In this manner, the information in the KG is disseminated between both domains, which can be considered as a flow of knowledge in the hybrid interaction sequence. Suppose the information can be disseminated within L hop layers. At the l th hop layer ($0 \leq l \leq L$), the information of each entity and its cross-domain neighbor entities via various relations will be disseminated to the next hop layer $l + 1$ and is used to update the entity representation. The process of cross-domain information dissemination is defined in the following equation:

$$\widetilde{h_{e_k^{l+1}}} = \sigma \left[f_0 \left(\widetilde{h_{e_k^l}} \right) + \frac{1}{|N_i(k)|} \sum_{r \in R} \sum_{p \in N_i(k)} f_i \left(\alpha_p \cdot \widetilde{h_{e_p^l}} \right) + \frac{1}{|N_c(k)|} \sum_{r \in R} \sum_{q \in N_c(k)} f_c \left(\beta_q \cdot \widetilde{h_{e_q^l}} + c_q \cdot \widetilde{h_{e_q^l}} \right) \right], \quad (6)$$

where e_k is any entity in the entity set E ; $N_i(k)$ is the neighbor entity set of entity e_k from the same domain as e_k ; and $N_c(k)$ is the neighbor entity set of entity e_k from the complementary domain. In

other words, if e_k belongs to A domain, $N_i(k)$ is the neighbor entity set belonging to domain A , and $N_c(k)$ is the neighbor entity set belonging to domain B . f_0 , f_i , and f_c represent the transformation functions for initialization, in-domain, and cross-domain, respectively; α_p is the in-domain attention weight calculated between each entity $e_p \in N_i(k)$ and the sequence representation h_A ; β_q is the cross-domain attention weight calculated between each entity $e_q \in N_c(k)$ and the transferred behavioral representation $h_{(A \rightarrow B)}$; c_q shows the sum of similarities between entity e_q and each entity $e_p \in N_i(k)$, which is defined as $c_q = \sum_{p \in N_i(k)} e_p \cdot e_q$; $\widetilde{h}_{e_k^l}$ denotes the cross-domain disseminated representation of entity e_k at the l th hop layer, which aggregates the information from itself and cross-domain neighbor entities as the new representation for the next hop layer. At the first hop layer, the cross-domain disseminated representation is assigned by the gated entity representation, i.e., $\widetilde{h}_{e_k^0} = h_{e_{A_k}}$ if the entity e_k is from domain A , otherwise $\widetilde{h}_{e_k^0} = h_{e_{B_k}}$:

$$\begin{aligned} c &= \sigma \left(W_c \cdot \text{concat}[h_A, h_{(A \rightarrow B)}, h_{E_0}] + b_c \right) \\ h_{e_{A_k}} &= c \odot \left(\alpha_{A_k} \cdot h_{e_{A_k}^0} \right) \\ h_{e_{B_k}} &= (1 - c) \odot \left(\beta_{B_k} \cdot h_{e_{B_k}^0} \right), \end{aligned} \quad (7)$$

where W_c is the parameter; b_c is the bias term; $h_{e_{A_k}}$ is the gated entity representation of entity $e_{A_k} \in E_A$, $h_{e_{B_k}}$ is the gated representation of entity $e_{B_k} \in E_B$. c is a cross-domain information gate to handle the situation where the proportion of entities from different domains in the KG is different (e.g., when there are 1,000 entities from domain A , but only 10 entities from domain B). So we define the gated entity representations $h_{e_{A_k}}$ for domain A and $h_{e_{B_k}}$ for domain B based on their initial entity representations, respectively, which aims at balancing information from both domains. α_{A_k} is the attention weight of e_{A_k} for sequence representation h_A ; β_{B_k} is the attention weight of e_{B_k} for the transferred behavioral representation $h_{(A \rightarrow B)}$. These attention weights act as an information controller to identify entities of different importance in the KG, the definitions of which are shown in the following equation:

$$\begin{aligned} \alpha_{A_k} &= \text{softmax} \left(\mathbf{v}_1^\top \tanh \left(W_{A_1} \cdot h_A + W_{A_2} \cdot h_{e_{A_k}^0} \right) \right) \\ \beta_{B_k} &= \text{softmax} \left(\mathbf{v}_2^\top \tanh \left(W_{B_1} \cdot h_{(A \rightarrow B)} + W_{B_2} \cdot h_{e_{B_k}^0} \right) \right), \end{aligned} \quad (8)$$

where $h_{e_{A_k}^0}$ and $h_{e_{B_k}^0}$ are the initialized entity representations of entity e_{A_k} and e_{B_k} , respectively. h_A is the sequence representation and $h_{(A \rightarrow B)}$ is the transferred behavioral representation as mention above. \mathbf{v}_1 , \mathbf{v}_2 , W_{A_1} , W_{A_2} , W_{B_1} , and W_{B_2} are learnable parameters.

3.7 Mode Switch

Recall that $P(M_{S_B}|S, \langle E, R \rangle)$ and $P(M_{G_B}|S, \langle E, R \rangle)$ are the probabilities of conducting recommendations under *sequence mode* and *graph mode*, respectively. We model the mode switch as a binary classifier. Specifically, we first combine the sequence representation h_B , the transferred behavioral representation $h_{(A \rightarrow B)}$ and the sum of all transferred entity representation $\sum_k (h_{e_k^T})$ (where $h_{e_k^T} \in h_{ET}$). Then, we employ a softmax regression to transform the total representation into the mode probability distributions, as follows:

$$P(M_{G_B}|S, \langle E, R \rangle), P(M_{S_B}|S, \langle E, R \rangle) = \text{softmax} \left(W_m \cdot \text{concat} \left[h_B, h_{(A \rightarrow B)}, \sum_k (h_{e_k^T}) \right] + b_m \right), \quad (9)$$

where W_m is the weight matrix and b_m is the bias term.

3.8 Graph Recommendation Decoder

The graph recommendation decoder evaluates the probabilities of recommending items involved in the KG. Here, we directly use the representations of entities to learn the attention weights and take these weights as the final predicted recommendation probability. The recommendation probability for each item $B_{j+1} \in E_B$ is computed as follows:

$$P(B_{j+1}|M_{-G_B}, S, \langle E, R \rangle) = \begin{cases} 0 & \text{if } B_{j+1} \notin \mathbb{B} \\ \text{softmax} \left(h_{e_{B_{j+1}}}^T \right) & \text{if } B_{j+1} \in \mathbb{B}, \end{cases} \quad (10)$$

where $h_{e_{B_{j+1}}}^T$ is the transferred entity representation of $e_{B_{j+1}}$ (corresponding to $h_{e_k}^T$ in Equation (5) when e_k is an item entity from domain B). It is worth noting that this process only calculates the probability in the item set, and that the size of the item set \mathbb{B} is m . Note that $e_{B_{j+1}} \in E_B$ is an item entity corresponding to item $B_{j+1} \in \mathbb{B}$. The recommendation probabilities are set to zero for those items that do not exist in \mathbb{B} .

3.9 Sequence Recommendation Decoder

The sequence recommendation decoder evaluates the probabilities of items in the sequence item set. We first concatenate the sequence representation h_B and the transferred behavioral representation $h_{(A \rightarrow B)}$ into the hybrid representation c_S , i.e., $c_S = [h_B, h_{(A \rightarrow B)}]^T$. Then, the recommendation probability for each item $B_{j+1} \in \mathbb{B}$ is computed as follows:

$$P(B_{j+1}|M_{-S_B}, S, \langle E, R \rangle) = \begin{cases} 0 & \text{if } B_{j+1} \notin \mathbb{B} \\ \text{softmax} \left(W_{j+1} c_S + b_I \right) & \text{if } B_{j+1} \in \mathbb{B}, \end{cases} \quad (11)$$

where $W_{j+1} \in W_I$ is the weight matrix, b_I is the bias term. The recommendation probabilities are set to zero for those items that do not exist in the item set \mathbb{B} .

3.10 Objective Function

Our goal is to maximize the prediction probability for each domain given a hybrid interaction sequence. Therefore, we define the negative log-likelihood loss function as follows:

$$L_R(\theta) = L_{R_A}(\theta) + L_{R_B}(\theta), \quad (12)$$

where θ are all parameters in MIFN. Specifically, $L_{R_A}(\theta)$ and $L_{R_B}(\theta)$ can be derived as follows:

$$\begin{aligned} L_{R_A}(\theta) &= -\frac{1}{|\mathbb{S}|} \sum_{S \in \mathbb{S}} \sum_{A_i \in S} \log P(A_{i+1}|S, \langle E, R \rangle) \\ L_{R_B}(\theta) &= -\frac{1}{|\mathbb{S}|} \sum_{S \in \mathbb{S}} \sum_{B_j \in S} \log P(B_{j+1}|S, \langle E, R \rangle), \end{aligned} \quad (13)$$

where \mathbb{S} is the set of all hybrid interaction sequences in our training set, and $P(A_{i+1}|S, \langle E, R \rangle)$ or $P(B_{j+1}|S, \langle E, R \rangle)$ are the next prediction probabilities, which are as defined in Equation (2).

4 EXPERIMENTAL SETUP

4.1 Research Questions

We evaluate MIFN on four e-commerce datasets. We aim at answering the following questions in our experiments:

Table 2. Dataset Statistics

Domain	#Items	#Train	#Test	#Valid	Avg_Seq_Len
Movie	36,845	44,732	19,861	9,274	11.98
Book	63,937				
Food	29,207	25,766	17,280	7,650	9.91
Kitchen	34,886				

- (RQ1) How does MIFN perform compared with the state-of-the-art methods in terms of Recall and MRR? (See Section 5.1.)
- (RQ2) Does the KTU help to improve the performance of recommendations? And does the performance differ from the situation when we only allow for the flow of behavioral information? (See Section 5.2.)
- (RQ3) Does the knowledge graph construction method have a big effect on the overall recommendation results? (See Section 5.3.)
- (RQ4) Is MIFN able to provide better recommendations by incorporating the flow of knowledge across domains? (See Section 5.4.)
- (RQ5) Is the overall performance sensitive to the hyperparameters? (See Section 5.5.)

4.2 Datasets

We conduct experiments on the Amazon e-commerce collection,³ which consists of user interactions (e.g., userid, itemid, ratings, timestamps) from multiple domains and some item meta information (e.g., descriptions, images, product associations). Compared with other recommendation datasets, the Amazon dataset contains overlapping user interactions in multiple domains, which is suitable for **cross-domain sequential recommendation** (CDSR). Specifically, we pick two pairs of complementary domains “Movie-Book” domains and “Food-Kitchen” domains for experiments. For the “Movie-Book” dataset, the “Movie” domain contains movie watching records. The “Book” domain covers book reading records. For the “Food-Kitchen” dataset, the “Food” domain contains food purchase records. The “Kitchen” domain contains furniture purchase records. We follow the settings of Ma et al. [47] to process the data. To satisfy cross-domain characteristics, we first pick users who have interactions in both domains. Since we do not target cold-start users or items in this work, we only keep users who have more than 10 interactions and items whose frequency is larger than 10. To satisfy sequential characteristics which consists of many user interactions within a period of time, we first order the interactions by time for each user, then we split the sequences from each user into several small sequences with each sequence containing interactions within a period, i.e., a month for the “Movie-Book” dataset, and a year for the “Food-Kitchen” dataset. We also require that each sequence contains at least three items from each domain. The statistics of the processed datasets are shown in Table 2.

For knowledge graph construction, we use the Amazon product data to mine the relations of the items.⁴ The data are collected from large-scale user logs, which contain the following relations: (1) “Also_buy” (users also buy item X when buying item Y .); (2) “Also_view” (users also view item X when viewing item Y .); (3) “Buy_together” (users buy item X and Y together frequently); (4) “Buy_after_viewing” (users buy item X after they buy Y); (5) “Is_the_same_category” (item X and Y belong to the same category). Additionally, for the “Movie-Book” dataset, we also

³<https://www.amazon.com/>.

⁴<https://jmcauley.ucsd.edu/data/amazon>.

Table 3. KG Statistics

Domain	#Entities	#Relations	#Triples
Movie	65,418	6	3,911,284
Book	315,770		40,048,795
Food	50,273	5	3,822,123
Kitchen	82,552		7,836,064

crawl the relation “Adapted_from” between books and movies from Wikipedia.⁵ For example, $\langle \text{movie}, \text{Adapted_from}, \text{book} \rangle$ means that the movie is adapted from the book. We align knowledge entities with Wikipedia titles by fully matching. The statistics of the knowledge information are given in Table 3.

For evaluation, we use the last interacted item in each sequence for each domain as the ground truth item, respectively. We randomly select 80% of each user’s interactions as the training set, 10% as the validation set, and the remaining 10% as the test set.

4.3 Baseline Methods

We compare the proposed model MIFN with baselines from four categories: (1) traditional recommendation methods, (2) SR methods, (3) cross-domain recommendation methods, (4) and knowledge-aware recommendation methods.

4.3.1 Traditional Recommendation Methods. We adapt three commonly used traditional recommendation methods to SRs:

- POP: This method recommends the most popular items in which items are ranked based on their popularity. It is a simple baseline, but is commonly used owing to its simplicity yet effectiveness [26].
- Item-KNN: This method is inspired by the classical KNN model; it looks for items that are similar to other items that have been clicked by a user in the past, where similarity is defined as the cosine similarity between the vector of sequences [64].
- BPR-MF: This method follows the idea of MF with a pairwise ranking objective via stochastic gradient descent [61]. Following Hidasi et al. [27], we represent a new sequence by the average latent factors of items that appeared in the sequence so far.

4.3.2 SR Methods. A number of SR methods have been proposed in the last few years. In this work, we construct/select baselines that are fair (use the same information, similar architectures, etc.) to compare with:

- GRU4REC: It is the first attempt to use GRU for SRs. It utilizes session-parallel mini-batch training strategy and employs a ranking-based loss functions [27].
- HRNN: This method combines the extra user’s information into GRU networks and proposes a hierarchical RNN model based on GRU4REC [57].
- NARM: This method takes an attention mechanism into consideration to capture both sequential-level preferences and the user’s main purpose [40].
- STAMP: This method constructs two network structures to capture a user’s general preferences and the current preferences of the last click within the current sequence [45].
- SRGNN: This method constructs each sequence as a directed graph, where the items in the sequence are entities and the transition relationship between adjacent items represents edge.

⁵<https://en.wikipedia.org/>.

By modeling the complex transitions, each session graph and item embeddings of all items involved in each graph can be obtained through gated graph neural networks [82].

4.3.3 Cross-Domain Recommendation Methods. We use three popular cross-domain recommendation methods for comparison.

- NCF-MLP++: This is a deep learning based method where the model learns the inner product of the traditional CF by using **multilayer perceptron** (MLP) in each domain. The user representations are shared between both domains while item representations are private in each domain, and the final recommendations are aggregated from both domain recommendations probabilities. We adopt the implementation in [47].
- Conet: This method transfers information between different domains by a cross-stitch network [54], where information in each domain is captured by neural CF model [26].
- π -Net: This method is the only one that considers cross-domain characteristics for SRs. We take this as the fairest baseline to compare with. It designs a new gating mechanism to recurrently extract and share useful information across different domains [47].

4.3.4 Knowledge-Aware Recommendation Methods. As knowledge-aware recommendation methods, we choose two baseline methods.

- KGAT: This method explicitly models the high-order connectivities in KG in an end-to-end fashion. It recursively propagates the embeddings from an entity's neighbors (which can be users, items, or attributes) to refine the entity's embedding, and employs an attention mechanism to discriminate the importance of the neighbors [78].
- KSR: This method incorporates KGs into SRs, and it combines the sequential user preference captured by an RNN and attribute-level preferences captured by Key-Value Memory Networks to get the final representation of user preference [35].

4.4 Evaluation Metrics

We target the top-K recommendations in this work, so we adopt two widely used ranking-based metrics [14, 25, 51, 58]: MRR@K and Recall@K. Specifically, we report $K = 5, 10, 20$.

- Recall: This measures the proportion of the top-K recommended items that are in the evaluation set. It does not consider the actual rank of the item as long as it is amongst the list of recommend items.
- MRR: This is the average of reciprocal ranks of the relevant items. And the reciprocal rank is set to zero if the ground truth item is not in the list of recommended items. MRR takes the order of recommendation ranking into account. Since each sample has only one ground truth item, we choose MRR as the ranking metric instead of others, e.g., NDCG.

4.5 Implementation Details

For most of the baseline methods, we find the best settings using grid search on the validation set. For those with too many hyperparameters, we follow the reported optimal hyperparameter settings from the original publications that introduced them. For our model, the embedding size and the hidden size are set to 256, and the number of entities N in KG is set to 200. As for the parameter hop layer H , we conduct some preliminary experiments and find that when the hop layer is limited to 1, most of the sequences cannot be connected, but when the hop layer increases to 2, around 75% of the sequences can be connected, and when it continues to increase to 5 hops even 7 hops, there is not much growth, so we limit parameter hop layer to 2. We initialize the model parameters randomly using the Xavier method. We take Adam as our optimization method. MIFN

is implemented in TensorFlow and trained on a GeForce GTX TitanX GPU. The code and dataset used to run the experiments in this article are available at <https://github.com/mamuyang/MIFN>.

4.6 Discussion

In addition to calculating the prediction probability, we also explore the *mode switch* module to calculate the mode selection probability between *sequence mode* and *graph mode*. We assume that if an item does not exist in the item set, it can just be generated under the *graph mode*. Here, we can jointly train another mode prediction loss as follows, which adopts the negative log-likelihood loss:

$$\begin{aligned} L_{M_A}(\theta) &= -\frac{1}{|\mathbb{S}|} \sum_{S \in \mathbb{S}} \sum_{A_i \in S} [(1 - \mathbb{1}(A_{i+1} \in \mathbb{A})) \log P(M_{S_A}|S, \langle E, R \rangle)] \\ L_{M_B}(\theta) &= -\frac{1}{|\mathbb{S}|} \sum_{S \in \mathbb{S}} \sum_{B_j \in S} [(1 - \mathbb{1}(B_{j+1} \in \mathbb{B})) \log P(M_{S_B}|S, \langle E, R \rangle)] \\ L_M(\theta) &= L_{M_A}(\theta) + L_{M_B}(\theta), \end{aligned} \quad (14)$$

where $\mathbb{1}(item \in itemset)$ is the indicator function that equals 1 if this item is in the item set and 0 otherwise. $L_M(\theta)$ is the total mode loss for domain A and B.

Finally, we adopt a joint-learning strategy, and the final loss combines both recommendation loss and mode loss:

$$L(\theta) = L_R(\theta) + L_M(\theta). \quad (15)$$

Experimental results for this module are shown in Tables 4 and 5, where MIFN represents the model without mode prediction loss, MIFN+ L_M represents the model with this mode prediction loss. An analysis of these results is provided in Section 5.2.

5 RESULTS AND ANALYSIS

5.1 Overall Performance (RQ1)

We report the results of MIFN compared with the baseline methods on the “Movie-Book” and “Food-Kitchen” datasets. The results of all methods are given in Tables 4 and 5, respectively. From the results, we have the following main observations.

First, MIFN outperforms some single-domain SR methods (e.g., STAMP, NARM, SRGNN) and the knowledge-aware methods (KSR) on all datasets. Particularly, on the “Movie-Book” dataset, the largest increase over NARM is 7.5% and 12.4% in terms of MRR@5 and Recall@10 on the “Movie” domain, and on the “Book” domain, the largest increase is 6.2% and 9.1% in terms of MRR@20 and Recall@5. And the increase over KSR on the “Movie” domain is 4.6% and 6.9% in terms of MRR@5 and Recall@10, on the “Book” domain, the increase is 2.1% and 2.9% in terms of MRR@20 and Recall@5. On the “Food-Kitchen” dataset, the largest increase over NARM is 6.5% and 9.3% in terms of MRR@20 and Recall@20 on the “Food” domain, and on the “Kitchen” domain, the increase is 9.1% and 12.3% in terms of MRR@5 and Recall@10. And the increase over KSR is 2.7% and 3.9% in terms of MRR@20 and Recall@10 on the “Food” domain, and on the “Kitchen” domain, the increase is 1.6% and 4.1% in terms of MRR@20 and Recall@20. These improvements demonstrate that jointly considering both cross-domain behavior and knowledge is helpful for SR.

Second, MIFN outperforms the cross-domain recommendation baseline π -Net, which just makes use of information at user behavior level. Specifically, MIFN outperforms π -Net in terms of all metrics on both domains. It demonstrates that considering both knowledge and user behavior level information is better than only behavioral information. Meanwhile, it also proves the effectiveness

Table 4. Experimental Results (%) on the Amazon (“Movie-Book”) Dataset

Methods	Movie-domain						Book-domain					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
POP	0.10	0.11	0.13	0.20	0.29	0.58	0.11	0.14	0.16	0.20	0.44	0.75
BPR-MF	0.51	0.57	0.64	0.89	1.35	2.26	1.44	1.64	1.77	2.51	3.97	5.97
ItemKNN	1.05	1.27	1.48	2.11	3.84	6.99	1.35	1.64	1.95	2.88	5.10	9.69
GRU4REC	12.80	12.86	12.88	13.69	14.11	14.43	13.87	13.92	13.95	14.64	15.02	15.34
HRNN	13.38	13.43	13.45	13.95	14.25	14.58	14.57	14.61	14.62	14.99	15.25	15.46
NARM	13.80	13.85	13.86	14.20	14.53	14.80	15.25	15.26	15.27	15.57	15.66	15.78
STAMP	12.44	12.56	12.63	13.66	14.58	15.68	11.53	11.56	11.57	11.82	12.00	12.20
SRGNN	11.77	11.84	11.88	12.66	13.18	13.87	15.12	15.14	15.15	15.46	15.61	15.77
NCF-MLP++	1.64	1.86	2.03	2.95	4.61	7.24	1.76	1.98	2.11	3.20	4.84	7.34
Conet	1.43	1.73	2.01	2.83	5.20	9.24	1.17	1.36	1.51	2.13	3.54	5.77
π -Net	14.49	14.52	14.54	14.88	15.10	15.37	15.75	15.76	15.77	15.94	16.02	16.09
KGAT	4.16	4.35	4.48	5.81	7.25	9.18	5.93	6.19	6.36	7.91	9.92	12.32
KSR	14.18	14.23	14.26	14.91	15.28	15.73	15.84	15.87	15.89	16.51	16.72	16.92
MIFN-linear	14.00	14.07	14.10	14.88	14.91	15.22	15.32	15.34	15.35	15.79	15.99	16.07
MIFN-KTU	14.20	14.25	14.28	14.85	15.26	15.44	14.87	14.90	14.91	15.54	15.80	16.07
MIFN+ L_M	14.73	14.75	14.81	14.87	15.96	16.02	15.75	15.97	15.99	16.87	16.96	17.05
MIFN	14.84	14.87	14.88	15.13	16.34[†]	16.56[†]	16.05	16.16	16.23[†]	16.99[†]	17.03[†]	17.13[†]

Bold face indicates the best result in terms of the corresponding metric. Significant improvements over the results of the baseline (KSR) which performs best on most metrics are marked with [†] (t-test, $p < .05$).

of the KTU module of MIFN. With this module, MIFN is able to capture cross-domain knowledge and conduct information transfer in the KG so as to learn better sequence representations.

Third, we can observe that the results of MIFN in the “Book” domain are better than those in the “Movie” domain on the “Movie-Book” dataset. We believe that this is because the data are less sparse in the “Book” domain compared to the “Movie” domain. There is more space for improvement in the “Book” domain. With more interaction data, the models can identify more user preference characteristics in the dense domain so as to transfer it to the sparse domain through both the user behavioral information flow and the knowledge information flow. At the same time, we also find that the results in the “Food” domain are better than the “Kitchen” domain on the “Food-Kitchen” dataset. The final performance is not only related to the difference in sparsity between the datasets, but also to the characteristics of the domain and the degree of correlation between different domains.

Fourth, π -Net outperforms other sequential baselines, which means that cross domain information is beneficial to both domains. At the same time, knowledge aware methods outperform other sequential baselines, which also means that knowledge information can improve recommendation performance. Furthermore, it seems that considering knowledge is more useful than modeling cross-domain characteristics, as KSR slightly outperforms π -Net.

Fifth, the sequential methods achieve much better results in general. This is because RNN-based methods are able to capture the sequential characteristics and can obtain the better representations while the traditional methods neglect this information. What’s more, it seems that STAMP obtains lower results than the sequential method NARM in the “Movie-Book” dataset, while it performs

Table 5. Experimental Results (%) on the Amazon (“Food-Kitchen”) Dataset

Methods	Food-domain recommendation						Kitchen-domain recommendation					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
POP	0.40	0.49	0.56	0.82	1.50	2.15	0.22	0.25	0.27	0.40	0.60	1.47
BPR-MF	0.82	0.87	0.94	1.35	1.82	2.61	0.41	0.47	0.50	0.62	1.04	1.47
ItemKNN	1.55	1.98	2.43	3.28	6.70	12.47	1.13	1.44	1.90	2.60	4.77	11.08
GRU4REC	8.10	8.23	8.29	9.46	10.38	11.26	8.36	8.39	8.41	8.70	8.93	9.25
HRNN	7.22	7.35	7.45	8.49	9.47	10.93	7.81	7.86	7.88	8.29	8.61	9.12
NARM	9.43	9.54	9.62	10.34	11.86	12.23	8.41	8.44	8.46	8.69	8.91	9.21
STAMP	9.28	9.38	9.44	10.22	10.91	11.81	8.52	8.55	8.57	8.81	9.05	9.28
SRGNN	7.31	7.49	7.60	8.68	10.02	11.67	7.20	7.26	7.29	7.90	8.36	8.87
NCF-MLP++	2.01	2.24	2.42	3.45	5.19	8.02	0.87	1.03	1.17	1.72	3.00	4.99
Conet	3.38	3.64	3.82	5.07	7.07	9.73	3.30	3.55	3.71	5.09	7.03	9.47
π -Net	9.56	9.67	9.75	10.59	10.46	12.54	8.57	8.60	8.62	8.89	9.12	9.42
KGAT	4.74	4.97	5.09	6.49	8.18	9.94	4.85	5.09	5.23	6.96	8.77	10.81
KSR	9.79	9.91	9.98	10.82	11.78	12.77	9.03	9.07	9.08	9.39	9.62	9.92
MIFN-linear	9.65	9.76	9.89	10.73	11.34	12.81	9.00	9.05	9.17	9.33	9.36	9.41
MIFN-KTU	9.43	9.65	9.83	10.50	11.16	12.53	8.29	8.33	8.36	8.95	9.17	9.52
MIFN+ L_M	9.86	9.88	10.03	10.93	11.96	13.14	9.05	9.17	9.18	9.28	9.89	10.24
MIFN	9.91	10.16	10.25	11.20	12.25	13.27[†]	9.18	9.21	9.23	9.72[†]	10.01[†]	10.33

Bold face indicates the best result in terms of the corresponding metric. Significant improvements over the results of the baseline (KSR) which performs best on most metrics are marked with [†] (t-test, $p < .05$).

better in the “Kitchen” domain of “Food-Kitchen” dataset. We believe that this is because of differences in the datasets, e.g., we found that the user preferences in the kitchen domain are relatively more focused. And the method SRGNN performs worse than most sequential methods. This is because that SRGNN just employs the transition relation between adjacent items to construct the graph so as to get the representations of different items, however it does not consider the relation between the non-adjacent items (the other sequential methods do consider this), which may also affect item representations.

5.2 Ablation Study (RQ2)

To verify the effectiveness of the proposed modules, we design the ablation study to compare several model variants. The results are listed in Tables 4 and 5.

- (1) MIFN is the best performing variant; this variant includes both the BTU and KTU modules, and is trained with the recommendation loss L_R only.
- (2) MIFN-KTU is MIFN without the KTU module and performs information transfer only at the level of behavioral information;
- (3) MIFN-linear is MIFN with the KTU module but it replaces the GRU propagation function with a linear mapping layer;
- (4) MIFN+ L_M is MIFN by adding the mode switch loss.

First, by removing KTU, the performance of MIFN-KTU is dramatically less than that of MIFN, which confirms that considering knowledge flow can improve the cross-domain recommendations. In addition, the results of MIFN-KTU are worse than those of the knowledge-aware method KSR,

Table 6. Analysis of the Knowledge Graph Construction Algorithm on the Amazon (“Movie-Book”) Dataset

Ratios	Movie-domain recommendation						Book-domain recommendation					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
30%	16.46	16.83	16.99	21.27	24.33	28.97	18.51	18.70	18.85	25.82	27.32	31.38
50%	19.53	20.43	20.83	30.33	37.07	42.67	25.04	25.68	25.98	36.08	40.75	45.12
70%	23.41	23.43	23.45	42.54	47.99	50.61	27.12	28.31	28.41	48.93	56.64	57.31
90%	33.45	34.99	35.09	65.81	76.83	77.97	40.01	40.86	43.00	79.83	80.50	83.86
100%	67.04	67.56	67.69	89.57	93.30	95.15	83.08	83.46	83.62	91.97	94.79	97.02

The different ratios represent the different proportions of the predicted ground truth items appeared in the KG.

while MIFN outperforms KSR on all domains. This indicates that the KTU module is able to make good use of the cross-domain knowledge and is able to better capture user preferences by modeling the cross-domain knowledge flow.

Second, by replacing the GRU propagation function (Equation (5)) with a linear mapping layer, the performance of MIFN-linear drops a little compared with the full model MIFN. This demonstrates that the GRU propagation is also useful for the combination of cross-domain knowledge for CDSR. At the same time, we find that the MIFN-linear outperforms MIFN-KTU in terms of most metrics. This indicates that the cross-domain graph convolution mechanism without GRU propagation is also capable of capturing cross-domain knowledge information flow. But the performance of MIFN-linear is still lower than the KSR in terms of most metrics in both datasets and MIFN outperforms KSR in terms of all metrics of both datasets. This again shows that GRU propagation is more effective for cross-domain knowledge representation.

Third, if we jointly train the recommendation loss L_R and the mode switch loss L_M , the performance drops a little but its performance is still than that of the baselines. The switch loss L_M assumes that if the next item does not exist in the item set, it must be recommended under the *graph mode*, which makes the model tend to recommend items existing in the graph. However, since there already exists a similar supervision signal in L_R , which assumes that each item is recommended under the *graph mode*, the *sequence mode*, or a combination of both. Further adding the L_M loss introduces unnecessary bias toward *graph mode*.

5.3 Influence of the Knowledge Graph Construction Algorithm (RQ3)

The number of triples in the complete KG is large (see Table 3), so we propose a knowledge graph construction method as detailed in Algorithm 1 to build KG for each sequence. To study the effect of the knowledge graph construction method, we design an experiment aimed at analyzing the effect of the ratios of the ground truth items in the constructed KG on the final recommendation performance. We achieve this by simulating and controlling the ratios artificially. Specifically, we add the ground truth items to the extracted entities according to the specified ratios in advance. The results are given in Tables 6 and 7.

First, we can see that the performance increases as the ratio of ground truth item appeared in the KG increases on both datasets. For instance, when the ratio varies from 30% to 100%, the value of MRR@20 increases from 16.99% to 67.69% in the “Movie” domain and the value of Recall@20 increases from 22.41% to 95.45% in the “Kitchen” domain. This demonstrates that a good knowledge graph construction algorithm is of vital importance.

Table 7. Analysis of the Knowledge Graph Construction Algorithm on the Amazon (“Food-Kitchen”) Dataset

Ratios	Food-domain recommendation						Kitchen-domain recommendation					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
30%	11.75	12.30	12.72	17.67	21.77	27.99	10.96	11.51	11.99	13.71	15.52	22.41
50%	22.14	22.61	22.74	35.21	38.51	42.47	22.75	23.20	23.74	34.56	37.86	41.64
70%	26.01	26.74	27.35	41.52	47.11	56.17	25.46	26.24	26.79	37.54	43.51	51.60
90%	36.58	37.50	38.11	59.72	66.57	75.61	31.29	32.88	33.40	55.21	66.78	74.17
100%	47.40	48.51	48.60	85.15	92.89	94.13	43.78	45.57	45.77	80.16	92.68	95.45

The different ratios represent the different proportions of the predicted ground truth items appeared in the KG.

Second, we notice that when the ratio increases linearly, the results do not increase linearly. On both datasets, the increase is relatively slow from 30% to 90%. However, when we simulate the ratio from 90% to 100%, the performance is greatly improved in terms of both MRR and Recall on the “Movie-Book” dataset. We believe that this is because when the ratios reaches a certain value, the model can easily capture the characteristics of recommended items from the *graph mode* in most cases and relies mostly on the KG to do recommendations. We also notice that with 100% ratio, Recall is improved largely on “Food-Kitchen” dataset while MRR is not. We think this is because of the density of the KG. As shown in Table 3, there are more triples in the “Movie-Book” dataset, especially the “Book” domain. The richer knowledge makes it relatively easier to rank the ground truth items.

Currently, using Algorithm 1, the ratios of the ground truth items in the KG of the “Movie-Book” and “Food-Kitchen” datasets are 14% and 12%, respectively, both of which are relatively low. Algorithm 1 is only based on the entity distance calculated using the pretrained entity representations in the KG, which is insufficient. To this end, we think an important future research direction in order to further improve MIFN is to design a more effective knowledge graph construction algorithm.

5.4 Qualitative Analysis with Case Studies (RQ4)

To analyze the recommendation results with and without a flow of knowledge, we list some examples from the “Movie-Book” dataset. Figure 5 shows recommendations when the extracted KG is relevant to the current user preference, and Figure 6 shows recommendations when the extracted KG is irrelevant. Figures 5(a) and 6(a) are recommendations from MIFN, while Figures 5(b) and 6(b) are recommendations from MIFN-KTU. In each figure, the orange color represents the interactions in the “Movie” domain, and the blue color represents the interactions in the “Book” domain. The meaning of the different colored fonts and lines are explained in the legend. The green tick indicates that the recommendation is correct, and the red cross indicates it is wrong.

From Figure 5, we can observe that when using the extracted KG (Figure 5(a)), MIFN can give correct recommendations for both domains, however, the recommendation is wrong in the “Book” domain when the KG is not used (Figure 5(b)). Furthermore, it should be noted that the mode switch probabilities in Figure 5(a) are different for the two domains. The probability of *sequence mode* is 1.0 in the “Movie” domain, but in the “Book” domain, the probability of *graph mode* is 0.75, which means that the recommendation of the “Movie” domain comes from the item set, while the recommendation mostly relies on the KG for the “Book” domain. From the KG, the book entity

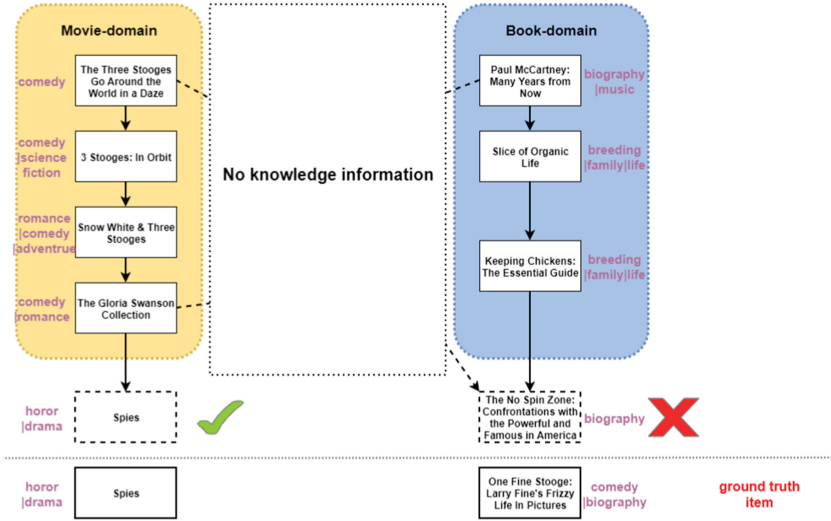
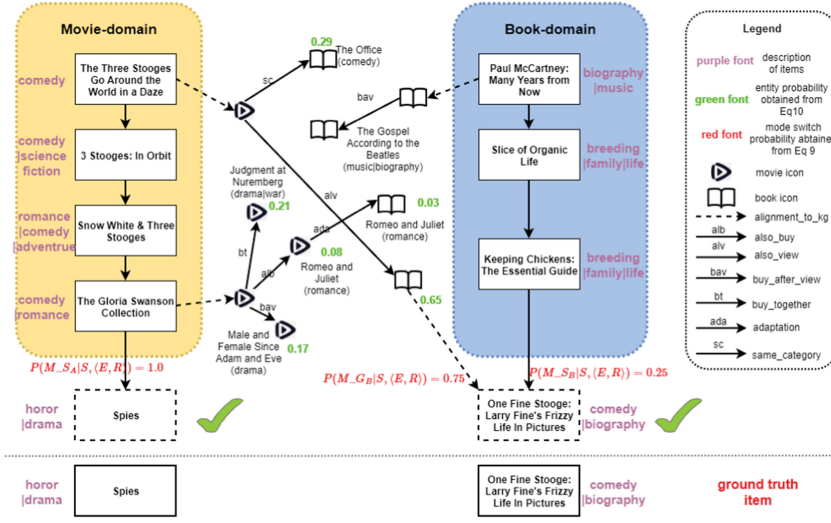
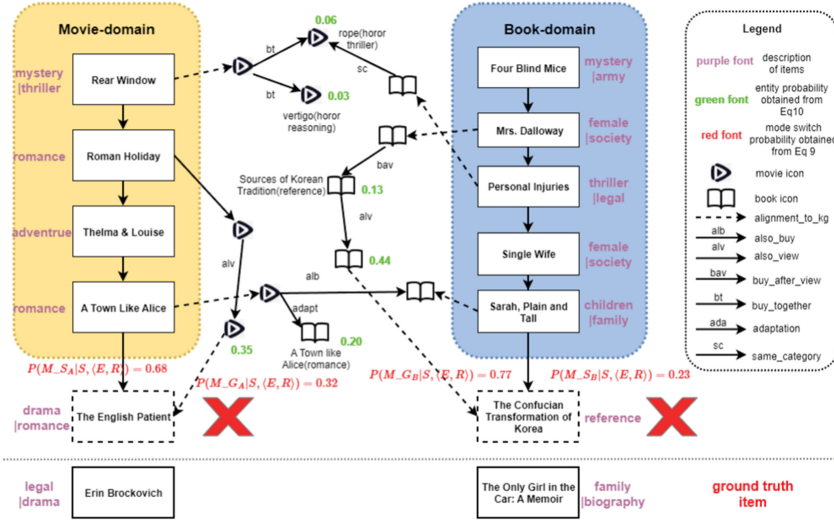
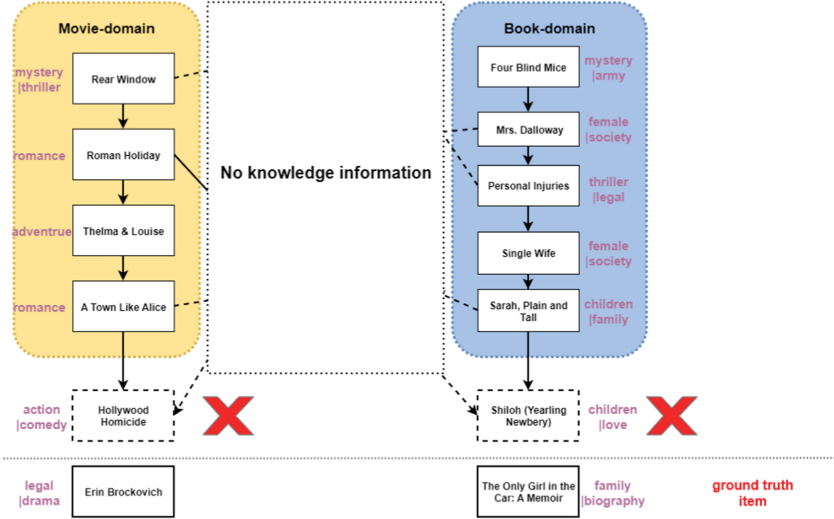


Fig. 5. Case study of MIFN and MIFN-KTU when the extracted KG is relevant to the current user preference.

“One Fine Stooge: Larry Fine’s Frizzy Life In Pictures” gets the highest recommendation score of 0.65 in the “Book” domain. The reason is that there exists a knowledge triple, that is, people who watch the movie “The Three Stooges Go Around the World in a Daze” will also view the book “One Fine Stooge: Larry Fine’s Frizzy Life In Pictures.” This knowledge is well transferred by the flow of knowledge to obtain a better recommendation for the “Book” domain. In contrast, as shown in Figure 5(b), the recommendation is wrong because the model only relies on the flow of behavioral information to recommend items from the item set.



(a) Recommendation from MIFN.



(b) Recommendation from MIFN-KTU.

Fig. 6. Case study of MIFN and MIFN-KTU when the extracted KG is irrelevant to the current user preference.

On the other hand, we also found that the flow of knowledge is not always helpful. As shown in Figure 6(a), MIFN still gives a wrong recommendation even by modeling knowledge information flow with KTU. MIFN recommends “The confucian Transformation of Korea” with the *graph mode* recommendation probability of 0.77, which means, in this case, it still relies mostly on the KG to do recommendation. However, we can observe that most of the extracted entities in the KG are thriller movies or reference books which are not relevant to the current user preference. As a result, MIFN performs somewhat worse than MIFN-KTU, because although MIFN-KTU also recommends the wrong item, the recommended item seems more relevant to the user preference,

Table 8. Analysis of the Number of Entities N on the Amazon (“Movie-Book”) Dataset

N	Movie-domain recommendation						Book-domain recommendation					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
50	14.34	14.37	14.39	14.83	15.04	15.37	15.66	15.67	15.67	15.86	15.95	16.04
100	14.38	14.43	14.44	14.88	15.09	15.46	15.70	15.71	15.72	15.90	16.02	16.11
150	14.50	14.51	14.53	15.08	16.23	16.34	16.00	16.01	16.05	16.78	16.93	17.00
180	14.70	14.73	14.74	15.15	16.20	16.42	15.98	16.03	16.07	16.95	17.04	17.18
200	14.84	14.87	14.88	15.13	16.34	16.56	16.05	16.16	16.23	16.99	17.03	17.13

Different N represents the different number of entities of the extracted KG.

Table 9. Analysis of the Number of Entities N on the Amazon (“Food-Kitchen”) Dataset

N	Food-domain recommendation						Kitchen-domain recommendation					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
100	9.56	9.77	9.96	10.56	11.34	12.85	8.64	8.77	8.99	8.92	9.00	9.29
150	9.67	9.81	9.98	10.79	11.82	13.00	9.08	9.12	9.15	9.58	9.63	9.97
170	9.98	10.12	10.19	11.24	12.23	13.15	9.06	9.10	9.21	9.89	10.10	10.38
180	9.78	9.96	10.06	11.19	12.16	13.17	9.06	9.08	9.11	9.89	10.14	10.20
200	9.91	10.16	10.25	11.20	12.25	13.27	9.18	9.21	9.23	9.72	10.01	10.33

Different N represents the different number of entities of the extracted KG.

i.e., “children|family.” This suggests that the quality of the extracted KG has a large impact on the final recommendation performance, which further verifies the conclusion in Section 5.3.

5.5 Hyperparameters Analysis (RQ5)

We conduct experiments to investigate the impact of the hyper-parameters in the MIFN framework, including the number of entities N and the maximum hop count H .

The number of entities N is used to limit the size of the KG during KG construction. The results of MIFN for different numbers of N are shown in Tables 8 and 9, with N ranging within [50, 100, 150, 180, 200] for the “Movie-Book” dataset and [100, 150, 170, 180, 200] for the “Food-Kitchen” dataset. The maximum hop count H is also used to limit the KG size. We show the results in Tables 10 and 11 when $H = 1, 2, 3, 5$ for both datasets. From these results, we can observe that when values of N and H drop, the performance does not decrease dramatically. This demonstrates that MIFN is not sensitive to both hyperparameters.

6 CONCLUSION AND FUTURE WORK

In this article, we study how to incorporate knowledge into the **cross-domain sequential recommendation** task. We present MIFN, which jointly models two types of information flow across domains, i.e., of behavioral information and of knowledge. To verify the effectiveness of MIFN, we conduct experiments on datasets from four Amazon domains. The results demonstrate that MIFN outperforms other state-of-the-art baselines. Through extensive analysis experiments, we confirm

Table 10. Analysis of the Maximum Hop Count H on the Amazon (“Movie-Book”) Dataset

H	Movie-domain recommendation						Book-domain recommendation					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
1	14.69	14.71	14.72	14.82	15.31	15.67	15.80	15.81	15.83	16.06	16.37	16.86
2	14.84	14.87	14.88	15.13	16.34	16.56	16.05	16.16	16.23	16.99	17.03	17.13
3	14.88	14.89	14.91	15.34	16.39	16.66	16.01	16.13	16.22	16.80	16.99	17.04
5	14.79	14.80	14.83	15.16	16.38	16.60	15.98	16.07	16.19	16.48	16.69	17.08

Different H represents the different hop layer in which entities and relations can be extracted.

Table 11. Analysis of the Maximum Hop Count H on the Amazon (“Food-Kitchen”) Dataset

H	Food-domain recommendation						Kitchen-domain recommendation					
	MRR			Recall			MRR			Recall		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
1	9.77	9.81	9.95	10.97	11.78	12.81	9.10	9.13	9.19	9.45	9.80	10.00
2	9.91	10.16	10.25	11.20	12.25	13.27	9.18	9.21	9.23	9.72	10.01	10.33
3	9.94	10.33	10.40	11.11	11.96	12.97	9.09	9.14	9.15	9.47	9.64	9.98
5	9.87	10.21	10.35	11.16	12.01	12.98	9.08	9.13	9.16	9.67	10.00	10.28

Different H represents the different hop layer in which entities and relations can be extracted.

that the flow of knowledge helps improve the recommendation performance in general, which means mixed flow of information can be used to enhance the CDSR performance.

There are also some limitations in this work. Currently the knowledge graph construction algorithm used to extract graphs is simple. So the quality of extracted graph is limited, which we show can be made more effective by including as much relevant knowledge as possible. And we only focus on the applications of MIFN to the scenario with two domains, which can be extended to multiple domains simultaneously. What’s more, some prior correlations are required between different domains. Now we assume that the users are overlapped between different domains during modeling the information flow, which limits the cross-domain research.

As to future work, MIFN can be enhanced in at least four directions correspondingly. First, we will further study knowledge graph construction algorithms to improve the quality of the KG without increasing its size. Second, MIFN is limited to information flow between two domains in this work. Therefore, we want to study how to make cross-domain recommendations across multiple domains. Third, it is worth trying to study the cross-domain transfer learning so as to improve the performance of both domains when there is no relationship between them. Fourth, we have only experimented on e-commerce datasets from Amazon as other datasets that meet our requirements do not seem to be available. We hope our work can inspire other researchers working on different recommendation scenarios.

DATA AND CODE

To facilitate reproduction of the results in the article, we are sharing the code and resources used to produce our results at <https://github.com/mamuyang/MIFN>.

REFERENCES

- [1] Fabian Abel, Eelco Herder, Geert-Jan Houben, Nicola Henze, and Daniel Krause. 2013. Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction* 23, 2 (2013), 169–209.
- [2] Qingyao Ai, Wahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms* 11, 9 (2018), 137.
- [3] Georgios Alexandridis, Georgios Siolas, and Andreas Stafylopatis. 2017. ParVecMF: A paragraph vector-based matrix factorization recommender system. <https://arxiv.org/abs/1706.07513>.
- [4] Georgios Alexandridis, Thanos Tagaris, Giorgos Siolas, and Andreas Stafylopatis. 2019. From free-text user reviews to product recommendation using paragraph vectors and matrix factorization. In *Proceedings of the 2019 World Wide Web Conference*. 335–343.
- [5] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2007. Distributed collaborative filtering with domain specialization. In *Proceedings of the 2007 ACM Conference on Recommender Systems*. 33–40.
- [6] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2008. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction* 18, 3 (2008), 245–286.
- [7] Ye Bi, Liqiang Song, Mengqiu Yao, Zhenyu Wu, Jianming Wang, and Jing Xiao. 2020. DCDIR: A deep cross-domain recommendation system for cold start users in insurance domain. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1661–1664.
- [8] Ye Bi, Liqiang Song, Mengqiu Yao, Zhenyu Wu, Jianming Wang, and Jing Xiao. 2020. A heterogeneous information network based cross domain insurance recommendation system for cold start users. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2211–2220.
- [9] Veronika Bogina and Tsvi Kuflik. 2017. Incorporating dwell time in session-based recommendations with recurrent Neural networks. In *Proceedings of the 2017 Conference on Recommender Systems*. 57–59.
- [10] Shuo Chen, Josh L. Moore, Douglas Turnbull, and Thorsten Joachims. 2012. Playlist prediction via metric embedding. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 714–722.
- [11] Tianwen Chen and Raymond Chi-Wing Wong. 2021. An efficient and effective framework for session-based social recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 400–408.
- [12] Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten de Rijke. 2020. Improving end-to-end sequential recommendations with intent-aware diversification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 175–184.
- [13] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiayi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. 108–116.
- [14] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan S. Kankanhalli. 2018. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *Proceedings of the 2018 World Wide Web Conference*. 639–648.
- [15] Paolo Cremonesi, Antonio Tripodi, and Roberto Turrin. 2011. Cross-domain recommender systems. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*. 496–503.
- [16] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential user-based recurrent neural network recommendations. In *Proceedings of the 11th ACM Conference on Recommender Systems*. 152–160.
- [17] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*. 278–288.
- [18] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskis, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *Proceedings of the Spanish Conference on Information Retrieval*. 24.
- [19] Ignacio Fernández-Tobías, Iván Cantador, and Laura Plaza. 2013. An emotion dimensional model based on social tags: Crossing folksonomies and enhancing recommendations. In *Proceedings of the International Conference on Electronic Commerce and Web Technologies*. 88–100.
- [20] Ignacio Fernández-Tobías, Iván Cantador, Paolo Tomeo, Vito Walter Anelli, and Tommaso Di Noia. 2019. Addressing the user cold start with cross-domain collaborative filtering: Exploiting item metadata in matrix factorization. *User Modeling and User-Adapted Interaction* 29, 2 (2019), 443–486.
- [21] Chen Gao, Xiangning Chen, Fuli Feng, Kai Zhao, Xiangnan He, Yong Li, and Depeng Jin. 2019. Cross-domain recommendation without sharing user-relevant data. In *Proceedings of the World Wide Web Conference*. 491–502.
- [22] Sharon Givon and Victor Lavrenko. 2009. Predicting social-tags for cold start book recommendations. In *Proceedings of the 3rd ACM Conference on Recommender Systems*. 333–336.
- [23] Lei Guo, Li Tang, Tong Chen, Lei Zhu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2021. DA-GCN: A domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*.

- [24] Ruining He and Julian McAuley. 2016. Fusing similarity models with Markov chains for sparse sequential recommendation. In *Proceedings of the 2016 IEEE 16th International Conference on Data Mining*. 191–200.
- [25] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 355–364.
- [26] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182.
- [27] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations*.
- [28] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 241–248.
- [29] Guangneng Hu. 2019. Personalized neural embeddings for collaborative filtering with text. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2082–2088.
- [30] Guangneng Hu and Xinyu Dai. 2017. Integrating reviews into personalized ranking for cold start recommendation. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 708–720.
- [31] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 667–676.
- [32] Guang-Neng Hu, Xin-Yu Dai, Feng-Yu Qiu, Rui Xia, Tao Li, Shu-Jian Huang, and Jia-Jun Chen. 2018. Collaborative filtering with topic and social latent factors incorporating implicit feedback. *ACM Transactions on Knowledge Discovery from Data* 12, 2 (2018), 1–30.
- [33] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. 2013. Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22nd International Conference on World Wide Web*. 595–606.
- [34] Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. 2019. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 573–581.
- [35] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 505–514.
- [36] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*.
- [37] Adit Krishnan, Mahashweta Das, Mangesh Bendre, Hao Yang, and Hari Sundaram. 2020. Transfer learning via contextual invariants for one-to-many cross-domain recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1081–1090.
- [38] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. 2052–2057.
- [39] Chenliang Li, Xichuan Niu, Xiangyang Luo, Zhenzhong Chen, and Cong Quan. 2019. A review-driven neural model for sequential recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2866–2872.
- [40] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.
- [41] Pan Li and Alexander Tuzhilin. 2019. DDTCDR: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 331–339.
- [42] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2017. CCCFNet: A content-boosted collaborative filtering neural network for cross domain recommender systems. In *Proceedings of the 26th International Conference on World Wide Web Companion*. 817–818.
- [43] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. Explainable outfit recommendation with joint outfit matching and comment generation. *IEEE Transactions on Knowledge and Data Engineering* 32, 8 (2019), 1502–1516.
- [44] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. 2016. Context-aware sequential recommendation. In *Proceedings of the 2016 IEEE 16th International Conference on Data Mining*. 1053–1058.
- [45] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1831–1839.

- [46] Babak Loni, Yue Shi, Martha A. Larson, and Alan Hanjalic. 2014. Cross-domain collaborative filtering with factorization machines. In *Proceedings of the 36th European Conference on IR Research on Advances in Information Retrieval*. 656–661.
- [47] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. π -Net: A parallel information-sharing network for shared-account cross-domain sequential recommendations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 685–694.
- [48] Renfeng Ma, Qi Zhang, Jiawen Wang, Lizhen Cui, and Xuanjing Huang. 2018. Mention recommendation for multimodal microblog with cross-attention memory network. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 195–204.
- [49] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly learning explainable rules for recommendation with knowledge graph. In *Proceedings of the World Wide Web Conference*. 1210–1221.
- [50] Weizhi Ma, Min Zhang, Chenyang Wang, Cheng Luo, Yiqun Liu, and Shaoping Ma. 2018. Your tweets reveal what you like: Introducing cross-media content information into multi-domain recommendation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3484–3490.
- [51] Lei Mei, Pengjie Ren, Zhumin Chen, Liqiang Nie, Jun Ma, and Jian-Yun Nie. 2018. An attentive interaction network for context-aware recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 157–166.
- [52] Wenjing Meng, Deqing Yang, and Yanghua Xiao. 2020. Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1091–1100.
- [53] Nima Mirbakhsh and Charles X. Ling. 2015. Improving top-n recommendation for cold-start users via cross-domain information. *ACM Transactions on Knowledge Discovery from Data* 9, 4 (2015), 1–19.
- [54] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3994–4003.
- [55] WeiKe Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*. 230–235.
- [56] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *ACM Computing Surveys* 51, 4 (2018), 66.
- [57] Massimo Quadrana, Alexandros Karatzoglou, Balzs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the 11th ACM Conference on Recommender Systems*. 130–137.
- [58] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. RepeatNet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 4806–4813.
- [59] Pengjie Ren, Zhaochun Ren, Fei Sun, Xiangnan He, Dawei Yin, and Maarten de Rijke. 2020. NLP4REC: The WSDM 2020 workshop on natural language processing for recommendations. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 907–908.
- [60] Ruiyang Ren, Zhaoyang Liu, Yaliang Li, Wayne Xin Zhao, Hui Wang, Bolin Ding, and Ji-Rong Wen. 2020. Sequential recommendation with self-attentive multi-adversarial network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 89–98.
- [61] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. 452–461.
- [62] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*. 811–820.
- [63] Shaghayegh Sahebi and Peter Brusilovsky. 2013. Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. In *Proceedings of the 21st Conference on User Modeling, Adaptation and Personalization*. 289–295.
- [64] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*. 285–295.
- [65] Guy Shani, David Heckerman, and Ronen I. Brafman. 2005. An MDP-based recommender system. *Journal of Machine Learning Research* 6, 9 (2005), 1265–1295.
- [66] Bracha Shapira, Lior Rokach, and Shirley Freilikhman. 2013. Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction* 23, 2–3 (2013), 211–247.
- [67] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 555–563.

- [68] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1441–1450.
- [69] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. 17–22.
- [70] Amit Tiroshi, Shlomo Berkovsky, Mohamed Ali Kâafar, Terence Chen, and Tsvi Kuflik. 2013. Cross social networks interests predictions based on graph features. In *Proceedings of the 7th ACM Conference on Recommender Systems*. 319–322.
- [71] Amit Tiroshi and Tsvi Kuflik. 2012. Domain ranking for cross domain collaborative filtering. In *Proceedings of the 20th International Conference on User Modeling, Adaptation, and Personalization*. 328–333.
- [72] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make it a chorus: Knowledge-and time-aware item modeling for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 109–118.
- [73] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 417–426.
- [74] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge graph convolutional networks for recommender systems. In *Proceedings of the World Wide Web Conference*. 3307–3313.
- [75] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. A collaborative session-based recommendation approach with parallel memory modules. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 345–354.
- [76] Pengfei Wang, Yu Fan, Long Xia, Wayne Xin Zhao, Shaozhang Niu, and Jimmy Huang. 2020. KERL: A knowledge-guided reinforcement learning model for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 209–218.
- [77] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 403–412.
- [78] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 950–958.
- [79] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item silk road: Recommending items from information domains to social users. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 185–194.
- [80] Yaqing Wang, Chunyan Feng, Caili Guo, Yunfei Chu, and Jenq-Neng Hwang. 2019. Solving the sparsity problem in recommendations via cross-domain item embedding based on co-clustering. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 717–725.
- [81] Yufei Wen, Lei Guo, Zhumin Chen, and Jun Ma. 2018. Network embedding based recommendation method in social networks. In *Proceedings of the 11th Web Conference*. 11–12.
- [82] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 346–353.
- [83] Xiang Wu, Qi Liu, Enhong Chen, Liang He, Jingsong Lv, Can Cao, and Guoping Hu. 2013. Personalized next-song recommendation in online karaokes. In *Proceedings of the 7th ACM Conference on Recommender Systems*. 137–140.
- [84] Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 285–294.
- [85] Guang Yang, Xiaoguang Hong, Zhaohui Peng, and Yang Xu. 2020. Long short-term memory with sequence completion for cross-domain sequential recommendation. In *Proceedings of the Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*. 378–393.
- [86] Ghim-Eng Yap, Xiao-Li Li, and S. Yu Philip. 2012. Effective next-items recommendation via personalized sequential pattern mining. In *Proceedings of the 17th International Conference on Database Systems for Advanced Applications*. 48–64.
- [87] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 353–362.
- [88] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor Sheng, Jiajie Xu, Deqing Wang, Guanfang Liu, and Xiaofang Zhou. 2019. Feature-level deeper self-attention network for sequential recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 4320–4326.

- [89] Cheng Zhao, Chenliang Li, Rong Xiao, Hongbo Deng, and Aixin Sun. 2020. CATN: Cross-domain recommendation for cold-start users via aspect transfer network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 229–238.
- [90] Wayne Xin Zhao, Jinpeng Wang, Yulan He, Ji-Rong Wen, Edward Y. Chang, and Xiaoming Li. 2016. Mining product adopter information from online reviews for improving product recommendation. *ACM Transactions on Knowledge Discovery from Data* 10, 3 (2016), 1–23.
- [91] Lin Zheng, Naicheng Guo, Weihao Chen, Jin Yu, and Dazhi Jiang. 2020. Sentiment-guided sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1957–1960.
- [92] Fuzhen Zhuang, Yingmin Zhou, Fuzheng Zhang, Xiang Ao, Xing Xie, and Qing He. 2018. Cross-domain novelty seeking trait mining for sequential recommendation. <https://arxiv.org/abs/1803.01542>.
- [93] Andrew Zimdars, David Maxwell Chickering, and Christopher Meek. 2001. Using temporal data for making recommendations. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*. 580–588.

Received December 2020; revised July 2021; accepted September 2021