UvA-DARE (Digital Academic Repository)

Mixed Information Flow for Cross-domain Sequential Recommendations

Ma, M.; Ren, P.; Chen, Z.; Ren, Z.; Zhao, L.; Ma, J.; de Rijke, M.

Mixed Information Flow for Cross-domain Sequential Recommendations

MUYANG MA, Shandong University, China
PENGJIE REN, Shandong University, China
ZHUMIN CHEN, Shandong University, China
ZHAOCHUN REN, Shandong University, China
LIFAN ZHAO, Shandong University, China
JUN MA, Shandong University, China
MAARTEN DE RIJKE, University of Amsterdam and Ahold Delhaize, The Netherlands

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 paper, 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 mixed information flow network is able to further improve recommendation performance in different domains by modeling mixed information flow.

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

Additional Key Words and Phrases: Cross-domain recommendation, Sequential recommendation, Knowledge base, Graph transfer

Pengjie Ren and Zhumin Chen are the corresponding authors.
Authors' addresses: Muyang Ma, Shandong University, Jinan, China, muyang0331@gmail.com; Pengjie Ren, Shandong University, Jinan, China, jay.ren@outlook.com; Zhumin Chen, Shandong University, Jinan, China, chenzhumin@sdu.edu.cn; Zhaochun Ren, Shandong University, Qingdao, China, zhaochun.ren@sdu.edu.cn; Lifan Zhao, Shandong University, Qingdao, China, mogician233@outlook.com; Jun Ma, Shandong University, Jinan, China, majun@sdu.edu.cn; Maarten de Rijke, University of Amsterdam and Ahold Delhaize, Amsterdam and Zaandam, The Netherlands, m.derijke@uva.nl.

1 INTRODUCTION

Sequential recommendation (SR) aims to predict the next item that the user is most likely to interact with based on her/his past sequential behavior (e.g., clicks on items) [48]. Recently, cross-domain sequential recommendation (CDSR) has emerged as a way to promote recommendation performance by leveraging and combining information from different domains [39]. 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.
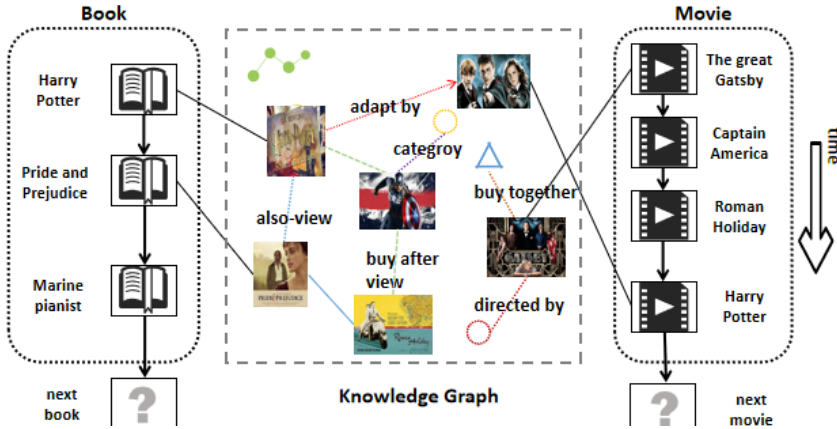

Fig. 1. Illustration of a connection between a user's behavioral information across domains via a knowledge graph. Lines in different colors represent different connections.

Zhuang et al. [79] and Ma et al. [39] 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. For example, as illustrated in Figure 1, assume 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. 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 [35, 50]. Of particular relevance to us is work that has proposed to incorporate knowledge and combine it with behavioral information for SR [see, e.g., 26, 27, 70]. 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 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, 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) sequential recommendation, (2) cross-domain recommendation, and (3) knowledge-aware recommendation.

2.1 Sequential recommendations

Early work on recommender systems (RSs) typically use collaborative filtering (CF) to generate recommendations [24] 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 chainss (MCs) [7, 18, 53, 80] and Markov decision processs (MDPs) [56, 71] have been used to predict users' next action given information about their past behavior [66, 73]. All these methods take into account the sequential characteristics. However, there are considerable challenges with the size of the state when considering the entire sequence [47].

Recurrent neural networks (RNNs) have been introduced to SR to handle variable-length sequential data. Hidasi et al. [21] are the first to leverage RNNs for SR. They utilize session-parallel mini-batch training and employ ranking-based loss functions to train the model. Then, Tan et al. [60] 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. [32] incorporate an attention mechanism into the encoder to capture the users' main preference in the current sequence. Ren et al. [49] point out the repeat consumption in SR, where the same item is re-consumed repeatedly over time. Quadrana et al. [48] propose a hierarchical RNN model that can relay and evolve latent hidden states of the RNNs across user sequences. Donkers et al. [11] explicitly model user information in a gated architecture with extra input layers for gated recurrent unit (GRU). Memory enhanced RNN has been well studied for SR recently. Chen et al. [8] introduce a memory mechanism to SR and design

a memory-augmented neural network integrated with the insights of CF. Wang et al. [65] 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. [70] 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. [75] propose a feature-level deeper self-attention network to capture transition patterns between features of items by integrating various heterogeneous features. Sun et al. [59] 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. [22] 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. [36] incorporate contextual information into SR and propose a context-aware RNN model to capture external situations and lengths of time intervals. Bogina and Kuflik [6] 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. [40] propose a cross-attention memory network for multi-modal tweets via both textual and visual information. Li et al. [31] 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. [78] 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. [51] 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 [4, 46] in RS. It aims to mitigate the lack of data by exploiting user preference and item attributes in domains distinct but related to the target domain [15].

Traditional methods for cross-domain recommendation can be grouped into two main categories [13]. One category of methods *aggregates information* across different domains. According to different aggregating strategies, such methods can be further divided into three groups [15]. 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 [3, 54], leveraging users' social influence [1, 14], linking users' preference by a multi-domain graph [10, 61] or user behavioral information features [38, 42]. 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 [62] and Shapira et al. [57] 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 to model the weights assigned to recommendations coming from different domains. For example, Givon and Lavrenko [17] 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. [25] propose tensor-based factorization to share latent features between different domains by using the same parameters in both factorization models. Li et al. [30] 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 [44] 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. Elkahky et al. [12] propose a multi-view deep learning recommendation system by using rich auxiliary features to represent users from different domains. Then, Lian et al. [34] 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. [23] propose a model using a cross-stitch network [45] to learn complex user behavioral information based on neural CF [20]. Wang et al. [68] propose to combine user behavioral information in information domains and user-user connection in social domains to do recommendation. Wang et al. [69] embed item-level information and cluster-level correlative information from different domains into a unified framework. Gao et al. [16] transfer item embeddings across domains without sharing user-relevant data. Li and Tuzhilin [33] 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. [29] 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. [76] 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. [5] 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. Despite the fact that the listed methods 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. [79] 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. Ma et al. [39] 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 to explore extra knowledge to promote sequence representation across domains.

2.3 Knowledge-aware recommendations

Considerable efforts have been made to utilize side-information, especially knowledge graphs, to enhance the performance of recommendations. Zhao et al. [77] 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. [74] 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 [28], there have been a number of publications that propose variants of GCNs for recommendation by considering different types of information. Wang et al. [63] simulate users' hierarchical preferences over knowledge entities by extending users' potential preferences along links in a KG. Wang et al. [64] consider the connections among items based on higher-order entity features in KGs. Wang et al. [67] explicitly model the high-order connections in KGs by employing an attention mechanism to discriminate the importance of the neighbors. Ma et al. [41] 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. [72] 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 has also been considered in SRs. Huang et al. [27] 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. [58] model users' social influence with a graph-attention neural network, which dynamically infers the influencers based on the users' current preference. Huang et al. [26] introduce a taxonomy-aware memory-based multi-hop reasoning architecture by incorporating taxonomy data as structural knowledge to enhance the reasoning capacity.

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 paper.

3.1   Task formulation

Cross-domain sequential recommendation (CDSR) aims to predict 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 $\mathbb{A} = \{A_1, A_2, A_3, \ldots, A_n\}$ denote the item set for domain A, which consist of n unique items. Similarly, let $\mathbb{B} = \{B_1, B_2, B_3, \ldots, B_m\}$ 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 $S = [A_1, B_1, B_2, \ldots, A_i, \ldots, B_j, \ldots]$, 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 $\langle e_{A_1}, Is\_the\_same\_category, e_{B_3} \rangle$ means that entity e_{A_1} from domain A has the same category as entity e_{B_3} from domain B. Since we aim to link 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.

Table 1. Summary of main symbols and notation used in the paper.

$\mathbb{A}$	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., $S = [A_1, B_1, B_2, \ldots, A_i, \ldots, B_j, \ldots]$.
$\mathbb{S}$	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.
$h_{(A \to B)_i}$	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.

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 Eq. 1:

$$P(A_{i+1}|S, \langle E, R \rangle) \sim f_A(S, \langle E, R \rangle)$$
$$P(B_{j+1}|S, \langle E, R \rangle) \sim f_B(S, \langle E, R \rangle), \quad (1)$$

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 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 $f_B(S, \langle E, R \rangle)$.

3.2 Knowledge graph construction

In this work, we extract data (entities and relations) from the Amazon product collection[1] 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 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

---

[1]https://jmcauley.ucsd.edu/data/amazon
[2]https://en.wikipedia.org/

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 $\vartheta_1^0 = S_A$, $\vartheta_2^0 = S_B$, $connected = false$;
 3: **for** $k \in range(H)$ **do**
 4:    Extract related triples $\vartheta_1^k = \{(h, r, t) | (h, r, t) \in \langle E, R, E \rangle$ and $h \in \vartheta_1^{k-1}\}$;
 5:    Extract related triples $\vartheta_2^k = \{(h, r, t) | (h, r, t) \in \langle E, R, E \rangle$ 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:        θ = Select_triples($\vartheta_1^k, \vartheta_2^k, S_A, S_B$);
 9:        A = Construct_Adjacency_matrix(θ);
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 that are related to items involved in the hybrid interaction sequence S within H hops. We stop extracting more hops if it meets the requirement that there is a path for any given pair of items from both domains. Finally, we construct the relational adjacency matrix for the extracted KG.

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 ϑ_1^0 or ϑ_2^0 with the item entities (see line 2). At each hop, we extract all triples connected to the entities in the current KG (i.e., ϑ_1^k and ϑ_2^k), where $\vartheta_1^k = \{(h, r, t) | (h, r, t) \in \langle E, R, E \rangle$ and $h \in \vartheta_1^{k-1}\}$, and $\vartheta_2^k = \{(h, r, t) | (h, r, t) \in \langle E, R, E \rangle$ and $h \in \vartheta_2^{k-1}\}$, $k = 1, 2, \ldots, H$, where $\vartheta_1^0 = S_A$ and $\vartheta_2^0 = S_B$ (see line 4 to 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 hop reaches the maximum hop H, we stop extracting other triples from the complete KG and construct the relational adjacency matrix (see line 6 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}, e_9 in the path $A_3 \rightarrow e_8 \rightarrow e_{26} \rightarrow e_9 \rightarrow B_6$. 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 triples (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 Eq. 1) by taking two recommendation modes into consideration, as shown as in Eq. 2:

$$
\begin{aligned}
P(A_{i+1}|S, \langle E, R \rangle) &= \\
&P(M\_S_A|S, \langle E, R \rangle)P(A_{i+1}|M\_S_A, S, \langle E, R \rangle) + P(M\_G_A|S, \langle E, R \rangle)P(A_{i+1}|M\_G_A, S, \langle E, R \rangle) \\
P(B_{j+1}|S, \langle E, R \rangle) &= \\
&P(M\_S_B|S, \langle E, R \rangle)P(B_{j+1}|M\_S_B, S, \langle E, R \rangle) + P(M\_G_B|S, \langle E, R \rangle)P(B_{j+1}|M\_G_B, S, \langle E, R \rangle).
\end{aligned}
\tag{2}
$$

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. $P(M\_S_A|S, \langle E, R \rangle)$ and $P(M\_G_A|S, \langle E, R \rangle)$ represent the probabilities under the *sequence mode* and the *graph mode* in domain A, respectively, $P(A_{i+1}|M\_S_A, S, \langle E, R \rangle)$ and $P(A_{i+1}|M\_G_A, S, \langle E, R \rangle)$ 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.
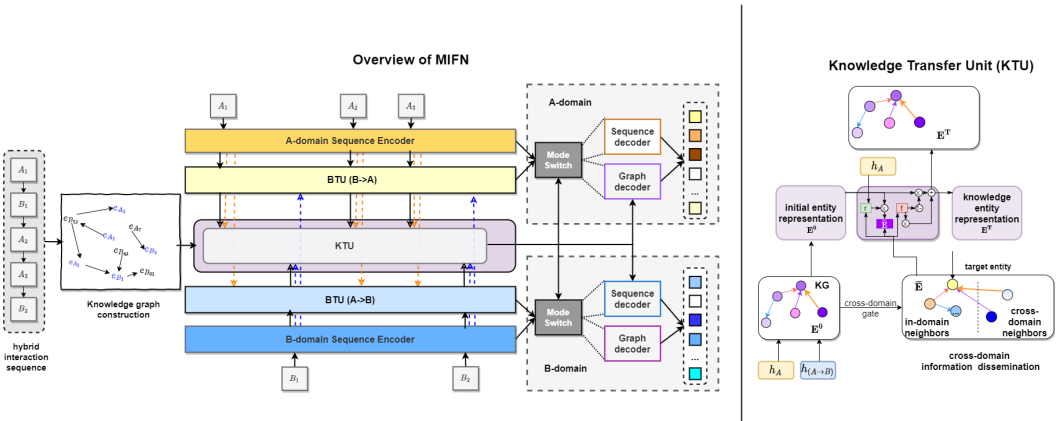
Fig. 2. An overview of MIFN. Section 3.3 contains a walkthrough of the model.

As shown in the left side of Figure 2, 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 to grasp useful knowledge from the KG and propagates it to both domains. The mixed recommendation 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 Eq. 10. The *sequence recommendation decoder* evaluates the probability of clicking items, corresponding to Eq. 11.

3.4 Sequence encoder

As with existing studies [21, 22], 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 $H_A = \{h_{A_1}, h_{A_2}, \ldots, h_{A_i}, \ldots, h_{A_n}\}$ for domain A, and $H_B =$

$\{h_{B_1}, h_{B_2}, \ldots, h_{B_j}, \ldots, h_{B_m}\}$ 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 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 [39]. For example, a user who has read the *Harry Potter* book (e.g., "*Harry Potter and the Philosopher's Stone*" or "*Harry Potter and the Chamber of Secrets*" and so on) has also watched the "*Pirates of the Caribbean*" movie 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 fantastic 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., $h_{A_i \to B}$, as follows:

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

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 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. $h_{A_i \to B}$ can be seen as a combination of $\widehat{h_{A_i}}$ and $h_{A_{i-1} \to B}$ 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.

$h_{A_i \to B}$ 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:

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

After that, we can obtain the transferred behavioral representation $h_{(A \to B)_i}$ 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 be not 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 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 KTU, as shown in the right side of the Figure 2.

For each hybrid interaction sequence S, we get the item representations $\{h_{A_1}, h_{A_2}, \ldots, h_{A_i}, \ldots\}$ from the sequence encoder (Section 3.4) and the transferred behavioral representations $\{h_{(A \to B)_1}, h_{(A \to B)_2}, \ldots, h_{(A \to B)_i}, \ldots\}$ 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 $h_{(A \to B)}$. We also obtain the relational adjacency matrix A of the KG, which consists of N entities and the corresponding relations (§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 Eq. 5:

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

The explanations for the main parts of Eq. 5 are as follows:

(i) **Gated functions**. r and f are the update gate function and the forget gate function, which aim to regulate 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**. $\widehat{h_{e_k}}$ 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 Eq. 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 $\widehat{h_{e_k}}$ 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 [28, 63, 64]. 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 ($0 \le l \le L$), 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 Eq. 6:

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

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; $N_c(k)$ is the neighbor entity set of entity e_k from the complementary domain. 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 $h_{(A \to B)}$; 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 $\widetilde{h_{e_k^0}} = h_{e_{B_k}}$:

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

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 1000 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 to balance 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 $h_{(A \to B)}$. These attention weights act as an information controller to identify entities of different importance in the KG, the definitions of which are shown in Eq. 8:

$$
\begin{aligned}
s_{A_k} &= \mathbf{v_1}^\top \tanh \left(W_{A_1} \cdot h_A + W_{A_2} \cdot h_{e_{A_k}^0} \right) \\
\alpha_{A_k} &= \frac{\exp\left(s_{A_k}\right)}{\sum_i \exp\left(s_{A_i}\right)} \\
s_{B_k} &= \mathbf{v_2}^\top \tanh \left(W_{B_1} \cdot h_{(A \to B)} + W_{B_2} \cdot h_{e_{B_k}^0} \right) \\
\beta_{B_k} &= \frac{\exp\left(s_{B_k}\right)}{\sum_j \exp\left(s_{B_j}\right)},
\end{aligned}
\tag{8}
$$

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 $h_{(A \to B)}$ 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 $h_{(A \to B)}$ and the sum of all transferred entity representation $\sum_k (h_{e_k^T})$ (where $h_{e_k^T} \in h_{E^T}$). Then, we employ a softmax regression to transform the total representation into the mode probability distributions, as follows:

$$
P(M\_G_B | S, \langle E, R \rangle), P(M\_S_B | S, \langle E, R \rangle) = \text{softmax} \left(W_m \cdot concat \left[h_B, h_{(A \to B)}, \sum_k (h_{e_k^T}) \right] + b_m \right), \tag{9}
$$

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:

$$P(B_{j+1}|M\_G_B, S, \langle E, R \rangle) = \begin{cases} 0 & \text{if } B_{j+1} \notin \mathbb{B} \\ \dfrac{\exp(h_{e_{B_{j+1}}^T})}{\sum_k^m \exp(h_{e_k^T})} & \text{if } B_{j+1} \in \mathbb{B}, \end{cases} \tag{10}$$

where $h_{e_{B_{j+1}}^T}$ is the transferred entity representation of $e_{B_{j+1}}$ (corresponding to $h_{e_k^T}$ in Eq. 5 when e_k is an item entity from domain B). m is the number of the item set $\mathbb{B}$. 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 $h_{(A \to B)}$ into the hybrid representation c_S, i.e., $c_S = \left[h_B, h_{(A \to B)} \right]^\top$. Then, the recommendation probability for each item $B_{j+1} \in \mathbb{B}$ is computed as follows:

$$P(B_{j+1}|M\_S_B, S, \langle E, R \rangle) = \begin{cases} 0 & \text{if } B_{j+1} \notin \mathbb{B} \\ \dfrac{\exp(W_{j+1}c_S + b_I)}{\sum_k^m \exp(W_k c_S + b_I)} & \text{if } B_{j+1} \in \mathbb{B}, \end{cases} \tag{11}$$

where $W_k \in W_I$ 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:

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

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

$$L_{R_A}(\theta) = -\frac{1}{|\mathbb{S}|} \sum_{S \in \mathbb{S}} \sum_{A_i \in S} \log P(A_{i+1}|S, \langle E, R \rangle)$$

$$L_{R_B}(\theta) = -\frac{1}{|\mathbb{S}|} \sum_{S \in \mathbb{S}} \sum_{B_j \in S} \log P(B_{j+1}|S, \langle E, R \rangle), \tag{13}$$

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 Eq. 2.

Additionally, MIFN incorporates a *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 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:

$$L_{M_A}(\theta) = -\frac{1}{|\mathbb{S}|} \sum_{S \in \mathbb{S}} \sum_{A_i \in S} [(1 - \mathbb{1}(A_{i+1} \in \mathbb{A})) \log P(M\_S_A | S, \langle E, R \rangle)]$$

$$L_{M_B}(\theta) = -\frac{1}{|\mathbb{S}|} \sum_{S \in \mathbb{S}} \sum_{B_j \in S} [(1 - \mathbb{1}(B_{j+1} \in \mathbb{B})) \log P(M\_S_B | S, \langle E, R \rangle)] \tag{14}$$

$$L_M(\theta) = L_{M_A}(\theta) + L_{M_B}(\theta),$$

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:

$$L(\theta) = L_R(\theta) + L_M(\theta). \tag{15}$$

4 EXPERIMENTAL SETUP

4.1 Research questions

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

(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.)

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. [39] 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

---

[3]https://www.amazon.com/

sequence contains at least 3 items from each domain. The statistics of the processed datasets are shown in Table 2.

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				

For knowledge graph construction, we use the Amazon product data to mine the relations of the items.[4] The data is 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 crawl the relation "Adapted_from" between books and movies from Wikipedia.[5] For example, ⟨movie, Adapted_from, book⟩ 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 shown in Table 3.

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

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 five categories: (1) traditional recommendation methods, (2) cross-domain recommendation methods, (3) sequential recommendation methods, (4) cross-domain sequential recommendation methods, and (5) 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 [20].

---

[4]https://jmcauley.ucsd.edu/data/amazon
[5]https://en.wikipedia.org/

- 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 [55].
- BPR-MF: This method follows the idea of MF with a pairwise ranking objective via stochastic gradient descent [52]. Following Hidasi et al. [21], we represent a new sequence by the average latent factors of items that appeared in the sequence so far.

4.3.2 Cross-domain recommendation methods. We use two 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 [39].
- Conet: This method transfers information between different domains by a cross-stitch network [45], where information in each domain is captured by neural CF model [20].

4.3.3 Sequential recommendation 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 [21].
- HRNN: This method combines the extra user's information into GRU networks and proposes a hierarchical RNN model based on GRU4REC [48].
- NARM: This method takes an attention mechanism into consideration to capture both sequential-level preferences and the user's main purpose [32].
- 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 [37].

4.3.4 Cross-domain Sequential recommendation methods.
- 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 [39].

4.3.5 Knowledge-aware recommendation methods.
- SRGNN: This method constructs each sequence as a directed graph, where the items in the sequence are entities and transition relationship between adjacent items represents edge which is considered as knowledge information. 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 [70].
- 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 [27].

4.4 Evaluation metrics

We target the top-K recommendations in this work, so we adopt two widely used ranking-based metrics [9, 19, 43, 49]: 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.

Table 4. Experimental results (%) on the Amazon ("Movie-Book") dataset. **Bold face** indicates the best result in terms of the corresponding metric. Significant improvements over the best baseline results are marked with † (t-test, $p < .05$).

| 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
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
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
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
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
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-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**†

- 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, the hop layer H is set to 2, and the number of entities N in KG is set to 200. 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 paper are available at https://github.com/mamuyang/MIFN.

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 shown in Table 4 and 5, respectively. From the results, we have the following main observations.

First, MIFN outperforms the single-domain SR methods (e.g., STAMP, NARM) and the knowledge-aware methods (e.g., SRGNN, 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"

Table 5. Experimental results (%) on the Amazon ("Food-Kitchen") dataset. **Bold face** indicates the best result in terms of the corresponding metric. Significant improvements over the best baseline results are marked with [†] (t-test, $p < .05$).

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
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
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
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
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
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-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**[†]

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 sequential 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 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 is less sparse in the "Book" domain compared to that in the "Movie" domain. And the results in the "Food" domain are better than the "Kitchen" domain on the "Food-Kitchen" dataset. Again, we think it is because the data sparsity difference as the users have more interactions in the "Food" domain than in the "Kitchen" 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.

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 than traditional methods and cross-domain methods. 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. Besides, it seems that STAMP obtains lower results than the sequential method NARM in the "Movie-Book" dataset, while it performs 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 knowledge aware 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 Table 4 and Table 5.

(1) MIFN is the best performing variant, which includes both the BTU and KTU modules, and 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+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, 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, 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 towards *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

Table 6. Analysis of the knowledge graph construction algorithm on the Amazon ("Movie-Book") dataset. The different ratios represent the different proportions of the predicted ground truth items appeared in the KG.

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

Table 7. Analysis of the knowledge graph construction algorithm on the Amazon ("Food-Kitchen") dataset. The different ratios represent the different proportions of the predicted ground truth items appeared in the KG.

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

add the ground truth items to the extracted entities according to the specified ratios in advance. The results are shown in Table 6 and Table 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.

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 3 shows recommendations when the extracted KG is relevant to the current user preference, and Figure 4 shows recommendations when the extracted KG is irrelevant. Figure 3(a) and 4(a) are recommendations from MIFN, while Figure 3(b) and 4(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 3, we can observe that when using the extracted KG (Figure 3(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 3(b)). Furthermore, it should be noted that the mode switch probabilities in Figure 3(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 "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 3(b), the recommendation is wrong because the model only relies on the flow of behavioral information to recommend items from the item set.

On the other hand, we also found that the flow of knowledge is not always helpful. As shown in Figure 4(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, 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.

6 CONCLUSION AND FUTURE WORK

In this paper, 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 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.

As to future work, MIFN can be enhanced from at least two directions. First, currently we extract KGs with Algorithm 1 currently, which we show can be made more effective by including as much relevant knowledge as possible. Hence, we will further study the knowledge graph construction

(a) Recommendation from MIFN.

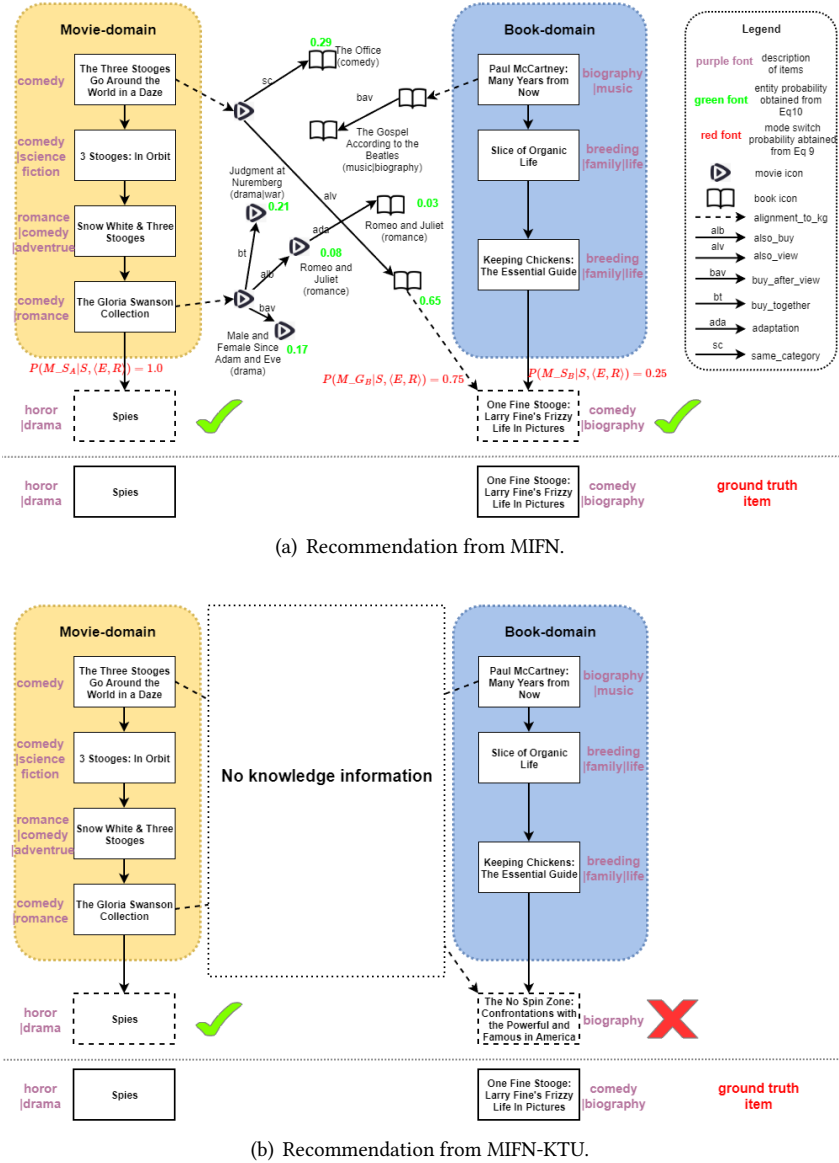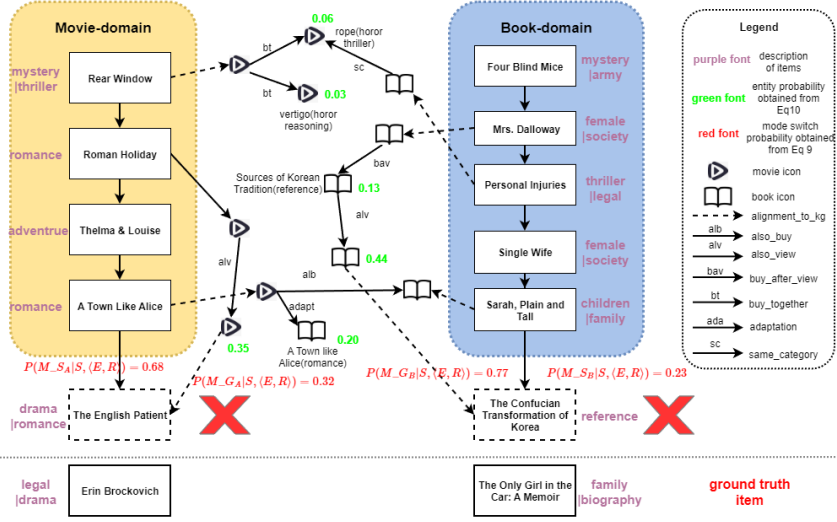(b) Recommendation from MIFN-KTU.

Fig. 3. Case study of MIFN and MIFN-KTU when the extracted KG is relevant to the current user preference.

algorithm 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.

DATA AND CODE

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

(a) Recommendation from MIFN.
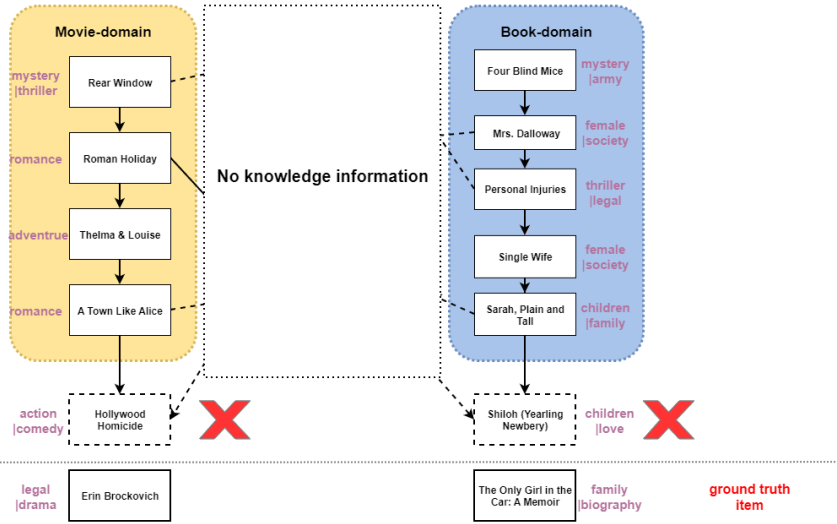
(b) Recommendation from MIFN-KTU.

Fig. 4. Case study of MIFN and MIFN-KTU when the extracted KG is irrelevant to the current user preference.

ACKNOWLEDGMENTS

the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

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. *UMUAI 2013* 23 (2013), 169–209.

[2] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms 2018* 11 (2018), 137.

[3] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2007. Distributed collaborative filtering with domain specialization. In *RecSys 2007*. 33–40.

[4] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2008. Mediation of user models for enhanced personalization in recommender systems. *UMUAI 2008* 18 (2008), 245–286.

[5] 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 *SIGIR 2020*. 1661–1664.

[6] Veronika Bogina and Tsvi Kuflik. 2017. Incorporating dwell time in session-based recommendations with recurrent Neural networks. In *RecSys 2017*. 57–59.

[7] Shuo Chen, Josh L Moore, Douglas Turnbull, and Thorsten Joachims. 2012. Playlist prediction via metric embedding. In *SIGKDD 2012*. 714–722.

[8] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *WSDM 2018*. 108–116.

[9] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan S. Kankanhalli. 2018. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *WWW 2018*. 639–648.

[10] Paolo Cremonesi, Antonio Tripodi, and Roberto Turrin. 2011. Cross-domain recommender systems. In *ICDMW 2011*. 496–503.

[11] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential user-based recurrent neural network recommendations. In *RecSys 2017*. 152–160.

[12] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW 2015*. 278–288.

[13] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *CERI 2012*. 24.

[14] Ignacio Fernández-Tobías, Iván Cantador, and Laura Plaza. 2013. An emotion dimensional model based on social tags: Crossing folksonomies and enhancing recommendations. In *EC-Web 2013*. 88–100.

[15] 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. *UMUAI 2019* 29 (2019), 443–486.

[16] 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 *WWW 2019*. 491–502.

[17] Sharon Givon and Victor Lavrenko. 2009. Predicting social-tags for cold start book recommendations. In *RecSys 2009*. 333–336.

[18] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *ICDM 2016*. 191–200.

[19] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *SIGIR 2018*. 355–364.

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

[21] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *ICLR 2016*. –.

[22] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *RecSys 2016*. 241–248.

[23] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative cross networks for cross-domain recommendation. In *CIKM 2018*. 667–676.

[24] 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. *TKDD 2018* 12 (2018), 1–30.

[25] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. 2013. Personalized recommendation via cross-domain triadic factorization. In *WWW 2013*. 595–606.

[26] 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 *WSDM 2019*. 573–581.

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

[28] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR 2017*.

[29] 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 *SIGIR 2020*. 1081–1090.

[30] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. In *IJCAI 2009*. 2052–2057.

[31] Chenliang Li, Xichuan Niu, Xiangyang Luo, Zhenzhong Chen, and Cong Quan. 2019. A review-driven neural model for sequential recommendation. In *IJCAI 2019*. 2866–2872.

[32] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *CIKM 2017*. 1419–1428.

[33] Pan Li and Alexander Tuzhilin. 2019. DDTCDR: Deep dual transfer cross domain recommendation. In *WSDM 2020*. 331–339.

[34] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2017. CCCFNet: A content-boosted collaborative filtering neural network for cross domain recommender systems. In *WWW 2017*. 817–818.

[35] 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. *TKDE 2019* 32, 8 (2019), 1502–1516.

[36] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. 2016. Context-aware sequential recommendation. In *ICDM 2016*. 1053–1058.

[37] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-term attention/memory priority model for session-based recommendation. In *SIGKDD 2018*. 1831–1839.

[38] Babak Loni, Yue Shi, Martha A. Larson, and Alan Hanjalic. 2014. Cross-domain collaborative filtering with factorization machines. In *ECIR 2014*. 656–661.

[39] 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 *SIGIR 2019*. 685–694.

[40] Renfeng Ma, Qi Zhang, Jiawen Wang, Lizhen Cui, and Xuanjing Huang. 2018. Mention recommendation for multimodal microblog with cross-attention memory network. In *SIGIR 2018*. 195–204.

[41] 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 *WWW 2019*. 1210–1221.

[42] 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 *IJCAI 2018*. 3484–3490.

[43] Lei Mei, Pengjie Ren, Zhumin Chen, Liqiang Nie, Jun Ma, and Jian-Yun Nie. 2018. An attentive interaction network for context-aware recommendations. In *CIKM 2018*. 157–166.

[44] Nima Mirbakhsh and Charles X Ling. 2015. Improving top-n recommendation for cold-start users via cross-domain information. *TKDD 2015* 9 (2015), 1–19.

[45] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *CVPR 2016*. 3994–4003.

[46] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer learning in collaborative filtering for sparsity reduction. In *AAAI 2010*. 230–235.

[47] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *CSUR 2018* 51 (2018), 66.

[48] Massimo Quadrana, Alexandros Karatzoglou, Balzs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *RecSys 2017*. 130–137.

[49] 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 *AAAI 2019*. 4806–4813.

[50] 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 *WSDM 2020*. 907–908.

[51] 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 *SIGIR 2020*. 89–98.

[52] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI 2009*. 452–461.

[53] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW 2010*. 811–820.

[54] 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 *UMAP 2013*. 289–295.

[55] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW 2001*. 285–295.

[56] Guy Shani, David Heckerman, and Ronen I Brafman. 2005. An MDP-based recommender system. *J Mach Learn Res 2005* 6 (2005), 1265–1295.

[57] Bracha Shapira, Lior Rokach, and Shirley Freilikhman. 2013. Facebook single and cross domain data for recommendation systems. *UMUAI 2013* 23 (2013), 211–247.

[58] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *WSDM 2019*. 555–563.

[59] 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 *CIKM 2019*. 1441–1450.

[60] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *RecSys 2016*. 17–22.

[61] Amit Tiroshi, Shlomo Berkovsky, Mohamed Ali Kâafar, Terence Chen, and Tsvi Kuflik. 2013. Cross social networks interests predictions based on graph features. In *RecSys 2013*. 319–322.

[62] Amit Tiroshi and Tsvi Kuflik. 2012. Domain ranking for cross domain collaborative filtering. In *UMAP 2012*. 328–333.

[63] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *CIKM 2018*. 417–426.

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

[65] 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 *SIGIR 2019*. 345–354.

[66] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for nextbasket recommendation. In *SIGIR 2015*. 403–412.

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

[68] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item silk road: Recommending items from information domains to social users. In *SIGIR 2017*. 185–194.

[69] 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 *WSDM 2019*. 717–725.

[70] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *AAAI 2019*. 346–353.

[71] Xiang Wu, Qi Liu, Enhong Chen, Liang He, Jingsong Lv, Can Cao, and Guoping Hu. 2013. Personalized next-song recommendation in online karaokes. In *RecSys 2013*. 137–140.

[72] Yikun Xian, Zuohui Fu, S Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *SIGIR 2019*. 285–294.

[73] Ghim-Eng Yap, Xiao-Li Li, and S Yu Philip. 2012. Effective next-items recommendation via personalized sequential pattern mining. In *DASFAA 2012*. 48–64.

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

[75] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level deeper self-attention network for sequential recommendation. In *IJCAI 2019*. 4320–4326.

[76] 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 *SIGIR 2020*. 229–238.

[77] 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. *TKDD 2016* 10 (2016), 1–23.

[78] Lin Zheng, Naicheng Guo, Weihao Chen, Jin Yu, and Dazhi Jiang. 2020. Sentiment-guided sequential recommendation. In *SIGIR 2020*. 1957–1960.

[79] Fuzhen Zhuang, Yingmin Zhou, Fuzheng Zhang, Xiang Ao, Xing Xie, and Qing He. 2018. Cross-domain novelty seeking trait mining for sequential recommendation. *arXiv preprint arXiv:1803.01542* (2018).

[80] Andrew Zimdars, David Maxwell Chickering, and Christopher Meek. 2001. Using temporal data for making recommendations. In *UAI 2001*. 580–588.