Information Sciences 669 (2024) 120550

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Transfer learning in cross-domain sequential recommendation

Zitao Xu [a], Weike Pan [a,*], Zhong Ming [a,b,c]

[a] College of Computer Science and Software Engineering, Shenzhen University, China
[b] Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), China
[c] College of Big Data and Internet, Shenzhen Technology University, China

ARTICLE INFO

Keywords:
Cross-domain recommendation
Sequential recommendation
Attentive learning
Transfer learning

ABSTRACT

Sequential recommendation captures users' dynamic preferences by modeling the sequential information of their behaviors. However, most existing works only focus on users' behavior sequences in a single domain, and when there is insufficient data in the target domain, the recommendation performance may not be satisfactory. We notice that a user's interests are usually diverse, for which the items he/her interacts with in a period of time may be from multiple domains. Moreover, there are also item transition patterns across sequences from different domains, which means that a user's interaction in one domain may affect his/her interaction in the other domains next time. In this paper, we aim to improve the performance of sequential recommendation in the target domain by introducing users' behavior sequences from multiple source domains, and propose a novel solution named transfer via joint attentive preference learning (TJAPL). Specifically, we tackle the studied problem from the perspective of transfer learning and attentive preference learning (APL). For target-domain APL, we adopt the self-attention mechanism to capture the users' dynamic preferences in the target domain. Furthermore, to address the scarcity challenge posed by limited target-domain data, we introduce users' behavioral sequences in the source domain, and devise cross-domain user APL to transfer and share the users' overall preferences from multiple source domains to the target domain. We also design cross-domain local APL that specializes in capturing the item transition patterns across different domains for knowledge transfer. These modules are all based on the attention mechanism and thus can accelerate the training by parallel computation. Notice that our TJAPL can be applied to scenarios with multiple source domains, while transferring knowledge from multiple domains is potentially helpful in practical applications. Extensive empirical studies indicate that our TJAPL significantly outperforms thirteen recent and competitive baselines.

1. Introduction

Recommender systems have been recognized in playing an irreplaceable role in matching user needs with rich resources and helping users alleviate the information overload issue. Traditional recommendation systems typically use collaborative filtering (CF) methods [1,2] to model user preferences in a static way. However, these methods ignore the order in users' behavior sequences, and since both users' preferences and items' popularity are dynamically changing over time, traditional methods can not be well suitable for certain scenarios.

* Corresponding author.
E-mail addresses: xuzitao2018@email.szu.edu.cn (Z. Xu), panweike@szu.edu.cn (W. Pan), mingz@szu.edu.cn (Z. Ming).

https://doi.org/10.1016/j.ins.2024.120550
Received 27 November 2022; Received in revised form 23 March 2024; Accepted 3 April 2024

Available online 10 April 2024
0020-0255/

Fig. 1. An example of a user's behavior sequences in the book domain (the top half) and the movie domain (the bottom half).

To solve this problem, sequential recommendation has been proposed to predict the next likely preferred item based on user's behavior sequence. It considers a user's interactions as a dynamic sequence and mainly focuses on capturing the transition patterns within the sequence. Markov chains (MCs) have been widely adopted to capture the sequential pattern among successive items on early sequential recommendation studies [3,4], but such methods may fail to learn more complex item transition patterns for long behavior sequences. With the advancement of deep learning such as neural networks, an increasing number of studies attempt to employ powerful neural models to capture the sequential information. GRU4Rec [5] is a very first adaption of recurrent neural networks (RNNs) which models users' behavior sequences step by step. Caser [6] utilizes convolutional neural networks (CNNs) to capture sequential patterns by sliding filters. Recently, attention mechanism [7] have shown its effectiveness to capture the long-term dependencies [8,9]. It's believed that sequential recommendation has become increasingly important because of its high accuracy and practicability. However, most existing studies only focus on users' behavior sequences in one single domain and suffer from the data scarcity issue commonly existed in recommender systems.

Cross-domain recommendation (i.e., recommendation across one target domain and one or more source domains) has been proposed to address the data scarsity challenge by introducing the relatively richer information from the source domain(s) [10]. Some recent works focus on sharing and transferring knowledge across different domains. For example, EMCDR [11] transfers user's preferences between different domains by learning a mapping function. CoNet [12] transfers knowledge between different domains through a collaborative cross-network. However, most existing cross-domain recommendation methods don't take into account users' sequential information, which are thus unable to model users' dynamic preferences from the behavior sequences.

Cross-domain sequential recommendation is a new and emerging problem. In this work, we aim to combine the sequential information with some other domain data, which not only captures the dynamic preferences from users' behavior sequences, but also effectively alleviates the data scarsity issue that usually occurs in a single domain. Moreover, we notice that the items a user interacts with may come from multiple domains in a period of time, due to the diversity of users' preferences. There are also item transition patterns across sequences from different domains, which implies that a user's interaction in one domain may affect his/her next interaction in another domain.

Scenarios of cross-domain sequential recommendation are common in real-world situations. For example, from the top part of Fig. 1, we can see that the user has recently read three "Harry Potter" (the books), and following the idea of sequential recommendation, we will recommend the sequel of "Harry Potter" (the book) for him/her, which obviously has a higher probability of satisfying his/her needs. From the bottom half of Fig. 1, we can observe that in the movie domain, the user recently tends to watch fantasy movies, but due to the uncertainty of the user's intention, there is no strong connection between these movies. If we account for the user's recent behaviors in the book domain, we can find that he/she is interested in "Harry Potter" (the books), so it may be a good suggestion to recommend "Harry Potter" (the movie) for him/her. It can be seen that if both the sequential information and the cross-domain behaviors are considered, the recommendation performance will be effectively improved.

Cross-domain sequential recommendation faces two primary challenges: i) how to address the data scarcity problem on users' behavior sequences in the target domain, and ii) how to extract the correlation between users' preferences in the target domain and source domains. There are relatively limited works on cross-domain sequential recommendation, and most methods rely on RNNs [13,14] to model users' behavior sequences, which have limited capability in capturing the complex associations between domains and are challenging to parallelize. Moreover, most existing methods capture the users' preferences within one single domain, neglecting the item transition patterns across sequences from different domains, i.e., a user's interaction in one domain may influence his/her next interaction in other domains. Furthermore, in real-world recommender systems, users often interact in more than two domains. Transferring knowledge from multiple domains is potentially helpful in capturing a user's preferences more comprehensively. However, existing methods often focus on the associations of one single source domain to a certain target domain.

As a response, we propose a novel solution named transfer via joint attentive preference learning (TJAPL). Specifically, we tackle the studied problem from the perspective of transfer learning and attentive preference learning (APL). Our TJAPL contains target-domain APL (TD-APL) and cross-domain APL, where the latter is further divided into cross-domain user APL (CD-UAPL) and cross-domain local APL (CD-LAPL). In particular, we treat the self-attention sequential recommendation (SASRec) model [8] as TD-APL to model the users' behavior sequences and capture their dynamic preferences in the target domain. Furthermore, considering that users may have similar interests in multiple domains in a period of time, we propose CD-UAPL to share and transfer the users' overall preferences from more than one source domain to the target domain, leveraging the sequential behaviors from the source domains to address the scarcity problem. We also propose CD-LAPL to capture the item transition patterns across sequences from different domains and generate the users' cross-domain local attentive preferences. Notice that these modules are all based on attention mechanism thus can accelerate the training by parallel computation. Moreover, it can be applied to scenarios with more than one source domain, which is also demonstrated to be effective in the experiments.

The main technical contributions of this work can be briefly summarized as follows:

- We study a new and important problem, i.e., cross-domain sequential recommendation, and propose a novel method named transfer via joint attentive preference learning (TJAPL), which addresses the challenges well by transferring knowledge from more than one source domain to a target domain.
- We design cross-domain user attentive preference learning (CD-UAPL) to deal with the data scarcity problem by leveraging the users' overall preferences from the source domains, and design cross-domain local APL (CD-LAPL) to extract the transition patterns across different domains.
- We conduct extensive empirical studies on three cross-domain datasets, where the results show that our TJAPL significantly outperforms in all cases. We also conduct ablation studies to explore the contribution of various components of our TJAPL.

2. Related work

In this section, we briefly describe the related works from four categories: (i) general recommendation, (ii) cross-domain general recommendation, (iii) sequential recommendation, and (iv) cross-domain sequential recommendation.

2.1. General recommendation

In early works, users' preferences are commonly modeled by using CF methods. Matrix factorization (MF) based methods [1,2,15] is one main branch of CF methods. Specifically, it projects users and items into a shared vector space and then predicts a user's rating on an item using the inner product of the two corresponding vectors. Neighborhood-based methods are another line of work [16,17], which make recommendations according to (item, item) or (user, user) similarities. Recently, deep learning (DL) based models [18–21] have been adopted to improve the recommendation performance. For instance, NCF [19] adopts multi-layer perceptron (MLP) to capture user preferences while CDAE [20] and CVAE [21] use an autoencoder (AE) to predict users' ratings. Although general recommendation methods have proven to be effective, their recommendation performance depends on whether the training data is sufficient. Without a large amount of users' behavior data, how to recommend new items to the right users and how to make personalized recommendations for new users are the problems that recommendation systems need to face in real-world scenarios.

2.2. Cross-domain general recommendation

To alleviate the data scarcity issue in a typical domain, cross-domain recommendation was proposed. In cross-domain recommendation, the most important concern is determining what knowledge to transfer between domains and how to transfer the knowledge. A representative branch of transferring knowledge is the mapping-based methods [22,23,11], which model the connection between two domains by explicitly learning a mapping function. For example, EMCDR [11] learns users' preferences in different domains separately, and then transfers the overlapped users' preferences across domains by a mapping function. DDTCDR [23] captures users' preferences and preserves the relations between users across different latent spaces by introducing a latent orthogonal mapping. SSCDR [22] proposes a semi-supervised strategy which can learn mapping functions using some non-overlapping data. Another approach to transferring knowledge is based on multi-domain collaborative training [24,12,25,26]. For example, CMF [24] factorizes matrices from multiple domains simultaneously and enables knowledge transfer by sharing the users' latent factors. CoNet [12] transfers knowledge between different domains by developing a collaborative cross-network.

2.3. Sequential recommendation

Unfortunately, the methods mentioned in Section 2.1 and Section 2.2 are not suitable for sequential recommendation because they ignore the order of users' interactions. MCs [3] which can extract the sequential patterns among successive items are widely employed in the early work of sequential recommendation. For instance, factorized personalized MCs (FPMC) [4] combine MCs and MF [2] to capture short-term preference and long-term preference, respectively. [27,28] adopt high-order MCs to model more historical interactions. Recently, RNNs have been introduced to sequential recommendation since their natural instincts to handle sequential data [5,29–31]. GRU4Rec [5] is one of the earliest RNN-based methods for sequential recommendation, which models users' behavior sequences by employing gated recurrent units (GRUs). GRU4Rec+ [29] adopts an additional sampling strategy and designs a new loss function to improve GRU4Rec. Caser [6] proposes a CNN-based method that learns the patterns in sequences by

Table 1
A summary of some related problems, i.e., general recommendation, cross-domain general recommendation, sequential recommendation and cross-domain sequential recommendation.

	Single-Domain	Cross-Domain
Non-Sequential	FISM [15], etc. (General recommendation, many)	CoNet [12], etc. (Cross-domain general recommendation, many)
Sequential	SASRec [8], etc. (Sequential recommendation, many)	π-Net [13], DA-GCN [38] **(Cross-domain sequential recommendation, few)**

two types of convolutional filters, i.e., horizontal ones and vertical ones. Moreover, multi-head attention mechanism [7] are employed to sequential recommendation which can avoid the vanishing gradient problem commonly existed in RNN-based models [32,8,9]. SASRec [8] applies self-attention blocks to extract the long-range dependencies across sequences. BERT4Rec [9] designs a Cloze task and adopts bidirectional attention networks to model users' behavior sequences. More recently, graph neural networks (GNNs) [33, 34] have been adopted to extract the structural information and transition patterns. For example, SRGNN [33] separates sequences into graph-structured data and then uses GNNs to capture complex sequential dependencies. There are also some works [32,35,36] focusing on capturing a user's long-term preference or global representation to generate the user's general interests. Although great progress has been made in these studies, none of them has considered introducing some source-domain data or transferring knowledge under cross-domain situations, and suffer from the same data sparsity issue mentioned in general recommendation.

2.4. Cross-domain sequential recommendation

Recently, for cross-domain sequential recommendation, π-Net [13] and its improved version PSJNet [14] have been proposed in a shared-account scenario. Specifically, π-Net devises a cross-domain transfer unit to capture and transfer knowledge across different domains at each timestamp. PSJNet [14] proposes a model framework which splits role-specific representations from the mixed users' behavior at each step, and joins the representations to obtain cross-domain representations. MIFN [37] employs GRUs to encode users' behavior sequences in each domain and integrates knowledge graphs to improve knowledge transfer across domains. However, these RNN-based models may not be sufficiently expressive to extract the complex associations between domains and make the models less effective. To address the above challenges, DA-GCN [38] employs GNN to model the complicated interaction relationships, as well as the explicit structural information. More recently, CD-SASRec [39] proposes an improved method based on SASRec [8] that fuses the source-domain aggregated vector into the item embedding in the target domain, in order to transfer information across domains. RecGURU [40] unifies user embeddings from different domains via an adversarial learning approach and generates a single global representation, which captures a user's overall preferences. However, these models neglect to explore the item transition patterns across different domains. Moreover, most of the studies consider using only one source domain to improve a target-domain performance, and are not readily applicable to multiple domains. We make a summary of the above four problems, i.e., general recommendation, cross-domain general recommendation, sequential recommendation and cross-domain sequential recommendation, in Table 1.

In this work, we base our target-domain attentive preference learning module on SASRec [8], which has been shown as an effective and efficient model in various previous works on sequential recommendation [35,36,41]. We aim to improve SASRec by leveraging rich source-domain data to alleviate the data sparsity problem, and transfer a user's overall preferences and sequential information from multiple source domains (rather than from one single domain) to a target domain.

3. Proposed method

In this section, we first formalize the studied task, i.e., cross-domain sequential recommendation. Then, we introduce the detailed components of our model, i.e., transfer via joint attentive preference learning (TJAPL). As shown in Fig. 2, our TJAPL mainly consists of a target-domain attentive preference learning module (TD-APL), a cross-domain user attentive preference learning module (CD-UAPL), and a cross-domain local attentive preference learning module (CD-LAPL). For ease of reading and understanding, we summarize the key notations and their explanations in Table 2.

3.1. Problem definition

For cross-domain sequential recommendation, we have the set of users $\mathcal{U}$, and we denote the set of items in the target domain as $\mathcal{I}$. Moreover, we have N source domains with same users and different item sets $\mathcal{I}^{S_n}$, $1 \leqslant n \leqslant N$. We define the target-domain behavior sequence of each user $u \in \mathcal{U}$ as $\mathcal{V} = \{v_1, v_2, \dots, v_L\}$ (ordered by the interaction time), which consists of L items from $\mathcal{I}$. And we will repeatedly append a padding item at the beginning of the sequence if the sequence length is shorter than L. Moreover, $\mathcal{V}_t = \{v_1, v_2, \dots, v_t\}$, $1 \leqslant t \leqslant L$ denotes a truncated behavior sequence at time step t with regard to sequence $\mathcal{V}$. Specifically, for the n-th source domain, we denote a truncated item sequence as $\mathcal{V}_{t'}^{S_n} = \{v_1^{S_n}, v_2^{S_n}, \dots, v_{t'}^{S_n}\}$, where t' is the most recent time step at which the user interacted with an item in the n-th source domain before the real moment corresponding to the time step t in the target domain. This is to ensure the causality of the user behaviors from the n-th source domain to the target domain. Cross-domain sequential recommendation aims to predict the next likely to be preferred item in the target domain (i.e., v_{t+1}) according to $\mathcal{V}_t$ and $\mathcal{V}_{t'}^{S_n}$ where $1 \leqslant n \leqslant N$. The left part of Fig. 2 is the input sequence of the target domain, and the right part is that of the n-th source domain.

Table 2
Important notations and their explanations.

Symbol	Explanation
$\mathcal{U}$	user set
$\mathcal{I}$	item set for the target domain
N	number of source domains
$\mathcal{I}^{S_n}$	S_n represents the n-th source domain; $\mathcal{I}^{S_n}$ is the item set for the n-th source domain
u	user $u \in \mathcal{U}$
v_i	the item that user interacted with at time step i in the target domain
$v_j^{S_n}$	the item that user interacted with at time step j in the n-th source domain
L	maximum sequence length
B	number of attention blocks
$\mathcal{V} = \{v_1, v_2, \ldots, v_L\}$	user's interaction sequence in the target domain
$\mathcal{V}_t = \{v_1, v_2, \ldots, v_t\}$	truncated item sequence at time step t with regard to sequence $\mathcal{V}$
$\mathcal{V}_{t'}^{S_n} = \{v_1^{S_n}, v_2^{S_n}, \ldots, v_{t'}^{S_n}\}$	truncated item sequence at time step t' for the n-th source domain
d	latent vector dimensionality
$\boldsymbol{u}, \boldsymbol{V}, \boldsymbol{V}^{S_n}$	embedding associate with u, $\mathcal{V}_t$, $\mathcal{V}_{t'}^{S_n}$
$\boldsymbol{p}_t$	position embedding at time step t
$\boldsymbol{f}_t$	target-domain attentive preference at time step t
$\boldsymbol{f}_t^u$	cross-domain user attentive preference at time step t
$\boldsymbol{f}_t^{S_n}$	cross-domain local attentive preference at time step t
$\boldsymbol{o}_t$	final representation of the user's preference at time step t
$r_{t,i}$	preference score of item i at time step t

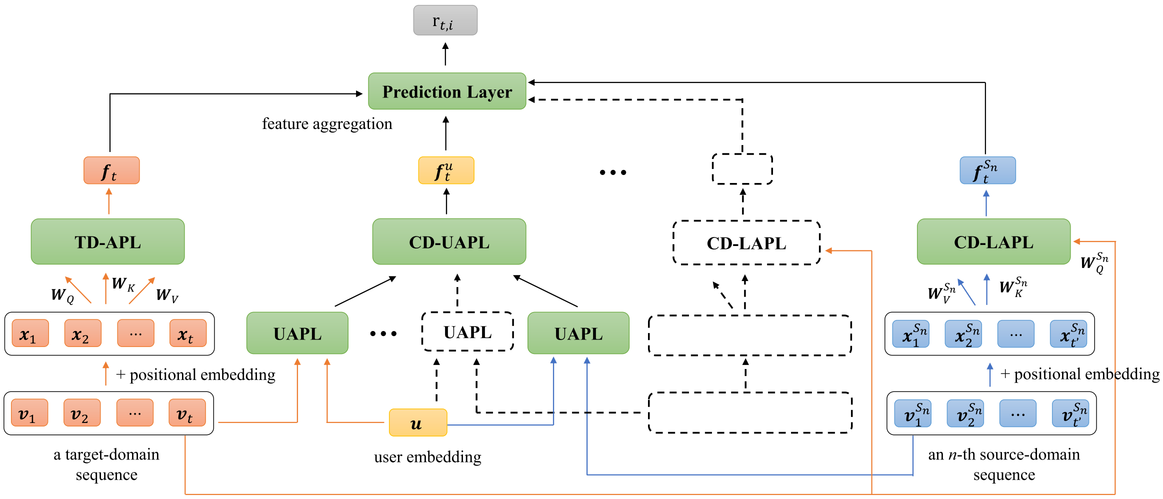


Fig. 2. The framework of our proposed TJAPL (transfer via joint attentive preference learning). TD-APL (target-domain APL) is fed with the embedding of a target domain sequence, which contains some self-attention blocks (see Eqs. (3)~(7)). CD-UAPL (cross-domain user APL) extracts a user's overall preference in all domains, where each domain includes a user attention layer (see Eqs. (8)~(11))) to capture the user preferences in the corresponding domain. CD-LAPL (cross-domain local APL) is fed with the embedding of a target-domain sequence and a source-domain sequence which consists of cross-domain attention blocks (see Eqs. (14)~(16)). Notice that each source domain contains its own CD-LAPL.

3.2. Target-domain attentive preference learning

We first denote $\boldsymbol{u}$ as the embedding vector of the user u in $\mathcal{U}$, where $\boldsymbol{u} \in \mathbb{R}^d$ is a learnable vector. Similarly, we denote $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_t\}$ as the embedding of the target-domain sequence $\mathcal{V}_t$ and $\boldsymbol{V}^{S_n} = \{\boldsymbol{v}_1^{S_n}, \boldsymbol{v}_2^{S_n}, \ldots, \boldsymbol{v}_{t'}^{S_n}\}$ as the embedding of the n-th source-domain sequence $\mathcal{V}_{t'}^{S_n}$.

We employ attention mechanism [7,8] to explore the sequential patterns in the target domain. Since the self-attention model can't consider the positions of the previous items, a learnable position embedding $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_L\} \in \mathbb{R}^{L\times d}$ should be added to the sequence embedding $\boldsymbol{V}$ and $\boldsymbol{V}^{S_n}$, then we obtain the position-aware input embedding $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t\}$ and $\boldsymbol{X}^{S_n} = \{\boldsymbol{x}_1^{S_n}, \boldsymbol{x}_2^{S_n}, \ldots, \boldsymbol{x}_{t'}^{S_n}\}$,

$$\boldsymbol{x}_i = \boldsymbol{v}_i + \boldsymbol{p}_i, \tag{1}$$

$$\boldsymbol{x}_i^{S_n} = \boldsymbol{v}_i^{S_n} + \boldsymbol{p}_i. \tag{2}$$

Next, we feed the sequence $\boldsymbol{X}$ into some stacked self-attention blocks (SABs). Omitting the residual connection layers and the normalization layers, each SAB is regarded as a self-attention layer $SAL(\cdot)$ followed by a feed-forward network $FFN(\cdot)$. Specifically, $SAL(\boldsymbol{X})$ can be formalized as:

$$\alpha_i = softmax\left(\boldsymbol{x}_t\boldsymbol{W}_Q\left(\boldsymbol{x}_i\boldsymbol{W}_K\right)^T\right), \forall i \in \{1,2,\ldots,t\}, \tag{3}$$

$$\boldsymbol{h}_t = \sum_{i=1}^{t}\alpha_i\left(\boldsymbol{x}_i\boldsymbol{W}_V\right), \tag{4}$$

where $\boldsymbol{x}_t\boldsymbol{W}_Q$, $\boldsymbol{x}_i\boldsymbol{W}_K$, and $\boldsymbol{x}_i\boldsymbol{W}_V$ stand for Query, Key, and Value, respectively. $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, $\boldsymbol{W}_V \in \mathbb{R}^{d\times d}$ are learnable parameters that improve the flexibility of the model. More clearly, the importance of Value is measured by using Query to match against Key. In this case, it refers to using the item which was interacted with at the last time step to match those items a user interacted with before, then obtain the item weighting information to generate the information used for prediction at the next time step, i.e., $\boldsymbol{h}_t \in \mathbb{R}^d$.

Then, we employ a two-layer $FFN(\boldsymbol{h}_t)$ to enable the model to explore the nonlinear features:

$$\boldsymbol{f}_t = ReLU\left(\boldsymbol{h}_t\boldsymbol{W}^{(1)} + \boldsymbol{b}^{(1)}\right)\boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)}, \tag{5}$$

where $\boldsymbol{W}^{(1)}, \boldsymbol{W}^{(2)} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)} \in \mathbb{R}^d$ are learnable parameters for the two-layer FFN. We utilize the same dropout and normalization layers as in [7] in this module.

Stacking the SAB is usually helpful for the model to extract the more complex sequential patterns. We denote the b-th ($b > 1$) SAB as:

$$\boldsymbol{h}_t^{(b)} = SAL(\boldsymbol{f}_t^{(b-1)}), \tag{6}$$

$$\boldsymbol{f}_t^{(b)} = FFN(\boldsymbol{h}_t^{(b)}). \tag{7}$$

Finally, we take the final output vector $\boldsymbol{f}_t^{(b)} \in \mathbb{R}^d$ from the top SAB as the target-domain attentive preference, which represents the current interests of the user at time step t in the target domain. In the remainder of this paper, we use $\boldsymbol{f}_t$ to denote $\boldsymbol{f}_t^{(b)}$ for simplicity. Notice that in contrast to RNNs, the computation of self-attention mechanism can be effectively parallelized.

3.3. Cross-domain user attentive preference learning

Although TD-APL can capture the dynamic preference from the target-domain sequence, due to the property of the self-attention mechanism, it will rely on the last interaction in a sequence to generate the relevant output. This makes the model overly focused on the short-term preferences of users, while capturing the user's overall preference is beneficial for making personalized and diverse recommendations. In addition, so far, we have focused only on the target-domain sequential information of users, and how to make use of the source-domain sequences is also one of the issues to be considered.

Inspired by the existing works on single-domain sequential recommendation that devotes to identifying the long-term preferences of users to generate the user's general interests. [32,35,15], we propose a novel CD-UAPL module on cross-domain sequential recommendation.

According to the attention mechanism introduced in Section 3.2, an effective approach is to take the learnable vector $\boldsymbol{u} \in \mathbb{R}^d$ (i.e., the embedding of user u) as the query in the attention layer, which means that the query is the same for the user u regardless of which time step t the current interaction is at. This is also beneficial for personalized recommendation, because each user has his/her own embedding vector. The target-domain user attentive preference can then be formalized as follows:

$$\beta_i = softmax\left(\boldsymbol{u}\boldsymbol{W}_{Q_u}\left(\boldsymbol{v}_i\boldsymbol{W}_{K_u}\right)^T\right), \forall i \in \{1,2,\ldots,t\}, \tag{8}$$

$$\boldsymbol{z}_t = \sum_{i=1}^{t}\beta_i\left(\boldsymbol{v}_i\boldsymbol{W}_{V_u}\right), \tag{9}$$

where $\boldsymbol{v}_i$ is the initial embedding of item i, $\boldsymbol{W}_{Q_u}$, $\boldsymbol{W}_{K_u}$, $\boldsymbol{W}_{V_u} \in \mathbb{R}^{d\times d}$ are the learnable parameters similar to $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, $\boldsymbol{W}_V$ in Eq. (3), and $\boldsymbol{u}\boldsymbol{W}_{Q_u}$ denotes the Query. $\boldsymbol{z}_t$ is the target-domain user attentive preference, which stands for the overall preference of user u up to time step t in the target domain.

Notice that we abandon the position information $\boldsymbol{P}$ which is also the difference between Eq. (8) and Eq. (3) in the attention layer besides the query condition. This is because the long-term preference is not sensitive to the position information of the interactions compared to the short-term dynamic preference [36].

Considering that a same user usually has similar preferences beneath his or her behaviors in different domains, e.g., in the book domain, the user prefers to read fantasy novels, and to watch fantasy movies in the movie domain (as shown in Fig. 1), then the items of the fantasy genre are often more attractive to him. Even though the types of items are different, they reflect the same user preference.

In addition, when the user's behavior in the target domain is highly sparse, using only his or her target-domain interactions may not capture the user's overall preference well. However, if we can additionally generate the user's overall preference from some dense source domains with more interactions, the recommendation performance will be improved. Hence, we formalize the user attentive preference in the n-th source domain as follows:

$$\beta_i^{S_n} = softmax\left(\boldsymbol{u}\boldsymbol{W}_{Q_u}^{S_n}\left(\boldsymbol{v}_i^{S_n}\boldsymbol{W}_{K_u}^{S_n}\right)^T\right), \forall i \in \{1,2,\dots,t'\}, \tag{10}$$

$$\boldsymbol{z}_t^{S_n} = \sum_{i=1}^{t'} \beta_i^{S_n}\left(\boldsymbol{v}_i^{S_n}\boldsymbol{W}_{V_u}^{S_n}\right), \tag{11}$$

where $\boldsymbol{v}_i^{S_n}$ is the initial input embedding of the n-th source-domain sequence, $\boldsymbol{W}_{Q_u}^{S_n}$, $\boldsymbol{W}_{K_u}^{S_n}$, $\boldsymbol{W}_{V_u}^{S_n} \in \mathbb{R}^{d\times d}$ are learnable parameters, and $\boldsymbol{z}_t^{S_n}$ denotes the user attentive preference in the n-th source domain.

We employ concatenation to aggregate all user attentive preference from different domains, and then feed the concatenation vector into MLP to get the final representation of cross-domain user attentive preference:

$$\boldsymbol{z} = concat\left[\boldsymbol{z}_t, \dots, \boldsymbol{z}_t^{S_N}\right], \tag{12}$$

$$\boldsymbol{f}_t^u = \boldsymbol{z}\boldsymbol{W}^{(u)} + \boldsymbol{b}^{(u)}, \tag{13}$$

where N denotes the number of source domains, $\boldsymbol{z} \in \mathbb{R}^{(1+N)d}$ denotes the concatenation of all user preferences and $\boldsymbol{W}^{(u)} \in \mathbb{R}^{(1+N)d\times d}$, $\boldsymbol{b}^{(u)} \in \mathbb{R}^d$ are learnable parameters. We take the final output vector $\boldsymbol{f}_t^u \in \mathbb{R}^d$ as the cross-domain user attentive preference.

3.4. Cross-domain local attentive preference learning

Considering that the next interaction of user in the target domain may be related to an item he/she recently interacted with in a certain source domain, we propose CD-LAPL to exploit the user's sequential information in multiple domains then transfer knowledge across different domains.

We adapt the attention block which is introduced in Section 3.2 to measure the importance of a user's previous interactions in a source domain to current interaction in the target domain, and explore the transition patterns across sequences from different domains. Specifically, we denote the input embedding of the target-domain item at the last time step $\boldsymbol{v}_t$ as query, and denote the position-aware input embedding of the n-th source-domain sequence $\boldsymbol{X}^{S_n}$ as key and value, and then the cross-domain attention layer can be formalized as follows:

$$\alpha_i^{S_n} = softmax\left(\boldsymbol{v}_t\boldsymbol{W}_Q^{S_n}\left(\boldsymbol{x}_i^{S_n}\boldsymbol{W}_K^{S_n}\right)^T\right), \forall i \in \{1,2,\dots,t'\}, \tag{14}$$

$$\boldsymbol{h}_t^{S_n} = \sum_{i=1}^{t'} \alpha_i^{S_n}\left(\boldsymbol{x}_i^{S_n}\boldsymbol{W}_V^{S_n}\right), \tag{15}$$

where $\boldsymbol{W}_Q^{S_n}$, $\boldsymbol{W}_K^{S_n}$, $\boldsymbol{W}_V^{S_n} \in \mathbb{R}^{d\times d}$ are learnable parameters. We also employ a two-layer $FFN(\cdot)$ to further improving the model performance:

$$\boldsymbol{f}_t^{S_n} = ReLU\left(\boldsymbol{h}_t^{S_n}\boldsymbol{W}^{S_n(1)} + \boldsymbol{b}^{S_n(1)}\right)\boldsymbol{W}^{S_n(2)} + \boldsymbol{b}^{S_n(2)}, \tag{16}$$

where $\boldsymbol{W}^{S_n(1)}, \boldsymbol{W}^{S_n(2)} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}^{S_n(1)}, \boldsymbol{b}^{S_n(2)} \in \mathbb{R}^d$ are weights and biases for the two-layer FFN, respectively. And each cross-domain attention block can also be regarded as a cross-domain attention layer followed by a FFN.

We take the top cross-domain attention block's output vector $\boldsymbol{f}_t^{S_n} \in \mathbb{R}^d$ as the cross-domain local attentive preference, which represents a user's cross-domain dynamic interests at the t-th time step reflected from the target domain and the n-th source domain. Notice that for N source domains, we will obtain N cross-domain local attentive preferences.

3.5. Prediction layer

To combine all the output vectors from TD-APL, CD-UAPL and CD-LAPL, we try different designs for feature aggregation such as concatenation, summation and maximum. In this paper, we employ concatenation to aggregate all features which is the optimal choice as we found in the empirical studies in Session 4.10.

$$\boldsymbol{o} = concat\left[\boldsymbol{f}_t, \boldsymbol{f}_t^u, \dots, \boldsymbol{f}_t^{S_N}\right], \tag{17}$$

where $\boldsymbol{o} \in \mathbb{R}^{(2+N)d}$ denotes the concatenation of all the output vectors. Then, the concatenation vector is fed into an MLP to obtain the final representation of the user's preference:

$$\boldsymbol{o}_t = \boldsymbol{o}\boldsymbol{W}^{(o)} + \boldsymbol{b}^{(o)}, \tag{18}$$

where $\boldsymbol{W}^{(o)} \in \mathbb{R}^{(2+N)d\times d}$ and $\boldsymbol{b}^{(o)} \in \mathbb{R}^d$ are learnable parameters, and $\boldsymbol{o}_t \in \mathbb{R}^d$ denotes the final representation of the user's preference. Finally, the prediction score of item i can be calculated as follows:

Algorithm 1: The learning procedure of transfer via joint attentive preference learning (TJAPL).

1: **Initialization:** Initialize model parameters Θ.
2: **repeat**
3: **for** *each epoch* **do**
4: Collect a batch of users and their corresponding sequences in the target domain and the source domains.
5: Calculate the target-domain attentive preference $\boldsymbol{f}_t$ of time step t via Equations (1) - (7).
6: Calculate the cross-domain user attentive preference $\boldsymbol{f}_t^u$ of time step t via Equations (8) - (13).
7: **for** $n \leftarrow 1$ **to** N **do**
8: Calculate the cross-domain local attentive preference $\boldsymbol{f}_t^{S_n}$ of time step t via Equations (14) - (16).
9: **end for**
10: Calculate the final representation of the user's preference $\boldsymbol{o}_t$ of time step t via Equations (17) - (18).
11: Predict the preference score $r_{t,i}$ of item i at each time step t via Equation (19).
12: Calculate the binary cross-entropy loss $\mathcal{L}$ via Equation (20).
13: Update the model parameters via $\nabla_\Theta \mathcal{L}$.
14: **end for**
15: **until** Convergence

$$r_{t,i} = \boldsymbol{o}_t(\boldsymbol{v}_i)^T. \tag{19}$$

We adopt Adam as the optimizer [42] and the binary cross-entropy loss function for our TJAPL can be formalized as:

$$\mathcal{L} = -\sum_{u\in\mathcal{U}}\sum_{t=1}^{L-1}\delta(v_{t+1})[\log(\sigma(r_{t,v_{t+1}}) + \log(1-\sigma(r_{t,j})], \tag{20}$$

where $j \in \mathcal{I}\backslash\mathcal{V}^u$ is a sampled negative item and σ is the sigmoid function. The indicator function $\delta(v_{t+1}) = 1$ only if v_{t+1} is not a padding item, and 0 otherwise.

3.6. The learning algorithm

Algorithm 1 describes the training procedure of our TJAPL. First, we calculate the target-domain attentive preference $\boldsymbol{f}_t$ through TD-APL (line 5), which is fed with the sequence of the target domain. Next, we calculate the cross-domain user attentive preference $\boldsymbol{f}_t^u$ through CD-UAPL (line 6) and the cross-domain local attentive preference $\boldsymbol{f}_t^{S_n}$ via CD-LAPL (lines 7 - 9), which take the target-domain sequence and the source-domain sequences as input. Then, we aggregate all features to obtain the final representation of user's preference $\boldsymbol{o}_t$ (line 10). Finally, we calculate the prediction score $r_{t,i}$ for item i at time step t (line 11). We optimize our proposed model by minimizing the loss function $\mathcal{L}$ (lines 12 - 13).

3.7. Discussions

Our TJAPL can be viewed as an attention-based model. In this section, we discuss our TJAPL with other related attention-based models. SASRec [8] is a seminar method which employs the attention mechanism to sequential recommendation. It is more efficient and has an advantage in capturing the long-range dependency compared with RNN-based models, but it cannot handle the cross-domain scenarios.

CD-SASRec [39] is a cross-domain version of SASRec. It adopts SASRec to learn the source-domain preference of the user, then fuses that preference into the target-domain item embedding, and finally uses SASRec in the target domain to complete the sequential recommendation task. This method can alleviate the problem of data sparsity, but it does not capture the cross-domain sequential dependency well, since what is fused in the target domain at each time step is the overall feature of the user in the source domain.

Considering that a same user typically has similar preferences in different domains, we further design CD-UAPL to exploit the user's preferences in multiple source domains. Then, we transfer them to the target domain in order to get a more comprehensive user's preference to alleviate the data sparsity problem. We believe that there is also some important sequential information in the source domain, because the next interaction of the user in the target domain is probably related to an item he/she recently interacted with in a certain source domain. Therefore, we also devise CD-LAPL to explore the transition patterns across sequences from different domains. This mechanism is pivotal for knowledge transfer, as it identifies patterns in how users transit from one item to another across different domains. By learning these patterns, we can fully exploit the rich behavioral data in the source domain to improve the target-domain recommendation performance. Notice that capturing sequential information of users is also the key to distinguishing cross-domain sequential recommendation from cross-domain general recommendation. Moreover, our TJAPL can effectively handle scenarios with multiple source domains. In practical applications, utilizing multiple source-domain data can provide richer information, which allows the model to capture user preferences comprehensively and alleviate the data sparsity problem.

4. Experiments

In this section, we introduce the experimental settings and conduct extensive empirical studies to answer the following six research questions:

(RQ1) What's the performance of our proposed TJAPL as compared with the state-of-the-art methods?

Table 3
Statistic details of datasets.

Dataset	# Overlapped-Users	# Items	# Interactions	Avg. Length	Density
Movie		59513	460226	42.11	0.07%
CD	10929	91169	344221	31.50	0.03%
Book		236049	607657	55.60	0.02%

(RQ2) How does our TJAPL perform when using different source domains? Is it beneficial to the model if we increase the number of source domains?
(RQ3) Does our TJAPL alleviate the data sparsity issue?
(RQ4) What's the influence of various components in our TJAPL?
(RQ5) How does the key parameters affect the performance of our TJAPL?
(RQ6) What's the impact of different feature aggregation methods in our TJAPL?

4.1. Datasets

We conduct empirical studies on Amazon,[1] which is a review data collected by [43] from the eponymous e-commerce website. The Amazon data is suitable for the study of cross-domain sequential recommendation compared with other recommendation data, since it contains overlapped users in multiple domains. We choose three datasets with different categories, i.e., "Movie", "CD" and "Book". Then we follow [8,36] and preprocess these three datasets as follows: 1) We assume that all user interactions with items are positive feedback and determine the order of interactions by their timestamps. 2) We only keep the users and items with no fewer than five related interactions. And we drop duplicated (user, item) pairs. 3) We only keep the sequence of a user who has interactions in all the three domains. 4) We use the leave-one-out method for evaluation, which splits the sequence of each user into three parts, i.e., the last interaction for test, the penultimate interaction for validation and the remaining interactions for training. The statistics of the processed datasets are shown in Table 3. We will make the scripts of data processing and the processed datasets publicly available once the paper is accepted.

4.2. Evaluation metrics

We apply two common ranking-based metrics for the evaluation of recommendation performance, i.e., HT@10 (hit ratio) and NDCG@10 (normalized discounted cumulative gain), where the former is equivalent to recall because each user has exactly one preferred item in the test data in our case. Specifically, HT@10 denotes to the proportion of ground-truth items presenting in the top-10 recommended lists, while NDCG@10 is sensitive to the exact ranking positions of the items in recommended lists. We follow the common strategy in [8,19] and sample 100 negative items as candidates to avoid heavy computation on all (user, item) pairs. These 100 negative items have not been interacted with by the users and are sampled according to their popularity to ensure that they are informative and representative [36].

4.3. Baselines

To justify the effectiveness of our TJAPL, we compare it with thirteen recent and competitive methods from four recommendation categories. We adopt one general recommendation method (i.e., BPRMF) and one cross-domain general recommendation method (i.e., CoNet) since these traditional models typically do not perform well in the sequential recommendation task. We adopt six sequential recommendation methods (i.e., FPMC, GRU4Rec, GRU4Rec+, Caser, GCSAN and SASRec), including MCs-based model, RNN-based model, CNN-based model, GNN-based model and attention-based model. For cross-domain sequential recommendation, we adopt five cross-domain sequential methods, including an RNN-based model π-net, a GNN-based model DA-GCN and three attention-based models (i.e., CD-SASRec, RecGURU, C^2DSR). These methods are all recently proposed and representative ones for the studied problem. Notice that we have also used most of those baselines in [44].

- BPRMF [2]. A classic model for general recommendation which optimizes the matrix factorization by a pairwise ranking loss.
- CoNet [12]. A neural transfer learning model for general cross-domain recommendation through a collaborative cross-network [45]. It adds cross-connection units on MLP to enable dual information transfer.
- FPMC [4]. A traditional method for sequential recommendation that combines matrix factorization (MF) and first-order MCs. This method mainly models sequential information through MCs.
- GRU4Rec [5]. An RNN-based method for sequential recommendation that employs GRU to model users' behavior sequences step by step.
- GRU4Rec+ [29]. An improved model based on GRU4Rec [5] that develops a new loss function and an additional sampling strategy.
- Caser [6]. A CNN-based model for sequential recommendation that adopts convolutional filters to the embeddings of the most recent items in order to capture high-order Markov chains.

[1] http://jmcauley.ucsd.edu/data/amazon/.

- GCSAN [34]. A GNN-based model which constructs directed graphs for the sequences and applies gated GNNs to obtain all node vectors involved in the session graphs.
- SASRec [8]. An attention based model that explores the sequential dependencies by adopting the attention mechanism. It also works as the target-domain attentive preference learning module in our TJAPL.
- π-Net [13]. An RNN-based model for cross-domain sequential recommendation which adopts a cross-domain transfer unit to capture and transfer user information across domains.
- DA-GCN [38]. A novel GNN-based model which links different domains by constructing a cross-domain sequence graph. It employs GNN to model the complicated interaction relationships, as well as the explicit structural information.
- CD-SASRec [39]. An improved method based on SASRec [8] for cross-domain sequential recommendation. It fuses the source-domain aggregated vector into the target-domain item embedding to transfer information across domains.
- RecGURU [40]. It employs a self-attentive autoencoder to derive latent user representations, and proposes an adversarial learning method to unify user embeddings generated from different domains into a single global generalized user representation, which captures the overall preferences of users.
- C^2DSR [46]. A novel model which adopts a graphical and attentional encoder to capture the item relationships, and devises two sequential objectives with a contrastive objective to jointly learn the single-domain and cross-domain user representations.

4.4. Implementation details

We implement GRU4Rec,[2] Caser,[3] SASRec,[4] π-net,[5] RecGURU[6] and C^2DSR[7] following the published codes of the original papers. The latent dimensionality d is selected from $\{10, 20, 30, 40, 50\}$ and configured as $d = 50$ for all baselines since we find that on such sparse datasets, these methods usually benefit from a larger value of d [6,8]. For our TJAPL, we use Adam optimizer with a learning rate of 0.001, and the mini-batch size is set to 128, the dropout rate is set to 0.5. For all datasets, we set the maximum length of a sequence L to 100. The negative sampling number is set to 2048 for GRU4Rec+, the vertical and horizontal filter numbers are set to 4 and 16, respectively, for Caser, and other key parameters are followed the suggestions of the corresponding papers or turned on the validation data. For the architecture of attention-based methods (i.e., SASRec, CD-SASRec, RecGURU, C^2DSR and our TJAPL), we adopt single-head attention layers and two attention blocks (i.e., $B = 2$). For the GNN-based methods (i.e., GCSAN, DA-GCN and C^2DSR), the depth of the GNN layer is set to 2. For the shared-account recommendation methods (i.e., π-Net and DA-GCN), the latent user number is set to 1.

For cross-domain recommendation methods, we only report the best performance of models with the corresponding source domain (i.e., when the target domain is Movie, we use CD or Book as a source domain to assist in training, and show only the best results). For our proposed TJAPL, since our model can be applied to a multi-domain scenario, we report the results of simultaneously utilizing two source domains, and we discuss the performance compared with one single source domain in Section 4.6. The source codes of our TJAPL are available at https://csse.szu.edu.cn/staff/panwk/publications/TJAPL/.

4.5. Overall performance comparison (RQ1)

Table 4 illustrates the experimental results of our TJAPL and the baselines on three datasets, where the results of most baselines are also reported in [44]. Moreover, to provide a more comprehensive comparison with the baselines, we study the general top-K recommendation performance with different values of K in {1, 5, 10}, which are reported in Table 5. We mark the best result in each column in bold and the second-best result in underline.

Firstly, we can observe that our proposed TJAPL outperforms all the baselines on all the three datasets, and gains 9.46% NDCG@10 and 8.43% HR@10 improvements on average against the strongest baseline, which demonstrates the capability of our TJAPL to model the sequential information with cross-domain data. Besides, the sequential recommendation methods outperform the general recommendation baseline, which indicates the importance of extracting sequential information from users' behavior. And the cross-domain sequential recommendation methods outperform most traditional sequential recommendation methods, which demonstrates the significance of taking into account the cross-domain information. Moreover, the attention-based models achieve outstanding performances in both sequential recommendation and cross-domain sequential recommendation, which demonstrates the superiority of the attention mechanism in modeling dynamic preference. Furthermore, among the three datasets, the "Movie" dataset has the most significant improvement, which is probably because the "Movie" dataset is more tightly related to the other domains, i.e., a user's interaction sequences in the "Book" and "CD" domains are likely to influence his/her next interaction in the "Movie" domain, so knowledge transfer is more effective. Additionally, the cross-domain sequential recommendation methods achieve relatively small improvements on the "Book" dataset, since the source domain ("Movie" or "CD") is sparser (as is shown in Table 3). And our TJAPL can still achieve superior performance on the "Book" dataset because it can utilize both the "Movie" domain and the "CD" domain as source domains simultaneously, which demonstrates the effectiveness of knowledge transfer across multiple domains. From Table 5,

[2] https://github.com/hidasib/GRU4Rec.
[3] https://github.com/graytowne/caser_pytorch.
[4] https://github.com/kang205/SASRec.
[5] https://github.com/mamuyang/PINet.
[6] https://github.com/Chain123/RecGURU.
[7] https://github.com/cjx96/C2DSR.

Table 4
Recommendation performance of one general recommendation method (i.e., BPR), one general cross-domain recommendation method (i.e., CoNet), six sequential recommendation methods (i.e., FPMC, GRU4Rec, GRU4Rec+, Caser, GCSAN, SASRec), five cross-domain sequential recommendation method (i.e., π-Net, DA-GCN, CD-SASRec, RecGURU, C^{2}DSR) and our TJAPL leveraging two source domains, on three datasets. Notice that for CoNet, π-net, DA-GCN, CD-SASRec and RecGURU, C^{2}DSR, we report the better results when transferring knowledge from one of the other two source domains.

Method	Movie		CD		Book	
	NDCG@10	HT@10	NDCG@10	HT@10	NDCG@10	HT@10
BPRMF	0.0597	0.1256	0.0492	0.1142	0.0465	0.1088
CoNet	0.0675	0.1489	0.0756	0.1484	0.0764	0.1819
FPMC	0.0723	0.1697	0.0819	0.1785	0.0695	0.1416
GRU4Rec	0.1017	0.1984	0.1210	0.2247	0.1066	0.2162
GRU4Rec+	0.1133	0.2157	0.1440	0.2536	0.1293	0.2407
Caser	0.1231	0.2243	0.1267	0.2473	0.1163	0.2274
GCSAN	0.1576	0.2889	0.1783	0.3206	0.1291	0.2409
SASRec	0.1822	0.3234	0.1978	0.3569	0.1401	0.2607
π-Net	0.1113	0.2080	0.1265	0.2335	0.1042	0.2101
DA-GCN	0.1736	0.3124	0.1897	0.3458	0.1283	0.2375
CD-SASRec	0.1789	0.3173	0.2009	0.3614	0.1481	0.2737
RecGURU	0.1884	0.3433	0.2044	0.3649	0.1373	0.2556
C^{2}DSR	0.1922	0.3423	0.1978	0.3435	0.1486	0.2752
TJAPL	**0.2133**	**0.3769**	**0.2199**	**0.3907**	**0.1632**	**0.2984**

Table 5
The general top-K recommendation performance (i.e., K = 1, 5, 10).

Dataset	Metrics	GCSAN	SASRec	CD-SASRec	C^{2}DSR	TJAPL
Movie	NDCG@1	0.0522	0.0694	0.0664	0.0728	**0.0868**
	NDCG@5	0.1235	0.1411	0.1375	0.1524	**0.1691**
	NDCG@10	0.1576	0.1822	0.1789	0.1922	**0.2133**
	HT@5	0.1894	0.2125	0.2061	0.2281	**0.2538**
	HT@10	0.2889	0.3234	0.3173	0.3423	**0.3769**
CD	NDCG@1	0.0604	0.0737	0.0789	0.0756	**0.0896**
	NDCG@5	0.1278	0.1503	0.1587	0.1557	**0.1826**
	NDCG@10	0.1783	0.1978	0.2009	0.1978	**0.2199**
	HT@5	0.2113	0.2391	0.2441	0.2345	**0.2722**
	HT@10	0.3206	0.3569	0.3614	0.3435	**0.3907**
Book	NDCG@1	0.0479	0.0506	0.0555	0.0596	**0.0698**
	NDCG@5	0.0943	0.1082	0.1184	0.1221	**0.1394**
	NDCG@10	0.1291	0.1407	0.1481	0.1486	**0.1632**
	HT@5	0.1489	0.1656	0.1792	0.1829	**0.2069**
	HT@10	0.2409	0.2607	0.2737	0.2752	**0.2984**

we can observe that our TJAPL performs best in all cases, and the performance trends of various algorithms are similar across different values of the parameter K.

4.6. Influence of source domains (RQ2)

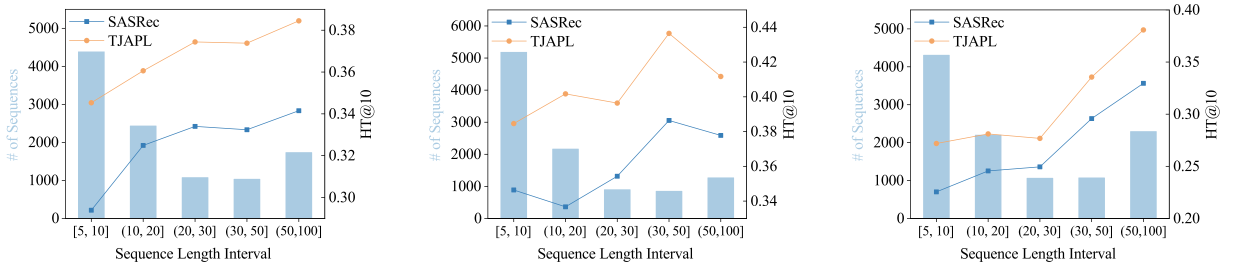
The recommendation performance of our TJAPL with different source domains is shown in Table 6. Notice that “Both” means leveraging both the other two domains for knowledge transfer and preference learning.

We can observe that the “Movie” dataset and the “CD” dataset achieve the best performance when leveraging the other two domains (i.e., “Both”) while the second best performance is obtained for the “Book” domain. This indicates that our model can effectively improve the recommendation performance by transferring knowledge from more than one source domain to a target domain. Our TJAPL is able to capture more user preference in a dense domain with more interaction data and then transfers it to a sparse domain. Hence, using two source domains performs better than using one single source domain. Moreover, when leveraging only a single domain, transferring knowledge from the “Book” domain seems more helpful because it contains more interaction data (as is shown in Table 3).

For the “Book” domain, it achieves the best performance when using the “Movie” domain as the source domain. The reason is that the “Movie” domain may be more tightly related to the “Book” domain and therefore performs better. Besides, leveraging the “CD” domain as the source domain also performs better than leveraging both domains [47]. The reason may be that for a user, data across multiple source domains is not always related, in which case the introduction of extra information and noise would make it less efficient than leveraging a single domain.

Table 6
Performance of different source domains, including knowledge transfer from one or two source domains.

Target Domain / Source Domain	Movie		CD		Book	
	NDCG@10	HT@10	NDCG@10	HT@10	NDCG@10	HT@10
Movie	—	—	0.2136	0.3842	**0.1684**	**0.3059**
CD	0.2024	0.3625	—	—	0.1612	0.2986
Book	0.2062	0.3687	0.2189	0.3895	—	—
Both	**0.2133**	**0.3769**	**0.2199**	**0.3907**	0.1632	0.2984



(a) Movie (b) CD (c) Book

Fig. 3. Performance of different target-domain sequence lengths on three datasets for SASRec and our TJAPL.

Furthermore, we can observe that our TJAPL still outperforms all the baselines on all the datasets (as shown in Table 4) when only leverages one single domain for knowledge transfer, which demonstrates the stability and the superiority of our TJAPL.

4.7. Performance analysis w.r.t. sparsity (RQ3)

In this subsection, we divide the users into groups according to the lengths of their behavior sequences in the target domain and identify the reasons for improvement by comparing the performance of SASRec and TJAPL in different user groups. Moreover, to study the impact of source-domain sequence lengths, we fix the target-domain sequence length interval and divide the users into groups according to their sequence lengths in the source domain. We report the results of "Movie" and "CD" when leveraging the "Book" domain for knowledge transfer, and leveraging the "Movie" domain for "Book". Notice that we only report the performance on HT@10 since the variation tendency of the NDCG@10 is similar to that of HT@10. The analysis validates the effectiveness of introducing cross-domain information in alleviating the data sparsity problem.

4.7.1. Performance w.r.t. target-domain sequence length

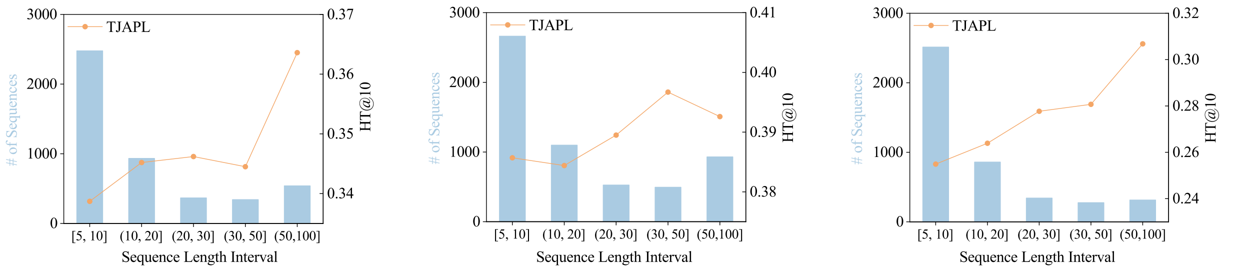
According to the users' sequence lengths in the target domain, We divide them into five user groups, and report the average HT@10 on each user group for SASRec and our TJAPL. Fig. 3 depicts the size of each user group and the corresponding HT@10 performance. It can be seen that the interaction data of the majority of users is sparse in the target domain. The group with the shortest sequence length interval contains the most users on all the datasets, and the number of users decreases as the sequence length interval of the group gets longer.

As shown in Fig. 3, our TJAPL achieves significant improvement on users within short sequence length intervals, with the relative largest improvement ranging from 17.48% to 20.57% on all the datasets. That's because the shorter users' sequence lengths indicate the sparser their interaction data, in which case the traditional single-domain method (i.e., SASRec) struggles to adequately capture users' preferences. In contrast, the introduction of the rich source-domain data can enhance users' preferences, and the knowledge transfer across domains seems to be more effective in this situation. This indicates the effectiveness of our TJAPL to alleviate the data sparsity issue in the target domain. Meanwhile, we also observe that our TJAPL achieves better performance than SASRec on all the user groups, which also demonstrates the superiority of our TJAPL in sequential recommendation. Furthermore, the performance gradually improves as the target-domain sequence length increases, but it shows a slight fluctuation in Fig. 3(b). The reason is that the recommendation performance is already relatively good in the interval (30,50] (as evident from the trend in SASRec).

4.7.2. Performance w.r.t. source-domain sequence length

To explore the impact of the source-domain sequence length on the recommendation performance of the target domain, we select the shortest target-domain sequence interval (i.e., the sparsest data) and then divide the users into groups according to their source-domain sequence lengths.

As is shown in Fig. 4, similar to the target domain, the user group with the shortest source-domain sequence length interval contains the most users on all the datasets, and the number of users decreases as the sequence length interval of the group gets longer. Moreover, we can find that the target-domain recommendation performance generally gets improvement as the source-domain sequence length increases. This is reasonable since the model can capture user's preference better in the source domain with more interaction data, so as to transfer a more comprehensive user's preference to the target domain.



(a) Movie (b) CD (c) Book

Fig. 4. Performance of different source-domain sequence lengths on three datasets for our TJAPL.

Table 7
Recommendation performance in ablation studies of our TJAPL with different architectures. Notice that 'T', 'U', 'C', 'U1', 'U2' represent TD-APL, CD-UAPL, CD-LAPL, target-domain UAPL and source-domain UAPL, respectively.

Setting / Architecture	Book → Movie		Book → CD		Movie → Book	
	NDCG@10	HT@10	NDCG@10	HT@10	NDCG@10	HT@10
T	0.1840	0.3291	0.1978	0.3569	0.1401	0.2607
T + U	0.1864	0.3531	**0.2201**	0.3881	0.1665	**0.3076**
T + U1	0.1822	0.3445	0.2154	0.3842	0.1629	0.2953
T + U2	0.1876	0.3459	0.2118	0.3775	0.1619	0.2924
T + C	0.1922	0.3586	0.2097	0.3756	0.1579	0.2883
T + C + U	**0.2062**	**0.3687**	0.2189	**0.3895**	**0.1684**	0.3059

4.8. Ablation study (RQ4)

We conduct an ablation study to evaluate the contribution of different components of our TJAPL, and the results are presented in Table 7. In particular, we only report the results of "Movie" and "CD" when leveraging the "Book" domain for knowledge transfer, and leveraging the "Movie" domain for "Book". We compare the separate effect of TD-APL (i.e., SASRec, denoted as 'T') with the joint effects that additionally add CD-UAPL (denoted as 'U') and CD-LAPL (denoted as 'C'). We also examine the effects of different domains on CD-UAPL, i.e., target-domain user attentive preference learning (denoted as 'U1') and source-domain user attentive preference learning (denoted as 'U2'). Moreover, we compare the joint effects of all the combination approaches.

Our observations are as follows.

- **'T + U' vs. T.** The integrated model with the addition of CD-UAPL always significantly outperforms the separate one, which demonstrates the importance of capturing the cross-domain user attentive preference and indicates the effectiveness of our CD-UAPL.
- **'T + U' vs. 'T + U1' or 'T + U2'.** CD-UAPL is considered as the combination of the target-domain and source-domain user attentive preference learning modules. We can find that 'T + U1' is generally more effective than 'T + U2' (except on "Movie") which means that users tend to generate the corresponding user preferences by applying their own target-domain data when it is sufficient. Furthermore, 'T + U' achieves the best overall performance, which indicates the benefit of combining the target-domain and source-domain user attentive preference.
- **'T + C' vs. T.** Without CD-LAPL (i.e., 'C'), we find that the performance is much worse. It confirms that this module can learn the cross-domain local attentive preference from the recent interactions of the target and source domain, which indicates the significance of capturing the transition patterns across sequences from different domains.
- **'T + C + U' vs. 'T + U' or 'T + C'.** We can see that almost all the best results are from 'T + C + U', which demonstrates the complementarity of these three parts. It captures the local attentive preference and user attentive preference from both the target and source domains, balancing these representations and improving the effect for sequential recommendation.

4.9. Influence of hyper-parameters (RQ5)

In this subsection, we explore the influence of two hyper-parameters (i.e., the latent dimensionality d and the number of attention blocks B) on the model performance. The results are presented in Fig. 5 and Fig. 6, respectively.

From Fig. 5, we observe that our model typically benefits from some relatively larger values of the dimensionality d, and it tends to be stable with $d \geq 40$ on all datasets. This means that a larger dimensionality does not always result in better performance due to the overfitting problem.

From Fig. 6, we observe that unlike SASRec, it is sufficient to get the best performance for our TJAPL in most cases by setting the number of attention blocks $B = 2$, and stacking more blocks may not further improve the performance. That' s because in the hierarchical structure, the feature learned by SASRec in the bottom attention block can be seen as the long-term preference, which is similar to the user attentive preference learned in our TJAPL, and the increased model capacity may lead to overfitting.

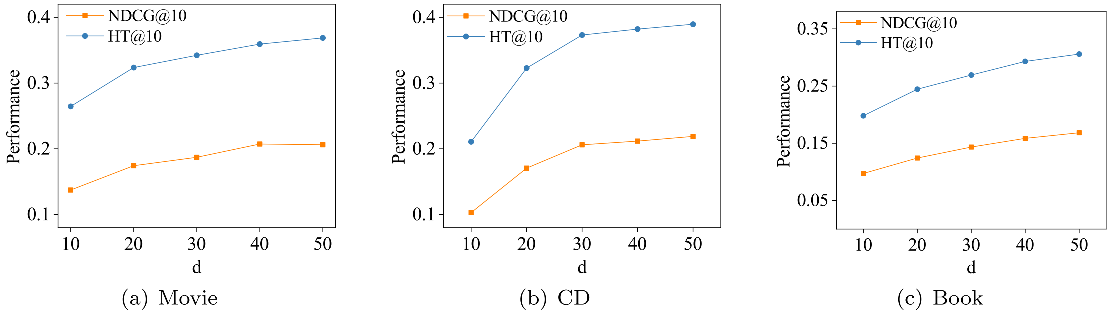


Fig. 5. Performance of different dimensionalities d on three datasets ($B = 2$).

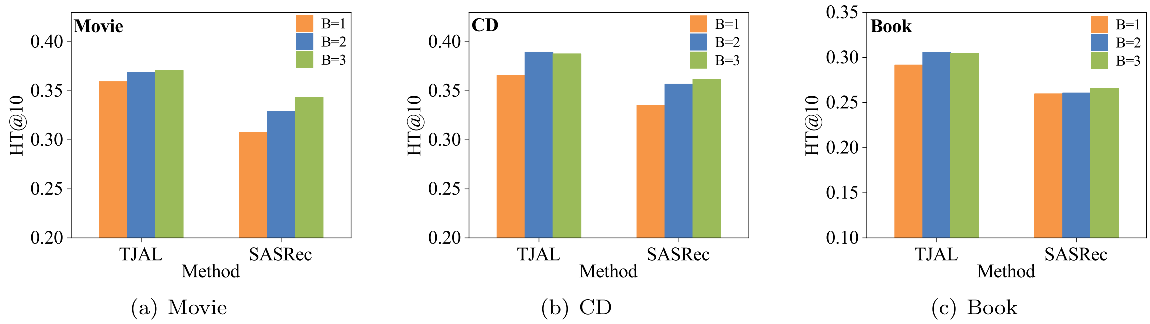


Fig. 6. Performance (HT@10) of different numbers of blocks B ($d = 50$) for SASRec and our TJAPL.

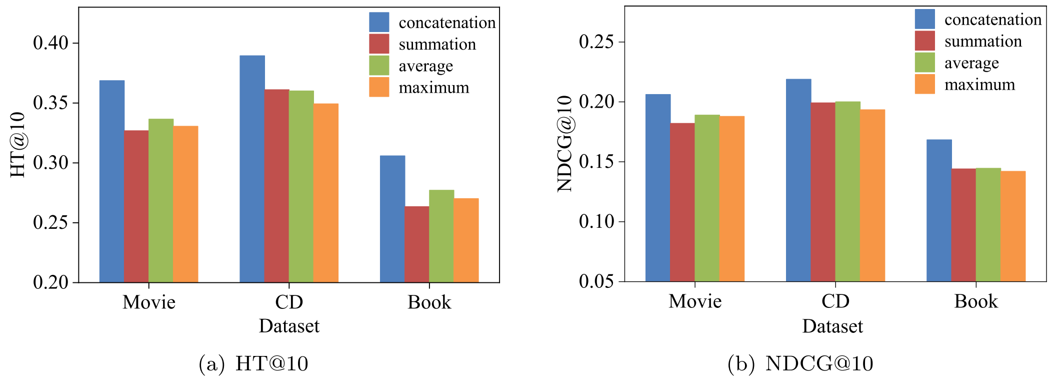


Fig. 7. Performance of different feature aggregation methods.

4.10. Aggregation methods comparison (RQ6)

In this subsection, we discuss the effects of different designs for feature aggregation in Eq. (17). As stated in Section 3.5, we employ concatenation to aggregate the features (i.e., target-domain attentive preference, cross-domain user attentive preference, and cross-domain local attentive preference) in the prediction layer. We replace the method of feature aggregation with summation, average and maximum, to examine their performance. As illustrated in Fig. 7, when employing concatenation to aggregate the features, our TJAPL achieves the best performance, while average performs better than summation and maximum (except on CD). It confirms that concatenation can effectively balance the information to aggregate all the features.

5. Conclusions and future work

In this work, we propose an effective transfer learning solution called transfer via joint attentive preference learning (TJAPL) to deal with a new and important problem, i.e., cross-domain sequential recommendation. We tackle the studied problem via attentive preference learning (ALP), including target-domain APL (TD-APL), cross-domain user APL (CD-UAPL) and cross-domain local APL (CD-LAPL). Specifically, we adopt the attention mechanism in TD-APL to effectively capture the dynamic preferences in the target domain. Moreover, we design CD-UAPL to enable knowledge transfer from multiple source domains to a target domain, leveraging the behavior sequences from the source domains to capture the user's overall preferences, and address the data scarcity problem. We also design CD-LAPL to explore the item transition patterns across sequences from different domains and capture the user's dynamic interests at each time step reflected from different domains. Furthermore, our TJAPL can be applied to a multi-domain scenario, which is more adaptable and flexible in real-world recommender systems. Extensive empirical studies on three real cross-domain datasets demonstrate that our TJAPL outperforms the competitive baselines in all cases.

In the future, we aim to improve our model in a multi-target cross-domain recommendation scenario, which suffers from a more serious negative transfer problem since the relatedness between the source and target domains may not be strong. Moreover, we are interested in studying our TJAPL in scenes of cross-domain or cross-organization privacy-aware federated recommendation [48], which can reduce the risk of privacy leakage from the introduction of rich source-domain data.

CRediT authorship contribution statement

Zitao Xu: Conceptualization, Investigation, Methodology, Software, Validation, Writing – original draft. **Weike Pan:** Conceptualization, Funding acquisition, Methodology, Supervision, Writing – review & editing. **Zhong Ming:** Funding acquisition, Resources, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

We thank the support of National Natural Science Foundation of China (Nos. 62172283 and 62272315), and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515010122).

References

[1] A. Paterek, Improving regularized singular value decomposition for collaborative filtering, in: Proceedings of KDD Cup and Workshop, 2007, pp. 39–42.
[2] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI '09, 2009, pp. 452–461.
[3] A. Zimdars, D.M. Chickering, C. Meek, Using temporal data for making recommendations, in: Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence, UAI '01, 2001, pp. 580–588.
[4] S. Rendle, C. Freudenthaler, L. Schmidt-Thieme, Factorizing personalized Markov chains for next-basket recommendation, in: Proceedings of the 19th International Conference on World Wide Web, WWW '10, 2010, pp. 811–820.
[5] B. Hidasi, A. Karatzoglou, L. Baltrunas, D. Tikk, Session-based recommendations with recurrent neural networks, in: Proceedings of the 4th International Conference on Learning Representations, ICLR '16, 2016.
[6] J. Tang, K. Wang, Personalized top-N sequential recommendation via convolutional sequence embedding, in: Proceedings of the 11th ACM International Conference on Web Search and Data Mining, WSDM '18, 2018, pp. 565–573.
[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NeurIPS '17, 2017, pp. 6000–6010.
[8] W. Kang, J.J. McAuley, Self-attentive sequential recommendation, in: Proceedings of the 18th IEEE International Conference on Data Mining, ICDM '18, 2018, pp. 197–206.
[9] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, P. Jiang, Bert4rec: sequential recommendation with bidirectional encoder representations from transformer, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, 2019, pp. 1441–1450.
[10] F. Zhu, Y. Wang, C. Chen, J. Zhou, L. Li, G. Liu, Cross-domain recommendation: challenges, progress, and prospects, in: Proceedings of the 30th International Joint Conference on Artificial Intelligence, IJCAI '21, 2021, pp. 4721–4728.
[11] T. Man, H. Shen, X. Jin, X. Cheng, Cross-domain recommendation: an embedding and mapping approach, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, in: IJCAI '17, vol. 17, 2017, pp. 2464–2470.
[12] G. Hu, Y. Zhang, Q. Yang, Conet: collaborative cross networks for cross-domain recommendation, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18, 2018, pp. 667–676.
[13] M. Ma, P. Ren, Y. Lin, Z. Chen, J. Ma, M.d. Rijke, 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, SIGIR '19, 2019, pp. 685–694.
[14] W. Sun, M. Ma, P. Ren, Y. Lin, Z. Chen, Z. Ren, J. Ma, M. De Rijke, Parallel split-join networks for shared account cross-domain sequential recommendations, IEEE Trans. Knowl. Data Eng. (2021), https://doi.org/10.1109/TKDE.2021.3130927.
[15] S. Kabbur, X. Ning, G. Karypis, FISM: factored item similarity models for top-N recommender systems, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13, 2013, pp. 659–667.
[16] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, WWW '01, 2001, pp. 285–295.
[17] F. Aiolli, Efficient top-N recommendation for very large scale binary rated datasets, in: Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, 2013, pp. 273–280.
[18] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, T.-S. Chua, Attentive collaborative filtering: multimedia recommendation with item- and component-level attention, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, 2017, pp. 335–344.

[19] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, WWW '17, 2017, pp. 173–182.

[20] Y. Wu, C. DuBois, A.X. Zheng, M. Ester, Collaborative denoising auto-encoders for top-n recommender systems, in: Proceedings of the 9th ACM International Conference on Web Search and Data Mining, WSDM '16, 2016, pp. 153–162.

[21] X. Li, J. She, Collaborative variational autoencoder for recommender systems, in: Proceedings of the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '17, 2017, pp. 305–314.

[22] S. Kang, J. Hwang, D. Lee, H. Yu, Semi-supervised learning for cross-domain recommendation to cold-start users, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, 2019, pp. 1563–1572.

[23] P. Li, A. Tuzhilin, Ddtcdr: deep dual transfer cross domain recommendation, in: Proceedings of the 13th International Conference on Web Search and Data Mining, WSDM '20, 2020, pp. 331–339.

[24] A.P. Singh, G.J. Gordon, Relational learning via collective matrix factorization, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, 2008, pp. 650–658.

[25] M. He, J. Zhang, P. Yang, K. Yao, Robust transfer learning for cross-domain collaborative filtering using multiple rating patterns approximation, in: Proceedings of the 11th ACM International Conference on Web Search and Data Mining, WSDM '18, 2018, pp. 225–233.

[26] Q. Cui, T. Wei, Y. Zhang, Q. Zhang, Herograph: a heterogeneous graph framework for multi-target cross-domain recommendation, in: Proceedings of the 3rd Workshop on Online Recommender Systems and User Modeling Co-Located with the 14th ACM Conference on Recommender Systems, RecSys '20, 2020, pp. 1–7.

[27] R. He, J. McAuley, Fusing similarity models with Markov chains for sparse sequential recommendation, in: Proceedings of the 16th IEEE International Conference on Data Mining, ICDM '16, 2016, pp. 191–200.

[28] R. He, W.-C. Kang, J. McAuley, Translation-based recommendation, in: Proceedings of the 11th ACM Conference on Recommender Systems, RecSys '17, 2017, pp. 161–169.

[29] B. Hidasi, A. Karatzoglou, Recurrent neural networks with top-k gains for session-based recommendations, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18, 2018, pp. 843–852.

[30] M. Quadrana, A. Karatzoglou, B. Hidasi, P. Cremonesi, Personalizing session-based recommendations with hierarchical recurrent neural networks, in: Proceedings of the 11th ACM Conference on Recommender Systems, RecSys '17, 2017, pp. 130–137.

[31] Y.K. Tan, X. Xu, Y. Liu, Improved recurrent neural networks for session-based recommendations, in: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS '16, 2016, pp. 17–22.

[32] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, J. Wu, Sequential recommender system based on hierarchical attention network, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI '18, 2018, pp. 3926–3932.

[33] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, T. Tan, Session-based recommendation with graph neural networks, in: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, AAAI '19, 2019, pp. 346–353.

[34] C. Xu, P. Zhao, Y. Liu, V.S. Sheng, J. Xu, F. Zhuang, J. Fang, X. Zhou, Graph contextualized self-attention network for session-based recommendation, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI '19, 2019, pp. 3940–3946.

[35] Y. He, Y. Zhang, W. Liu, J. Caverlee, Consistency-aware recommendation for user-generated item list continuation, in: Proceedings of the 13th International Conference on Web Search and Data Mining, WSDM '20, 2020, pp. 250–258.

[36] J. Lin, W. Pan, Z. Ming, Fissa: fusing item similarity models with self-attention networks for sequential recommendation, in: Proceedings of the 14th ACM Conference on Recommender Systems, RecSys '20, 2020, pp. 130–139.

[37] M. Ma, P. Ren, Z. Chen, Z. Ren, L. Zhao, P. Liu, J. Ma, M. de Rijke, Mixed information flow for cross-domain sequential recommendations, ACM Trans. Knowl. Discov. Data 16 (4) (2022) 1–32.

[38] L. Guo, L. Tang, T. Chen, L. Zhu, Q.V.H. Nguyen, H. Yin, DA-GCN: a domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation, in: Proceedings of the 13th International Joint Conference on Artificial Intelligence, IJCAI '21, 2021, pp. 2483–2489.

[39] N. Alharbi, D. Caragea, Cross-domain self-attentive sequential recommendations, in: Proceedings of International Conference on Data Science and Applications, ICONDATA '22, 2022, pp. 601–614.

[40] C. Li, M. Zhao, H. Zhang, C. Yu, L. Cheng, G. Shu, B. Kong, D. Niu, Recguru: adversarial learning of generalized user representations for cross-domain recommendation, in: Proceedings of the 15th ACM International Conference on Web Search and Data Mining, WSDM '22, 2022, pp. 571–581.

[41] T. Zhang, P. Zhao, Y. Liu, V.S. Sheng, J. Xu, D. Wang, G. Liu, X. Zhou, Feature-level deeper self-attention network for sequential recommendation, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI '19, 2019, pp. 4320–4326.

[42] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations, ICLR '15, 2015.

[43] J. McAuley, C. Targett, Q. Shi, A. van den Hengel, Image-based recommendations on styles and substitutes, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15, 2015, pp. 43–52.

[44] Z. Xu, W. Pan, Z. Ming, A multi-view graph contrastive learning framework for cross-domain sequential recommendation, in: Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23, 2023.

[45] I. Misra, A. Shrivastava, A. Gupta, M. Hebert, Cross-stitch networks for multi-task learning, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '16, 2016, pp. 3994–4003.

[46] J. Cao, X. Cong, J. Sheng, T. Liu, B. Wang, Contrastive cross-domain sequential recommendation, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22, 2022, pp. 138–147.

[47] S. Yao, Z. Feng, J. Song, L. Jia, Z. Zhong, M. Song, Chemical property relation guided few-shot molecular property prediction, in: Proceedings of the 8th International Joint Conference on Neural Networks, 2022, pp. 1–8.

[48] Z. Lin, W. Pan, Q. Yang, Z. Ming, Recommendation framework via fake marks and secret sharing, ACM Trans. Inf. Syst. (2022).