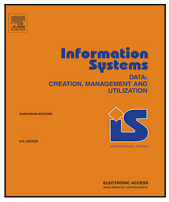
Information Systems 124 (2024) 102388

Contents lists available at ScienceDirect

Information Systems

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



Enhancing cross-market recommendations by addressing negative transfer and leveraging item co-occurrences

Zheng Hu [a], Satoshi Nakagawa [b], Shi-Min Cai [a], Fuji Ren [a], Jiawen Deng [a,*]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[b] The University of Tokyo, Tokyo 113-8656, Japan

ARTICLE INFO

Keywords:
Recommender systems
Cross-market recommendation
Transfer learning

ABSTRACT

Real-world multinational e-commerce companies, such as Amazon and eBay, serve in multiple countries and regions. Some markets are data-scarce, while others are data-rich. In recent years, cross-market recommendation (CMR) has been proposed to bolster data-scarce markets by leveraging auxiliary information from data-rich markets. Previous CMR algorithms have employed techniques such as sharing market-agnostic parameters or incorporating inter-market similarity to optimize the performance of CMR. However, the existing approaches have several limitations: (1) They do not fully utilize the valuable information on item co-occurrences obtained from data-rich markets (such as the consistent purchase of mice and keyboards). (2) They ignore the issue of negative transfer stemming from disparities across diverse markets. To address these limitations, we introduce a novel attention-based model that exploits users' historical behaviors to mine general patterns from item co-occurrences and designs market-specific embeddings to mitigate negative transfer. Specifically, we propose an attention-based user interest mining module to harness the potential of common items as bridges for mining general knowledge from item co-occurrence patterns through rich data derived from global markets. In order to mitigate the adverse effects of negative transfer, we decouple the item representations into market-specific embeddings and market-agnostic embeddings. The market-specific embeddings effectively model the inherent biases associated with different markets, while the market-agnostic embeddings learn generic representations of the items. Extensive experiments conducted on seven real-world datasets illustrate our model's effectiveness.[1] Our model outperforms the suboptimal model by an average of 4.82%, 6.82%, 3.87%, and 5.34% across four variants of two metrics. Extensive experiments and analysis demonstrate the effectiveness of our proposed model in mining general item co-occurrence patterns and avoiding negative transfer for data-sparse markets.

1. Introduction

Online shopping has become the mainstream way of e-commerce today. Compared with physical stores, online businesses provide convenience for customers [1,2]. Online shopping systems need recommendation algorithms to help users solve the problem of information overload [3–5]. Traditional academic and industrial recommendation methods utilize data from a single market to train models and serve the corresponding market. In the era of globalization, multinational e-commerce behemoths such as Amazon, Shopee, and eBay have expanded their operations across multiple regions, thereby gaining access to a wealth of diverse market data. To fully exploit the potential inherent in these multiple markets, the concept of cross-market recommendation (CMR) has emerged [6]. Intuitively, these parallel markets share common items while having distinct users. It is important to highlight that while certain markets boast abundant data resources, others grapple with the challenges posed by data scarcity. In response to this, researchers have recently put forth CMR models that adeptly harness information derived from parallel markets, thereby enhancing the performance of recommendation systems.

As illustrated in Fig. 1, two significant limitations are typically overlooked by previous CMR methods. First, the widespread cross-market co-occurrence of items is ignored by most existing methods. These shared items across parallel markets act as intuitive bridges for knowledge transfer. The historical behaviors of users reveals patterns of item co-occurrence that could serve as a foundation for knowledge transfer between markets. Unfortunately, current methods do not explicitly mine and utilize this essential general knowledge. Second, market-specific user interests generate interference in the knowledge transfer process, leading to a degradation in recommendation performance. This issue, widely known as "negative transfer", remains largely

* Corresponding author.
E-mail address: dengjw2016@gmail.com (J. Deng).

[1] Our codes and checkpoints are available at https://github.com/laowangzi/ACMR.

https://doi.org/10.1016/j.is.2024.102388
Received 21 November 2023; Received in revised form 1 March 2024; Accepted 12 April 2024
Available online 16 April 2024

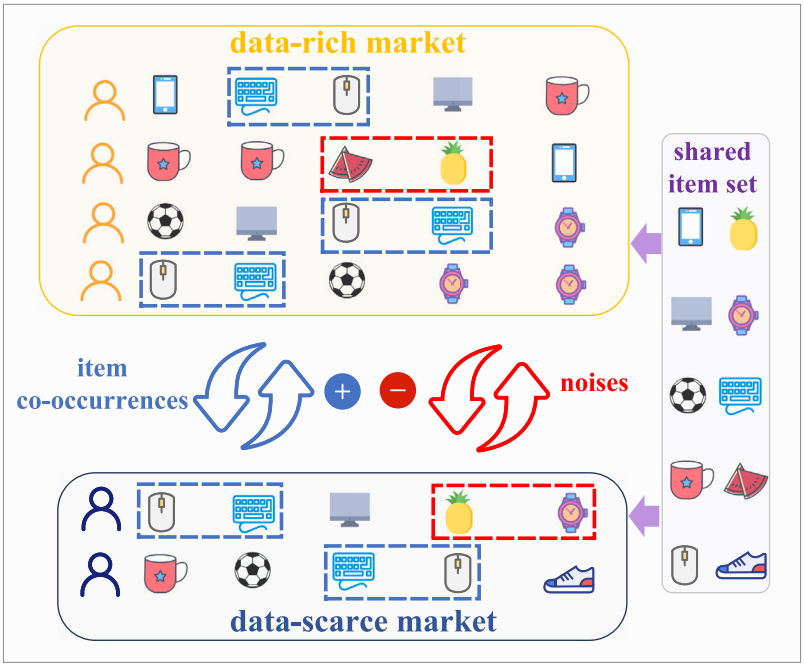


Fig. 1. A simple illustration of item co-occurrences and noises in CMR. The blue dashed box delineates the co-occurrence of a mouse and keyboard, symbolizing a piece of general knowledge that is widely applicable across diverse markets. Conversely, the red dashed box illustrates market noises, which are a result of market-specific variations in user interests. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

unaddressed in prior research. For example, Hamed and Mohammad et al. [6] proposed a NeuMF [7]-based method, named FOREC, which is designed for CMR in a market-agnostic parameter way. They choose a market as the primary source for training the NeuMF model as the shared bottom and then fork multi-layer perceptrons as specific heads for each target market. However, NeuMF-based models face challenges in capturing item co-occurrence relationships, which are sources of direct information from the additional markets. Cao et al. [8] proposed an item similarity-based method, called M^3Rec, which mines two inter- and intra-market similarities using multiple markets data. Then, they leverage the similarities as prior knowledge to fine-tune all local markets. However, this approach overlooks the need to mitigate negative transfer effects that may arise when inter-market similarity exhibits a negative correlation.

To address these challenges, we propose an **A**ttentive **C**ross-**M**arket **R**ecommendation model called **ACMR**, which improves recommendation performance by capturing item co-occurrence across markets and avoiding negative transfer. Unlike the cross-market knowledge transfer strategies used in previous studies, our method seeks to mine prevalent market-agnostic item co-occurrence in users' historical behaviors. Specifically, we use the self-attention mechanism to capture the general knowledge of item co-occurrences provided by the massive amounts of data from parallel markets. Therefore, our model can achieve better knowledge transfer and effectively utilize information from global markets. Furthermore, we introduce an *Explicit User Modeling* component, which leverages attention-processed item sequences to model user interests. To prevent negative transfer resulting from mutual noisy interference of diverse markets, we decouple item representations into market-agnostic embeddings and market-specific embeddings. Market-specific embeddings are used to model the bias of the specific market to avoid the negative transfer problem, while market-agnostic embedding learns the inherent characteristics of items. Our contributions are as follows:

- We introduce the ACMR, which enhances knowledge transfer by mining widespread item co-occurrences across parallel markets. We alleviate the negative transfer problem in the CMR task by decoupling the item representation into a market-specific representation and a market-agnostic representation. Our model maximizes the reuse of global market information while avoiding mutual interference between markets.
- Extensive experiments are conducted on seven national markets across multiple continents. We compare our model among five types of baselines using two metrics, which include traditional recommendation models, attention-based recommendation models, cross-market recommendation models, cross-domain recommendation models, and multi-domain recommendation models. The experimental results consistently highlighted the superior performance of the proposed model.
- We conduct thorough ablation experiments to showcase the effectiveness of our proposed key components. Our empirical findings validate that our attention-based model is adept at capturing item co-occurrence across various markets. We also find that market-specific embeddings significantly mitigate the negative transfer problem in data-sparse markets. Visualization of the learned item representations reveals that our method effectively maps item representations to a consistent vector space, which demonstrates the effectiveness of ACMR in cross-market item representation learning.

The remainder of this paper is organized as follows: Section 2 discusses related works. The preliminaries and symbol notions are given in Section 3. Section 4 describes the details and training process of ACMR. The experimental results and analysis are provided in Section 5. Section 6 is the presentation of conclusions and future work.

2. Related work

2.1. Cross-market and cross-domain recommendation

Cross-domain recommendation (CDR) and CMR share the same goal: they both aim to improve recommendation performance by leveraging external information from other categories or markets. Similar to general cross-domain tasks [9,10], the key to CDR and CMR is effectively leveraging knowledge transfer to improve the performance of the model in the target domain. However, the assumptions of CMR and CDR are different. CMR assumes that user sets

in each market are disjoint while sharing the same item set across markets. For CDR, the situation is reversed: the users are shared across the domains while the items are disjoint. For example, Contrastive Cross-Domain Sequential Recommendation (C^2DSR) [11] jointly mine the single- and cross-domain user preferences by maximizing the mutual information between the domains. Personalized Transfer of User Preferences for Cross-domain Recommendation (PTUPCDR) [12] proposes a meta-network that generates personalized bridge functions to transfer personalized preferences across domains. In order to obtain de-biased representations, CDRIB [13] utilizes the information bottleneck principle to encode domain-shared information, allowing for recommendations to be made in both source and target domains by jointly considering domain interactions and de-bias pre-trained representations. MetaCAR [14] proposes a Content-Aware cross-domain recommendation method to improve model performance through meta-augmentation. Further research has focused on mitigating biases and transferring knowledge across different domains [15,16]. Cross-domain recommendation models are commonly used to address the cold-start problem in the target domain [17–19]. Multi-domain recommendation (MDR) shares similar core technology with CDR, but with the goal of improving performance across all interactive domains, rather than just transferring knowledge in a specific direction [20]. CDR methods can be adapted to the MDR problem by treating each domain as the target domain. Additionally, multi-task approaches such as Shared-Bottom [21], Cross-Stitch [22], and MMoE [23] can be applied to tackle the MDR problem by considering each domain as a task. The work on MDR is largely inspired by multi-task learning. For example, CMoIE [24] extends the MMoE framework by incorporating conflict resolution modules. STAR [25] separates the model parameters into shared and domain-specific parts, utilizing a star topology with shared centered parameters and domain-specific parameters for different domains.

To our knowledge, the most closely related work are two CMR methods: FOREC [6] and M^{3}Rec [8]. The FOREC first pre-train a market-agnostic NeuMF on multiple markets as the shared bottom. They claim this step generates a generalized recommendation model with significative internal representations, which maximize the reusability of parameters translating into target market adaptation. Then, the FOREC forks multi-layer perceptrons as the market-specific head and fine-tunes the model on the target market. In addition, Bhargav et al. [26] proposed to add market-adapted embeddings on the basis of FOREC to further improve the market adaptation performance. However, their architecture fails to sufficiently decouple market-specific embeddings and fails to capture item co-occurrences in global markets. The M^{3}Rec considers the CMR problem from the perspective of item similarity. They utilize EASE^R [27] to learn the intra-market similarity on global markets. They apply the node2vector [28] on the items' co-occurrence weighted matrix to capture the item correlation as the inter-market item similarity. After obtaining the intra- and inter-market similarity, they leverage them as prior knowledge to fine-tune all local markets.

2.2. Attention-based recommendation

Some works utilize the attention mechanism's excellent sequence modeling ability to mine the users' interests according to the user–item interaction sequence. For example, Deep Interest Network (DIN) [29] employs the attention mechanism adaptively calculating the representation vector of user interests by considering the relevance of historical behaviors given a candidate item. Deep Interest Evolution Network (DIEN) [30] uses GRU [31] to model user behavior sequences, considering sequence information based on DIN. Contrastive Graph Self-Attention Network (CGSNet) [32] aggregates item representations from three distinct graph encoders through an attention-based fusion module as the global perspective. Meanwhile, it designs a self-attention subnetwork to learn the complex item transition information from the local perspective. Finally, it introduces a contrastive learning paradigm based on the two perspectives. Attention mechanism has also been introduced into recommendation algorithms based on knowledge graphs. For example, Shimizu et al. [33] propose an explainable recommendation framework based on a knowledge graph attention network, which utilizes the side information of items and realizes high recommendation accuracy. Chen et al. [34] introduced an attention-based knowledge graph recommendation framework, which utilizes a Collaborative Guidance Mechanism to extract information from historical behaviors and knowledge, resulting in more accurate and tailored recommendations. The transformer architecture increasingly combines with recommendation algorithms. For example, Bert4Rec [35] employs deep bidirectional self-attention to model the user interaction sequence. UNBERT [36] utilizes the transformer encoder to model the content of news at the word level and the user behaviors at the new level.

Compared with the existing CMR methods, we creatively employ the attention-based model, which captures the co-occurrences of items across markets and avoids the negative transfer. Different from the existing recommendation models based on the attention mechanism, we redesign the model and modify the pre-training task according to the requirements of the CMR task. The problem formulation and details of our model are in the following sections.

3. Preliminaries

3.1. Problem formulation

In this section, we give the definitions of CMR and notations. The symbol notations used in this paper are defined in Table 1. Assuming there are m parallel markets $M = \{M_1, M_2 .... M_m\}$. Denote the item sets as $I = \{I_1, I_2 ... I_m\}$ and the user sets as $U = \{U_1, U_2 ... U_m\}$. All markets share the same item set. For the item set of each market, it can be expressed as:

$$\{I_p \in I \mid \forall\ p\ \in [1,\ 2,\ ...m]\} \tag{1}$$

There is a user–item interaction matrix $Y_t \in \{0,1\}^{|U_t| \times |I_t|}$ for each market $M_t = (U_t, I_t)$. In the matrix Y_t, $y^t_{uv} = 1$ represents that the user u likes the item v. The remains in Y_t are set to 0. We take the $y_{uv} = 1$ records out of the user–item interaction matrix Y_t. Then we group these records by users to generate each user's historical behaviors $(s_1, s_2 .... s_z)$, where $z = \sum_{i=1}^{m} |U_i|$. It is worth noting that in the definition of this paper, the historical behaviors are not arranged in chronological order.

The problem can be described as follows: Given the parallel markets and the historical behavior data of users, our goal is to utilize the global market data to predict users' purchase probability in a target market and generate recommendation sequences based on the prediction results.

3.2. Transformer Layer Brief

The *Transformer Layer* is a bidirectional attention mechanism that calculates attention scores between any two vectors [37]. As illustrated in the right part of Fig. 2, the *Transformer Layer* contains two sub-layers: the multi-head self-attention sub-layer and the position-wise feed-forward network. Residual connection [38] and layer normalization [39] are applied for both sub-layers individually. That is, the calculation process of each sub-layer is $LayerNorm(x + Sublayer(x))$, where $Sublayer(x)$ is the functions as in Eq. (3) and Eq. (4). The following is a brief introduction to the sub-layers.

Multi-Head Self-Attention. This sub-layer aims to capture the contextual representation of each item in the input sequence [40]. The scaled dot-product attention [37] is defined as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{2}$$

Table 1
Symbol notion.

Symbol	Definition
$M = (M_1, M_2, ..M_m)$	the market set
U_t	the user set of market t
I_t	the item set of market t
$s_u = \{v_1, v_2..., v_n\}$	historical behaviors of user u
E_i, E_m	item and market-specific embedding matrix
Q, K, V	the projection matrices corresponding to query, key and value
$T^l = [t_1^l, t_2^l..., t_n^l]$	the (l+1)th input of the *Transformer layer*
W, b	the learnable projection matrix and bias
$\hat{y}$	the predicted user–item interaction probability
$\mathcal{Y}^+$	the observed interactions
$\mathcal{Y}^-$	the negative samples
ED	the embedding dimension
SL	the max input user interactions length
L	the number of the *Transformer Layers*
H	the number of the attention heads

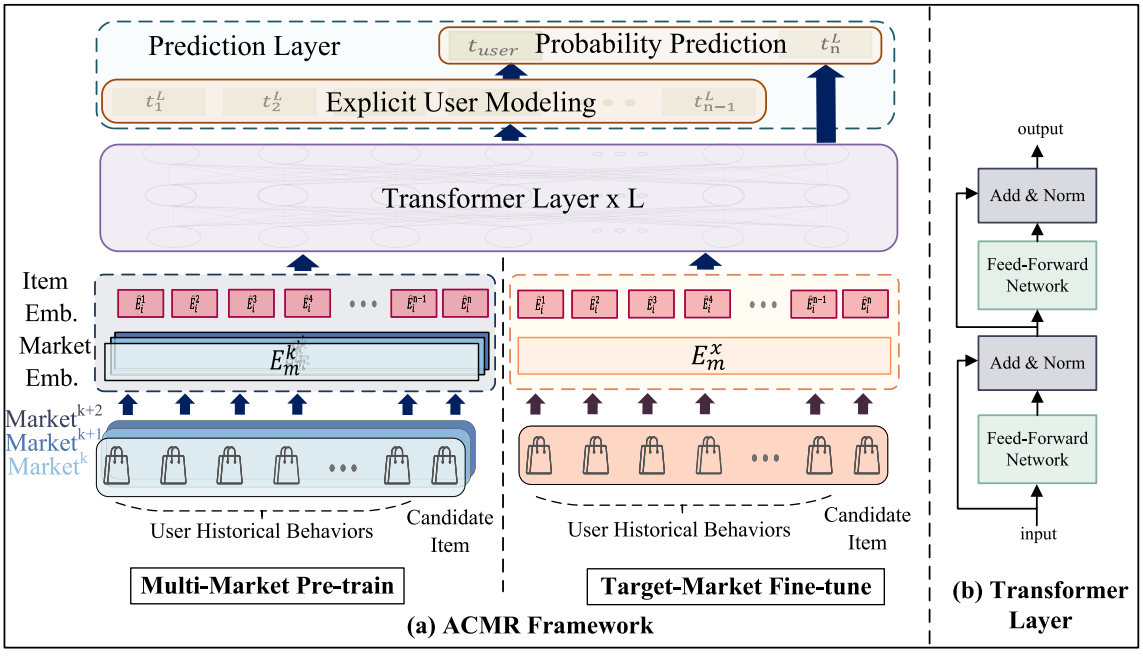


Fig. 2. The architecture of the proposed ACMR is displayed, illustrating (a) the framework and (b) the transformer layer utilized in it. The pre-training and fine-tuning stages are depicted, with distinct colors representing various markets. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where Q, K, V are matrix correspondingly representing query, key, and value. These matrices are linearly projected from T^l as in Eq. (3). Let $T^l \in \mathbb{R}^{n \times d}$ denote the $(l+1)$th input. The multi-head self-attention (MH) applies g parallel attention functions to produce the output representations, which are concatenated and linear projected:

$$\begin{aligned} MH(T^l) &= Concat(head_1, head_2 ... head_g) W^O \\ head_i &= Attention(T^l W_i^Q, T^l W_i^K, T^l W_i^V) \end{aligned} \tag{3}$$

where $W_i^Q \in \mathbb{R}^{d \times \frac{d}{h}}$, $W_i^K \in \mathbb{R}^{d \times \frac{d}{h}}$, $W_i^V \in \mathbb{R}^{d \times \frac{d}{h}}$ are learnable parameter matrix for each head. $W^O \in \mathbb{R}^{d \times d}$ is a projection matrix for the concatenated result.

Position-wise Feed-Forward Network. This sub-layer comprises two linear projections, with a ReLU activation function intervening. This configuration is applied uniformly and independently to each position. Let $T^l = [t_1^l; ...; t_n^l]$, the calculation process of this sub-layer is:

$$\begin{aligned} C^l &= LayerNorm(T^l + Dropout(MH(T^l))) \\ F(C^l) &= [FFN(c_1^l); ...; FFN(c_n^l)] \\ FFN(x) &= RELU(xW^1 + b^1)W^2 + b^2 \end{aligned} \tag{4}$$

where W^i, b^i are learnable parameters. We omit the layer subscript l for convenience. While the linear projections are shared across different positions, they use different parameters from layer to layer.

4. Method

Let $\hat{s}_i$ denotes an input sequence of ACMR, which is constructed by concatenating a user's historical behaviors s_i and a candidate item. In this section, we present the details of the proposed ACMR model. Fig. 2 shows the overall architecture of ACMR, which is composed of an *Embedding Layer*, L stacked bidirectional *Transformer Layers* and a *Prediction Layer*. We will cover each component in detail next. This section also describes how ACMR is trained and optimized on multi-market data.

4.1. Embedding layer

For a given item, the corresponding embeddings include the market-agnostic embedding and the market-specific embedding. The final input representation is constructed by summing them. For market-agnostic

embeddings, we randomly initialize the learnable matrix $E_i \in \mathbb{R}^{|V|\times d}$, where d is the embedding dimension.

Market-specific Embedding. Intuitively, the representation of the same item in different markets should be similar but not exactly alike. To this end, we try to inject market information into the input representations. Inspired by the idea of the position embedding and the segment embedding [37,41], we create a learnable parameter matrix $E_m \in \mathbb{R}^{|M|\times d}$ as the market-specific embedding. The market-specific embeddings model the bias of specific markets so that the same item has different representations in parallel markets. In this way, the shared E_i can learn the unbiased general knowledge of the items, which reduces the negative transfer caused by the mutual interference between the markets.

Denote a set of market-agnostic embeddings retrieved by an input sequence as $\hat{E}_i^i \in \mathbb{R}^{n\times d}$, where n is the length of the sequence. Assuming the corresponding market-specific embedding is E_m^k. We broadcast the market-specific embedding as $\hat{E}_m^k \in \mathbb{R}^{n\times d}$ and add to the market-agnostic embedding to construct the input representation.

$$\hat{E}_{input}^i = \hat{E}_i^i + \hat{E}_m^k \tag{5}$$

4.2. Stacked transformer layers

In this paper, the primary objective of the *Transformer Layer* is to capture general item co-occurrences using the self-attention mechanism, thereby acquiring market-agnostic generic knowledge. The self-attention mechanism is widely employed in the field of natural language processing (NLP) to model token sequences and learn general knowledge [41,42]. In the domain of recommendation systems, the attention mechanism evaluates the significance of various items in the item sequence by utilizing learned weights. This aids the model in comprehensively representing the item sequence [29,35,43]. Drawing inspiration from these studies, we employ the *Transformer Layer* to capture item co-occurrence patterns present in user behavior sequences across different markets. Overall, the Embedding Layer's market embeddings furnish market-specific information, thereby enhancing the diversity of item representations. Meanwhile, the *Transformer Layer* empowers the model to learn the generality in user behavior through the self-attention mechanism.

In order to capture more complex interactions between items, we stack L *Transformer Layers*. However, The risk of overfitting increases as the network goes deeper. We apply dropout [44] to avoid overfitting. In summary, ACMR refines the representation sequence as follows:

$$\begin{aligned} T^{l+1} &= Trm(T^l),\ l \in [0,\dots,L-1] \\ Trm(T^l) &= LayerNorm(C^l + Dropout(F(C^l))) \\ C^l &= LayerNorm(T^l + Dropout(MH(T^l))) \end{aligned} \tag{6}$$

where $T^l \in \mathbb{R}^{n\times d}$ denote the $(l+1)$th input and $T^0 = \hat{E}_{input}$. $Trm()$ note the *Transformer Layer* described in Section 3.2.

4.3. Prediction layer

After the hierarchical interaction of L layers across all positions in the previous module, we get the final item representation sequence T^L. The representation of each position contains the implicit context information. In order to adapt to the recommendation task, in this section, we explicitly model the user and predict the purchase probability.

Explicit User Modeling. In order to make a personalized recommendation, we generate an explicit user representation based on T^L, which models the user interests. In the NLP field, Sentence-Bert [45] experimented with three pooling methods to derive semantically meaningful sentence embedding: Using the output representation of the special token, computing a max-over-time of the output vectors, and computing the mean of all output vectors. Compared to using special tokens and taking the maximum value, the average strategy does not make distinct choices regarding which part of the sequence is more important than others. As a result, it captures general information without overly focusing on specific features [46]. Taking inspiration from this, in the field of recommendation systems, we aim to retain the complete historical interest information of users. Therefore, we adopt the average pooling strategy for explicit user modeling. By doing so, similar users are positioned closely in the vector space, which aligns with the concept of user modeling in collaborative recommendation [47,48].

$$t_{user} = MeanPooling(t_1^L, t_2^L, \dots, t_{n-1}^L) \tag{7}$$

where $[t_1^L, t_2^L, \dots, t_{n-1}^L] \in T^L$ are the item representations corresponding to the user's historical behaviors.

Probability Prediction. We concatenate the user representation t_{user} and the candidate item representation t_n^L as the input of this layer. Drawing on the design of prediction layers used in downstream tasks within the field of NLP [41], and similar to earlier recommendation algorithms that employ self-attention mechanisms for modeling item sequences [35], we employ a single-layer feedforward network as the prediction layer. Additionally, we utilize a *Sigmoid* activation function to estimate the probability of a user engaging with the candidate item.

$$\hat{y} = \sigma\left(Concat(t_{user}, t_n^L)W + b\right) \tag{8}$$

where $W \in \mathbb{R}^{2d\times 1}$ and b are learnable parameters. σ is the *Sigmoid* activate function. We empirically find that increasing the number of layers of the fully connected layer does not improve the performance. Presumably, because the stacked *Transformer Layers* already have enough fitting ability.

4.4. Model training

The training process of ACMR consists of two steps: pre-training and fine-tuning. The same loss function is used in both stages as follows,

$$\mathcal{L} = \sum_{(s_u,i)\in\mathcal{Y}^+\cup\mathcal{Y}^-} y_{s_u,i}\log(\hat{y}_{s_u,i}) + \left(1 - y_{s_u,i}\right)\log\left(1 - \hat{y}_{s_u,i}\right) \tag{9}$$

Algorithm 1: Model Training Algorithm

Input: I, M, target market M_t, parameter set Θ, global data D_{train} and D_{valid}
Hyperparameter: *iter*
Output: Θ'

```
model parameter Θ initialization;
S ← metrics are initialized as 0;
Pre-train Stage
for i in iter do
    Θ ← Train(Θ, I, M, D_train);
    S' ← Evaluate(Θ, I, M, D_valid);
    if S' > S then
        Θ' ← Θ;
        S ← S'
    end
end
Fine-tune Stage
S ← 0;
D_train ← Filter(D_train, M_t);
D_valid ← Filter(D_valid, M_t);
for i in iter do
    Θ ← Train(Θ, I, M_t, D_train);
    S' ← Evaluate(Θ, I, M_t, D_valid);
    if S' > S then
        Θ' ← Θ;
        S ← S'
    end
end
```

Table 2
Statistics of the preprocessed dataset. Markets are arranged from left to right in order of the number of interactions from largest to smallest. The "Avg.length" means the average length of the user interactions.

	CA	UK	FR	DE	MX	JP	IN	total
#User	4668	3352	1838	1851	1878	487	239	14 313
#Item	5735	3251	1879	2179	1645	955	470	8304
#Ratings	44 779	31 547	17 624	17 300	17 095	4485	2015	134 845
#Avg.length	9.6	9.4	9.6	9.3	9.1	9.2	8.4	9.2

where $\mathcal{Y}^+$ denotes the observed interactions in Y, and $\mathcal{Y}^-$ denotes the negative instances, which are sampled from unobserved interactions. The target label $y_{s_{u,i}}$ values 0 or 1 denoting whether u has interacted with i. We adopt mini-batch Adam [49] to train the model and update the parameters.

The distinction between pre-training and fine-tuning lies in the data utilized. During the pre-training phase, our model is trained on data from various parallel markets, resulting in a market-agnostic model. This model yields generalized recommendation performance and latent item representations encompassing universal knowledge. Furthermore, the market-specific embedding models the biases present in different markets in this phase. The fine-tuning phase exclusively employs data from the target market to eliminate noise from other markets and customize the model to fit the target market. In essence, the initial pre-training on global markets facilitates the acquisition of general knowledge, while subsequent fine-tuning directs the model's attention towards the specific market.

In contrast to previous pre-trained models [35,41,50,51], we employ the same task for both pre-training and fine-tuning stages. Earlier research utilized the *Cloze* task [52] during pre-training primarily to prepare for various downstream tasks. However, our model focuses on a single downstream task, rendering the use of different tasks unnecessary across the two phases. Furthermore, utilizing a consistent training task addresses the performance gap caused by inconsistent tasks between these stages. In our setting, the pre-training process is generic, allowing for easy deployment in new markets by simply loading pre-trained parameters and fine-tuning them for the target market. Additionally, within the pre-train and fine-tune paradigm we have adopted, our model can be efficiently deployed in a cold-start target market by fine-tuning the globally pre-trained model for local market differences.

5. Experiment and discussion

5.1. Experimental setup

Dataset. Following FOREC [6], our model is assessed on the electronics category of the XMarket dataset,[2] comprising seven parallel markets originating from various regions across three continents: Germany (DE), Canada (CA), Japan (JP), India (IN), France (FR), Mexico (MX), and the United Kingdom (UK). Same as the previous works [6,8,53,54], we filtered the users and items that there exist less than five interactions.[3] These selected parallel markets exhibit variations in size, culture, and evaluation of the CMR model's performance. The detailed information of the preprocessed datasets is shown in Table 2. To gain a better understanding of the variation in data quantity across different markets, we arrange the markets in Table 2 from left to right based on the number of ratings. Notably, CA and UK are the largest markets with significantly higher interaction volumes compared to other markets. Conversely, JP and IN are the smallest in terms of data availability.

Baselines. We use several popular recommendation models as baseline methods for comparison, which could be categorized into five classes: (1) Traditional methods: NeuMF [7] and Wide&Deep [55], (2)Cross-market methods: FOREC [6] and M^{3}Rec [8], (3) Attention-based method: Bert4Rec [35], (4) Cross-domain methods: CoNet [15] and DDTCDR [16], (5) Multi-domain (task) methods: Cross-Stitch [22] and MMoE [23].

- **NeuMF** is a neural network-based collaborative filtering model. It ensembles matrix factorization (MF) and multi-layer perceptron (MLP) so that it unifies the strengths of linearity of MF and non-linearity of MLP for modeling the user–item latent structures.
- **Wide&Deep** is a popular two-tower recommendation model based on neural networks. It jointly trains *wide* linear models and *deep* neural networks to combine the benefits of memorization and generalization for recommender systems.
- **Bert4Rec** employs the deep bidirectional self-attention to the sequential recommendation task. It adopts the *Cloze* objective to train the model, predicting the random masked items in the sequence by joint conditioning on their left and right context.
- **FOREC** is a recommendation model for CMR, which is a combination of a NeuMF as the shared bottom across parallel markets and several fully connected layers as the market-specific head. Different from the origin paper using one specific market to train the bottom, we use all markets except the target market as source markets to train the bottom in our implementation. We experimentally observed that our implementation performed better.
- **M^{3}Rec** is the state-of-the-art cross market recommendation method. It first calculates two global item similarities: intra- and inter-market similarities. It learns the intra-market similarity by adopting linear models with closed-form solutions and then captures the high-order inter-market similarity by the random walk. Then, it incorporates the global item similarities and conducts the market adaptation operation for each target market.
- **CoNet** is a deep transfer learning approach for cross-domain recommendation. It leverages neural networks as the base model and incorporates cross-connections between hidden layers of two base networks to enable dual knowledge transfer across domains, effectively addressing the data sparse issue in recommender systems.
- **DDTCDR** is a cross-domain recommendation model that utilizes dual learning to transfer information between two related domains in order to provide recommendations. It incorporates a novel latent orthogonal mapping technique to extract user preferences across multiple domains while preserving relations between users and combines it with an autoencoder approach to extract the latent essence of feature information.
- **Cross-Stitch** aims to learn shared representations by introducing a new sharing unit called the "cross-stitch" unit. The unit combines activations from multiple networks and can be trained end-to-end, allowing the network to learn an optimal combination of shared and domain-specific representations across multiple domains.
- **MMoE** adapts the Mixture-of-Experts (MoE) structure by sharing expert submodels across all domains while utilizing a gating network trained to optimize each domain, allowing for simultaneous learning of multiple goals and domains.

Implementation Details. For FOREC,[4] NeuMF,[5] CoNet,[6] DDTCDR[7] and Bert4Rec,[8] we use the code provided by the corresponding authors.

[2] https://xmrec.github.io/
[3] The processed data is available at https://github.com/hamedrab/FOREC
[4] https://github.com/hamedrab/FOREC
[5] https://github.com/hexiangnan/neural_collaborative_filtering
[6] https://github.com/njuhugn/CoNet
[7] https://github.com/lpworld/DDTCDR
[8] https://github.com/FeiSun/BERT4Rec

Table 3
The overall experimental results. Specifically, the "++" notation signifies the model trained on all parallel markets. The optimal results are designed in bold. Sub-optimal results are annotated with underlining.

	Models	Recall													
		DE		*JP*		*IN*		*FR*		*CA*		*MX*		*UK*	
		R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10
single	NeuMF	0.4262	0.5445	0.2936	0.4045	0.4895	0.5397	0.3993	0.5375	0.4385	0.5511	0.6070	0.6879	0.4821	0.5915
	Wide&Deep	0.4424	0.5861	0.3470	0.4579	0.4686	0.5313	0.4020	0.5484	0.4732	0.5936	0.6395	0.7215	0.5122	0.6300
	Bert4Rec	0.4349	0.5483	0.2936	0.3696	0.4686	0.5397	0.3835	0.5152	0.4331	0.5207	0.6102	0.6741	0.4949	0.5897
multiple	NeuMF++	0.5386	0.7023	0.4476	0.5995	0.5481	0.7154	0.5554	0.7317	0.4061	0.5426	0.6709	0.7667	0.5420	0.6655
	Wide&Deep++	0.5586	0.7044	0.4024	0.5893	0.5439	0.7154	0.5636	0.7426	0.3969	0.5353	0.6751	0.7651	0.5575	0.6894
	Bert4Rec++	0.4932	0.5921	0.4065	0.5092	0.4142	0.5732	0.5010	0.6180	0.3487	0.4539	0.6549	0.7316	0.4812	0.5763
	CoNet	0.5197	0.6067	0.4189	0.5175	0.5439	0.5941	0.4755	0.5811	0.3935	0.4002	0.6874	0.7513	0.5301	0.6143
	DDTCDR	0.5500	0.6126	0.4559	0.5175	0.6109	0.6444	0.5343	0.5952	0.4644	0.5831	0.7210	0.7508	0.5325	0.6438
	MMoE	0.5375	0.6202	0.4168	0.5072	0.5272	0.5774	0.5169	0.5996	0.4621	0.5874	0.6821	0.7343	0.5382	0.6122
	Cross-Stitch	0.5321	0.6397	0.3922	0.5010	0.5356	0.5941	0.5484	0.6382	0.4916	0.5968	0.7034	0.7529	0.5495	0.6432
	FOREC	0.6347	0.7612	0.5893	0.6878	0.6820	0.7656	0.6479	0.7747	0.4963	0.5886	0.7587	0.8093	0.6357	0.7395
	M^{3}Rec	0.6185	0.7039	0.2731	0.3552	0.4769	0.5313	0.6512	0.7431	0.4033	0.4792	0.7539	0.8104	0.5817	0.6628
	ACMR	**0.6611**	**0.7931**	**0.6308**	**0.7888**	**0.7163**	**0.8033**	**0.6919**	**0.8239**	**0.5196**	**0.6359**	**0.7941**	**0.8739**	**0.6481**	**0.7636**
	Models	NDCG													
		DE		*JP*		*IN*		*FR*		*CA*		*MX*		*UK*	
		N@5	N@10	N@5	N@10	N@5	N@10	N@5	N@10	N@5	N@10	N@5	N@10	N@5	N@10
single	NeuMF	0.3020	0.3404	0.1911	0.2271	0.3524	0.3675	0.2745	0.3189	0.3173	0.3540	0.5096	0.5358	0.3758	0.4113
	Wide&Deep	0.3140	0.3603	0.2507	0.2861	0.3558	0.3751	0.2779	0.3250	0.3452	0.3843	0.5301	0.5567	0.4001	0.4381
	Bert4Rec	0.3126	0.3494	0.2053	0.2297	0.3636	0.3868	0.2603	0.3029	0.3170	0.3456	0.5129	0.5337	0.3860	0.4167
multiple	NeuMF++	0.3789	0.4325	0.2819	0.3312	0.3157	0.3705	0.3845	0.4416	0.2738	0.3182	0.5637	0.5945	0.4257	0.4658
	Wide&Deep++	0.3981	0.4456	0.2664	0.3268	0.3339	0.3896	0.3960	0.4543	0.2669	0.3119	0.5641	0.5934	0.4293	0.4721
	Bert4Rec++	0.3577	0.3900	0.2692	0.3031	0.2596	0.3121	0.3658	0.4034	0.2420	0.2761	0.5620	0.5868	0.3883	0.4191
	CoNet	0.3996	0.4279	0.3060	0.3375	0.4967	0.5130	0.3638	0.3984	0.3357	0.3380	0.6264	0.6472	0.4356	0.4628
	DDTCDR	0.4822	0.5023	0.3961	0.4160	0.5660	0.5767	0.4564	0.4763	0.3604	0.3988	0.6772	0.6868	0.4251	0.4610
	MMoE	0.4622	0.4889	0.3360	0.3654	0.4928	0.5083	0.4469	0.4738	0.3315	0.3721	0.6492	0.6662	0.4754	0.4993
	Cross-Stitch	0.4280	0.4628	0.3137	0.3492	0.4654	0.4844	0.4390	0.4678	0.3884	0.4225	0.6549	0.6711	0.4687	0.4988
	FOREC	0.4821	0.5233	0.4187	0.4510	0.5703	0.5983	0.4818	0.5238	0.3685	0.3986	0.6595	0.6759	0.4967	0.5306
	M^{3}Rec	0.4998	0.5278	0.2292	0.2553	0.4317	0.4489	**0.5220**	0.5518	0.3174	0.3419	0.6792	0.6976	0.4855	0.5118
	ACMR	**0.5032**	**0.5462**	**0.4731**	**0.5243**	**0.6066**	**0.6345**	0.5194	**0.5622**	**0.3933**	**0.4310**	**0.6938**	**0.7198**	**0.5172**	**0.5546**

For Wide&Deep, MMoE, Corss-Stitch and M^{3}Rec, we implement them with PyTorch according to the original papers. We employ PyTorch for the implementation of our ACMR model. To adapt Cross-Stitch and MMoE for the CMR task, we leverage shared user and item embeddings across different markets. The recommendation task in the target market and other markets are treated as different tasks that are trained simultaneously. During model training, the embeddings from different markets are averaged and used as the input for the model. In line with Bonab et al. [6], we have adapted DDTCDR for CMR by establishing connections between the item features in the MLP networks of the two markets. Similarly, we have made analogous modifications to the network structure of CoNet.

Hyperparameters Setting. We use Adam [49] to optimize all the models. For common hyperparameters in all models, we test the batch size of [512, 1024, 2048], the learning rate of [1e−3, 5e−4, 1e−4, 5e−5], and the latent dimension of [8, 16, 32, 64, 128]. We consider the ℓ_2 regularization in [1e−5, 1e−6, 1e−7]. In order to avoid overfitting, we apply a fixed dropout rate of 0.3 to all models. All other hyperparameters either follow the suggestion from the methods' authors or are tuned on the validation sets. We report the results of each baseline under its optimal hyperparameter settings. We apply the early stopping strategy with 50 epochs for all baselines and our model. To mitigate the impact of random variations, we randomly split the datasets and conduct five independent replicate experiments. The reported results are the average performance across these five independent replicate experiments. We train our model with a learning rate of 1e−3, ℓ_2 regularization of $1e{-}7$, batch size of 1024, and the latent dim of 32. After tuning the hyperparameters, we set the layer number $L = 4$, the head number $g = 8$, and the maximum sequence length as 50. For all models, We randomly sampled 4 negative instances per positive instance in the training set. The training device employed was a single 24G NVIDIA TITAN RTX.

Evaluation Metrics. We employ two commonly used metrics, namely $NDCG@K$ and $Recall@K$, to assess the quality of the rank lists generated by all the methods. The evaluation of these metrics is conducted at two specific cut-off points: 5 and 10. Similar to previous works [6,7], we construct the ground truth using the purchasing behavior by considering an item as relevant if the user gives a rating. We follow a long line of literature and use the leave-one-out strategy for validation and test [7,55–58]. Specifically, for each user, we randomly sample one interaction for validation and one for testing. In addition, we follow the literature and sample 99 negative items for each user in our evaluations.

5.2. Experimental results & discussion

Table 3 shows the experimental results of the ACMR against the baselines in terms of $Recall@K$ and $NDCG@K$ on all markets. The discrepancy between "single" and "multiple" in Table 3 lies in the volume of data employed for model training. "Single" denotes the practice of training the target market, whereas "multiple" refers to training the model on data sourced from all parallel markets. According to the results, we have the following insightful observations:

- Our ACMR achieves the best performance on almost all markets and metrics, significantly outperforming the state-of-the-art CMR model M^{3}Rec. Compared with the second-best model, ACMR has an average improvement of 4.82%, 3.87%, 6.82% and 5.34% on $Recall@5$, $NDCG@5$, $Recall@10$ and $NDCG@10$, respectively. This observation indicates that our model can best make full use of the information on parallel markets. In addition, we find that CMR models (ACMR, FOREC, and M^{3}Rec) generally perform better than traditional methods and attention-based methods. This observation indicates that the CMR method can make full use of multi-market information than other methods.

- Comparing CMR models (ACMR, FOREC, and M^{3}Rec), we observe that ACMR has a significant improvement in two data-scarce markets: JP and IN, with an improvement of 16.25%, 6.05% in terms of $NDCG$ and 14.68%, 4.92% in terms of $Recall$. This observation indicates that our model can best protect the small markets from the interference of other markets and avoid negative transfers while benefiting from global information.
- Our model exhibits a 0.49% lower performance compared to the optimal model M^{3}Rec on $NDCG@5$ in the FR market. Nevertheless, it is noteworthy that the performance of the M^{3}Rec model displays considerable volatility across different markets. Notably, our model surpasses M^{3}Rec by 106.4% on the JP dataset regarding $NDCG@5$. This disparity in performance can be attributed to M^{3}Rec's reliance on item similarity to transfer global market information. In certain markets, especially those with substantial differences from other markets (e.g., JP and other European markets), item similarity can result in negative transfer, leading to a decline in performance. Therefore, considering the stability of performance and the generalizability, our model remains the best.
- When comparing the performance of traditional recommendation models, such as NeuMF and Wide&Deep, trained on single-market ("single") and multi-market ("multiple") datasets, it is evident that they generally exhibit superior performance when trained on multi-market data. Intuitively, more market data brings more user–item interaction information, which is beneficial to improving the performance of recommendations. However, traditional recommendation algorithms cannot make full use of multi-market data because they cannot block out noises from other markets.
- Our findings indicate that the performance of the modified CDR models consistently falls short when compared to the CMR models. We attribute this discrepancy to two factors: Firstly, the foundational assumption of CDR, which posits that user interests are stable across domains (e.g., users fond of romantic films are likely to enjoy romantic books), is not applicable within the CMR task. CDR models are designed to capture the cross-domain interest consistency, yet such congruence is absent in the CMR task, thereby impeding knowledge transfer for CDR models. Secondly, the lack of domain-specific information in CDR models poses a significant challenge for accurately modeling domain characteristics. Many CDR models, such as CoNet and DDTCDR, utilize domain-specific attributes (like directors of movies or genres of books) as inputs, facilitating the distinction between domains. Conversely, the CMR task often lacks corresponding market-specific features for both items and users, which impairs the models' ability to effectively distinguish between distinct markets.
- We observe that the modified multi-domain learning models, namely Cross-Stitch and MMoE, generally underperform compared to the CMR model. This underperformance may be attributed to the inherent design of multi-domain learning models, which are constructed to simultaneously learn from multiple tasks within various domains by capitalizing on their mutual interdependencies [59]. However, since the CMR task involves user-decoupled recommendation scenarios, the correlation between recommendation tasks across different markets is relatively weak. On the other hand, we notice that Cross-Stitch achieves suboptimal performance on the CA dataset. We posit that this is due to two main reasons. Firstly, the unique attributes of the CA dataset itself must be considered; being the largest among the datasets, it exhibits the least significant data imbalance when serving as the target market. Secondly, the architecture of Cross-Stitch, with its simple yet efficient mechanism for transferring network information, has demonstrated commendable performance in various fields, including computer vision and recommendation systems. It is low coupling to specific scenarios and robust generalization capabilities [22] make it more easy to adaptation within the CMR context.

5.3. Ablation experiments

To gain a deeper comprehension of the effects of market-specific embedding, multiple markets data, and fine-tuning, we perform ablation experiments. The results of these experiments are presented in Fig. 3. Based on the obtained findings, we have the following observations:

- **w/o market-specific emb.** As depicted in Fig. 3, when market-specific embeddings are removed, there is an average performance reduction of 2.27% and 1.61% across two metrics in seven market datasets. Significantly, we note a marked enhancement in the performance of the model in two of the most data-sparse markets, JP and IN, upon the integration of market-specific embeddings (with an average increase of 3.36% for JP and 9.00% for IN across two metrics). Smaller markets are intuitively more susceptible to interference from other markets in CMR due to their lack of sufficient data for fine-tuning pre-trained models. Our proposed market-specific embeddings effectively reduce noise from other markets, thereby preventing negative transfer while still leveraging auxiliary data from those markets for data-sparse markets. Concurrently, we find that the improvement brought by market-specific embeddings is not significant for relatively large markets (FR, CA, MX, and UK), with a slight decrease even observed in the DE market. We posit that this is because data-rich markets can effectively fine-tune the model to fit the target market, and the issue of negative transfer is relatively minor for these markets. Therefore, we conclude that market-specific embeddings are more beneficial for data-sparse markets.
- **w/o multiple markets.** The deep blue bars in Fig. 3 shows the results without training the model with multi-market data. When training only using single-market data, the $Recall@5$, $Recall@10$, $NDCG@5$ and $NDCG@10$ on average are reduced by 30.99%, 27.36%, 32.91% and 34.88%. ACMR trained on multi-market data performs better across all markets in all metrics compared to using single-market data. This observation indicates that ACMR is suitable for the CMR task and is effective in transferring knowledge across markets.
- **w/o fine-tuning.** As the orange bars shown in Fig. 3, ACMR without fine-tuning performs worse on the $Recall@5$, $Recall@10$, $NDCG@5$ and $NDCG@10$ by 11.22%, 7.69%, 15.27% and 17.28%. This observation justifies the need for fine-tuning. Fine-tuning prompts the model to focus on the current market and filter out noises from other markets. We found that fine-tuning has a larger performance improvement for markets with small data volumes and a smaller improvement of about 1% for the largest market, CA. This observation indicates that small markets are more susceptible to interference, and fine-tuning is more necessary for small markets.

5.4. Item representation visualization

To further investigate the functioning of cross-market pre-training and market-specific embeddings, we employ the UMAP [60] algorithm to convert the item representations of ACMR into low dimension and visualize them, as depicted in Fig. 4. Embedding vectors for different markets are drawn in different colors. To reduce randomness caused by parameter initialization, we maintain fixed randomly generated seeds for model parameters. Fig. 4(a) presents the visualization results of item embeddings after respectively training our model on seven datasets, while Fig. 4(b) illustrates the visualization results of item embeddings without market-specific embeddings after pre-training our model on all parallel market data. Furthermore, Fig. 4(c) displays the visualization results of item representations after pre-training our model on all parallel market data. By comparing Fig. 4, we have the following observations:

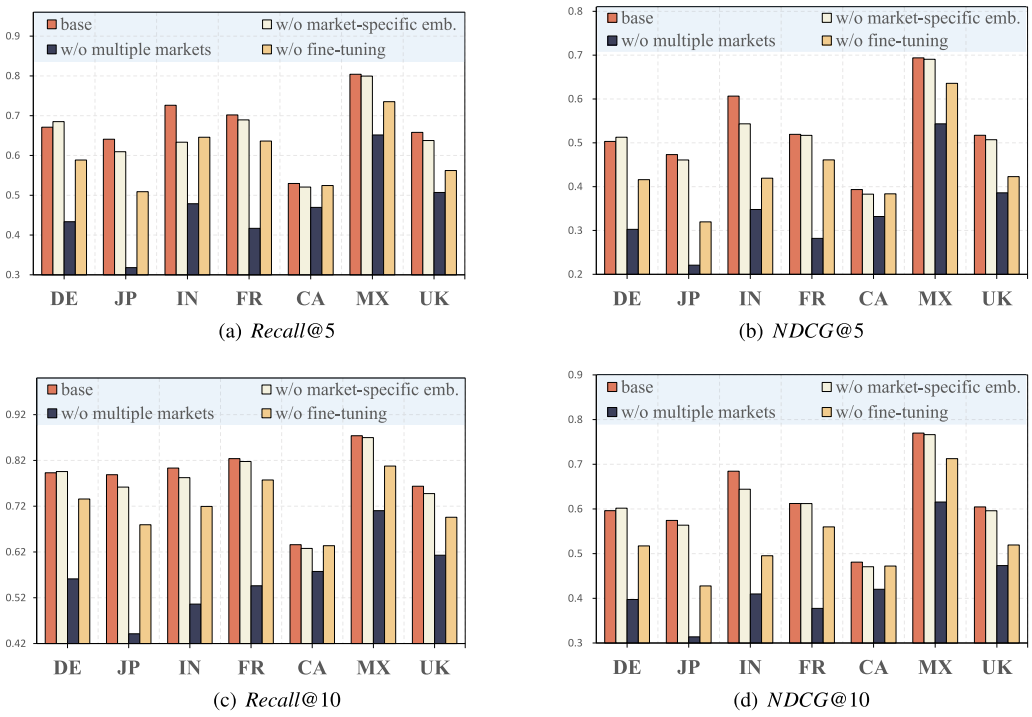


Fig. 3. The results of ablation experiments.

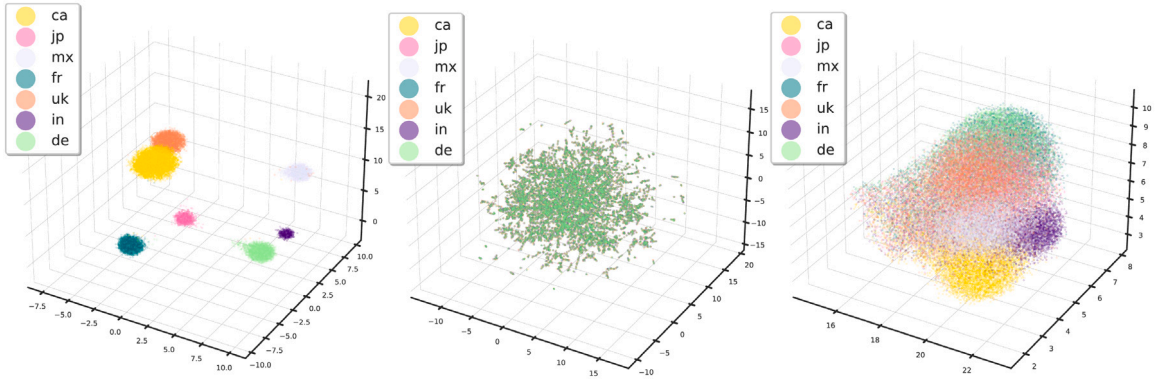


Fig. 4. UMAP visualization of item representations.

- **Shared Vector Space.** As depicted in Figs. 4(b) and 4(c), the item representations disperse within a unified vector space through cross-market training. In contrast, Fig. 4(a) demonstrates that training in separate markets yields item embeddings distributed across distinct vector spaces. Similar to cross-lingual word embedding in the NLP field [61,62], projecting item representations into the same vector space proves advantageous for knowledge transfer. This discovery signifies the effectiveness of our model in capturing item co-occurrence relationships across diverse markets and facilitating efficient knowledge transfer.
- **Modeling Market Bias.** By comparing Figs. 4(b) and 4(c), it is evident that market-specific embedding captures the various biases among different markets. Notably, although both figures present that item representations are distributed in the same vector space, the inclusion of market-specific embeddings results in a more balanced distribution of item representations within that space. Furthermore, market-specific embeddings also enable modeling the similarity between markets, enhancing the model's ability to differentiate the co-occurrence patterns of items across distinct markets.

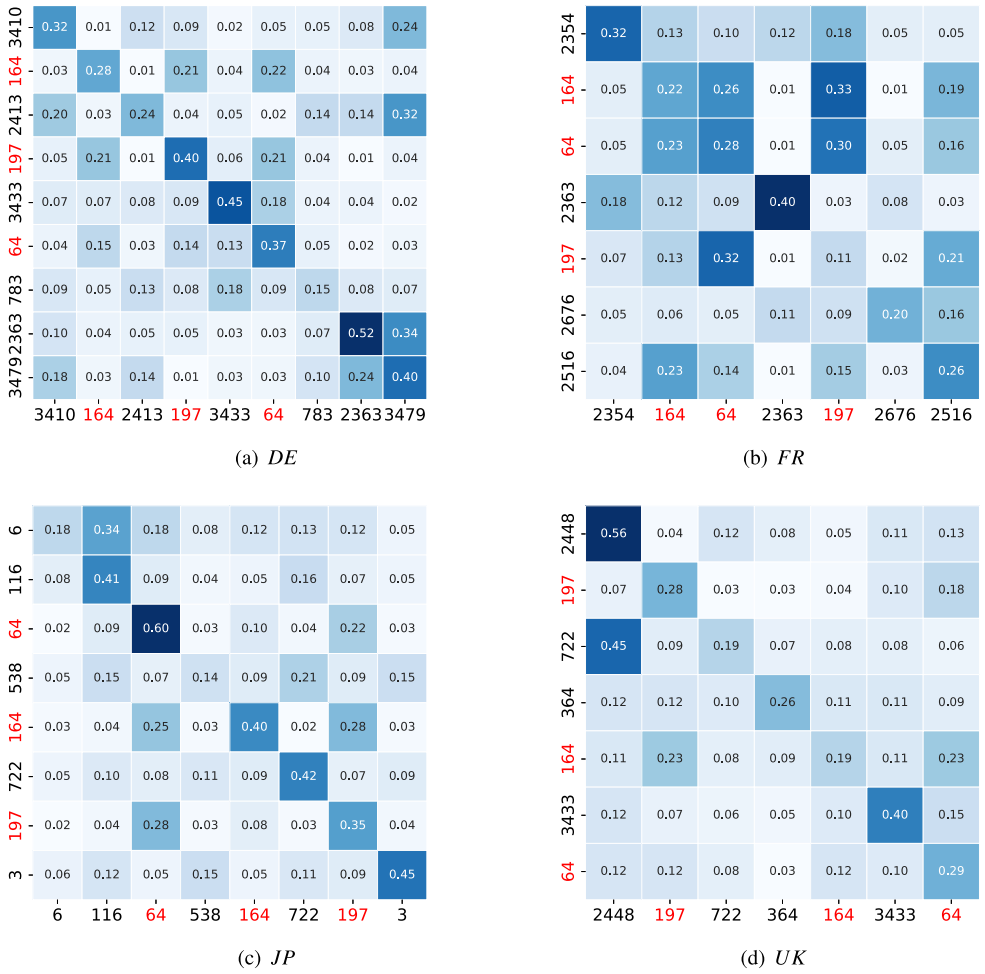
5.5. Item co-occurrences study

In this section, our objective is to explore the efficacy of our proposed method in capturing item co-occurrence patterns across diverse markets. Our investigation is twofold: Initially, we undertake an experiment wherein the pre-training dataset excludes the target market's training set, incorporating only data from other markets. The outcomes of this experiment are detailed in Table 4. Subsequently, we conduct a case study to examine the impact of the attention mechanism in modeling item co-occurrences. For this purpose, we select users.[9] from various markets, all of whom have historical behavior pattern with

[9] Specifically, the selected users are #7811 from DE, #4380 from UK, #2603 from FR, and #5 from JP.

Table 4
The term "Pre-train∗" refers to this process of pre-training on a non-target market dataset. Subsequently, the term "Fine-tune∗" denotes the process of fine-tuning within the target market using parameters derived from the aforementioned pre-training phase.

	DE		JP		IN		FR		CA		MX		UK	
	R@5	N@5	R@5	N@5	R@5	N@5	R@5	N@5	R@5	N@5	R@5	N@5	R@5	N@5
Pre-train∗	0.4602	0.3220	0.3778	0.2469	0.2175	0.1344	0.4678	0.3345	0.1733	0.1106	0.5702	0.4985	0.4477	0.3433
Fine-tune∗	0.6600	0.5014	0.6036	0.4699	0.6359	0.5362	0.6795	0.5147	0.5085	0.3826	0.7310	0.6417	0.6190	0.4951
ACMR	0.6611	0.5032	0.6308	0.4731	0.7163	0.6066	0.6919	0.5194	0.5196	0.3933	0.7941	0.6938	0.6481	0.5172



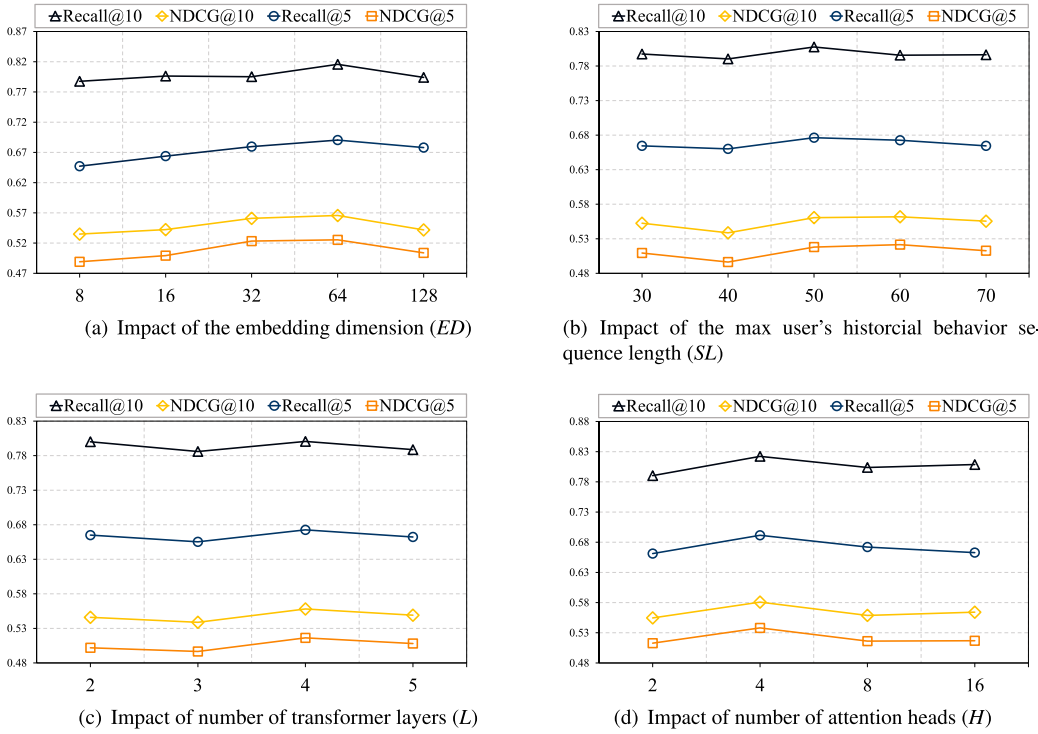
(a) DE (b) FR

(c) JP (d) UK

Fig. 5. Heatmaps of attention coefficients. Darker colors represent higher attention weights. The number on the axis represents the item ID. The item co-occurrence pattern is marked in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

items #64, #164, and #197. We average the attention coefficients of all heads in the last layer of the model and then plot them as heatmaps, as shown in Fig. 5 Drawing on the results of these experiments, we derive several significant observations.

- Observing the row "Pre-train∗" in Table 4, it is evident that our model continues to perform well on the target test set, even when pre-trained on a dataset that lacks the target market data. In some instances, such as DE and UK, the model's performance is comparable to the cross-domain and multi-domain baselines. We attribute this outcome to our model's capability to effectively capture item co-occurrence. Despite being trained exclusively on data from other markets, our model can grasp general knowledge, specifically item co-occurrence, thereby offering a generalization capability on the target market.
- A comparison of the rows "Fine-tune∗" and "ACMR" in Table 4 reveals that "Fine-tune*" delivers competitive performance. "Fine-tune∗" represents the market cold-start scenario, meaning, for a new market, it signifies the outcome of fine-tuning a model pre-trained on data from other markets. This observation illustrates our model's capacity for general knowledge transfer and its ability for swift fine-tuning deployment to new markets in potential real-world scenarios.
- An examination of Fig. 5 shows that the mutual attention coefficients of the general item co-occurrence pattern in these four markets are relatively high. This implies that the attention mechanism identifies the cross-market item co-occurrence through higher attention weights. For instance, in the DE market, the two items with the highest attention coefficient for item #64, besides itself, are #164 and #197. Similar trends can also be observed for FR, JP, and UK. This observation provides an intuitive demonstration of the effectiveness of the attention mechanism in capturing general item co-occurrence patterns.



(a) Impact of the embedding dimension (*ED*)

(b) Impact of the max user's historcial behavior sequence length (*SL*)

(c) Impact of number of transformer layers (*L*)

(d) Impact of number of attention heads (*H*)

Fig. 6. Results of hyperparameter sensitivity experiments.

5.6. Hyperparameter sensitivity experiment

To explore the effect of hyperparameters on model performance, we perform hyperparameter sensitivity experiments, including the embedding dimension (*ED*), the max user's historical behavior sequence length (*SL*), the transformer layer number L and the attention head number H. For simplicity, we only report the results of DE while the situation is similar in other markets. We will analyze the impact of each hyperparameter on model performance in detail below.

- **The impact of *ED*.** The results of the hyperparameter sensitivity experiments for *ED* are presented in Fig. 6(a), which show that the model's performance initially improves and then declines as the dimension increases. However, it is important to note that larger dimensions require more memory and computing time. While high-dimensional embeddings offer stronger representation capabilities, they also pose a higher risk of overfitting. Taking into account the trade-off between time efficiency and model performance, we have decided to utilize 32 dimensions as the embedding dimension.
- **The impact of *SL*.** *SL* refers to the maximum length of the user historical behavior sequence that is fed into the model. The experimental results for different values of *SL* are presented in Fig. 6(b). We conducted experiments with *SL* values of 30, 40, 50, 60, and 70. It was observed that as *SL* increases, the model performance initially improves and then deteriorates. A shorter *SL* may not accurately capture the user's interests, while a longer *SL* increases computational complexity and may introduce noise. Therefore, we believe that the choice of *SL* should consider both the average length of user behavior sequences and time efficiency.
- **The impact of *L*.** The experimental results of hyperparameter sensitivity for L are displayed in Fig. 6(c). It is evident from the results that as the number of transformer blocks increases, the metrics initially improve and then decline. The transformer architecture possesses strong fitting capabilities, but excessive blocks can lead to overfitting issues. Taking into account the balance between performance and time efficiency, we believe that opting for three or four blocks is suitable for the current datasets.
- **The impact of *H*.** The multi-head mechanism enables the model to project embeddings into multiple subspaces, enabling it to focus on different aspects of information. We conducted tests to assess the model's performance and time efficiency with different numbers of heads, H, ranging from 2 to 16. The results, as illustrated in Fig. 6(d), indicate that the model achieves the highest performance when $H = 4$. Increasing the number of heads beyond this value does not enhance performance.

6. Conclusion & future work

In this paper, we proposed a novel model, ACMR, which employs transformer encoder blocks for the CMR task. In order to both utilize cross-market information and eliminate the mutual interference between different markets, we designed market-specific embeddings to model each market. We modified the structure of the transformer block and designed an explicit user modeling component to make it suitable for recommendation tasks. We conducted extensive experiments on commodity datasets from seven countries on three continents. Our model outperforms the second-best model by 4.82%, 6.82%, 3.87%, and 5.34% in terms of four metrics, respectively. The experimental results show that our model is the state-of-the-art CMR model. We conducted ablation experiments and hyperparameter sensitivity tests to analyze the effectiveness of our model and the influence of hyperparameter settings. The experimental results indicate that our model is able to learn the general knowledge of items and effectively transfer information across parallel markets.

In the future, at the model design level, we will explore how to incorporate user-side information (e.g., age, gender, and language) and item-side information (e.g., category, review, and price) into ACMR. We hope that more auxiliary information helps improve the performance and facilitates knowledge transfer. At the market research level, we will further explore how to model the bias of markets in more tiny granularity and find ways to visualize market similarities to improve the interpretability of algorithms.

CRediT authorship contribution statement

Zheng Hu: Writing – original draft, Software, Methodology, Funding acquisition, Conceptualization. **Satoshi Nakagawa:** Visualization, Resources, Data curation. **Shi-Min Cai:** Writing – review & editing, Supervision, Investigation. **Fuji Ren:** Writing – review & editing, Supervision, Project administration. **Jiawen Deng:** Writing – review & editing, Supervision, Data curation.

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.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Grant Nos. T2293771, 61673086) and the Ministry of Education of Humanities and Social Science Project, China (Grant No. 21JZD055) and the Project of Chengdu Science and Technology Bureau (No. 2023-CY02-00003-GX). The funders had no role in the study design, data collection, analysis, decision to publish, or preparation of the manuscript.

References

[1] P. Rita, T. Oliveira, A. Farisa, The impact of e-service quality and customer satisfaction on customer behavior in online shopping, Heliyon 5 (10) (2019) e02690.
[2] D.N. Jutla, P. Bodorik, Y. Zhang, Pecan: An architecture for users' privacy-aware electronic commerce contexts on the semantic web, Inf. Syst. 31 (4–5) (2006) 295–320.
[3] Y. Liu, P. Zhao, X. Liu, M. Wu, L. Duan, X. Li, Learning user dependencies for recommendation, in: IJCAI, ijcai.org, 2017, pp. 2379–2385.
[4] T. Zhou, L. Lü, Y.-C. Zhang, Predicting missing links via local information, Eur. Phys. J. B 71 (2009) 623–630.
[5] J. Liu, T. Zhou, B. Wang, Research progress of personalized recommendation system, Prog. Nat. Sci. 19 (1) (2009) 1–15.
[6] H.R. Bonab, M. Aliannejadi, A. Vardasbi, E. Kanoulas, J. Allan, Cross-market product recommendation, in: CIKM, ACM, 2021, pp. 110–119.
[7] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T. Chua, Neural collaborative filtering, in: WWW, ACM, 2017, pp. 173–182.
[8] J. Cao, X. Cong, T. Liu, B. Wang, Item similarity mining for multi-market recommendation, in: SIGIR, ACM, 2022, pp. 2249–2254.
[9] S.D. Veeramachaneni, A.K. Pujari, V. Padmanabhan, V. Kumar, A hinge-loss based codebook transfer for cross-domain recommendation with non-overlapping data, Inf. Syst. 107 (2022) 102002.
[10] Y. Zhang, X. Ma, S. Wan, H. Abbas, M. Guizani, CrossRec: Cross-domain recommendations based on social big data and cognitive computing, Mob. Netw. Appl. 23 (6) (2018) 1610–1623.
[11] J. Cao, X. Cong, J. Sheng, T. Liu, B. Wang, Contrastive cross-domain sequential recommendation, in: CIKM, ACM, 2022, pp. 138–147.
[12] Y. Zhu, Z. Tang, Y. Liu, F. Zhuang, R. Xie, X. Zhang, L. Lin, Q. He, Personalized transfer of user preferences for cross-domain recommendation, in: WSDM, ACM, 2022, pp. 1507–1515.
[13] J. Cao, J. Sheng, X. Cong, T. Liu, B. Wang, Cross-domain recommendation to cold-start users via variational information bottleneck, in: ICDE, IEEE, 2022, pp. 2209–2223.
[14] H. Xu, C. Li, Y. Zhang, L. Duan, I.W. Tsang, J. Shao, Metacar: Cross-domain meta-augmentation for content-aware recommendation, IEEE Trans. Knowl. Data Eng. 35 (8) (2023) 8199–8212.
[15] G. Hu, Y. Zhang, Q. Yang, Conet: Collaborative cross networks for cross-domain recommendation, in: CIKM, ACM, 2018, pp. 667–676.
[16] P. Li, A. Tuzhilin, DDTCDR: deep dual transfer cross domain recommendation, in: WSDM, ACM, 2020, pp. 331–339.
[17] J. Cao, J. Sheng, X. Cong, T. Liu, B. Wang, Cross-domain recommendation to cold-start users via variational information bottleneck, in: ICDE, IEEE, 2022, pp. 2209–2223.
[18] J. Cao, S. Li, B. Yu, X. Guo, T. Liu, B. Wang, Towards universal cross-domain recommendation, in: WSDM, ACM, 2023, pp. 78–86.
[19] W. Liu, X. Zheng, J. Su, L. Zheng, C. Chen, M. Hu, Contrastive proxy kernel stein path alignment for cross-domain cold-start recommendation, IEEE Trans. Knowl. Data Eng. 35 (11) (2023) 11216–11230.
[20] L. Luo, Y. Li, B. Gao, S. Tang, S. Wang, J. Li, T. Zhu, J. Liu, Z. Li, S. Pan, MAMDR: A model agnostic learning framework for multi-domain recommendation, in: ICDE, IEEE, 2023, pp. 3079–3092.
[21] S. Ruder, An overview of multi-task learning in deep neural networks, 2017, CoRR, arXiv:1706.05098.
[22] I. Misra, A. Shrivastava, A. Gupta, M. Hebert, Cross-stitch networks for multi-task learning, in: CVPR, IEEE Computer Society, 2016, pp. 3994–4003.
[23] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, E.H. Chi, Modeling task relationships in multi-task learning with multi-gate mixture-of-experts, in: KDD, ACM, 2018, pp. 1930–1939.
[24] S. Wang, Y. Li, H. Li, T. Zhu, Z. Li, W. Ou, Multi-task learning with calibrated mixture of insightful experts, in: ICDE, IEEE, 2022, pp. 3307–3319.
[25] X. Sheng, L. Zhao, G. Zhou, X. Ding, B. Dai, Q. Luo, S. Yang, J. Lv, C. Zhang, H. Deng, X. Zhu, One model to serve all: Star topology adaptive recommender for multi-domain CTR prediction, in: CIKM, ACM, 2021, pp. 4104–4113.
[26] S. Bhargav, M. Aliannejadi, E. Kanoulas, Market-aware models for efficient cross-market recommendation, in: ECIR (1), in: Lecture Notes in Computer Science, vol. 13980, Springer, 2023, pp. 134–149.
[27] H. Steck, Embarrassingly shallow autoencoders for sparse data, in: WWW, ACM, 2019, pp. 3251–3257.
[28] A. Grover, J. Leskovec, Node2vec: Scalable feature learning for networks, in: KDD, ACM, 2016, pp. 855–864.
[29] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, K. Gai, Deep interest network for click-through rate prediction, in: KDD, ACM, 2018, pp. 1059–1068.
[30] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, K. Gai, Deep interest evolution network for click-through rate prediction, in: AAAI, AAAI Press, 2019, pp. 5941–5948.
[31] J. Chung, Ç. Gülçehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, CoRR, arXiv:1412.3555.
[32] F. Wang, X. Lu, L. Lyu, CGSNet: Contrastive graph self-attention network for session-based recommendation, Knowl.-Based Syst. 251 (2022) 109282.
[33] R. Shimizu, M. Matsutani, M. Goto, An explainable recommendation framework based on an improved knowledge graph attention network with massive volumes of side information, Knowl.-Based Syst. 239 (2022) 107970.
[34] Y. Chen, Y. Yang, Y. Wang, J. Bai, X. Song, I. King, Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation, in: ICDE, IEEE, 2022, pp. 299–311.
[35] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, P. Jiang, BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer, in: CIKM, ACM, 2019, pp. 1441–1450.
[36] Q. Zhang, J. Li, Q. Jia, C. Wang, J. Zhu, Z. Wang, X. He, UNBERT: user-news matching BERT for news recommendation, in: IJCAI, ijcai.org, 2021, pp. 3356–3362.
[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: NeurIPS, 2017, pp. 5998–6008.
[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, IEEE Computer Society, 2016, pp. 770–778.
[39] L.J. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, 2016, CoRR, arXiv:1607.06450.
[40] Z. Qiu, X. Wu, J. Gao, W. Fan, U-BERT: pre-training user representations for improved recommendation, in: AAAI, AAAI Press, 2021, pp. 4320–4327.
[41] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT (1), Association for Computational Linguistics, 2019, pp. 4171–4186.
[42] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: NeurIPS, 2020.
[43] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, J. Wu, Sequential recommender system based on hierarchical attention networks, in: IJCAI, ijcai.org, 2018, pp. 3926–3932.
[44] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[45] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT-networks, in: EMNLP/IJCNLP (1), Association for Computational Linguistics, 2019, pp. 3980–3990.
[46] M. Zhang, Y. Wu, W. Li, W. Li, Learning universal sentence representations with mean-max attention autoencoder, in: EMNLP, Association for Computational Linguistics, 2018, pp. 4514–4523.
[47] M.D. Ekstrand, J.T. Riedl, J.A. Konstan, et al., Collaborative filtering recommender systems, Found. Trends® Hum.–Comput. Interact. 4 (2) (2011) 81–173.

[48] J. Wang, A.P. de Vries, M.J.T. Reinders, Unifying user-based and item-based collaborative filtering approaches by similarity fusion, in: SIGIR, ACM, 2006, pp. 501–508.
[49] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR (Poster), 2015.
[50] H. Bao, L. Dong, S. Piao, F. Wei, BEiT: BERT pre-training of image transformers, in: ICLR, OpenReview.net, 2022.
[51] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, H. Wu, ERNIE: enhanced representation through knowledge integration, 2019, CoRR, arXiv:1904.09223.
[52] W.L. Taylor, "Cloze procedure": A new tool for measuring readability, Journalism Q. 30 (4) (1953) 415–433.
[53] J. Zou, Y. Chen, E. Kanoulas, Towards question-based recommender systems, in: SIGIR, ACM, 2020, pp. 881–890.
[54] P. Han, S. Shang, A. Sun, P. Zhao, K. Zheng, X. Zhang, Point-of-interest recommendation with global and local context, IEEE Trans. Knowl. Data Eng. 34 (11) (2022) 5484–5495.
[55] H. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, H. Shah, Wide & deep learning for recommender systems, in: DLRS@RecSys, ACM, 2016, pp. 7–10.
[56] Y. Ge, S. Xu, S. Liu, Z. Fu, F. Sun, Y. Zhang, Learning personalized risk preferences for recommendation, in: SIGIR, ACM, 2020, pp. 409–418.
[57] Q. Zhang, F. Ren, Double bayesian pairwise learning for one-class collaborative filtering, Knowl.-Based Syst. 229 (2021) 107339.
[58] P. Li, A. Tuzhilin, DDTCDR: deep dual transfer cross domain recommendation, in: WSDM, ACM, 2020, pp. 331–339.
[59] Y. Li, X. Tian, T. Liu, D. Tao, On better exploring and exploiting task relationships in multitask learning: Joint model and feature learning, IEEE Trans. Neural Netw. Learn. Syst. 29 (5) (2018) 1975–1985.
[60] L. McInnes, J. Healy, UMAP: uniform manifold approximation and projection for dimension reduction, 2018, CoRR, arXiv:1802.03426.
[61] F. Ren, H. Shi, Parallel machine translation: Principles and practice, in: ICECCS, IEEE Computer Society, 2001, pp. 249–259.
[62] M. Artetxe, G. Labaka, E. Agirre, Learning principled bilingual mappings of word embeddings while preserving monolingual invariance, in: EMNLP, The Association for Computational Linguistics, 2016, pp. 2289–2294.