Proxy-Aware Cross-Domain Sequential Recommendation

Shitong Xiao[1], Rui Chen[1], Qilong Han[1], Riwei Lai[1], Hongtao Song[*,1], Li Li[2]
[1]*College of Computer Science and Technology, Harbin Engineering University*
[2]*Department of Electrical and Computer Engineering, University of Delaware*
shitong479@gmail.com, {ruichen, hanqilong, lai, songhongtao}@hrbeu.edu.cn, lilee@udel.edu

2023 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-8867-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/IJCNN54540.2023.10191129

***Abstract*—Cross-domain sequential recommendation (CDSR) aims to predict the next item that a user is most likely to interact with based on past sequential behavior from multiple domains. Existing works on CDSR usually transfer knowledge across different domains by linking items between domains via common users, which suffers from the following limitations: (1) due to the inherent differences between the domains, transferring information across domains can be affected by different representations of related items in different domains. (2) None of existing studies consider the time interval information among items, which is essential in sequential recommendation to capture user intents over time. In this work, we propose a novel cross-domain sequential recommendation model to address the above challenges. Specifically, we first design a shared proxy item encoder to generate a universal representation for each item in all domains by using its textual descriptions. Then, we develop a time-interval-aware attention encoder to represent sequences by considering the time interval information. Moreover, we present a contrastive learning auxiliary task to enhance a cross-domain sequence by weighing the importance of the items in the auxiliary domain with respect to the objective domain. Experiments demonstrate the superiority of our proposed method from various aspects.**

***Index Terms*—Cross-Domain Sequential Recommendation; Contrastive Learning; Universal Representation**

I. Introduction

In the real world, there is a strong correlation and causality between users' behaviors, and sequential recommendation is used to predict what users are most interested in at the next moment by modeling their past interactions. Various methods using recurrent neural networks (RNNs) [1], convolutional neural networks (CNNs) [2], and/or self-attention mechanisms [3] have been proposed to learn good representations of user intents and characterize sequential user-item interactions. However, users' intents may vary over time in practice. It is difficult to capture such changes based on limited historical interactions. For example, from Fig. 1 one can infer that the user is a history and war fan by observing his/her interaction records in the movie domain. However, using the movie domain alone is difficult to generate a recommendation of "Baby Goes Away" for the user, which is an emotional movie.

To solve this challenge, cross-domain sequential recommendation (CDSR) has gained a lot of attention in recent years.

This work was supported by the National Key R&D Program of China under Grant No. 2020YFB1710200, and the National Natural Science Foundation of China under Grant No. 62072136.

*Corresponding author

Fig. 1. An illustration of user interactions in the domains of books and movies.

CDSR collectively captures a user's preferences by combining his/her historical interactions in different domains, which can enrich the data in the target domain for more accurate recommendations. The key challenge is to learn and transfer shared information across domains. The pioneering works are the π-Net [4] and PSJNet [5], which rely on elaborate gating mechanisms to transfer single-domain information. To capture the complex relationships among associated items, DA-GCN [6], TiDA-GCN [7] and MIFN [8] construct an interaction graph (or knowledge graph) between items in different domains to guide information transfer across domains. Some other works, such as RecGURU [9] and UniSRec [10], aim to reduce cross-domain information transfer differences by generating generalized representations.

While these latest efforts lead to encouraging results, we argue that there are still several problems to be considered in order to further enhance model performance: (1) the above methods use common users as a bridge to link items in two domains to achieve better recommendations. However, it is difficult to use such item-level linkages to transfer the underlying user preferences between the two domains. For example, in Fig. 1, the user watched "Genghis Khan" in the movie domain and read "The Fifteenth Year of the Ten Thousand Lives" in the book domain. It is difficult to capture their connection based solely on the item-level linkage. Instead, we can learn from their textual descriptions that they jointly reveal the user's interest in history. (2) For the target domain, some items in the auxiliary domain are very different or even completely irrelevant to a target domain sequence. Adding such auxiliary information without filtering may not be helpful to making recommendations in the target domain, and may even introduce noise that adversely affects recommendation

performance. How to find advantageous information in the auxiliary domain remains a challenge in CDSR. (3) Existing CDSR studies largely overlook temporal information between items, which is an important signal to capture changes of user intents. For example, in Fig. 1, the user's interactions in the book domain in September shows that the user prefers history books, and this dynamic interest also influences his interactions in the movie domain in the following month.

To address the above limitations in CDSR, we propose a proxy-aware cross-domain sequential recommendation model named **P-CDSR**. We first encode each item's textual description using a proxy encoder to obtain a more generalized representation, which is used for cross-domain sequences to transfer shared information across domains by revealing higher-level semantic interconnections. Then, we develop a time-interval-aware attention encoder, where the sequential orders and relative time intervals between any two items are both exploited. Finally, to better understand the correlation between items from different domains in a cross-domain sequence, we propose a contrastive learning auxiliary task to enhance the representation of cross-domain sequences.

We summarize our key contributions as follows:

- We propose a proxy item encoder to encode the textual description of each item in a cross-domain sequence, which helps reveal the higher-level semantic relevance behind the items.
- We develop a time-interval-aware attention encoder to model the time interval information between items.
- We design a contrastive learning auxiliary task to further enhance cross-domain sequence representations.
- We perform extensive experiments on two public datasets and demonstrate that our proposed solution substantially and consistently outperforms a wide range of state-of-the-art competitors.

II. Related Work

A. Sequential Recommendation

Sequential recommendation (SR) generally exploits the temporal order of users' historical interactions to predict the next items with which users are likely to interact. Early SR models commonly rely on Markov chains to model the transition of user-item interactions in a sequence to predict the next interaction. With the development of the neural networks, GRU4Rec [1] first applies Gated Recurrent Units (GRU) to the session-based recommendation. Caser [2] uses CNN to model high-order item-item relationships.

With the Transformer's emergence [11], the self-attention mechanism becomes the mainstream modeling approach for SR. The most classic is the SASRec model [3], which first applies the self-attention mechanism to SR. LSSA [12] proposes a long- and short-term self-attention network to consider both long-term preferences and sequential dynamics. In order to solve the problem of sparse data, CoCoRec [13] and CAFE [14] enhance recommendation performance by fusing category information in SR. Although these studies have made great progress, none of them has considered capturing the feature correlations along the time dimension, which makes them less effective in capturing user evolving preferences. Therefore, there has lot of recent work that has focused on modeling temporal information in sequences. For instance, TiSAS [15] enhances SASRec with the time-interval information, TGSRec [16] jointly models collaborative signals and temporal effects for SR, and TimelyRec [17] proposes two different user preference time patterns and learns them jointly.

B. Cross-Domain Sequential Recommendation

Cross-domain sequential recommendation (CDSR) aims to predict future interactions based on users' historical interactions from multiple domains. It is first studied in π-Net [4], which rely on elaborate gating mechanisms to transfer single-domain information. PSJNet [5] is an extension of π-Net that introduces a parallel split join the scheme to transfer different user intents between domains. CDNST [18] transfers users' novelty-seeking traits learned from his/her historical interactions in the source domain to the target domain. In order to capture the complex relationships among associated items in both domains, DA-GCN [6] first introduced graph neural network to cross-domain sequence recommendation, and MIFN [8] constructs a knowledge graph to guide the connection between items from other domains to transfer information across domains. In addition, $\mathbf{C}^2$DSR [19] proposes a graphical and attentional encoder to exploit both intra- and inter-sequence item relationships in CDSR. Some other works propose a framework that is able to fine-tune a large pre-trained embedding network to adapt to downstream tasks in the target domain, such as UniSRec [10] and PeterRec [20]. However, all the above methods still suffer from the two notable problems discussed previously.

C. Contrastive Learning

Contrastive learning has recently achieved great success in various research domains, including Computer Vision, Natural Language Processing, Recommendation, etc. Its goal is to make the distance between a given anchor point and positive samples as small as possible and the distance to negative samples as large as possible by spatial transformation. Early methods are proposed for the computer vision problems, such as MoCo [21] and SimCLR [22]. For contrastive learning in the sequential recommendation, S^3Rec [23] pre-trains sequential recommendation by contrastive learning with four self-supervised tasks defined on historical items and their attributes. CL4SRec [24] uses three data augmentation methods suitable for sequential recommendation to generate positive samples for contrastive learning. DuoRec [25] proposes a contrast regularization method to enhance the uniformity of the distribution of sequence representations.

III. METHODOLOGY

In this section, we first formulate the CDSR problem, and then present our proposed P-CDSR framework in detail. Its architecture is illustrated in Fig. 2. There are five components

in the framework: (1) an embedding initialization component that initializes item embeddings, their absolute positions, and relative time intervals; (2) a proxy item encoder component, which represents an item by its document and utilizes the pre-trained language model to learn the document embeddings; (3) a time-interval-aware attention encoder, which encodes the representation of each interaction; (4) a contrastive learning module used to filter useful information in auxiliary domains to generate better cross-domain sequence representations and (5) a prediction component conducts item recommendation by a joint training strategy to simultaneously optimize the cross-entropy objective function in both domains.

A. Preliminaries

Let $\mathcal{A} = \{a_1, a_2, \cdots, a_n\}$ and $\mathcal{B} = \{b_1, b_2, \cdots, b_m\}$ be the set of items in domains A and B, where n and m are the total number of items in $\mathcal{A}$ and $\mathcal{B}$, respectively. Let $\mathcal{D} = \{d_{a_1}, \cdots, d_{a_n}, d_{b_1}, \cdots, d_{b_m}\}$ denote the set of item documents (i.e., items' textual descriptions) in both domains, where d_{a_i} and d_{b_j} denote the documents for items a_i and b_j, respectively. In our setting, $\mathcal{S}$ denotes the overall interaction sequence set, and $\mathcal{S}_u$ denotes the interaction sequence set of user u, which includes S_A, S_B, and S_C. In particular, $S_A = [A_1, A_2, \cdots, A_i, \cdots]$ and $S_B = [B_1, B_2, \cdots, B_j, \cdots]$, which denote the interaction sequences of user u in domains A and B, respectively. S_C denotes the cross-domain interaction sequence by merging S_A and S_B in chronological order. Each interaction A_i (or B_j) in the sequence is defined as a triplet. For example, $A_i = (a_i, d_{a_i}, t_i)$, where $a_i \in \mathcal{A}$, $d_{a_i} \in \mathcal{D}$, and t_i is the timestamp when user u interacted with item a_i.

Based on these notations, we are ready to define the CDSR problem. Given S_A, S_B and S_C, the CDSR task recommends the next items with the following probabilities:

$$\mathbf{P}(a_{i+1}|S_A, S_B, S_C) \sim f_A(S_A, S_B, S_C), \tag{1}$$

$$\mathbf{P}(b_{j+1}|S_A, S_B, S_C) \sim f_B(S_A, S_B, S_C), \tag{2}$$

where $\mathbf{P}(a_{i+1}|S_A, S_B, S_C)$ and $\mathbf{P}(b_{j+1}|S_A, S_B, S_C)$ are the probabilities of recommending item a_{i+1} and b_{j+1} for domains A and B under the condition S_A, S_B, and S_C, respectively, and $f_A(S_A, S_B, S_C)$ and $f_B(S_A, S_B, S_C)$ represent the learned functions utilized to estimate $\mathbf{P}(a_{i+1}|S_A, S_B, S_C)$ and $\mathbf{P}(b_{j+1}|S_A, S_B, S_C)$, respectively.

B. Embedding Initialization

In the previous section, we have defined three different interaction sequences for each user. For these three different sequences of items, we build three parameter matrices as embedding look-up tables for embedding initialization: embedding matrix $\mathbf{E}_A \in \mathbb{R}^{n\times d}$ and $\mathbf{E}_B \in \mathbb{R}^{m\times d}$ denote the representations of items in domain A and domain B, respectively, where d is the dimension of the embeddings. The item embedding table for cross-domain sequences will be described in detail in the next section. To identify the order information of a sequence, we define a learnable position embedding matrix $\mathbf{P} \in \mathbb{R}^{l\times d}$, where l is the max length of an interaction sequence.

To further enhance the item representation learning, we incorporate the time interval information into the sequence. Specifically, given an interactive time sequence $[t_1, t_2, \cdots, t_l]$, we use $t_{pq} = |t_p - t_q|$ to denote the time interval between item p and item q. Hence, we have the time interval matrix $\mathbf{T} \in \mathbb{R}^{l\times l}$:

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1l} \\ t_{21} & t_{22} & \cdots & t_{1l} \\ \vdots & \vdots & \vdots & \vdots \\ t_{l1} & t_{l2} & \cdots & t_{ll} \end{bmatrix}. \tag{3}$$

Similarly, each time interval can be represented as an embedding, and the corresponding time interval embedding matrix is denoted by $\mathbf{R} \in \mathbb{R}^{l\times l\times d}$.

C. Proxy Embedding Encoder

UniSRec [10] points out that representing items by explicit IDs greatly limits the expressiveness of items from domain items and is difficult to reveal the user intent behind the items in a cross-domain sequence. In addition, according to CAFE [14] and CoCoRec [13], information that reveals higher-level semantics, such as categories and descriptions of items, can provide insight into the user intent behind the items. Inspired by the above approaches, for an item in the cross-domain sequence, we represent it by its document instead of explicit item IDs. Take item a_i as an example. We denote its document by $d_{a_i} = \{w_1, w_2, w_3, \cdots, w_L\}$, where w_k denotes the k-th word in the document, and L is the maximum document length.

Inspired by the paragraph vector models, we learn the distributed representations for items' documents by constructing language models. We utilize the widely used **BERT** [26] to learn universal text representations to represent items. For item a_i, we use its document d_{a_i} as input:

$$c_{a_i} = \mathbf{BERT}([CLS]; w_1, w_2, \cdots, w_L), \tag{4}$$

where $c_{a_i} \in \mathbb{R}^d$ is the final hidden vector corresponding to the first input token ([CLS]), and ";" denotes the concatenation operation. Following this approach, we get a representation of each item's document. Then we get the embedding table for cross-domain: $\mathbf{E}_C = [c_{a_1}, \cdots, c_{a_n}, c_{b_1}, \cdots, c_{b_m}]$, where $\mathbf{E}_C \in \mathbb{R}^{(n+m)\times d}$.

D. Time-Interval-Aware Attention Encoder

In previous studies, GRU [1] and self-attention [3] are commonly used to encode sequences in cross-domain sequential recommendation. These methods capture the sequentiality of items in sequences, but ignore the fact that different time intervals of items have different effects. For instance, items with more recent timestamps will have more influence on the next item. To further enhance the item representation learning, we incorporate time interval information into the items' representations and devise a time-interval-aware attention encoder. The encoder is composed of multiple layers of self-attention block and point-wise feed-forward network block, and the details are described as follows:

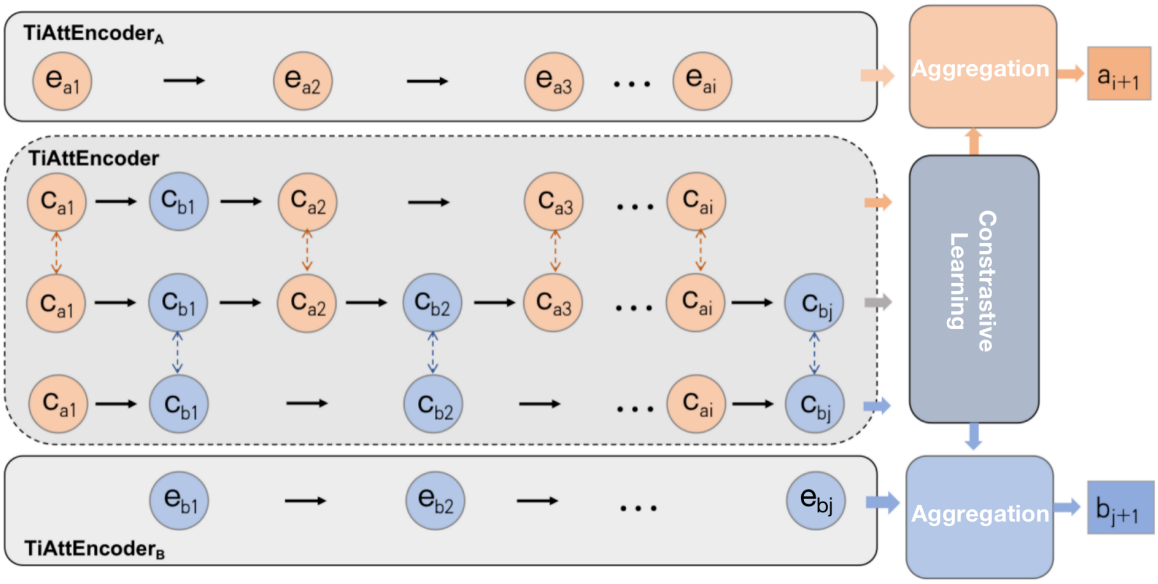


Fig. 2. The overall architecture of our P-CDSR framework.

1) Time-Interval-Aware Attention Layer: For a sequence S, we inject a positional embedding to each item in the sequence to get the initial input embedding $\mathbf{M}^0$ for the attention layer, which is formulated as follows:

$$\mathbf{M}^0 = [e_1 + p_1, e_2 + p_2, \cdots, e_l + p_l]. \tag{5}$$

Then we adopt a scaled dot-product attention [11] to model sequence relationships, which is denoted by f_{att}. Due to the recursive nature of these multiple layers of blocks, we use the xth layer to explain the mechanism. For the xth layer input $\mathbf{M}_p^x$ of item p, its output from the time interval-aware attention encoder is calculated as:

$$\mathbf{Z}_p^x = \sum_{q=1}^{l} f_{att}(\mathbf{M}_p^x \mathbf{W}_Q^x, \mathbf{M}_q^x \mathbf{W}_K^x) \cdot \mathbf{M}_q^x \mathbf{W}_V^x, \tag{6}$$

where the projection matrices $\mathbf{W}_Q^x, \mathbf{W}_K^x, \mathbf{W}_V^x \in \mathbb{R}^{d \times d}$ are learnable parameters. $\mathbf{Z}_p^x \in \mathbb{R}^d$ is the output of item p after the x_{th} attention layer.

Inspired by [14], we apply the effect of temporal information to the sequence by constructing a masking score on f_{att} in Equation 6. Formally, the re-weighted attention weights are calculated by:

$$f_{att}\left(\mathbf{Q}_p, \mathbf{K}_q\right) = \frac{\exp\left(w_{pq}\right) \cdot \theta_{pq}}{\sum_{k=1}^{l} \exp\left(w_{pk}\right) \cdot \theta_{pk}}, w_{pq} = \frac{\mathbf{Q}_p \mathbf{K}_q^T}{\sqrt{d}}, \tag{7}$$

where θ =1 for a standard scaled dot-product attention and $\sqrt{d}$ is a scale factor. We learn θ_{pq} from $\mathbf{M}_p^k$, $\mathbf{M}_q^k$, and the time interval t_{pq} between item p and item q:

$$\theta_{pq} = \left(\mathbf{M}_p^x \mathbf{W}_Q^x + \mathbf{M}_q^x \mathbf{W}_K^x + r_{pq}\right) \mathbf{W}_L^x + \mathbf{b}_L \tag{8}$$

where $\mathbf{W}_L^k \in \mathbb{R}^d$, $\mathbf{b}_L \in \mathbb{R}^1$, and time interval embedding $r_{pq} \in \mathbb{R}^d$ is from the time interval embedding table $\mathbf{R} \in \mathbb{R}^{l \times l \times d}$.

2) Point-Wise Feed-Forward Layer: To prevent overfitting and achieve a stable training process, the next layer $\mathbf{M}^{x+1}$ is generated from $\mathbf{Z}^x$ with residual connections [27], layer normalization [28], and point-wise feed-forward networks [11] as follows:

$$\mathbf{M}^{x+1} = \mathrm{LN}\left(\mathbf{M}^x + \mathrm{FFN}^x\left(\mathbf{Z}^x\right)\right). \tag{9}$$

Then we use function $\mathrm{TiAttEncoder}$ to denote the overall encoding process. For different sequence inputs, we get:

$$\begin{aligned} \mathbf{H}_A &= \mathrm{TiAttEncoder}_A(\mathbf{E}_A), \\ \mathbf{H}_B &= \mathrm{TiAttEncoder}_B(\mathbf{E}_B), \\ \mathbf{H}_C &= \mathrm{TiAttEncoder}_C(\mathbf{E}_C), \end{aligned} \tag{10}$$

where $\mathbf{H}_A \in \mathbb{R}^{l \times d}$, $\mathbf{H}_B \in \mathbb{R}^{l \times d}$, $\mathbf{H}_C \in \mathbb{R}^{l \times d}$ denote sequential output in cross-domain, A and B domains respectively.

E. Contrastive Learning

To better understand the correlations between items from different domains in a cross-domain sequence, we propose a contrastive learning auxiliary task to enhance the representations of cross-domain sequences.

First, to identify which items from domain B are helpful for domain A's recommendations in a cross-domain sequence, we calculate the weights of each other. For domain A, the formulation is as follows:

$$\begin{aligned} \mathbf{\Phi}_B &= softmax(\mathbf{H}_B \cdot \overline{\mathbf{H}}_A), \\ \overline{\mathbf{H}}_A &= \mathrm{mean}(\mathbf{H}_A), \end{aligned} \tag{11}$$

where $\mathbf{H}_A$, $\mathbf{H}_B$ are the output representations of S_A, S_B in Equation 10 respectively. Then for the anchor sequence $S_C = [A_1, B_1, A_2, B_2, \cdots, A_i, B_j]$, an interaction B_i is replaced with the mask [m] if $\phi_{B_i} < 0.5$. The masked sequence is thus: $\widehat{S_C^A} = [A_1, B_1, A_2, [m], \cdots, A_i, [m]]$, and we use $\mathrm{TiAttEncoder}_C$ in Equation 10 to encode the sequence and the output is $\widehat{\mathbf{H}_C^A}$.

Then for the anchor sequence S_C, we divide its output $\mathbf{H}_C$ into two parts by different domains. For example, for domain A, we have:

$$\begin{aligned} h_C^A = \boldsymbol{f}_C^A(\mathbf{H}_C), \quad \widehat{h_C^A} = \boldsymbol{f}_C^A(\widehat{\mathbf{H}_C^A}), \\ \boldsymbol{f}_C^A(\cdot) = \text{Mean}\left(\{\mathbf{H}_{C_k} : C_k \in \mathcal{A}\}\right), \end{aligned} \tag{12}$$

where $\boldsymbol{f}_C^A(\cdot)$ is a function that extracts and calculates the mean value of the output representation corresponding to the items in the sequence belonging to domain A.

Finally, we follow the InfoNCE loss [29] in contrastive learning to construct our auxiliary tasks, which we can see as follow:

$$\mathcal{L}_C^A = -\log \frac{\exp\left(\text{sim}\left(h_C^A, \widehat{h_C^A}\right)\right)}{\exp\left(\text{sim}\left(h_C^A, \widehat{h_C^A}\right)\right) + \sum_{S_{C'}} \exp\left(\text{sim}\left(h_C^A, h_{C'}^A\right)\right)}, \tag{13}$$

where sim(·) is used to calculate the similarity between two vectors, and $\mathcal{S}_{C'}$ denotes a cross-domain sequence of other users within the same batch. Similarly, we can calculate $\mathcal{L}_C^B$ for domain B.

F. Model Training and Evaluation

We consider two types of sequences: single-domain sequences (i.e., S_A and S_B) and cross-domain sequences (i.e., S_C). We design different training objectives for different types of sequences. In particular, we employ the padding technique on the input sequences to simplify the calculation. For example, $S_A = [A_1, [\text{pad}], A_2, [\text{pad}], A_3]$, $S_B = [[\text{pad}], B_1, [\text{pad}], B_2, [\text{pad}]]$, $S_C = [A_1, B_1, A_2, B_2, A_3]$.

1) Single-Domain Training Objective: For single-domain item prediction, we utilize the most common training strategy used in sequential recommendation to optimize our encoder. For domain A, to make use of the auxiliary information of the cross-domain sequence, we sum the representations of the corresponding items in the two sequences and calculate the prediction probabilities for all items in domain A. The formulation is given as follows:

$$\begin{aligned} \mathcal{L}_A &= \sum_k \mathcal{L}_A\left(S_A, k\right), \\ \mathcal{L}_A\left(S_A, k\right) &= -\log \mathbf{P}_A\left(a_{k+1} \mid [A_1, A_2, \cdots, A_k]\right), \\ \mathbf{P}_A\left(a_{k+1} \mid [A_1, A_2, \cdots, A_k]\right) &= softmax(\mathbf{H}_{A_k} W_A + \mathbf{H}_{C_k} W_A), \end{aligned} \tag{14}$$

where $\mathbf{H}_{A_k} \in \mathbb{R}^{1\times d}$ and $\mathbf{H}_{C_k} \in \mathbb{R}^{1\times d}$ are sequential representations at position k in single-domain and cross-domain, $\mathbf{W}_A \in \mathbb{R}^{d\times n}$ is the learnable parameter matrix for prediction.

2) Cross-Domain Training Objective: For cross-domain sequences, unlike single-domain sequences, which are mixed sequences of items from different domains in chronological order, there may be cases where it is dominated by a data-rich domain. So to avoid this situation, we follow C^{2}DSR [19] and propose a balanced objective function as follows:

$$\mathcal{L}_C = \sum_k \mathcal{L}_C(S_C, k), \tag{15}$$

$$\mathcal{L}_C(S,k) = \begin{cases} -\log \mathrm{P}_C^A\left(a_{k+1} \mid [A_1, B_1, A_2, \ldots, B_k]\right), \\ -\log \mathrm{P}_C^B\left(b_{k+1} \mid [A_1, B_1, A_2, \ldots, B_k]\right), \end{cases}$$

$$\mathrm{P}_C^A\left(a_{k+1} \mid [A_1, B_1, A_2, \ldots, B_k]\right) = softmax\left(\mathbf{H}_{C_k}\mathbf{W}_A\right),$$

$$\mathrm{P}_C^B\left(b_{k+1} \mid [A_1, B_1, A_2, \ldots, B_k]\right) = softmax\left(\mathbf{H}_{C_k}\mathbf{W}_B\right),$$

where we decompose the loss function into two equally independent parts, domain A and domain B, for optimization, maintaining predictive power for one domain even if the cross-domain sequences contain multiple consecutive terms in the other domain.

3) Overall Training Objective:

$$\mathcal{L} = \underbrace{\lambda\left(\mathcal{L}_A + \mathcal{L}_B + \mathcal{L}_C\right)}_{\text{Sequential training objective}} + \underbrace{(1-\lambda)\left(\mathcal{L}_C^A + \mathcal{L}_C^B\right)}_{\text{Contrastive learning objective}} \tag{16}$$

where $\mathcal{L}_A, \mathcal{L}_B, \mathcal{L}_C$ denote the sequential recommendation tasks, $\mathcal{L}_C^A, \mathcal{L}_C^B$ denote the contrastive learning auxiliary tasks, and λ denotes a hyperparameter to balance these two types of tasks.

Evaluation. In the evaluation stage, take domain A as an example. We sum the representations of the corresponding items in single-domain and cross-domain sequences to calculate the prediction probability and select the item with the highest prediction score in domain A as the recommended next item:

$$\mathbf{P}_A\left(a_{i+1} \mid S_A, S_B, S_C\right) = \text{softmax}\left(\mathbf{H}_{A_i}\mathbf{W}_A + \mathbf{H}_{A_i}\mathbf{W}_C\right) \tag{17}$$

We can follow a similar idea to generate the next recommendation for domain B.

TABLE I
STATISTICS OF TWO EVALUATION DATASETS.

Scenarios	#Items	#Train	#Valid	#Test	#Avg.length
Food	29,207	34,117	2,722	2,747	9.91
Kitchen	34,886		5,451	5,659	

IV. EXPERIMENTS

In this section, we introduce our experimental setup and present the empirical results. Our experiments are designed to answer the following research questions:

- **RQ1:** How does P-CDSR perform compared with other state-of-the-art methods?
- **RQ2:** How do different components in P-CDSR contribute to model performance?
- **RQ3:** How does the key hyperparameter λ affect P-CDSR's performance?

A. Experimental Setup

1) Datasets: The experiments are conducted on the *Amazon*[1] e-commerce collection, which includes millions of customers and products as well as rich metadata such as reviews,

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

TABLE II
THE PERFORMANCE(%) COMPARISON WITH BASELINES. ALL IMPROVEMENTS ARE SIGNIFICANT WITH p-VALUE < 0.05 BASED ON PAIRED t-TESTS.

Method	Food-domain recommendation						Kitchen-domain recommendation					
	MRR	NDCG		HR			MRR	NDCG		HR		
		@5	@10	@1	@5	@10		@5	@10	@1	@5	@10
BPRMF	4.10	3.55	4.03	2.42	4.51	5.95	2.01	1.45	1.85	0.73	2.18	3.43
ItemKNN	3.92	3.51	3.97	2.41	4.59	5.98	1.89	1.28	1.75	0.58	1.99	3.26
NCF-MLP	4.49	3.94	4.51	2.68	5.10	6.86	2.18	1.57	2.03	0.91	2.23	3.65
CoNet	4.13	3.61	4.14	2.42	4.77	6.35	2.17	1.50	2.11	0.95	2.07	3.71
GRU4Rec	5.79	5.48	6.13	3.63	7.12	9.11	3.06	2.55	3.10	1.61	3.50	5.22
SASRec	7.30	6.90	7.79	4.73	8.92	11.68	3.79	3.35	3.93	1.92	4.78	6.62
TiSASRec	7.88	7.23	8.24	5.07	9.36	12.44	3.98	3.41	4.09	2.11	4.88	6.84
π-Net	7.68	7.32	8.13	5.25	9.25	11.75	3.53	2.98	3.73	1.57	4.34	6.67
MIFN	8.55	8.28	9.01	6.02	10.43	12.71	4.09	3.57	4.29	2.21	4.86	7.08
C^2DSR	8.91	8.65	9.71	5.84	11.24	14.54	4.65	4.16	4.94	2.51	5.74	8.18
P-CDSR	**9.87**	**9.57**	**10.72**	**6.66**	**12.34**	**15.94**	**4.78**	**4.37**	**5.08**	**2.69**	**6.06**	**8.27**
Improv.	10.77%	10.64%	10.4%	14.04%	9.79%	9.63%	2.8%	5.05%	2.83%	7.17%	5.57%	1.1%

product descriptions, and product categories. In our experiments, we choose a pair of complementary domains, namely the "Food-Kitchen" domains. To satisfy the characteristics of CDSR, we extract the users who have interactions in both domains. To guarantee that each user/item has enough interactions, we filter out the users and items with less than 10 interactions. To extract item documents, we follow the three-step process. First, we extract text descriptions (e.g., category, title) for each item from its metadata. Then, we concatenate the words from the text description to form a topic string. Finally, stopwords and duplicate words are removed from the topic string. We follow the settings used in $\mathbf{C}^2$DSR [19] to process the data for a fair comparison. For each dataset, the users' latest interaction sequences are equally divided into the validation/test sets, and the remaining interaction sequences are used as the training set. The statistics of the processed datasets are shown in Table I.

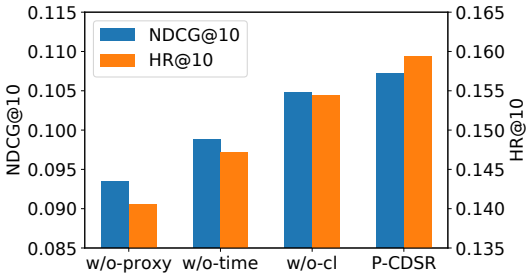
2) Evaluation Metrics: Based on previous studies, we report the performance using the leave-one-out method. We guarantee unbiased evaluations according to Rendles [30] and calculate 1,000 scores (including 999 negative samples and 1 positive sample) for each validation/test case. Then, we report the top-K recommendation performance of the 1,000 ranking lists in terms of mean reciprocal rank (MRR), normalized discounted cumulative gain (NDCG@$\{5, 10\}$), and hit ratio (HR@$\{1, 5, 10\}$).

B. Baseline Algorithms

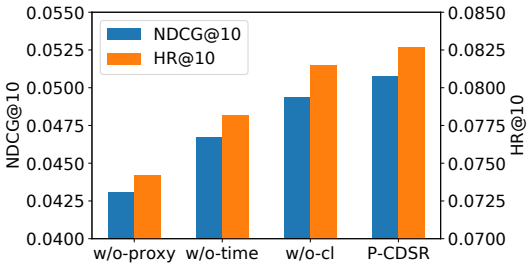
To demonstrate the effectiveness of our solution, we compare it with four classes of baseline methods, including traditional recommendation methods, sequential recommendation methods, cross-domain recommendation methods, and cross-domain sequential recommendation methods.

- **BPRMF** [31] is a matrix-factorization-based method optimized by the BPR loss.
- **Item-KNN** [32] is based on the idea of KNN, which recommends the next item based on the most recent items the user interacted with.
- **GRU4Rec** [1] is the first method to use GRU to capture the temporal relationships in sequential recommendation.
- **SASRec** [3] is a self-attention-based sequential recommendation model used as our backbone network.
- **TiSASRec** [15] is a sequential recommender based on a self-attention mechanism considering time interval information.
- **NCF-MLP** [33] uses the classic neural collaborative filtering (NCF) model for cross-domain recommendations, which learns user/item representations by two separate MLP networks with a shared initialized user embedding layer.
- **CoNet** [34] proposes a novel transfer learning approach for cross-domain recommendation using neural networks as the base model.
- **π-Net** [4] is the first to propose the cross-domain sequential recommendation task and to use gated recurrent networks to capture the information flow across different domains.
- **MIFN** [8] propose a mixed information flow network for CDSR to consider both the flow of behavioral information and the flow of knowledge extracted from the constructed knowledge graph.
- **$\mathbf{C}^2$DSR** [19] is a cross-domain sequential recommendation model that includes a graphical and attentional encoder to exploit both intra- and inter-sequence item relationships.

1) Implementation Details: For a fair comparison, our model keeps the same hyperparameter setting as $\mathbf{C}^2$DSR [19]. The embedding size is fixed to 64, the dropout rate is fixed to 0.3, the L_2 regularizer coefficient is selected from $\{0.0001, 0.00005, 0.00001\}$, the learning rate is selected from $\{0.001, 0.0005, 0.0001\}$, and the training epoch is fixed to 100 to get the best result. Besides, the trade-off factor λ is selected from 0.1 to 1 with step length 0.1. We initialize the model parameters randomly using the Xavier method. We take Adam as our optimization method. We implement our model in PyTorch. The hyperparameters of all baseline algorithms are

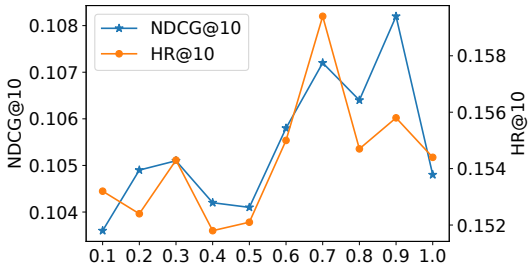


(a) Amazon-Food

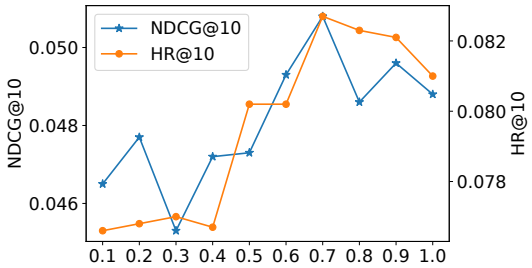


(b) Amazon-Kitchen

Fig. 3. Impact of different components on P-CDSR.



(a) Effect of λ on Amazon-Food



(b) Effect of λ on Amazon-Kitchen

Fig. 4. Impact of the hyperparameter λ on P-CDSR.

carefully tuned by grid search. All experiments were run on a workstation with an Intel Xeon Platinum 2.40GHz CPU, an NVIDIA RTX 8000 GPU, and 500GB RAM. Our code is publically available at https://github.com/XXX-1310/P-CDSR.

C. Comparison with Baselines (RQ1)

We report the main results in Table II, where the best results are boldfaced, and the second-best results are underlined. From the results, we have the following main observations.

- For the sequential recommendation baselines, we can observe that the models based on self-attention outperform the GRU-based models. In addition, the result from TiSASRec shows that time interval information has a positive effect on model performance.
- While both π-Net and GRU4Rec use GRU as the basic framework, π-Net achieves better performance, which suggests that identifying the differences between domains and transferring this knowledge across domains can improve the recommendation performance.
- Our solution significantly outperforms all baselines on all metrics. We attribute such improvements to a few reasons. First, using the generalized representations of items' documents as a proxy to bridge between the two domains can better facilitate the interaction between the two domains to reveal users' true preferences. Second, the introduction of the contrastive learning task to enhance cross-domain sequences' representations can provide advantageous information to improve the recommendation performance in both domains.

D. Impact of Different Components (RQ2)

To verify the benefits of different components in P-CDSR, we conduct an ablation study with several variants:

- **w/o-proxy** represents items in cross-domain sequences based on explicit item IDs.
- **w/o-time** leverages standard self-attention to model sequentiality without any additional temporal information.
- **w/o-cl** removes the contrastive learning loss from P-CDSR and only utilizes recommendation loss.

Fig. 3 shows the performance of the different variants. It can be seen that replacing explicit IDs with item documents to represent items improves the performance by a significant margin. We deem that this is because representing items in a cross-domain sequence through the shared proxy encoder helps transfer shared information across domains by revealing higher-level semantic interconnections. We can also observe that augmenting cross-domain sequence representations via contrastive learning and incorporating time interval information into sequence representations are all beneficial to model performance.

E. Impact of Key Hyperparameter (RQ3)

In this section, we study the impact of the key hyperparameter λ on model performance. The hyperparameter λ is introduced as a trade-off parameter to balance recommendation and contrastive learning tasks. We examine different λ values from 0.1 to 1 with step length 0.1, where $\lambda = 1$ means using just the recommendation tasks for training. The experimental results are shown in Fig. 4. We can observe that our model performs worst when setting $\lambda = 0.1$ and best when $\lambda = 0.7$, indicating that the auxiliary task can positively contribute to model performance.

V. Conclusion

In this work, we proposed a novel P-CDSR model for cross-domain sequential recommendation. To transfer shared

information across domains, we designed the proxy embedding encoder to generate generalized representations for items by their documents from all domains. We also put forward a temporal-interval-aware attention encoder to incorporate temporal interval information with the sequential order. Furthermore, we introduced a contrastive learning auxiliary task to enhance a cross-domain sequence by bringing its related items closer together. We conducted a comprehensive experimental study to show that our P-CDSR framework consistently and significantly outperforms a large number of state-of-the-art competitors on two public benchmark datasets.

REFERENCES

[1] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016.

[2] J. Tang and 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)*, 2018.

[3] W. Kang and J. J. McAuley, "Self-attentive sequential recommendation," in *Proceeding of the 18th IEEE International Conference on Data Mining (ICDM)*, 2018.

[4] M. Ma, P. Ren, Y. Lin, Z. Chen, J. Ma, and M. de 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)*, 2019.

[5] W. Sun, Y. L. Pengjie Ren, M. Ma, Z. Chen, Z. Ren, J. Ma, and M. de Rijke, "Parallel split-join networks for shared account cross-domain sequential recommendations," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2022.

[6] L. Guo, L. Tang, T. Chen, L. Zhu, Q. V. H. Nguyen, and H. Yin, "DA-GCN: A domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[7] L. Guo, J. Zhang, L. Tang, T. Chen, L. Zhu, and H. Yin, "Time interval-enhanced graph neural network for shared-account cross-domain sequential recommendation," *IEEE Transactions on Neural Networks and Learning Systems (TNN)*, 2022.

[8] M. Ma, P. Ren, Z. Chen, Z. Ren, L. Zhao, P. Liu, J. Ma, and M. de Rijke, "Mixed information flow for cross-domain sequential recommendations.," *ACM Trans. Knowl. Discov. Data (TKDD)*, 2022.

[9] C. Li, M. Zhao, H. Zhang, C. Yu, L. Cheng, G. Shu, B. Kong, and 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)*, 2021.

[10] Y. Hou, S. Mu, W. X. Zhao, Y. Li, B. Ding, and J. Wen, "Towards universal sequence representation learning for recommender systems," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2022.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31th conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[12] C. Xu, J. Feng, P. Zhao, F. Zhuang, D. Wang, Y. Liu, and V. S. Sheng, "Long- and short-term self-attention network for sequential recommendation," *Neurocomputing*, 2021.

[13] R. Cai, J. Wu, A. San, C. Wang, and H. Wang, "Category-aware collaborative sequential recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2021.

[14] J. Li, T. Zhao, J. Li, J. Chan, C. Faloutsos, G. Karypis, S. Pantel, and J. J. McAuley, "Coarse-to-fine sparse sequential recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2022.

[15] J. Li, Y. Wang, and J. J. McAuley, "Time interval aware self-attention for sequential recommendation," in *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*, 2020.

[16] Z. Fan, Z. Liu, J. Zhang, Y. Xiong, L. Zheng, and P. S. Yu, "Continuous-time sequential recommendation with temporal graph collaborative transformer," in *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)*, 2021.

[17] J. Cho, D. Hyun, S. Kang, and H. Yu, "Learning heterogeneous temporal patterns of user preference for timely recommendation," in *Proceedings of the 30th International Conference on World Wide Web (WWW)*, 2021.

[18] F. Zhuang, Y. Zhou, H. Ying, F. Zhang, X. Ao, X. Xie, Q. He, and H. Xiong, "Sequential recommendation via cross-domain novelty seeking trait mining," *Journal of Computer Science & Technology (J. Comput. Sci. Technol.)*, 2020.

[19] J. Cao, X. Cong, J. Sheng, T. Liu, and B. Wang, "Contrastive cross-domain sequential recommendation," in *Proceedings of the 31st ACM International Conference on Information& Knowledge Management (CIKM)*, 2022.

[20] F. Yuan, X. He, A. Karatzoglou, and L. Zhang, "Parameter-efficient transfer from sequential behaviors for user modeling and recommendation," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR)*, 2020.

[21] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[22] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[23] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J. Wen, "S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, 2020.

[24] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, J. Zhang, B. Ding, and B. Cui, "Contrastive learning for sequential recommendation," in *Proceedings of 38th IEEE International Conference on Data Engineering (ICDE)*, 2022.

[25] R. Qiu, Z. Huang, H. Yin, and Z. Wang, "Contrastive learning for representation degeneration problem in sequential recommendation," in *Proceedings of 15th ACM International Conference on Web Search and Data Mining (WSDM)*, 2022.

[26] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, 2016.

[29] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, 2018.

[30] W. Krichene and S. Rendle, "On sampled metrics for item recommendation.," in *Proceedings of the The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2020.

[31] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.

[32] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the Tenth International World Wide Web Conference (WWW)*, 2001.

[33] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web (WWW)*, 2017.

[34] G. Hu, Y. Zhang, and Q. Yang, "Conet: Collaborative cross networks for cross-domain recommendation," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*, 2018.