Transfer Learning in Cross-Domain Sequential Recommendation

Zitao Xu[1] Weike Pan[1]* Zhong Ming[1,2,3]

[1]College of Computer Science and Software Engineering, Shenzhen University

[2]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ)

[3]College of Big Data and Internet, Shenzhen Technology University

Problem Definition

- **Input**:
  - A target-domain behavior sequence $\mathcal{V}_u = \{v_1, v_2, \ldots, v_t, \ldots, v_L\}$ (ordered by the interaction time) for each user $u \in \mathcal{U}$.
  - N source-domain behavior sequences, e.g., for the n-th source domain, $\mathcal{V}_u^{S_n} = \{v_1^{S_n}, v_2^{S_n}, \ldots, v_t^{S_n}, \ldots, v_L^{S_n}\}$ (ordered by the interaction time) for the same user u.
- **Goal**: Predict the next possible preferred item for each user u in the target domain according to $\mathcal{V}_u$ and $\mathcal{V}_u^{S_n}$ where $1 \leqslant n \leqslant N$.

Motivation

The limitation of existing cross-domain sequential recommendation (CDSR) methods:

- There are relatively limited works on CDSR, and most methods rely on RNNs which have limited capability in capturing the complex associations between domains.
- Most existing methods capture the users' preferences within one single domain, neglecting the item transition patterns across sequences from different domains.
- Existing methods often focus on the associations of one single source domain to a certain target domain.

Overall of Our Solution

- We study a new and important problem, i.e., cross-domain sequential recommendation, and propose a novel solution named transfer via joint attentive preference learning (TJAPL).
- Specifically, we tackle the studied problem from the perspective of transfer learning and attentive preference learning (APL).

Advantages of Our Solution

- We treat the self-attention sequential recommendation (SASRec) model [Kang and McAuley, 2018] as target-domain attentive preference learning (TD-APL) to model the users' behavior sequences and capture their dynamic preferences in the target domain.
- We propose cross-domain user attentive preference learning (CD-UAPL) to share and transfer the users' overall preferences from more than one source domains to the target domain, leveraging the sequential behaviors from the source domains to address the scarcity problem.
- We also propose cross-domain local attentive preference learning (CD-LAPL) to capture the item transition patterns across sequences from different domains and generate the users' cross-domain local attentive preferences.

Notations (1/2)

Table: Some Notations and explanations (cont.).

Symbol	Explanation
$\mathcal{U}$	user set
$\mathcal{I}$	item set for the target domain
N	number of source domains
$\mathcal{I}^{S_n}$	S_n represents the n-th source domain; $\mathcal{I}^{S_n}$ is the item set for the n-th source domain
u	user $u \in \mathcal{U}$
v_i	the item that user interacted with at time step i in the target domain
$v_j^{S_n}$	the item that user interacted with at time step j in the n-th source domain
L	maximum sequence length
B	number of attention blocks
$\mathcal{V} = \{v_1, v_2, \ldots, v_L\}$	user's interaction sequence in the target domain
$\mathcal{V}_t = \{v_1, v_2, \ldots, v_t\}$	truncated item sequence at time step t with regard to sequence $\mathcal{V}$

Notations (2/2)

Table: Some Notations and explanations (cont.).

Symbol	Explanation
$\mathcal{V}_{t'}^{S_n} = \{v_1^{S_n}, v_2^{S_n}, \ldots, v_{t'}^{S_n}\}$	truncated item sequence at time step t' for the n-th source domain
d	latent vector dimensionality
$\boldsymbol{u}$, $\boldsymbol{V}$, $\boldsymbol{V}^{S_n}$	embedding associate with u, $\mathcal{V}_t$, $\mathcal{V}_{t'}^{S_n}$
$\boldsymbol{p}_t$	position embedding at time step t
$\boldsymbol{f}_t$	target-domain attentive preference at time step t
$\boldsymbol{f}_t^u$	cross-domain user attentive preference at time step t
$\boldsymbol{f}_t^{S_n}$	cross-domain local attentive preference at time step t
$\boldsymbol{o}_t$	final representation of the user's preference at time step t
$r_{t,i}$	preference score of item i at time step t

TJAPL (1/3)

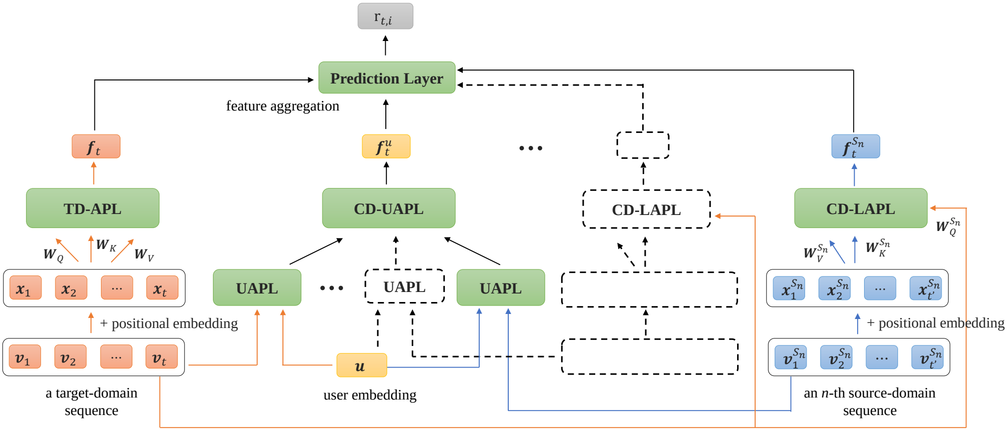


Figure: The framework of our proposed TJAPL (transfer via joint attentive preference learning).

TJAPL (2/3)

- TD-APL (target-domain APL) is fed with the embedding of a target domain sequence, which contains some self-attention blocks (see Eqs.(3$\sim$7)).
- CD-UAPL (cross-domain user APL) extracts a user's overall preference in all domains, where each domain includes a user attention layer (see Eqs.(8$\sim$11)) to capture the user preferences in the corresponding domain.

TJAPL (3/3)

- CD-LAPL (cross-domain local APL) is fed with the embedding of a target-domain sequence and a source-domain sequence which consists of cross-domain attention blocks (see Eqs.(14$\sim$16)). Notice that each source domain contains its own CD-LAPL.
- These modules are all based on attention mechanism thus can accelerate the training by parallel computation. Moreover, it can be applied to scenarios with more than one source domains.

Target-Domain Attentive Preference Learning (1/3)

- We employ attention mechanism [Vaswani et al., 2017, Kang and McAuley, 2018] to explore the sequential patterns in the target domain.
- A learnable position embedding $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_L\} \in \mathbb{R}^{L \times d}$ is added to the sequence embedding $\boldsymbol{V}$ and $\boldsymbol{V}^{S_n}$, then we obtain the position-aware input embedding $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t\}$ and $\boldsymbol{X}^{S_n} = \{\boldsymbol{x}_1^{S_n}, \boldsymbol{x}_2^{S_n}, \ldots, \boldsymbol{x}_{t'}^{S_n}\}$,

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

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

Target-Domain Attentive Preference Learning (2/3)

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

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

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

- It refers to using the item which was interacted with at the last time step to match those items a user interacted with before, then obtain the item weighting information to generate the information used for prediction at the next time step, i.e., $\boldsymbol{h}_t \in \mathbb{R}^d$.

Target-Domain Attentive Preference Learning (3/3)

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

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

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

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

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

- Finally, we take the final output vector $\boldsymbol{f}_t^{(b)} \in \mathbb{R}^d$ from the top SAB as the target-domain attentive preference, which represents the current interests of user at time step t in the target domain.

Cross-Domain User Attentive Preference Learning (1/4)

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

Cross-Domain User Attentive Preference Learning (2/4)

- We take the learnable vector $\boldsymbol{u} \in \mathbb{R}^d$ (i.e., the embedding of user u) as the query in the attention layer, which means that the query is the same for the user u regardless of which time step t the current interaction is at. This is beneficial for personalized recommendation, because each user has his/her own embedding vector.

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

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

- Notice that we abandon the position information $\boldsymbol{P}$ which is also the difference between Eq.(8) and Eq.(3) in the attention layer besides the query condition. This is because the long-term preference is not sensitive to the position information of the interactions compared to the short-term dynamic preference [Lin et al., 2020].

Cross-Domain User Attentive Preference Learning (3/4)

- Considering that a same user usually has similar preferences beneath his or her behaviors in different domains, similarly, we formalize the user attentive preference in the *n*-th source domain as follows:

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

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

Cross-Domain User Attentive Preference Learning (4/4)

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

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

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

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

Cross-Domain Local Attentive Preference Learning (1/3)

- Considering that the next interaction of user in the target domain may be related to an item he/she recently interacted with in a certain source domain, we propose CD-LAPL to exploit the user's sequential information in multiple domains then transfer knowledge across different domains.
- We adapt the attention block to measure the importance of a user's previous interactions in a source domain to current interaction in the target domain, and explore the transition patterns across sequences from different domains.

Cross-Domain Local Attentive Preference Learning (2/3)

- Specifically, we denote the input embedding of the target-domain item at the last time step $\boldsymbol{v}_t$ as query, and denote the position-aware input embedding of the n-th source-domain sequence $\boldsymbol{X}^{S_n}$ as key and value, and then the cross-domain attention layer can be formalized as follows:

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

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

- We also employ a two-layer $FFN(\cdot)$ to further improving the model performance:

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

Cross-Domain Local Attentive Preference Learning (3/3)

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

Prediction Layer (1/2)

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

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

  where $\boldsymbol{o} \in \mathbb{R}^{(2+N)d}$ denotes the concatenation of all the output vectors.

- Then, the concatenation vector is fed into an MLP to obtain the final representation of the user's preference:

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

  where $\boldsymbol{o}_t \in \mathbb{R}^d$ denotes the final representation of the user's preference.

Prediction Layer (2/2)

- Finally, the prediction score of item i can be calculated as follows:

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

We adopt Adam as the optimizer [Kingma and Ba, 2015] and the binary cross-entropy loss function for our TJAPL can be formalized as:

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

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

Algorithm 1: The learning procedure of TJAPL

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

Research Questions

- **RQ1:** What's the performance of our proposed TJAPL as compared with the state-of-the-art methods?
- **RQ2:** How does our TJAPL perform when using different source domains? Is it beneficial to the model if we increase the number of source domains?
- **RQ3:** Does our TJAPL alleviate the data sparsity issue?
- **RQ4:** What's the influence of various components in our TJAPL?
- **RQ5:** How does the key parameters affect the performance of our TJAPL?
- **RQ6:** What's the impact of different feature aggregation methods in our TJAPL?

Datasets (1/3)

- We conduct experiments on Amazon[1], which is a review data collected by [McAuley et al., 2015] from the eponymous e-commerce platform.
- The Amazon data contains overlapped users in multiple domains, which make it suitable for the study of CDSR compared with other datasets.
- We choose three datasets with different categories, i.e., "Movie", "CD" and "Book" from the Amazon data.

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

Datasets (2/3)

According to the setting in [Kang and McAuley, 2018, Lin et al., 2020], we preprocess the datasets as follows:

- (1) We suppose that the presence of review, check-in and purchase behaviors are positive feedback (i.e., a user interacted with an item) and use the timestamps to determine the order of the interactions.
- (2) We only keep the users and items with no fewer than five related interactions. And we discard later duplicated (user, item) pairs.
- (3) We only keep the sequence of a user who has interactions in all the three domains.
- (4) We adopt the leave-one-out evaluation by splitting each sequence into three parts, i.e., the last interaction for test, the penultimate interaction for validation and the remaining interactions for training.

Datasets (3/3)

Table: Statistical details of the datasets.

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

Evaluation Metrics

- We adopt two common ranking-based metrics:
  - HR@10 (hit ratio) refers to the proportion of the ground-truth items appearing in the top-10 recommended lists.
  - NDCG@10 (normalized discounted cumulative gain) is sensitive to the exact ranking positions of the items in the lists.
- Following the common strategy in [Kang and McAuley, 2018], we sample 100 negative items as candidates to avoid heavy computation on all the (user, item) pairs. These 100 negative items have not been interacted with by the corresponding users and are sampled according to their popularity to ensure that they are informative and representative [Lin et al., 2020].

Baselines (1/2)

- Two general recommendation methods:
  - BPRMF [Rendle et al., 2009]. A traditional model which optimizes a matrix factorization task using a pairwise ranking loss.
  - CoNet [Hu et al., 2018]. A neural transfer learning model which enables dual information transfer across domains by developing cross-connection units on MLPs.
- Six sequential recommendation methods:
  - FPMC [Rendle et al., 2010]. A classic method that combines matrix factorization and Markov chains to model the sequential pattern.
  - GRU4Rec [Hidasi et al., 2016]. An RNN-based method which explores the item dependencies over sequences by adopting GRUs.
  - GRU4Rec+ [Hidasi and Karatzoglou, 2018]. An improved model based on GRU4Rec which adopts a new loss function and an sampling strategy.
  - Caser [Tang and Wang, 2018]. A CNN-based model which employs horizontal and vertical convolutional filters to model the sequences.
  - GCSAN [Xu et al., 2019]. A GNN-based model which constructs directed graphs for the sequences and applies gated GNNs to obtain all node vectors involved in the session graphs.

Baselines (2/2)

- - SASRec [Kang and McAuley, 2018]. An attention based model that employs the attention mechanism to capture the dynamic preferences.
- Five cross-domain sequential recommendation methods:
  - π-Net [Ma et al., 2019]. An RNN-based model which devises a cross-domain transfer unit to extract and share the user information across different domains at each timestamp.
  - DA-GCN [Guo et al., 2021]. A GNN-based model which employs graph convolution networks to learn the complicated interaction relationships and the structural information in a cross-domain sequence graph.
  - CD-SASRec [Alharbi and Caragea, 2022]. An improved method based on SASRec [Kang and McAuley, 2018] which fuses the source-domain aggregated vector into the target-domain item embedding to transfer information across domains.
  - RecGURU [Li et al., 2022]. It employs a self-attentive autoencoder to derive latent user representations, and proposes an adversarial learning method to unify user embeddings generated from different domains into a single global generalized user representation, which captures the overall preferences of users.
  - C^2DSR [Cao et al., 2022]. A novel model which adopts a graphical and attentional encoder to capture the item relationships, and devises two sequential objectives with a contrastive objective to jointly learn the single-domain and cross-domain user representations.

Implementation Details

- For the general setting, we set the latent dimensionality d to 50, the mini-batch size to 128, the dropout rate to 0.5, the learning rate to 0.001 and the maximum length of a sequence L to 100.
- For the methods with Transformer architectures (i.e., SASRec, CD-SASRec and our TJAPL), we adopt single-head attention layers and two attention blocks.
- For the GNN-based methods (i.e., GCSAN and DA-GCN), we set the depth of the GNN layer to 2.
- For the shared-account recommendation methods (i.e., π-Net and DA-GCN), the latent user number is set to 1.
- For other key parameters, we reference the suggestions of the corresponding papers or tune them on the validation data.

- For cross-domain recommendation methods, we only report the performance of the best-performing model with the corresponding source domain. For our proposed TJAPL, since it can be applied to a multi-domain scenario, we report the results of simultaneously utilizing two source domains.

Overall Performance Comparison (1/3)

Table: Recommendation performance of our TJAPL and the baselines.

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

Overall Performance Comparison (2/3)

Form Table 4, we have the following observations:

- Our proposed TJAPL outperforms all the baselines on all the three datasets, and gains 9.46% NDCG@10 and 8.43% HR@10 improvements on average against the strongest baseline, which demonstrates the capability of our TJAPL to model the sequential information with cross-domain data.
- The sequential recommendation methods outperform the general recommendation baseline, which indicates the importance of extracting sequential information from users' behavior.
- The cross-domain sequential recommendation methods outperform most traditional sequential recommendation methods, which demonstrates the significance of taking into account the cross-domain information.
- The attention-based models achieve outstanding performances in both sequential recommendation and cross-domain sequential recommendation, which demonstrates the superiority of the attention mechanisms in modeling dynamic preference.

Overall Performance Comparison (3/3)

- The “Movie” dataset has the most significant improvement, which probably because the “Movie” dataset is more tightly related to the other domains, i.e., a user’s interaction sequences in the “Book” and “CD” domains are likely to influence his/her next interaction in the “Movie” domain, so knowledge transfer is more effective.
- The cross-domain sequential recommendation methods achieve relatively small improvements on the “Book” dataset, since the source domain (“Movie” or “CD”) is sparser (as is shown in Table 3). And our TJAPL can still achieve superior performance on the “Book” dataset because it can utilize both the “Movie” domain and the “CD” domain as source domains simultaneously, which demonstrates the effectiveness of knowledge transfer across multiple domains.

Influence of Source Domains (1/3)

The recommendation performance of our TJAPL with different source domains is shown in Table 5. Notice that "Both" means leveraging both the other two domains for knowledge transfer and preference learning.

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

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

Influence of Source Domains (2/3)

Form Table 5, we have the following observations:

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

Influence of Source Domains (3/3)

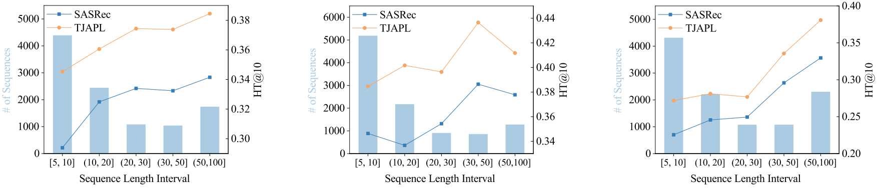
- For the “Book” domain, it achieves the best performance when using the “Movie” domain as the source domain. The reason is that the “Movie” domain may be more tightly related to the “Book” domain and therefore performs better. Besides, leveraging the “CD” domain as the source domain also performs better than leveraging both domains. The reason may be that for a user, data across multiple source domains is not always related, in which case the introduction of extra information and noise would make it less efficient than leveraging a single domain.
- Our TJAPL still outperforms all the baselines on all the datasets (as shown in Table 4) when only leverages one single domain for knowledge transfer, which demonstrates the stability and the superiority of our TJAPL.

Performance Analysis w.r.t. Sparsity

In this subsection, we conduct two experiments to verify the effectiveness of introducing cross-domain information to alleviate the data sparsity problem.

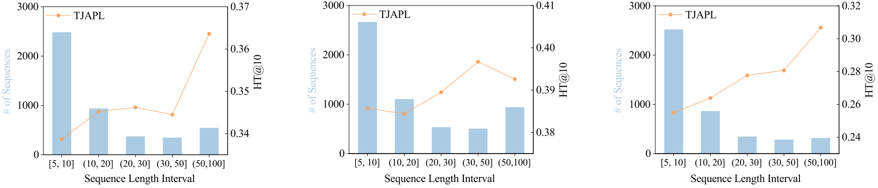
- (1) We divide users into groups according to their behavior sequence length in the target domain, and identify the reasons of improvement by comparing performance of SASRec and TJAPL in different user groups.
- (2) We divide the users into groups according to their behavior sequence length in the source domain while fixing the target-domain sequence length interval, and study how the source-domain sequence length affects the recommendation performance.

Performance Analysis w.r.t. Sparsity



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

Figure: Performance of different target-domain sequence lengths.



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

Figure: Performance of different source-domain sequence lengths.

Performance w.r.t. Target-Domain Sequence Length

Form Fig. 2, we have the following observations:

- The interaction data of the majority of users is sparse in the target domain. The group with the shortest sequence length contains the most users on all the datasets, and the number of users decreases as the sequence length interval of the group gets longer.
- TJAPL achieves significant improvement on users within short sequence length intervals, with relative largest improvement ranging from 17.48% to 20.57% on all the datasets. That's because the shorter users' sequence lengths indicate the sparser their interaction data, in which case the traditional single-domain method (i.e., SASRec) struggles to adequately capture users' preferences. In contrast, the introduction of the rich source-domain data can enhance users' preferences, and the knowledge transfer across domains seem to be more effective in this situation.
- Our TJAPL achieves better performance than SASRec on most user groups, which confirms the superiority of our TJAPL in sequential recommendation.

Performance w.r.t. Source-Domain Sequence Length

Form Fig. 3, We have the following observations:

- Similar to the target domain, user group with the shortest source-domain sequences length interval contains the most users in all the datasets, and the the number of users decrease as the sequences become longer.
- The recommendation performance in the target domain generally improves as the sequence length in the source domain increases. This is reasonable since the model can capture user's preference better in the source domain with more interaction data, so as to transfer a more comprehensive user's preference to the target domain.

Ablation Study (1/4)

We conduct an ablation study to evaluate the contribution of different components of our TJAPL.

- We report the results of “Movie” and “CD” when leveraging the “Book” domain for knowledge transfer, and leveraging the “Movie” domain for “Book”.
- We compare the separate effect of TD-APL (i.e., SASRec, denoted as ‘T’) with the joint effects that additionally add CD-UAPL (denoted as ‘U’) and CD-LAPL (denoted as ‘C’).
- We also examine the effects of different domains on CD-UAPL, i.e., target-domain user attentive preference learning (denoted as ‘U1’) and source-domain user attentive preference learning (denoted as ‘U2’).
- Moreover, we compare the joint effects of all the combination approaches.

Ablation Study (2/4)

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

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

Ablation Study (3/4)

Form Table 6, we have the following observations:

- **‘T + U’ vs. T.** The integrated model with the addition of CD-UAPL always significantly outperforms the separate one, which demonstrates the importance of capturing the cross-domain user attentive preference and indicates the effectiveness of our CD-UAPL.
- **‘T + U’ vs. ‘T + U1’ or ‘T + U2’.** CD-UAPL is considered as the combination of the target-domain and source-domain user attentive preference learning modules. We can find that ‘T + U1’ is generally more effective than ‘T + U2’ (except on “Movie”) which means that users tend to generate the corresponding user preferences by applying their own target-domain data when it is sufficient. Furthermore, ‘T + U’ achieves the best overall performance, which indicatesthe benefit of combining the target-domain and source-domain user attentive preference.

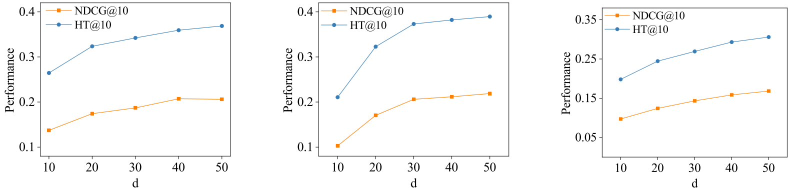
Ablation Study (4/4)

- **'T + C' vs. T.** Without CD-LAPL (i.e., 'C'), we find that the performance is much worse. It confirms that this module can learn the cross-domain local attentive preference from the recent interactions of the target and source domain, which indicates the significance of capturing the transition patterns across sequences from different domains.
- **'T + C + U' vs. 'T + U' or 'T + C'.** We can see that almost all the best results are from 'T + C + U', which demonstrates the complementarity of these three parts. It captures the local attentive preference and user attentive preference from both the target and source domains, balancing these representations and improving the effect for sequential recommendation.

Influence of Hyper-parameters (1/3)

- In this subsection, we explore the influence of two hyper-parameters (i.e., the latent dimensionality d and the number of attention blocks B) on the model performance.

Influence of Hyper-parameters (2/3)

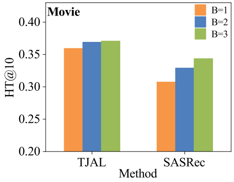


(a) Movie (b) CD (c) Book

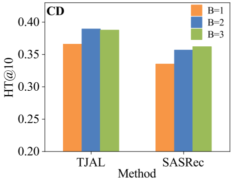
Figure: Performance of different dimensionalities d on three datasets ($B = 2$).

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

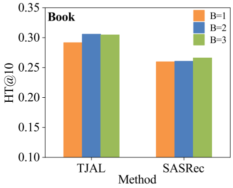
Influence of Hyper-parameters (3/3)



(a) Movie



(b) CD



(c) Book

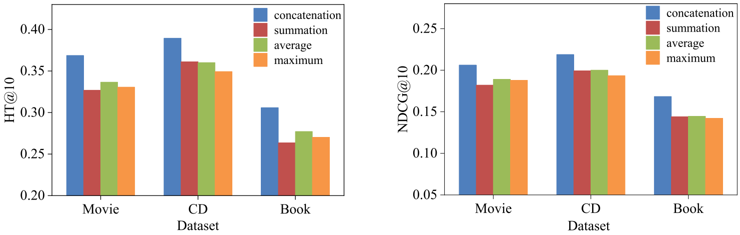
Figure: Performance of different numbers of blocks B ($d = 50$) for SASRec and our TJAPL.

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

Aggregation Methods Comparison (1/2)

- In this subsection, we discuss the effects of different designs for feature aggregation in Eq.(17). As stated in Section 3.5, we employ concatenation to aggregate the features (i.e., target-domain attentive preference, cross-domain user attentive preference, and cross-domain local attentive preference) in the prediction layer. We replace the method of feature aggregation with summation, average and maximum, to examine their performance.

Aggregation Methods Comparison (2/2)



(a) HT@10 (b) NDCG@10

Figure: Performance of different feature aggregation methods.

- When employing concatenation to aggregate the features, our TJAPL achieves the best performance, while average performs better than summation and maximum (except on CD). It confirms that concatenation can effectively balance the information to aggregate all the features.

Conclusions

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

Future Work

- We aim to apply our TJAPL to scenes of cross-domain or cross-organization privacy-aware federated recommendation [Lin et al., 2023], which can reduce the risk of privacy leakage from the introduction of rich source-domain data

Thank you!

- We thank the support of National Natural Science Foundation of China (Nos. 62172283 and 62272315), and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515010122).
- If you have any questions, please feel free to contact us.

Alharbi, N. and Caragea, D. (2022).
Cross-domain self-attentive sequential recommendations.
In *Proceedings of International Conference on Data Science and Applications*, ICONDATA '22, pages 601–614.

Cao, J., Cong, X., Sheng, J., Liu, T., and Wang, B. (2022).
Contrastive cross-domain sequential recommendation.
In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, pages 138–147.

Guo, L., Tang, L., Chen, T., Zhu, L., Nguyen, Q. V. H., and Yin, H. (2021).
DA-GCN: A domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation.
In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, IJCAI '21, pages 2483–2489.

Hidasi, B. and Karatzoglou, A. (2018).
Recurrent neural networks with top-k gains for session-based recommendations.
In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 843–852.

Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. (2016).
Session-based recommendations with recurrent neural networks.
In *Proceedings of the 4th International Conference on Learning Representations*, ICLR '16.

Hu, G., Zhang, Y., and Yang, Q. (2018).
Conet: Collaborative cross networks for cross-domain recommendation.
In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 667–676.

Kang, W. and McAuley, J. J. (2018).
Self-attentive sequential recommendation.
In *Proceedings of the 18th IEEE International Conference on Data Mining*, ICDM '18, pages 197–206.

Kingma, D. P. and Ba, J. (2015).
Adam: A method for stochastic optimization.
In *Proceedings of the 3rd International Conference on Learning Representations*, ICLR '15.

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

Lin, J., Pan, W., and Ming, Z. (2020).
Fissa: Fusing item similarity models with self-attention networks for sequential recommendation.
In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, pages 130–139.

Lin, Z., Pan, W., Yang, Q., and Ming, Z. (2023).
Recommendation framework via fake marks and secret sharing.
ACM Transactions on Information Systems, 41(2):40:1–40:37.

Ma, M., Ren, P., Lin, Y., Chen, Z., Ma, J., and Rijke, M. d. (2019).
π-net: A parallel information-sharing network for shared-account cross-domain sequential recommendations.
In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '19, pages 685–694.

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

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009).
BPR: Bayesian personalized ranking from implicit feedback.
In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461.

Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2010).
Factorizing personalized markov chains for next-basket recommendation.
In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 811–820.

Tang, J. and Wang, K. (2018).
Personalized top-N sequential recommendation via convolutional sequence embedding.
In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 565–573.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017).
Attention is all you need.
In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS '17, pages 6000–6010.

Xu, C., Zhao, P., Liu, Y., Sheng, V. S., Xu, J., Zhuang, F., Fang, J., and Zhou, X. (2019).
Graph contextualized self-attention network for session-based recommendation.
In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI '19, pages 3940–3946.