



Multi-domain sequential recommendation via multi-sequence and multi-task learning

Liwei Pan ^a, Weike Pan ^a,*, Zhong Ming ^{b,c}

^a College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

^b Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China

^c College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China



ARTICLE INFO

Keywords:

Multi-domain sequential recommendation
Multi-sequence
Multi-task learning

ABSTRACT

In recent years, more and more researchers and practitioners have focused on multi-domain CTR prediction and achieved great success. Though users' behaviors often exhibit sequentiality, little effort has been made on multi-domain sequential recommendation (MDSR). Most existing works on MDSR sort the interactions from all domains in chronological order and then predict the next interacted items in each domain. However, they neglect separate interaction sequences in each domain. Therefore, they cannot exploit the commonalities and differences among different domains well. Cross-domain sequential recommendation (CDSR) models are usually designed for performance improvement in one target domain rather than in each domain. Although extending a CDSR model to an MDSR one directly or indirectly is feasible, it will result in high time complexity. Meanwhile, they often ignore data imbalance across different domains, which might cause negative transfer.

As a response, we propose a novel MDSR solution called multi-sequence multi-task learning (MML). Our MML consists of three modules, including hybrid-domain sequential preference learning (HSPL), intra-domain sequential preference learning (ISPL) and multi-task learning & prediction (MLP). Specifically, HSPL aims to learn hybrid-domain sequential preferences. Meanwhile, we construct augmented sequences and leverage contrastive learning to learn more unbiased hybrid-domain sequential preferences for alleviating negative transfer. ISPL is designed to capture intra-domain sequential preferences. In the MLP module, three specific tasks and a behavior regularizer are leveraged to ensure that each module can learn the corresponding preferences sufficiently and enhance knowledge transfer among different domains. We conduct extensive experiments on some public datasets using different backbone models and show that our MML is able to achieve significantly better performance than the state-of-the-art methods in two or more domains in most cases. Meanwhile, our MML can achieve the same time complexity as the MDSR models only using hybrid interaction sequences. The source code can be found at <https://github.com/plw2019/MML>.

1. Introduction

Recommender systems play an increasingly crucial role in users' daily life. They are often designed to alleviate the problem of information overload and provide personalized services for each user. Most conventional recommendation methods mainly focus on

* Corresponding author.

E-mail addresses: panliwei2023@email.szu.edu.cn (L. Pan), panweike@email.szu.edu.cn (W. Pan), mingz@email.szu.edu.cn (Z. Ming).

user behavior data in one single domain (Hidasi, Karatzoglou, Baltrunas, & Tikk, 2016; Kang & McAuley, 2018; Lin, Pan, & Ming, 2020; Tang et al., 2025; Zhu et al., 2024). They ignore useful information from other domains, which may hurt the performance. To leverage knowledge from other domains, there are two main categories of methods, i.e., cross-domain sequential recommendation methods and multi-domain recommendation methods.

In recent years, **cross-domain sequential recommendation methods** (Lin et al., 2024; Park et al., 2024; Zhang, Duan, Guo, Xu, & Wang, 2023) have attracted more researchers' attention. By transferring knowledge from a source domain, these methods can alleviate the commonly existing cold-start and data sparsity problems in a target domain. For example, C²DSR (Cao, Cong, Sheng, Liu, & Wang, 2022) and Tri-CDR (Ma et al., 2024) leverage contrastive learning to transfer useful knowledge, and demonstrate excellent performance. However, there are still some limitations in these works. **Firstly**, they are all based on RNNs or Transformer rather than the state-of-the-art Mamba (Gu & Dao, 2023), which achieves a good balance between efficiency and effectiveness. **Secondly**, CDSR models usually predict the next item in the target domain by combining sequential preferences in the source and target domains via relatively complex attention mechanisms or gating networks. Meanwhile, in order to capture the sequential dependencies across two domains and avoid data leakage, for each predicted item in the target domain, a CDSR model needs to process a separate interaction sequence containing only the interactions in the source domain earlier than the predicted item (Xu, Pan & Ming, 2023). Considering these two points, transferring knowledge in most existing CDSR models might consume lots of computational resources. To solve these two problems, we propose a generic framework based on Mamba, which remains compatible with RNNs and Transformer. Moreover, we propose a novel and efficient method by leveraging sequential preferences in one domain to predict preferred items belonging to other domains so as to achieve knowledge transfer. **Thirdly**, they cannot be applied directly to multi-domain sequential recommendation. There are two approaches that we can utilize to extend a CDSR model to an MDSR one: (1) We can build a CDSR model for two domains. If there exist m domains, we then need to build at least $m(m - 1)/2$ pairs of CDSR models, which might cause a huge burden in model training and deployment. (2) We can also build extra encoders to capture sequential preferences from interaction sequences following the framework of C²DSR. However, due to the split operation of C²DSR as shown in Fig. 3, the encoders might consume lots of computational time when the hybrid interaction sequences are long. **Fourthly**, they often ignore the problem of data imbalance about the hybrid interaction sequences merged from different domains. They are designed to mainly build correlations between the hybrid interaction sequences in all domains and the isolated interaction sequences in each domain without considering the problem of data imbalance (Cao et al., 2022; Ma et al., 2024).

As a result, the learned hybrid-domain sequential preferences might be dominated by data-rich domains. Data imbalance mainly refers to two aspects: (1) The number of interacted items is different for each domain in a hybrid interaction sequence. By modeling the hybrid interaction sequence, the model might fail to capture preferences in a domain with sparse data. (2) The distribution of the time when the items are interacted with is different for each domain. For example, in a hybrid interaction sequence, the recently interacted items may belong to domain 1, while the previously interacted items may belong to domain 2. It will result in failing to capture the preferences in domain 2. If the biased preferences are transferred to the target domain (Li et al., 2023), it might cause negative transfer. There is an example as shown in Fig. 1. For user 1, a CDSR model might fail to learn sequential preferences in Book domain. It is because user 1 interacts with more items in Movie domain and user 1 recently interacts with items in Movie domain. Therefore, the CDSR model might fail to predict the target item in Book domain. For user 2, a CDSR model might fail to capture sequential preferences in Movie domain. Though user 2 interacts with the same number of items in Book and Movie domains, the most recently interacted items fall in Book domain. Therefore, the CDSR model might fail to predict items in Movie domain. To solve the third and fourth problems, we propose a specific split operation to reduce time consumption. Meanwhile, we construct some augmented hybrid interaction sequences for each user. By contrastive learning between the original sequences and the augmented ones, unbiased hybrid-domain sequential preferences can be obtained.

Recently, **multi-domain recommendation methods** (Fu et al., 2023; Ju et al., 2024; Tian et al., 2023; Wang et al., 2024; Zhao, Gao, Xu, & Chen, 2023) have attracted increasing attention from both researchers and practitioners. They aim to improve recommendation performance in multiple domains simultaneously (Jia et al., 2024; Li et al., 2023). The past few years have witnessed remarkable success in this field. **Firstly**, there are some multi-domain recommendation methods without using sequential information. For example, STAR (Sheng et al., 2021) leverages a domain-shared network and some domain-specific networks to learn domain-shared and domain-specific preferences, respectively. CAT-ART (Li, Xie et al., 2023) and HGDR (Liu et al., 2024) leverage matrix factorization and GNNs to learn domain-shared and domain-specific preferences for multi-domain recommendation, respectively. However, these methods only focus on CTR prediction and fail to capture useful sequential patterns. **Secondly**, multi-domain sequential recommendation considers users' interaction sequences and useful information from multiple domains simultaneously, which remains to be fully studied. There are three main challenges for multi-domain sequential recommendation: (1) How to obtain unbiased hybrid-domain sequential preferences is a crucial but often overlooked problem. (2) How to efficiently model both a hybrid interaction sequence from all domains and a separate interaction sequence in each domain poses a challenging problem. (3) How to transfer knowledge across different domains efficiently requires further investigation. The first proposed multi-domain sequential model, termed as DS-HRNN (Kim et al., 2019), utilizes hybrid interaction sequences as input (Hwang, Ju, Kang, Jang, & Yu, 2024; Sun et al., 2024). To solve disconnection in the same domain, DS-HRNN (Kim et al., 2019) adopts a supplementary loss and a behavior regularizer. However, it ignores the separate interaction sequences in each domain, which might result in failing to learn the intra-domain sequential preferences sufficiently. LLM-REC (Tang et al., 2023; Vats, Jain, Raja, & Chadha, 2024) and UPSR (Qu et al., 2023) apply large language models to MDSR. However, they both ignore the separate interaction sequences in each domain. Meanwhile, they use text descriptions rather than ID to represent an item, which may not capture the collaborative behavior information well. And, they all directly model the original hybrid interaction sequences from all domains. The data imbalance of the hybrid interaction sequences hinders accurate sequential preference learning. TJAPL (Xu, Pan, & Ming, 2024) transfers knowledge

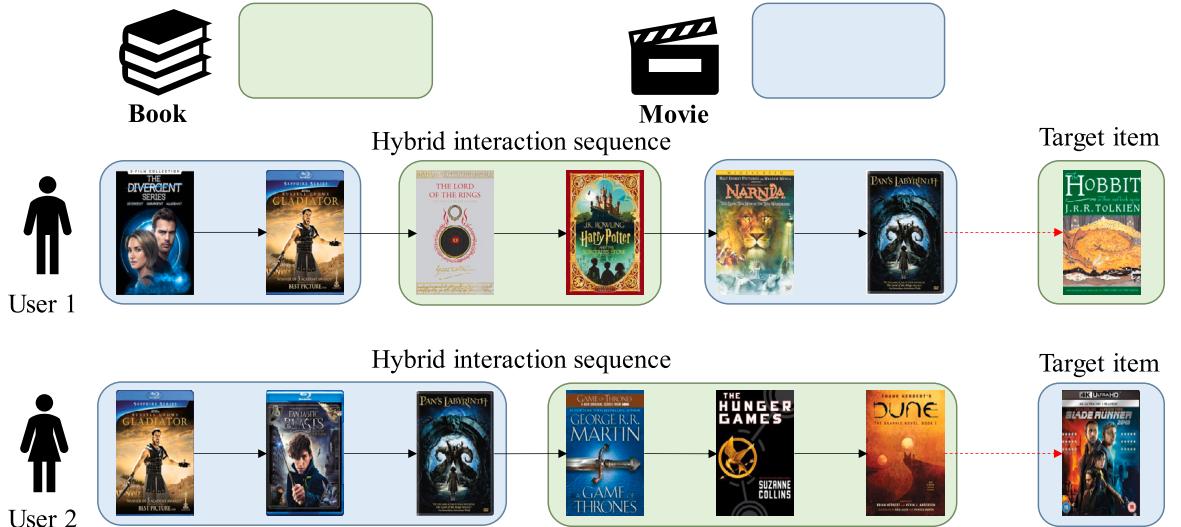


Fig. 1. Data imbalance in hybrid interaction sequences.

from multiple source domains to a target domain. However, TJAPL ignores the hybrid interaction sequences in all domains. Note that M³Rec (Chen, Qian, Li, & Lin, 2021) explores multiple tasks and multiple sequences in preference learning, but is not applicable to MDSR.

To address aforementioned three challenges in MDSR, we propose a novel solution, i.e., multi-sequence multi-task learning (MML). Our MML leverages contrastive learning to capture unbiased hybrid-domain sequential preferences for solving the first challenge. To solve the second challenge, our MML adopts a specific split operation and employs an efficient-yet-effective sequential model (i.e., Mamba4Rec) as a backbone model. Meanwhile, to solve the third challenge, our MML leverages a novel inter-domain item prediction task (i.e., **Task 3**) to transfer knowledge across different domains without increasing time complexity and consuming lots of computational resources. We conduct extensive experiments on some public datasets, where the results clearly demonstrate the superiority of our MML.

The main contributions and technical novelties of our MML are summarized as follows:

- We design two modules to capture hybrid-domain and intra-domain sequential preferences. Moreover, we design a specific split operation to model multiple interaction sequences efficiently. In order to solve the data imbalance problem, we leverage contrastive learning between the original interaction sequences and the augmented ones to obtain unbiased hybrid-domain sequential preferences.
- To learn the two kinds of sequential preferences sufficiently, we leverage three different kinds of tasks and a behavior regularizer. Specifically, **Tasks 1** and **2** ensure that hybrid-domain sequential preferences and intra-domain sequential preferences are fully learned, respectively. **Task 3** aims to transfer knowledge across different domains.
- Experimental results demonstrate the superiority of our generic, model-agnostic framework MML under three different backbone models (i.e., GRU4Rec, SASRec and Mamba4Rec).

2. Related work

2.1. Cross-domain sequential recommendation

Cross-domain sequential recommendation aims to predict the next interacted items by modeling users' historical interaction sequences from two domains (Chen, Xu, Pan, Yang, & Ming, 2024; Ding, Li, Chen, & Shou, 2023; Guo, Wang, Wang, Zhu, & Yin, 2023), which can be categorized into single-target and dual-target cross-domain sequential recommendation. The former aims to improve the recommendation performance in a target domain with the help of a source domain containing rich data. CD-SASRec (Alharbi & Caragea, 2022) transfers knowledge to a target domain by fusing a user's representational source-domain vector into item embeddings in the target domain. MGCL (Xu, Pan et al., 2023) designs a multi-view contrastive learning approach to transfer knowledge from a source domain to a target domain. Tri-CDR (Ma et al., 2024) designs a uniform framework to capture the correlations among source, target and hybrid interaction sequences. The latter aims to improve the recommendation performance in the target and source domains simultaneously. CDHRM (Wang, Guo, Chu, Hwang, & Feng, 2020) devises one cross-domain user-level and two intra-domain session-level RNNs to capture users' global interests and intra-session dynamics, respectively.

Table 1

Summary of the frequently used notations in the paper.

\mathcal{U}	User set of all domains.
\mathcal{V}	Item set of all domains.
\mathcal{D}	Domain set.
\mathcal{U}^d	User set of domain d .
\mathcal{V}^d	Item set of domain d .
S_u^H	User u 's hybrid interaction sequence of all domains.
\bar{S}_u^H	User u 's augmented hybrid interaction sequence of all domains.
S_u^d	User u 's interaction sequence of domain d .
\mathbf{M}_{hd}	The item embedding matrix containing hybrid-domain knowledge.
\mathbf{M}_{ta}	The item embedding matrix containing intra-domain knowledge.
L^h	The maximum length of hybrid interaction sequences of all domains.
L^d	The maximum length of interaction sequences of domain d .
n	The embedding size.
$\text{Encoder}^{hd}(\cdot)$	The hybrid-domain sequential encoder.
$\text{Encoder}^d(\cdot)$	The intra-domain sequential encoder of domain d .
\mathbf{X}_u^{hd}	The input matrix fed into $\text{Encoder}^{hd}(\cdot)$ about S_u^H .
$\bar{\mathbf{X}}_u^{hd}$	The input matrix fed into $\text{Encoder}^{hd}(\cdot)$ about \bar{S}_u^H .
\mathbf{X}_u^d	The input matrix fed into $\text{Encoder}^d(\cdot)$ about S_u^d .
$\mathbf{X}_u^{(B^{hd})}$	The hybrid-domain sequential preferences for user u .
$\bar{\mathbf{X}}_u^{(B^{hd})}$	The augmented hybrid-domain sequential preferences for user u .
$\mathbf{X}_u^{(B^a,d)}$	The intra-domain sequential preferences for user u in domain d .
\mathbf{X}	The input matrix fed into a Mamba block, which might be \mathbf{X}_u^{hd} , $\bar{\mathbf{X}}_u^{hd}$ or \mathbf{X}_u^d .
\mathbf{X}_o	The output of a Mamba block.
S_u^{t1}	User u 's predicted sequence for Task 1 .
$S_u^{t2,d}$	User u 's predicted sequence in domain d for Task 2 .
$S_u^{t3,d}$	User u 's predicted sequence in domain d for Task 3 .
$\tilde{r}_{u,i,t}$	The user u 's score to item i at time step t only considering the hybrid-domain sequential preferences.
$\tilde{p}_{u,i,t}^d$	The user u 's score to item i at time step t in domain d only considering the intra-domain sequential preferences.
$\tilde{z}_{u,i,t}^d$	The user u 's score to item i of other domains at time step t only considering user u 's interacted items in domain d .
r_{u,i,L^h+1}^d	User u 's score to item i of domain d at time step $L^h + 1$ by considering the hybrid-domain and intra-domain sequential preferences.

$C^2\text{DSR}$ ([Cao et al., 2022](#)) leverages a hybrid interaction sequence in all domains and a separate interaction sequence in each domain to capture domain-shared and domain-specific sequential preferences for each user, respectively. DREAM ([Ye, Li, & Yao, 2023](#)) models inter-sequence item relationship by a novel supervised contrastive learning approach. LEA-GCN ([Zhang et al., 2023](#)) leverages a lightweight GCN and a dual-channel external attention module to reduce the time complexity. However, these models share two common weaknesses. They become inapplicable when the number of domains exceeds two. Meanwhile, they ignore the problem of the data imbalance in the hybrid sequences. Though we can apply the multi-sequence learning method $C^2\text{DSR}$ to multi-domain recommendation, the time complexity will increase.

2.2. Multi-domain recommendation

Multi-domain recommendation aims to improve the recommendation performance in all domains simultaneously ([Li et al., 2023](#); [Luo et al., 2023](#); [Min, Luo, Lin, Huang, & Liu, 2023](#); [Wang et al., 2022](#); [Xu et al., 2023](#)). In early works, multi-task learning methods like MMoE ([Ma et al., 2018](#)), PLE ([Tang, Liu, Zhao, & Gong, 2020](#)) and MoSE ([Qin et al., 2020](#)) are applied to multi-domain recommendation by regarding recommendation performance in a domain as a task. The data within each task share the same distribution, while the data distribution varies across different domains. Consequently, these models will yield sub-optimal results. To solve these problems, some multi-domain recommendation models are proposed. STAR ([Sheng et al., 2021](#)) captures the domain-shared and domain-specific features by a domain-shared neural network and some domain-specific neural networks, respectively. AFT ([Hao et al., 2021](#)) leverages a generative adversarial neural network to learn feature transition among different domains. H^3Trans ([Xu et al., 2023](#)) leverages a hierarchical hyper-graph network to learn aggregated user representations and item representations. CAT-ART ([Li, Xie et al., 2023](#)) adopts contrastive learning and an auto-encoder to learn unbiased global user embeddings. Meanwhile, it alleviates the long-standing negative transfer problem by an attention-based method. But they all focus on CTR prediction and ignore the importance of the sequential patterns. MARIA ([Tian et al., 2023](#)) and MAGRec ([Ariza-Casabona, Twardowski, & Wijaya, 2023](#)) simply learn mixed behavior sequences by Transformer as users' profiles to improve CTR prediction.

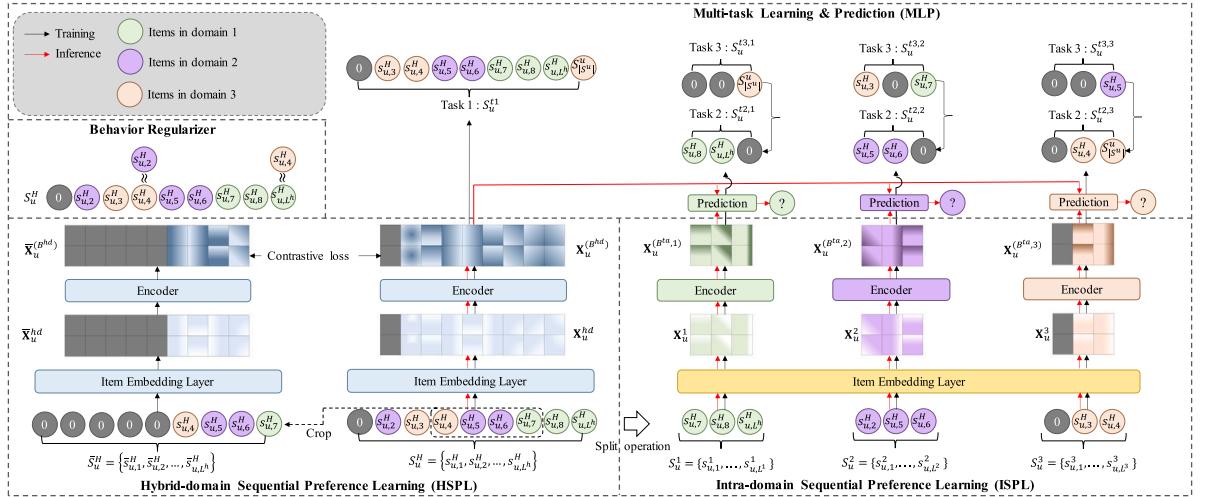


Fig. 2. The framework of the proposed multi-sequence multi-task learning (MML). Firstly, a whole hybrid interaction sequence S_u^H from all domains is split into three interaction sequences S_u^1, S_u^2 and S_u^3 for domains 1, 2 and 3, respectively. Secondly, hybrid-domain sequential preferences $X_u^{(B^d)}$ are learned by a hybrid-domain encoder. Meanwhile, augmented hybrid-domain sequential preferences $\bar{X}_u^{(B^d)}$ are learned by modeling an augmented hybrid interaction sequence \bar{S}_u^H that is obtained by a crop operation. The hybrid-domain sequential preferences $X_u^{(B^d)}$ become more unbiased by leveraging contrastive learning between $X_u^{(B^d)}$ and $\bar{X}_u^{(B^d)}$. Thirdly, each separate encoder is used to learn intra-domain sequential preferences $X_u^{(B^d,1)}, X_u^{(B^d,2)}$ and $X_u^{(B^d,3)}$ for domains 1, 2 and 3, respectively. Fourthly, HSPL is trained via **Task 1** and a behavior regularizer, and ISPL is trained via **Tasks 2** and **3**. Finally, MML predicts the user u 's next interacted item in domain d by taking $X_u^{(B^d)}$ and $X_u^{(B^d,d)}$ as the input, where $d \in \{1, 2, 3\}$. Note that **Tasks 1–3** represent hybrid-domain item prediction, intra-domain item prediction and inter-domain item prediction, respectively.

Without predicting the next interacted items based on users' historical interaction sequences, these methods fail to fully learn item-to-item correlation. DS-HRNN (Kim et al., 2019) leverages RNNs to model hybrid interaction sequences. But it overlooks the separate interaction sequences in each domain, which means the domain commonalities and domain differences cannot be fully captured. TJAPL (Xu et al., 2024) leverages a separate interaction sequence in each domain to learn a user's overall preferences and capture item transition patterns. However, TJAPL ignores the importance of hybrid interaction sequences. Meanwhile, TJAPL can only improve the recommendation performance in one target domain rather than in all domains simultaneously during a training epoch.

3. Methodology

3.1. Problem statement

In multi-domain sequential recommendation, we aim to predict the next interacted item for each user in each domain. Let $\mathcal{U}, \mathcal{V}, \mathcal{D}$ be the sets of users, items and domains, respectively. Let $\mathcal{U}^d, \mathcal{V}^d$ denote the user set and the item set, respectively, in domain $d \in \mathcal{D}$. Let S_u^H be the training hybrid interaction sequence of user u . By splitting S_u^H , a separate interaction sequence S_u^d for user u in domain d can be obtained. The probability of item $i \in \mathcal{V}^d$ being the next interacted item for user u in domain d can be represented as $P(i|S_u^1, S_u^2, \dots, S_u^{[D]}, S_u^H, d)$. Finally, in each domain d , the items from \mathcal{V}^d with the highest probabilities will be recommended to user u . Some commonly used notations are summarized in Table 1.

We propose our MML for multi-domain sequential recommendation. The framework of our MML is illustrated in Fig. 2. It mainly consists of three components, i.e., hybrid-domain sequential preference learning (HSPL), intra-domain sequential preference learning (ISPL), and multi-task learning & prediction (MLP), which are elaborated in Sections 3.4–3.6, respectively.

3.2. Item embedding layer

We use $S^u = \{S_1^u, S_2^u, \dots, S_{|\mathcal{S}^u|}^u\}$ to represent user u 's interaction sequence from all domains, obtained by sorting the interacted items in chronological order. Following the padding method in SASRec (Kang & McAuley, 2018), an input sequence $S_u^H = \{s_{u,1}^H, \dots, s_{u,i}^H, \dots, s_{u,L^H}^H\}$ is obtained, where $s_{u,i}^H \in \mathcal{V}$ and L^H denotes the maximum length of hybrid interaction sequences. Note that S_u^H does not contain the user u 's last interacted item. And for convenience, we use s_{u,L^H+1}^H to denote $S_{|\mathcal{S}^u|}^u$. Inspired by C²DSR (Cao et al., 2022) and EDDA (Ning et al., 2023), we use $\mathbf{M}_{hd} = [\mathbf{m}_1^{hd}, \dots, \mathbf{m}_i^{hd}, \dots, \mathbf{m}_{|\mathcal{V}|}^{hd}] \in \mathbb{R}^{|\mathcal{V}| \times n}$ to denote a learnable item matrix containing hybrid-domain knowledge, where n denotes the embedding size. We use $\mathbf{M}_{ta} = [\mathbf{m}_1^{ta}, \dots, \mathbf{m}_i^{ta}, \dots, \mathbf{m}_{|\mathcal{V}|}^{ta}] \in \mathbb{R}^{|\mathcal{V}| \times n}$ to represent a learnable item matrix containing intra-domain knowledge.

3.3. Encoder

We implement MML by leveraging three different sequential encoders (i.e., a GRU-based encoder, an attentional encoder and a Mamba-based encoder). In these encoders, the Mamba-based encoder (Gu & Dao, 2023) is a recently proposed one, which demonstrates effectiveness and efficiency. The GRU-based encoder is designed in a similar way to that of GRU4Rec (Hidasi et al., 2016). The attentional encoder is analogous to that of SASRec (Kang & McAuley, 2018). The Mamba-based encoder is designed by referring to that of Mamba4Rec (Liu, Lin, Wang, Liu & Caverlee, 2024). We will give brief descriptions about the Mamba-based encoder.

Mamba (Gu & Dao, 2023) is a state-of-the-art model for sequence modeling due to its efficiency and effectiveness. Compared with some Transformer-based models with quadratic computational complexity, Mamba can achieve linear computational complexity (Liu, Lin et al., 2024). The Mamba-based encoder consists of some Mamba layers, and a Mamba block is a core component of a Mamba layer. The main principle of Mamba is that it introduces an input-dependent selection mechanism based on traditional SSM (State Space Models) and obtains relevant knowledge according to the inputs (Gu & Dao, 2023). The specific details of a Mamba block are described as follows:

Firstly, $\mathbf{H}_x, \mathbf{H}_z \in \mathbb{R}^{L \times (e*n)}$ are obtained by performing linear projections on the input matrix $\mathbf{X} \in \mathbb{R}^{L \times n}$, where e denotes a block expansion factor for linear projections and L represents the length of an interaction sequence. Then, $\mathbf{H}'_x \in \mathbb{R}^{L \times (e*n)}$ is obtained through two operations (i.e., a 1D convolution $\text{Conv1d}(\cdot)$ and a SiLU activation $\text{SiLU}(\cdot)$), which can be formalized as $\mathbf{H}'_x = \text{SiLU}(\text{Conv1d}(\mathbf{H}_x))$. Next, $\bar{\mathbf{A}}, \bar{\mathbf{B}} \in \mathbb{R}^{L \times n \times p}$ and $\mathbf{C} \in \mathbb{R}^{L \times p}$ are obtained by performing a series of operations on \mathbf{H}'_x , where p denotes a state expansion factor for a selective SSM. Then, $\bar{\mathbf{A}}, \bar{\mathbf{B}}$ and \mathbf{C} are fed into the selective SSM (State Space Models). Through the selective SSM (i.e., $\text{SelectiveSSM}(\cdot)$), $\mathbf{H}_y \in \mathbb{R}^{L \times (e*n)}$ is obtained:

$$\mathbf{H}_y = \text{SelectiveSSM}(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{C})(\mathbf{H}'_x) \quad (1)$$

Finally, the output $\mathbf{X}_o \in \mathbb{R}^{L \times n}$ of the Mamba block is obtained:

$$\mathbf{X}_o = \text{Linear}(\mathbf{H}_y \odot \text{SiLU}(\mathbf{H}_z)) \quad (2)$$

where \odot denotes the element-wise product and $\text{Linear}(\cdot)$ denotes a linear operation. In a Mamba layer, the feed-forward networks and residual connections are leveraged similarly to SASRec (Kang & McAuley, 2018).

3.4. Hybrid-domain Sequential Preference Learning (HSPL)

Hybrid-domain sequential preferences are learned by a hybrid-domain sequential encoder $\text{Encoder}^{hd}(\cdot)$. If the hybrid-domain sequential encoder is a Mamba-based one, the hybrid-domain sequential preferences are captured by the following steps. Firstly, with a lookup over \mathbf{M}_{hd} , S_u^H can be transformed into an item embedding matrix $\mathbf{E}_u^{hd} = \left[\mathbf{m}_{s_{u,1}^H}^{hd}, \dots, \mathbf{m}_{s_{u,i}^H}^{hd}, \dots, \mathbf{m}_{s_{u,L^h}^H}^{hd} \right] \in \mathbb{R}^{L^h \times n}$. Same as the process in a GRU-based encoder, an input matrix $\mathbf{X}_u^{hd} \in \mathbb{R}^{L^h \times n}$ can be directly obtained without adding a position embedding matrix (Hidasi et al., 2016; Liu, Lin et al., 2024). In other words, \mathbf{X}_u^{hd} equals to \mathbf{E}_u^{hd} . If an attentional encoder is leveraged to capture sequential preferences, the input matrix \mathbf{X}_u^{hd} is obtained by adding the item embedding matrix \mathbf{E}_u^{hd} and the position matrix. Then, \mathbf{X}_u^{hd} is fed into $\text{Encoder}^{hd}(\cdot)$, and the hybrid-domain sequential preferences $\mathbf{X}_u^{(B^{hd})} = \left[\mathbf{x}_{u,1}^{(B^{hd})}, \mathbf{x}_{u,2}^{(B^{hd})}, \dots, \mathbf{x}_{u,L^h}^{(B^{hd})} \right] \in \mathbb{R}^{L^h \times n}$ for user u can be obtained:

$$\mathbf{X}_u^{(B^{hd})} = \text{Encoder}^{hd}(\mathbf{X}_u^{hd}) \quad (3)$$

where B^{hd} denotes the number of Mamba layers in $\text{Encoder}^{hd}(\cdot)$. However, this method might learn the biased hybrid-domain sequential preferences. As depicted in Fig. 2, if the user u 's hybrid interaction sequence is S_u^H , the learned hybrid-domain sequential preferences might largely focus on domain 1 but ignore domain 3. To address this problem, we propose to construct an augmented sequence and leverage contrastive learning. In Fig. 2, \bar{S}_u^H is the augmented sequence for user u . \bar{S}_u^H is obtained by cropping S_u^H , where $crop$ is the same as that of CL4SRec (Xie et al., 2022). \bar{S}_u^H is a continuous sub-sequence of S_u^H , and α controls the length ratio of the sub-sequence relative to the original sequence. When α is larger, \bar{S}_u^H tends to be similar to S_u^H . The reasons that our MML selects $crop$ as the augmentation method are as follows: (i) Compared with the original sequence, the number of items is changed in each domain for the augmented sequence. (ii) The positions of the interacted items are also changed in the original interaction sequences by the crop operation. It is expected to change the distributions of user u 's preferences as much as possible and alleviate the biased preferences caused by the interaction order and interaction number in each domain. After constructing \bar{S}_u^H , an input matrix $\tilde{\mathbf{X}}_u \in \mathbb{R}^{L^h \times n}$ fed into $\text{Encoder}^{hd}(\cdot)$ is obtained based on \bar{S}_u^H , and the augmented hybrid-domain sequential preferences $\tilde{\mathbf{X}}_u^{(B^{hd})} = \left[\tilde{\mathbf{x}}_{u,1}^{(B^{hd})}, \tilde{\mathbf{x}}_{u,2}^{(B^{hd})}, \dots, \tilde{\mathbf{x}}_{u,L^h}^{(B^{hd})} \right] \in \mathbb{R}^{L^h \times n}$ are learned from \bar{S}_u^H according to the aforementioned process. Then, contrastive learning is leveraged:

$$\mathcal{L}_{cl} = - \sum_{u \in \mathcal{U}} \log \frac{\exp \left(\text{sim} \left(\mathbf{x}_{u,L^h}^{(B^{hd})}, \tilde{\mathbf{x}}_{u,L^h}^{(B^{hd})} \right) / \tau \right)}{\sum_{u' \in B_u} \exp \left(\text{sim} \left(\mathbf{x}_{u,L^h}^{(B^{hd})}, \tilde{\mathbf{x}}_{u',L^h}^{(B^{hd})} \right) / \tau \right)} \quad (4)$$

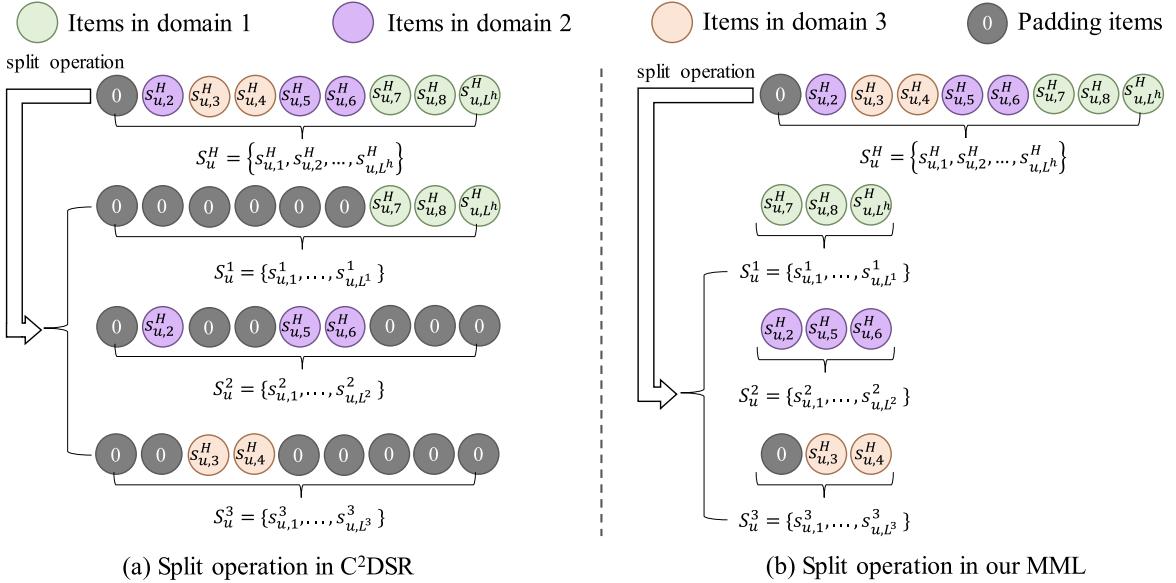


Fig. 3. The two different split operations for C²DSR and our MML.

where τ denotes the temperature coefficient, and B_u is the user set in one batch (Xie, Sun et al., 2022). Note that we use cosine similarity to measure similarity: $\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}\mathbf{v}^T}{\|\mathbf{u}\|\|\mathbf{v}\|}$. Contrastive learning can minimize the distance between $\mathbf{x}_{u,L^h}^{(B^{hd})}$ and $\bar{\mathbf{x}}_{u,L^h}^{(B^{hd})}$. Therefore, the distributions of the captured hybrid-domain sequential preferences $\mathbf{x}_{u,L^h}^{(B^{hd})}$ are changed. HSPL can capture more diverse and unbiased hybrid-domain sequential preferences.

3.5. Intra-domain Sequential Preference Learning (ISPL)

The user u 's interaction sequence $S_u^d = \{s_{u,1}^d, \dots, s_{u,i}^d, \dots, s_{u,L^d}^d\}$ in each domain $d \in \mathcal{D}$ can be obtained by splitting the hybrid interaction sequence S_u^H , where L^d denotes the maximum length of interaction sequences in domain $d \in \mathcal{D}$. The split method is different from that of C²DSR (Cao et al., 2022). The interacted items are put one by one according to the interacted domains. As illustrated in part (b) of Fig. 3, S_u^1 is obtained by splitting S_u^H , where $L^1 + \dots + L^d + \dots + L^{|\mathcal{D}|} = L^h$. The split operation for C²DSR is shown in part (a) of Fig. 3. If $s_{u,i}^H$ belongs to domain d , $s_{u,i}^d$ is equal to $s_{u,i}^H$; otherwise it will be a padding item. The lengths of all interaction sequences are the same, where $L^1 = \dots = L^d = \dots = L^{|\mathcal{D}|} = L^h$. The split operation of C²DSR is also applied to some CDSR models (e.g., DREAM (Ye et al., 2023) and CA-CDSR (Yin et al., 2024)). Because the lengths of separate interaction sequences are smaller by using our proposed method. Therefore, our MML is more efficient in modeling a separate interaction sequence in each domain, reducing the cost of computational resources. Meanwhile, our split method has broader applicability. For the two split operations, if our MML is based on Transformer, they can achieve comparable results. Because padding items can be masked during the calculation of attention scores. Meanwhile, the positions of two continuously interacted items are decided by position embeddings rather than the specific positions in interaction sequences. Even if some padding items are placed in the middle of the two items, a Transformer-based encoder still knows the two items are interacted continuously. However, if our MML is based on GRUs, our MML might struggle to capture sequential dependencies. Because some intervening padding items will increase the distance between the two continuously interacted items. Meanwhile, GRUs fail to capture long-range dependencies (Gu & Dao, 2023). If our MML is based on Mamba, padding items will have little effect on the recommendation performance due to its effectiveness in capturing long-range dependencies. It is worth noting that our split operation cannot be directly applied to C²DSR due to its specific training objective.

ISPL aims to capture intra-domain sequential preferences from the interaction sequences in each domain. Firstly, S_u^d can be transformed into $\mathbf{E}_u^d \in \mathbb{R}^{L^d \times n}$ with a lookup over \mathbf{M}_{ta} . If a Mamba-based encoder is leveraged to capture sequential preferences, an input matrix $\mathbf{X}_u^d \in \mathbb{R}^{L^d \times n}$ is equivalent to \mathbf{E}_u^d . The process is the same as that of HSPL. By feeding \mathbf{X}_u^d into an intra-domain encoder $\text{Encoder}^d(\cdot)$, the intra-domain sequential preferences $\mathbf{X}_u^{(B^{ta}, d)} = [\mathbf{x}_{u,1}^{(B^{ta}, d)}, \mathbf{x}_{u,2}^{(B^{ta}, d)}, \dots, \mathbf{x}_{u,L^d}^{(B^{ta}, d)}] \in \mathbb{R}^{L^d \times n}$ for user u in domain d can be obtained:

$$\mathbf{X}_u^{(B^{ta}, d)} = \text{Encoder}^d(\mathbf{X}_u^d) \quad (5)$$

where B^{ta} denotes the number of Mamba layers in $\text{Encoder}^d(\cdot)$. To learn the intra-domain sequential preferences in each domain, a separate intra-domain encoder is built for each domain $d \in \mathcal{D}$.

3.6. Multi-task Learning & Prediction (MLP)

To learn the two kinds of sequential preferences better, we adopt three tasks and a behavior regularizer to jointly train our MML. Different from Tri-CDR (Ma et al., 2024) which adopts a pretrain-then-fine-tune paradigm, our MML is trained using an end-to-end framework. HSPL is trained by **Task 1** and the behavior regularizer. ISPL is trained by **Tasks 2** and **3**. **Tasks 1** and **2** can ensure that our MML can fully exploit hybrid-domain sequential preferences and intra-domain sequential preferences. Meanwhile, **Task 3** can realize knowledge transfer across different domains.

Task 1 (Hybrid-domain Item Prediction). Given a hybrid interaction sequence for user u , **Task 1** aims to predict the user u 's next interacted item which can belong to any domain. The aim of this task is to capture the inter-domain item-to-item correlation. For **Task 1**, a negative item j of user u is randomly selected from $\mathcal{V}_d \setminus S^u$ and the corresponding positive item belongs to domain d . It can be formalized as:

$$\tilde{r}_{u,i,t} = \mathbf{x}_{u,t}^{(B^{hd})} (\mathbf{m}_i^{hd})^T \quad (6)$$

$$\mathcal{L}_1 = - \sum_{u \in \mathcal{U}} \sum_{i=1}^{L^h} \delta(s_{u,i}^{t1}) \left[\log \left(\sigma \left(\tilde{r}_{u,s_{u,i}^{t1},i} \right) \right) + \log \left(1 - \sigma \left(\tilde{r}_{u,j,i} \right) \right) \right] \quad (7)$$

where $\sigma(\cdot)$ is the sigmoid function. $\delta(\cdot)$ is an indicator function to judge whether an item is a padding item. If $s_{u,i}^{t1}$ is a padding item, $\delta(s_{u,i}^{t1})$ is 0, otherwise it is 1. For user u in all domains, the input sequence is S_u^H and the predicted sequence is $S_u^{t1} = \{s_{u,1}^{t1}, \dots, s_{u,i}^{t1}, \dots, s_{u,L^h}^{t1}\}$, where $s_{u,i}^{t1} = s_{u,i+1}^H$ for $i = 1, \dots, L^h$.

Task 2 (Intra-domain Item Prediction). Given an interaction sequence for user u in a single domain $d \in D$, this task is to predict the user u 's next interacted item in the domain d . Only the interactions in the current domain d are leveraged in training and prediction, so this task can fully capture the intra-domain sequential preferences in domain d . **Task 2** can be formalized:

$$\tilde{r}_{u,i,t}^d = \mathbf{x}_{u,t}^{(B^{ta,d})} (\mathbf{m}_i^{ta})^T \quad (8)$$

The loss of **Task 2** for all domains can be formalized as follows:

$$\mathcal{L}_2 = - \sum_{d \in D} \sum_{u \in \mathcal{U}^d} \sum_{i=1}^{L^d} \delta(s_{u,i}^{t2,d}) \left[\log \left(\sigma \left(\tilde{r}_{u,s_{u,i}^{t2,d},i}^d \right) \right) + \log \left(1 - \sigma \left(\tilde{r}_{u,j,i}^d \right) \right) \right] \quad (9)$$

In each domain, there exists **Task 2**. For user u in domain d , the input and predicted sequences are S_u^d and $S_u^{t2,d} = \{s_{u,1}^{t2,d}, \dots, s_{u,i}^{t2,d}, \dots, s_{u,L^d}^{t2,d}\}$, respectively. For **Task 2** in domain d , a negative item j of user u is randomly selected from $\mathcal{V}_d \setminus S^u$.

Task 3 (Inter-domain Item Prediction). Given an interaction sequence for user u in a single domain, this task is to predict the user u 's next interacted item which belongs to other domains. The predicted items are selected according to the hybrid interaction sequences. The specific details are shown in Fig. 2. For example, user u interacts with item $s_{u,2}^H$ in domain 2, and then interacts with item $s_{u,3}^H$ in domain 3. Therefore, when given the interaction sequence in domain 2 for user u and the last interacted item is $s_{u,2}^H$, **Task 3** aims to predict the next interacted item $s_{u,3}^H$ in domain 3. There are two motivations for **Task 3**. Firstly, if the i th and $(i+1)$ th interacted items w.r.t. user u belong to two different domains, the two items can be leveraged to build the correlations between the two domains. Therefore, **Task 3** can transfer knowledge across different domains to some extent. Secondly, domains are manually divided by categories (i.e., Book, CD and Movie) (Xu et al., 2023). If user u continuously interacts with two items. The two items might have some similarities. Though the two items belong to two different domains, the two items might be divided into the same domain by considering other side information (e.g., titles, text descriptions and so on). Meanwhile, there might exist some items belonging to multiple domains simultaneously. Therefore, **Task 3** can be regarded as a data augmentation method. It is because **Task 3** leverages more training data from other domains to help MML learn intra-domain sequential preferences in the current domain d . Though items in the predicted sequences and items in the input sequences might be divided into the same domain by considering more side information, there still exist distribution differences among different domains. Given user u 's intra-domain sequential preferences in domain d and predicted items belonging to another domain d' at time step t , **Task 3** projects intra-domain sequential preferences from one domain d to another domain d' , which is formalized as follows:

$$\tilde{\mathbf{x}}_{u,t}^{(B^{ta,d})} = \text{Linear}^{d'}(\mathbf{x}_{u,t}^{(B^{ta,d})}) \quad (10)$$

where $\text{Linear}^{d'}(\cdot)$ is a single-layer neural network, which maps intra-domain sequential preferences from one domain d to another domain d' . $\tilde{\mathbf{x}}_{u,t}^{(B^{ta,d})} \in \mathbb{R}^{1 \times n}$ denotes the projected intra-domain sequential preferences. Then, **Task 3** predicts the probability that user u interacts with item i in another domain d' :

$$\tilde{r}_{u,i,t}^d = \tilde{\mathbf{x}}_{u,t}^{(B^{ta,d})} (\mathbf{m}_i^{ta})^T \quad (11)$$

The loss function for **Task 3** in all domains can be formalized as follows:

$$\mathcal{L}_3 = - \sum_{d \in D} \sum_{u \in \mathcal{U}^d} \sum_{i=1}^{L^d} \delta(s_{u,i}^{t3,d}) \left[\log \left(\sigma \left(\tilde{r}_{u,s_{u,i}^{t3,d},i}^d \right) \right) + \log \left(1 - \sigma \left(\tilde{r}_{u,j,i}^d \right) \right) \right] \quad (12)$$

For user u in domain d , the input and predicted sequences are S_u^d and $S_u^{t3,d} = \{s_{u,1}^{t3,d}, \dots, s_{u,i}^{t3,d}, \dots, s_{u,L^d}^{t3,d}\}$, respectively. For **Task 3** in domain d , a negative item j of user u is also randomly selected from $\mathcal{V}_d \setminus S^u$. This task ensures that the intra-domain sequential preferences $\mathbf{x}_u^{(B^{ta},d)}$ belonging to domain d contain knowledge from other domains. Because if $\mathbf{x}_u^{(B^{ta},d)}$ contains little knowledge transferred from other domains, it is difficult for $\mathbf{x}_u^{(B^{ta},d)}$ to accurately predict items belonging to other domains.

Inspired by DS-HRNN (Kim et al., 2019), a behavior regularizer is adopted to build the correlations among items from different domains using the squared Euclidean norm $\|\cdot\|_2^2$. It only focuses on item embeddings containing hybrid-domain knowledge:

$$\mathcal{L}_{br} = \sum_{u \in \mathcal{U}} \sum_{i=2}^{L^h+1} p(s_{u,i}^H) \|\mathbf{m}_{s_{u,i-1}^H}^{hd} - \mathbf{m}_{s_{u,i-t}^H}^{hd}\|_2^2 \quad (13)$$

where $p(\cdot)$ is an indicator function. If $s_{u,i}^H$ and $s_{u,i-1}^H$ belong to different domains, $p(s_{u,i}^H)$ equals to 1, otherwise it is 0. $s_{u,i-t}^H$ denotes the last interacted item before item $s_{u,i}^H$, and the two items belong to the same domain. The final loss function is as follows:

$$\mathcal{L} = \lambda_{br} * \mathcal{L}_{br} + \lambda_{cl} * \mathcal{L}_{cl} + \sum_{i=1}^3 (\lambda_i * \mathcal{L}_i) \quad (14)$$

where $\lambda_i (i = 1, 2, 3)$ is a hyper-parameter which controls the magnitude of each loss of each task. λ_{br} and λ_{cl} denote the weights of the behavior regularizer and contrastive learning loss, respectively.

In order to predict the next interacted item for user u in domain d . Firstly, we can obtain the user u 's final sequential preferences \mathbf{x}_u^{final} by concatenating the hybrid-domain sequential preferences $\mathbf{x}_{u,L^h}^{(B^{hd})}$ and the intra-domain sequential preferences $\mathbf{x}_{u,L^d}^{(B^{ta},d)}$, which can be formalized as $\mathbf{x}_u^{final} = [\mathbf{x}_{u,L^h}^{(B^{hd})}, \mathbf{x}_{u,L^d}^{(B^{ta},d)}] \in \mathbb{R}^{1 \times 2n}$, where $[\cdot, \cdot]$ denotes the concatenation operation. We can then obtain the final item embedding $\mathbf{m}_i^{final} \in \mathbb{R}^{1 \times 2n}$ about item i from domain d by the same concatenation operation, i.e., $\mathbf{m}_i^{final} = [\mathbf{m}_i^{hd}, \mathbf{m}_i^{ta}]$. Finally, we use the following prediction rule:

$$r_{u,i,L^h+1}^d = \mathbf{x}_u^{final} (\mathbf{m}_i^{final})^T = \mathbf{x}_{u,L^h}^{(B^{hd})} (\mathbf{m}_i^{hd})^T + \mathbf{x}_{u,L^d}^{(B^{ta},d)} (\mathbf{m}_i^{ta})^T \quad (15)$$

where r_{u,i,L^h+1}^d represents the user u 's score to item i belonging to domain d at time step $L^h + 1$.

3.7. Discussion

Though our MML is efficient and effective, our MML still faces some challenges if it is deployed in a large-scale sequential recommendation system, especially in cases where there are a large number of domains and interactions. We discuss some specific challenges and potential solutions as follows:

- **Long sequential behavior modeling:** Though our MML is efficient, it is still challenging to model very long interaction sequences. For example, the training efficiency will decrease. Moreover, noise and outdated information may exist in long interaction sequences, which will hurt the recommendation performance. Following the method in SIM (Pi et al., 2020), we may select several of the most related interactions from a long interaction sequence for extracting long-term user interests.
- **Convergence:** Our MML might converge at different rates in different domains with varying levels of data richness, especially when there are lots of domains. Following the method of D3 (Jia et al., 2024), we may introduce different weights of the losses to different domains to alleviate this issue.
- **Side information:** Our MML only leverages item IDs in training and prediction. When there are different kinds of side information (e.g., prices, images and text descriptions) including domain-shared and domain-specific features, we may combine the item ID embeddings and feature embeddings. We can also fuse them by following the method in DIF-SR (Xie, Zhou & Kim, 2022) and ADI (Jiang et al., 2022).

4. Experiments

In this section, we aim to answer the following research questions: (RQ1) Does our MML achieve better performance than the state-of-the-art baselines? (RQ2) How do the hyper-parameters affect the performance of our MML? (RQ3) Is each module in our MML necessary? (RQ4) Can our MML alleviate the data sparsity and cold-start problems? (RQ5) Does our MML still demonstrate satisfactory recommendation performance without one of the three domains? (RQ6) Does our MML still achieve superior recommendation performance by leveraging other data augmentation methods? (RQ7) Is our MML more efficient from both theoretical and experimental perspectives? (RQ8) Can our MML alleviate the data imbalance problem via data augmentation and contrastive learning? (RQ9) Can our MML capture more unbiased hybrid-domain sequential preferences through contrastive learning? (RQ10) Can our MML achieve good generalization on some other multi-domain datasets and evaluation metrics?

4.1. Experimental setup

4.1.1. Datasets

The experiments are conducted on three Amazon datasets.¹ The domains are divided based on the category of the products. Three categories (i.e., Movie, CD and Book) are selected due to their abundant interactions. The data are processed following MGCL (Xu, Pan et al., 2023) and TJAPL (Xu et al., 2024). The process contains the following steps: (1) Subsequent duplicate (user, item) pairs are ignored. (2) The cold-start users and items with fewer than five interactions are discarded. (3) Only the users who have interactions in all domains are kept. The statistics of the processed data are shown in Table 2.

4.1.2. Baselines

Some state-of-the-art models are selected as baselines, including single-domain methods (i.e., MFBPR (Koren, Bell, & Volinsky, 2009), GRU4Rec (Hidasi et al., 2016), SASRec (Kang & McAuley, 2018), FMLP-Rec (Zhou, Yu, Zhao, & Wen, 2022) and Mamba4Rec (Liu, Lin et al., 2024)), cross-domain methods (i.e., CD-SASRec (Alharbi & Caragea, 2022), Tri-CDR (Ma et al., 2024), MGCL (Xu, Pan et al., 2023) and C²DSR (Cao et al., 2022)), and multi-domain methods (i.e., MMoE (Ma et al., 2018), PLE (Tang et al., 2020), GRU4Rec-hybrid, SASRec-hybrid, Mamba4Rec-hybrid, DS-HRNN (Kim et al., 2019), DS-TF, DS-Mamba, CAT-ART (Li, Xie et al., 2023) and TJAPL (Xu et al., 2024)). We implement CD-SASRec, DS-HRNN (Kim et al., 2019), DS-TF and DS-Mamba using PyTorch by ourselves, and obtain DS-TF and DS-Mamba by replacing RNNs in DS-HRNN (Kim et al., 2019) with Transformer and Mamba, respectively. Meanwhile, we implement GRU4Rec-hybrid, SASRec-hybrid and Mamba4Rec-hybrid by taking hybrid interaction sequences as the input of GRU4Rec, SASRec and Mamba4Rec, respectively. MMoE and PLE are implemented based on the DeepCTR-Torch library.² Other baselines are implemented based on the released codes provided by the corresponding authors. Some details about these baselines are described as follows:

Single-Domain Methods

- MFBPR (Koren et al., 2009): It applies matrix factorization to recommendation systems.
- GRU4Rec (Hidasi et al., 2016): It is a classic RNNs-based sequential recommendation model by applying GRUs.
- SASRec (Kang & McAuley, 2018): It leverages the self-attention mechanism to capture sequential patterns.
- FMLP-Rec (Zhou et al., 2022): It is an MLPs-based sequential recommendation model which leverages learnable filters to reduce noise.
- Mamba4Rec (Liu, Lin et al., 2024): It is a Mamba-based sequential recommendation model, which is more efficient than SASRec in training and inference. Meanwhile, it can achieve comparable recommendation performance to SASRec.

Cross-Domain Methods

- CD-SASRec (Alharbi & Caragea, 2022): It fuses an integrated user's source-domain vector into item embeddings in the target domain.
- Tri-CDR (Ma et al., 2024): It models the correlations among three types of sequences to transfer useful knowledge.
- MGCL (Xu, Pan et al., 2023): It leverages multi-view graph contrastive learning to transfer knowledge across two domains.
- C²DSR (Cao et al., 2022): It is a dual-target cross-domain sequential recommendation model. It leverages contrastive learning between a hybrid interaction sequence across all domains and an isolated interaction sequence in each domain.

Multi-Domain Methods

- MMoE (Ma et al., 2018): It leverages multiple experts to conduct multi-task learning.
- PLE (Tang et al., 2020): To alleviate negative transfer, it leverages task-shared and task-specific experts.
- GRU4Rec-hybrid: GRU4Rec is applied to MDSR by taking hybrid interaction sequences in all domains as input.
- SASRec-hybrid: SASRec is applied to MDSR by taking hybrid interaction sequences in all domains as input.
- Mamba4Rec-hybrid: Mamba4Rec is applied to MDSR by taking hybrid interaction sequences in all domains as input.
- DS-HRNN (Kim et al., 2019): It applies RNNs-based methods to capture sequential dynamics in multi-domain recommendation.
- DS-TF: It is implemented by replacing RNNs with Transformer in DS-HRNN (Kim et al., 2019).
- DS-Mamba: It is implemented by replacing RNNs with Mamba in DS-HRNN (Kim et al., 2019).
- CAT-ART (Li, Xie et al., 2023): It is a state-of-the-art multi-domain recommendation model for CTR prediction.
- TJAPL (Xu et al., 2024): It leverages a separate interaction sequence in each domain to learn a user's overall preferences and capture item transition patterns. Meanwhile, it realizes knowledge transfer across multiple domains.

¹ <https://jmcauley.ucsd.edu/data/amazon/>.

² <https://github.com/shenweichen/DeepCTR-Torch>.

Table 2

The statistics of the processed Amazon datasets.

Domain	Users	Items	Actions	Avg.Len.	Density
All domains	10 929	382 089	1 412 104	129.21	0.0003
Book	10 929	236 049	607 657	55.60	0.0002
CD	10 929	91 169	344 221	31.50	0.0003
Movie	10 929	59 513	460 226	42.11	0.0007

Table 3

Recommendation performance of the baselines and our MML on three domains.

Models	Book		CD		Movie	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
MFBPR	0.1312 ± 0.0008	0.2541 ± 0.0018	0.1916 ± 0.0026	0.3541 ± 0.0044	0.1165 ± 0.0009	0.2220 ± 0.0012
GRU4Rec	0.1364 ± 0.0025	0.2494 ± 0.0060	0.1647 ± 0.0034	0.3032 ± 0.0063	0.1445 ± 0.0048	0.2682 ± 0.0067
SASRec	0.1364 ± 0.0004	0.2545 ± 0.0012	0.1985 ± 0.0021	0.3552 ± 0.0015	0.1702 ± 0.0039	0.3052 ± 0.0064
FMLP-Rec	0.1068 ± 0.0064	0.2000 ± 0.0086	0.1622 ± 0.0136	0.2919 ± 0.0200	0.1881 ± 0.0053	0.3398 ± 0.0062
Mamba4Rec	0.1679 ± 0.0028	0.2958 ± 0.0051	0.2207 ± 0.0015	0.3792 ± 0.0042	0.1983 ± 0.0061	0.3384 ± 0.0101
CD-SASRec	0.1487 ± 0.0017	0.2743 ± 0.0022	0.1975 ± 0.0020	0.3528 ± 0.0035	0.1789 ± 0.0017	0.3177 ± 0.0020
Tri-CDR	0.1517 ± 0.0048	0.2775 ± 0.0098	0.1946 ± 0.0049	0.3493 ± 0.0060	0.1763 ± 0.0027	0.3265 ± 0.0058
MGCL	0.1488 ± 0.0077	0.2772 ± 0.0118	0.2100 ± 0.0090	0.3761 ± 0.0135	0.1949 ± 0.0069	0.3557 ± 0.0075
C ² DSR	0.1527 ± 0.0040	0.2809 ± 0.0055	0.2274 ± 0.0039	0.3889 ± 0.0065	0.1874 ± 0.0031	0.3341 ± 0.0047
MML _C +GRU4Rec	0.1702 ± 0.0068	0.2993 ± 0.0079	0.2252 ± 0.0064	0.3926 ± 0.0113	0.1913 ± 0.0079	0.3358 ± 0.0123
MML _C +SASRec	0.1853 ± 0.0021	0.3289 ± 0.0042	0.2637 ± 0.0025	0.4408 ± 0.0036	0.2358 ± 0.0021	0.4001 ± 0.0022
MML _C +Mamba4Rec	0.2036 ± 0.0068	0.3463 ± 0.0072	0.2680 ± 0.0110	0.4540 ± 0.0130	0.2457 ± 0.0092	0.4160 ± 0.0084
MMoE	0.0782 ± 0.0014	0.1638 ± 0.0021	0.1041 ± 0.0047	0.2008 ± 0.0105	0.0856 ± 0.0048	0.1646 ± 0.0153
PLE	0.0866 ± 0.0134	0.1762 ± 0.0197	0.1148 ± 0.0117	0.2270 ± 0.0054	0.1318 ± 0.0063	0.2910 ± 0.0102
GRU4Rec-hybrid	0.1314 ± 0.0032	0.2396 ± 0.0061	0.1731 ± 0.0038	0.3083 ± 0.0048	0.1505 ± 0.0064	0.2772 ± 0.0102
SASRec-hybrid	0.1402 ± 0.0008	0.2670 ± 0.0034	0.2061 ± 0.0027	0.3556 ± 0.0062	0.1832 ± 0.0017	0.3277 ± 0.0020
Mamba4Rec-hybrid	0.1569 ± 0.0041	0.2856 ± 0.0026	0.2244 ± 0.0022	0.3724 ± 0.0031	0.2033 ± 0.0046	0.3473 ± 0.0067
DS-HRNN	0.1446 ± 0.0006	0.2627 ± 0.0008	0.2037 ± 0.0016	0.3448 ± 0.0030	0.1925 ± 0.0030	0.3288 ± 0.0084
DS-TF	0.1576 ± 0.0018	0.2834 ± 0.0056	0.2238 ± 0.0046	0.3756 ± 0.0057	0.2143 ± 0.0020	0.3634 ± 0.0044
DS-Mamba	0.1601 ± 0.0022	0.2858 ± 0.0023	0.2267 ± 0.0012	0.3776 ± 0.0023	0.2202 ± 0.0032	0.3665 ± 0.0040
CAT-ART	0.1394 ± 0.0047	0.2661 ± 0.0059	0.2075 ± 0.0021	0.3783 ± 0.0032	0.1436 ± 0.0023	0.2723 ± 0.0033
TJAPL	0.1744 ± 0.0019	0.3116 ± 0.0026	0.2280 ± 0.0058	0.3977 ± 0.0076	0.1968 ± 0.0025	0.3548 ± 0.0031
MML+SASRec ^P	0.1859 ± 0.0081	0.3296 ± 0.0111	0.2571 ± 0.0049	0.4393 ± 0.0055	0.2356 ± 0.0031	0.4047 ± 0.0037
MML+GRU4Rec	0.1790 ± 0.0033	0.3125 ± 0.0049	0.2404 ± 0.0010	0.4121 ± 0.0010	0.2044 ± 0.0013	0.3589 ± 0.0015
MML+SASRec	0.1914 ± 0.0019	0.3362 ± 0.0013	0.2644 ± 0.0032	0.4455 ± 0.0037	0.2396 ± 0.0035	0.4079 ± 0.0037
MML+Mamba4Rec	0.2074 ± 0.0046*	0.3529 ± 0.0066*	0.2801 ± 0.0042*	0.4677 ± 0.0054*	0.2558 ± 0.0047*	0.4270 ± 0.0068*
MML vs. TJAPL	18.92%	13.25%	22.85%	17.60%	29.98%	20.35%

* Denotes statistically significant improvement (i.e., p -value ≤ 0.05) over the best baseline (i.e., TJAPL), which is determined by using the paired t-test.

4.1.3. Evaluation metrics

The experiments are evaluated via two widely used metrics i.e., HR@N and NDCG@N, where N is 10. HR@N considers presence while NDCG@N considers both the presence and positions in recommendation lists. These two metrics are formalized as follows:

$$\text{HR@N} = \frac{1}{|\mathcal{U}^d|} \sum_{u \in \mathcal{U}^d} \mathbf{1}(R_u \leq N) \quad (16)$$

$$\text{NDCG@N} = \frac{1}{|\mathcal{U}^d|} \sum_{u \in \mathcal{U}^d} \frac{\mathbf{1}(R_u \leq N)}{\log_2(R_u + 1)} \quad (17)$$

where R_u represents the rank of a preferred item of user u , and $\mathbf{1}(\cdot)$ is an indicator function. If the condition is satisfied, it returns 1; otherwise it returns 0.

4.1.4. Implementation details

For each domain, the training, validation and test data are obtained by splitting the historical interaction sequence of each user via the leave-one-out evaluation. We use the last interacted item for testing, the second most recently interacted item for validation and all the remaining items for training.

For a fair comparison, the common hyper-parameters are set to the same. Specifically, the dropout rate is set to 0.5, the L_2 regularization coefficient is set to 0.001, the batch size is set to 128, and the learning rate is set to 0.001. The embedding size is searched from {10, 20, 30, 40, 50}, and is finally set to 50 (Kang & McAuley, 2018; Tang & Wang, 2018; Xu, Pan et al., 2023). All models are trained by the Adam optimizer. For DS-HRNN, DS-TF, DS-Mamba, GRU4Rec-hybrid, SASRec-hybrid, Mamba4Rec-hybrid and our MML, the maximum hybrid sequence length is set to 300. For Tri-CDR, C²DSR and MML_C, the maximum hybrid sequence length is set to 200. For single-domain and cross-domain sequential models without hybrid sequences, the maximum sequence length

is set to 100. For the users' hybrid interaction sequences, only recently interacted 100 items can be retained in each domain. Because the maximum sequence length is 100 in single-domain sequential recommendation models, this rule can ensure fairness. For our MML, L^h is set to 300, L^d is set to 100 in all domains, λ_{br} is set to 0.001 after searching from {0.1, 0.01, 0.001}, τ is searched from {0.05, 0.1, 0.2, 0.5, 1}, α is searched from {0.1, 0.3, 0.5, 0.7, 0.9}, λ_1, λ_2 are set to 1, λ_3 is searched from {0.1, 0.5, 1.0, 1.5, 2.0}, λ_{cl} is set to 0.001 after searching from {0.1, 0.01, 0.001, 0.0001, 0.00001}, B^{hd} and B^{ta} are set to 2, the selective SSM state expansion factor p is set to 32, the kernel size for 1D convolution is set to 4, and the block expansion factor e for linear projections is set to 2. When the backbone model is GRU4Rec for MML, the number of GRU layers is set to 2. When the backbone model is SASRec for MML, the number of attention blocks and the number of attention heads are set to 2 and 1, respectively (Cao et al., 2022). Meanwhile, we keep the parameter settings for the backbone models of our MML the same as GRU4Rec, SASRec, Mamba4Rec and some baselines (i.e., CD-SASRec, Tri-CDR, MGCL, C²DSR, GRU4Rec-hybrid, SASRec-hybrid, Mamba4Rec-hybrid, DS-HRNN, DS-TF, DS-Mamba and TJAPL) using these backbone models. For DS-HRNN, DS-TF and DS-Mamba, the coefficients of both the supplementary loss and the behavior regularizer are searched from {0.001, 0.01, 0.1}. For MMoE, the number of experts is searched from {3, 5} and the hidden layers of expert networks are searched from {256 × 128, 512 × 256 × 128}. For PLE, the number of shared experts is set to 1, the number of specific experts is searched from {3, 5}, and the hidden layers of expert networks are searched from {256 × 128, 512 × 256 × 128}. Other hyper-parameters of the baselines are adjusted by the corresponding authors' suggestions in the original papers. For each user, 100 negative items that the user does not interact with in the corresponding domain are selected by popularity-based sampling (Xu, Pan et al., 2023).

Single-domain methods are trained separately in each domain. In cross-domain methods (Xu, Pan et al., 2023), for the same target domain, we build different models according to different source domains, and report the best results under optimal hyper-parameters in validation dataset. We run the experiments three times and report the average results and standard deviation in Table 3. Meanwhile, we conduct a significant test between the strongest baseline and our MML. And, we run the experiments three times and report the average results in Tables 4, 5, 6, 7, 8, 9, 10, 11, 12, 15 and 17.

MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec denote three variants of our MML leveraging three different kinds of encoders, where GRU4Rec, SASRec and Mamba4Rec respectively serve as the backbone models for sequential preference learning. MML+SASRec^P denotes a specific variant of MML+SASRec where all position embeddings are obtained with a lookup over the same position embedding matrix. For the position embeddings of each Transformer-based encoder in MML+SASRec^P, they are obtained with a lookup over a separate position embedding matrix. MML_C denotes that our MML is trained in two domains for CDSR.

4.2. Overall performance comparison (RQ1)

The performance of the baselines and our MML is shown in Table 3. Note that the best results and second-best results are marked in bold and underlined, respectively. We can have the following observations.

Firstly, compared with the strong baselines, our MML achieves better performance. It is because our MML considers the hybrid-domain, intra-domain sequential preferences in a joint manner. MML+GRU4Rec, MML+SASRec, MML+Mamba4Rec all achieve better recommendation performance than GRU4Rec, SASRec and Mamba4Rec, respectively. This phenomenon indicates that our MML is a generic MDSR framework, which can be adapted to different backbone models. Compared with some MDSR models (i.e., GRU4Rec-hybrid, SASRec-hybrid, Mamba4Rec-hybrid, DS-HRNN, DS-TF and DS-Mamba) only using hybrid interaction sequences, our MML achieves significant performance gains due to the full utilization of a separate interaction sequence in each domain and a hybrid interaction sequence in all domains. Though TJAPL considers a separate interaction sequence in each domain, TJAPL does not consider the hybrid interaction sequence in all domains, which results in failing to capture vital sequential patterns and hybrid-domain sequential preferences. Therefore, our MML performs better than TJAPL. Secondly, the cross-domain methods perform better than the single-domain methods in most cases, which showss that the former can alleviate the cold-start and data sparsity problems in the target domain by transferring knowledge from the source domain. Meanwhile, our MML_C achieves better recommendation performance compared with the state-of-the-art CDSR models. It is mainly because our MML_C can learn more unbiased and robust hybrid-domain sequential preferences. Therefore, our MML can also be reduced to MML_C trained in two domains. Thirdly, PLE and MMoE show relatively poor performance. Because they fail to capture sequential preferences without modeling interaction sequences. Meanwhile, they regard the recommendation performance of each domain as a single task, which does not consider distribution differences in different domains. Fourthly, MML+Mamba4Rec achieves better recommendation performance than MML+SASRec in all domains, which demonstrates the superiority of our MML based on Mamba. Fifthly, MML+SASRec achieves slightly better recommendation performance compared with MML+SASRec^P. Because the three domains are closely related, a domain-shared position embedding matrix can enhance knowledge transfer (Ruder, 2017). Finally, compared with the cross-domain methods, the multi-domain methods can gain the overall performance lift in most cases. Since multi-domain methods can be applied to scenarios with more than two domains, they can transfer more valuable knowledge.

4.3. Analyses of hyper-parameters (RQ2)

In this section, we explore the effect of seven different hyper-parameters on the recommendation performance. Firstly, we explore the impact of the embedding size (i.e., n). From Fig. 4, we can conclude that MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec achieve better recommendation performance when the embedding size is larger in most cases. Because if the embedding size is larger, more knowledge can be captured and transferred. Meanwhile, the recommendation performance of MML improves by a large margin when the embedding size increases from 10 to 20. After that, the recommendation performance increases slowly. It is because most

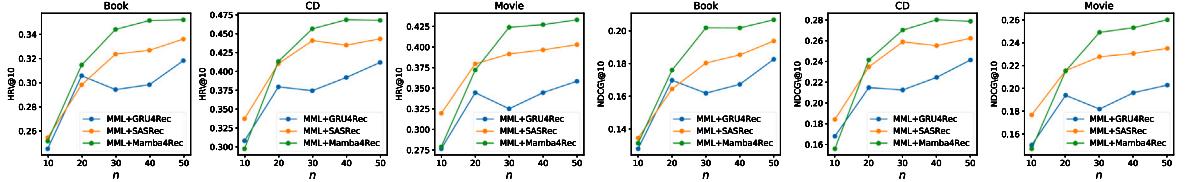


Fig. 4. Recommendation performance w.r.t. the embedding size.

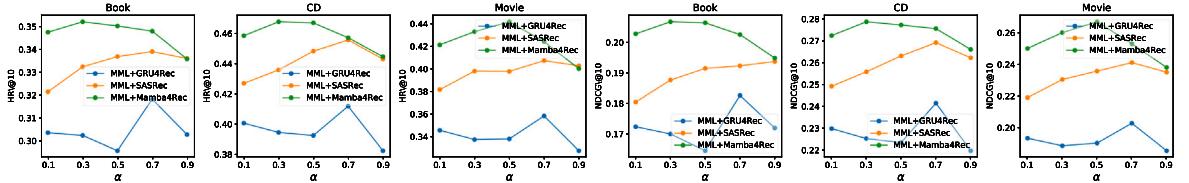


Fig. 5. Recommendation performance w.r.t. the augmentation ratio.

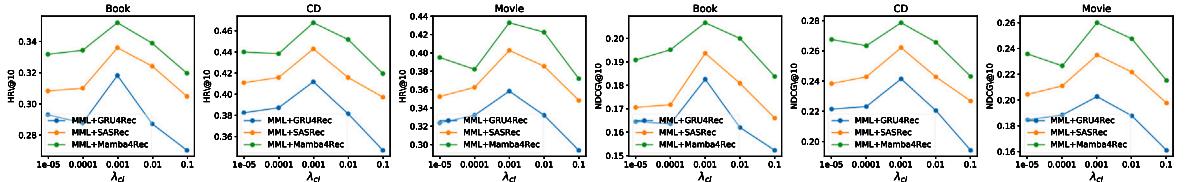


Fig. 6. Recommendation performance w.r.t. the weight of contrastive learning loss.

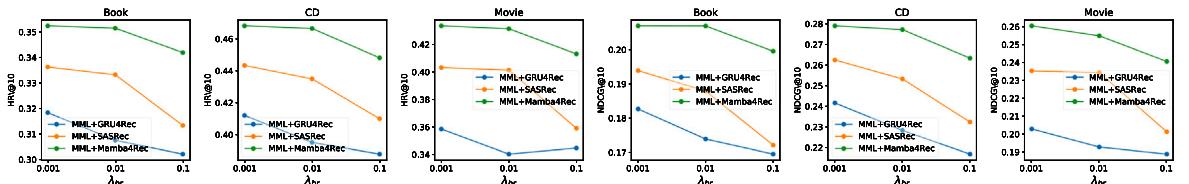


Fig. 7. Recommendation performance w.r.t. the weight of a behavior regularizer.

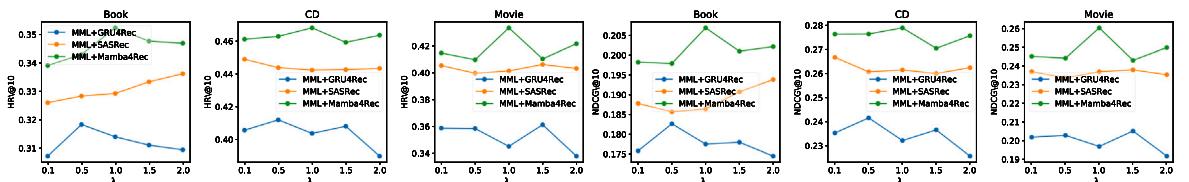


Fig. 8. Recommendation performance w.r.t. the weight of Task 3.

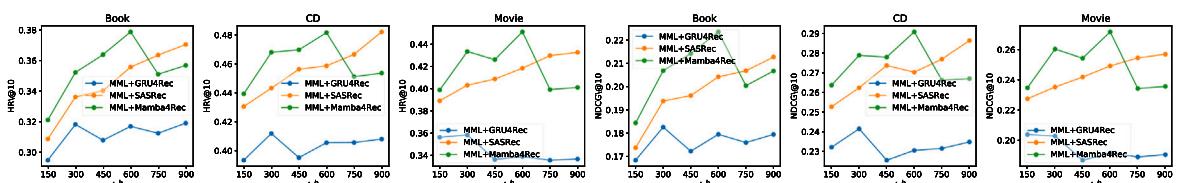


Fig. 9. Recommendation performance w.r.t. the maximum hybrid interaction sequence length.

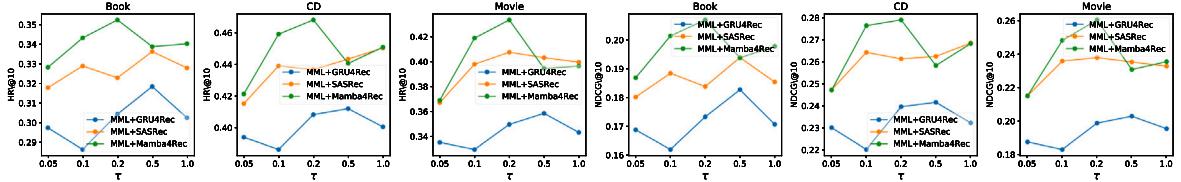


Fig. 10. Recommendation performance w.r.t. the temperature coefficient.

useful knowledge has been learned when the embedding size equals 20. When the embedding size is extremely small (i.e., 10), MML+Mamba4Rec is inferior to MML+SASRec. The small embedding size leads to the limited state capacity of Mamba (Gu & Dao, 2023), which results in failing to capture long-range dependencies (Gu & Dao, 2023). Meanwhile, MML+SASRec can capture long-range dependencies in one step via the self-attention mechanism while MML+Mamba4Rec performs state updates in a recurrent manner. Therefore, MML+SASRec can better capture long-range dependencies than MML+Mamba4Rec under low-dimensional embeddings.

Secondly, we explore the impact of the augmentation ratio α for the crop operation. From Fig. 5, we can conclude that a proper augmentation ratio is vital in data augmentation. For MML based on different backbone models (i.e., GRU4Rec, SASRec and Mamba4Rec), the optimal values are different. It is because GRU4Rec, SASRec and Mamba4Rec have different abilities in capturing preferences from the interaction sequences. For MML, the recommendation performance first increases and then decreases when the augmentation ratio becomes larger in most cases. If the augmentation ratio is small, the augmented hybrid-domain sequential preferences learned from the augmented interaction sequences might not reflect the users' true preferences. By contrastive learning, original hybrid-domain sequential preferences might be damaged, which hurts the recommendation performance. If the augmentation ratio is large, the augmented hybrid-domain sequential preferences learned from the augmented interaction sequences might be the same as the original hybrid-domain sequential preferences. Therefore, MML might fail to learn unbiased hybrid-domain sequential preferences by leveraging contrastive learning.

Thirdly, we explore the effect of the weight of the contrastive learning loss (i.e., λ_{cl}). The results are shown in Fig. 6. We can see that the recommendation performance becomes better with the increasing value of λ_{cl} , and then gradually decreases. If λ_{cl} is too small, the unbiased hybrid-domain sequential preferences cannot be obtained. If λ_{cl} is too large, the original hybrid-domain sequential preferences will be damaged.

Fourthly, we explore the effect of the weight of the behavior regularizer (i.e., λ_{br}). As shown in Fig. 7, we can conclude that when λ_{br} becomes larger, the recommendation performance will decrease. Because the knowledge contained in item embeddings might be damaged when λ_{br} is too large. Meanwhile, for MML+SASRec, MML+GRU4Rec and MML+Mamba4Rec, all optimal values of λ_{br} are 0.001. It is because the behavior regularizer focuses on item embeddings rather than encoders which are leveraged to capture sequential preferences. The main differences of the three MDSR models (i.e., MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec) are that they leverage different backbone models as encoders. Therefore, the optimal values of λ_{br} remain the same under different backbone models.

Fifthly, we explore the effect of the weight about Task 3 (i.e., λ_3). As shown in Fig. 8, for MML, the recommendation performance first improves then decreases in most cases. We can conclude that λ_3 cannot be too large or too small. If it is too large, the intra-domain sequential preferences might be damaged. If it is too small, the correlations among different domains might not be learned well, which hinders knowledge transfer.

Sixthly, we explore the effect of the maximum hybrid interaction sequence length L^h for the recommendation performance. Meanwhile, L^d is set to $\frac{L^h}{|\mathcal{D}|}$ (i.e., $\frac{L^h}{3}$). MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec have different abilities to extract sequential preferences from long interaction sequences. Therefore, the recommendation performance of MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec varies differently with L^h increasing as shown in Fig. 9. For MML+GRU4Rec, the recommendation performance improves by a large margin when L^h increases from 150 to 300. Because longer interaction sequences can ensure that the sequential preferences are captured more comprehensively. And then the recommendation performance of MML+GRU4Rec does not increase but instead decreases as L^h continues to increase. We think there are two reasons. Firstly, MML+GRU4Rec is difficult to capture sequential preferences and learn long-term dependencies from too long interaction sequences (Gu & Dao, 2023; Hidasi et al., 2016). Secondly, some too long interaction sequences might introduce some noise which damages the recommendation performance (Pi et al., 2020). For MML+SASRec, the recommendation performance improves when L^h becomes larger. We conclude that MML+SASRec can capture sequential preferences more comprehensively when L^h becomes larger. However, the recommendation performance of MML+Mamba4Rec does not improve and even decreases as L^h increases from 600 to 900. It might be because that MML+SASRec can filter more noise by a self-attention mechanism and learn more robust sequential preferences when L^h is larger.

Lastly, we explore the effect of the temperature coefficient τ . As shown in Fig. 10, with an increasing value of τ , the recommendation performance of MML first increases and then decreases in most cases. If τ is too small, MML might focus on samples which are similar but fail to consider some samples that are not similar. If τ is too large, negative samples and positive samples might not be distinguished well. Meanwhile, the optimal value of τ is different for our MML with different backbone models (i.e., GRU4Rec, SASRec and Mamba4Rec). The main reason is their different abilities to extract features.

Table 4

Recommendation performance of different variants of our MML+GRU4Rec.

Models	Book		CD		Movie	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
ISPL+GRU4Rec w/o \mathcal{L}_3	0.1295	0.2400	0.1662	0.3024	0.1466	0.2716
ISPL+GRU4Rec	0.1436	0.2625	0.1812	0.3314	0.1513	0.2816
HSPL+GRU4Rec w/o $\mathcal{L}_{br}, \mathcal{L}_{cl}$	0.1314	0.2396	0.1731	0.3083	0.1505	0.2772
HSPL+GRU4Rec w/o \mathcal{L}_{br}	0.1404	0.2566	0.1817	0.3209	0.1641	0.3013
HSPL+GRU4Rec w/o \mathcal{L}_{cl}	0.1432	0.2607	0.2019	0.3413	0.1926	0.3285
HSPL+GRU4Rec	0.1573	0.2875	0.2136	0.3711	<u>0.2018</u>	0.3500
MML+GRU4Rec w/o \mathcal{L}_{br}	<u>0.1753</u>	<u>0.3109</u>	<u>0.2313</u>	<u>0.3994</u>	0.1938	0.3466
MML+GRU4Rec w/o \mathcal{L}_{cl}	0.1672	0.2969	0.2210	0.3838	0.1847	0.3246
MML+GRU4Rec w/o \mathcal{L}_3	0.1638	0.2923	0.2239	0.3867	0.2002	<u>0.3508</u>
MML+GRU4Rec w/o $\mathcal{L}_{cl}, \mathcal{L}_{br}, \mathcal{L}_3$	0.1632	0.2909	0.2255	0.3902	0.1911	0.3377
MML+GRU4Rec	0.1790	0.3125	0.2404	0.4121	0.2044	0.3589

Table 5

Recommendation performance of different variants of our MML+SASRec.

Models	Book		CD		Movie	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
ISPL+SASRec w/o \mathcal{L}_3	0.1464	0.2702	0.2067	0.3604	0.1752	0.3088
ISPL+SASRec	0.1652	0.3032	0.2116	0.3773	0.1782	0.3219
HSPL+SASRec w/o $\mathcal{L}_{br}, \mathcal{L}_{cl}$	0.1402	0.2670	0.2061	0.3556	0.1832	0.3277
HSPL+SASRec w/o \mathcal{L}_{br}	0.1571	0.2901	0.2219	0.3780	0.2030	0.3632
HSPL+SASRec w/o \mathcal{L}_{cl}	0.1532	0.2786	0.2206	0.3704	0.2088	0.3575
HSPL+SASRec	0.1574	0.2970	0.2204	0.3994	0.2208	0.3952
MML+SASRec w/o \mathcal{L}_{br}	<u>0.1877</u>	<u>0.3312</u>	0.2551	0.4325	0.2272	0.3949
MML+SASRec w/o \mathcal{L}_{cl}	0.1728	0.3114	0.2430	0.4125	0.2009	0.3500
MML+SASRec w/o \mathcal{L}_3	0.1867	0.3260	<u>0.2625</u>	<u>0.4383</u>	<u>0.2354</u>	<u>0.3978</u>
MML+SASRec w/o $\mathcal{L}_{cl}, \mathcal{L}_{br}, \mathcal{L}_3$	0.1627	0.2994	0.2364	0.4075	0.1959	0.3449
MML+SASRec	0.1914	0.3362	0.2644	0.4455	0.2396	0.4079

Table 6

Recommendation performance of different variants of our MML+Mamba4Rec.

Models	Book		CD		Movie	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
ISPL+Mamba4Rec w/o \mathcal{L}_3	0.1682	0.2937	0.2209	0.3764	0.2040	0.3460
ISPL+Mamba4Rec	0.1734	0.3068	0.2240	0.3920	0.2021	0.3517
HSPL+Mamba4Rec w/o $\mathcal{L}_{pr}, \mathcal{L}_{cl}$	0.1569	0.2856	0.2244	0.3724	0.2033	0.3473
HSPL+Mamba4Rec w/o \mathcal{L}_{br}	0.1465	0.2665	0.2209	0.3631	0.2078	0.3548
HSPL+Mamba4Rec w/o \mathcal{L}_{cl}	0.1581	0.2830	0.2264	0.3779	0.2163	0.3624
HSPL+Mamba4Rec	0.1621	0.2976	0.2300	0.4024	0.2221	0.3825
MML+Mamba4Rec w/o \mathcal{L}_{br}	0.1957	<u>0.3394</u>	<u>0.2730</u>	<u>0.4541</u>	0.2478	0.4114
MML+Mamba4Rec w/o \mathcal{L}_{cl}	0.1868	0.3226	0.2647	0.4413	0.2284	0.3839
MML+Mamba4Rec w/o \mathcal{L}_3	<u>0.1962</u>	0.3361	0.2728	0.4529	<u>0.2542</u>	<u>0.4237</u>
MML+Mamba4Rec w/o $\mathcal{L}_{cl}, \mathcal{L}_{br}, \mathcal{L}_3$	0.1822	0.3213	0.2665	0.4382	0.2324	0.3869
MML+Mamba4Rec	0.2074	0.3529	0.2801	0.4677	0.2558	0.4270

4.4. Ablation study (RQ3)

We report the results about ablation study in [Tables 4–6](#) for MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec, respectively. Note that w/o represents removing the corresponding module or the corresponding task, where w/o \mathcal{L}_{br} , w/o \mathcal{L}_{cl} and w/o \mathcal{L}_3 denotes removing the behavior regularizer, contrastive learning and [Task 3](#), respectively. HSPL+GRU4Rec, HSPL+SASRec and HSPL+Mamba4Rec mean that HSPL is designed based on GRU4Rec, SASRec and Mamba4Rec, respectively. ISPL+GRU4Rec, ISPL+SASRec and ISPL+Mamba4Rec mean that ISPL is designed based on GRU4Rec, SASRec and Mamba4Rec, respectively. We have the following observations:

- For MML based on different backbone models (i.e., GRU4Rec, SASRec and Mamba4Rec), the performance of our MML will decrease in all cases without one of the two modules (i.e., ISPL and HSPL). Therefore, hybrid-domain and intra-domain sequential preferences are all important for improving the recommendation performance. HSPL only leverages hybrid interaction sequences,

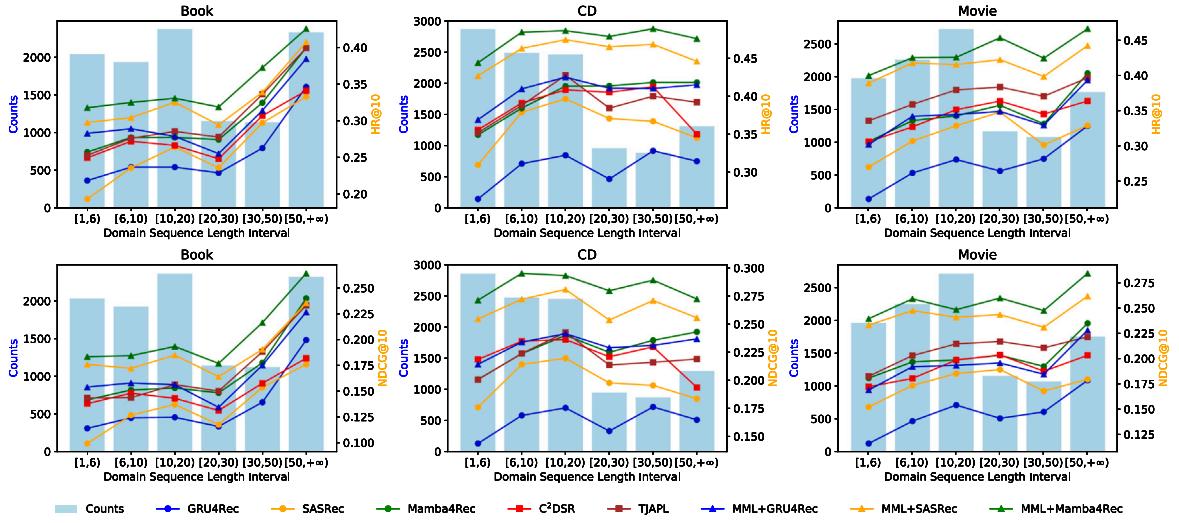


Fig. 11. Recommendation performance w.r.t. different interaction sequence lengths.

which leads to the problem that intra-domain sequential preferences might not be fully captured. ISPL only leverages separate interaction sequences in each domain, which fails to capture important sequential patterns in hybrid interaction sequences.

- For MML and ISPL, removing [Task 3](#) hurts the recommendation performance in nearly all cases. Therefore, [Task 3](#) helps learn the correlations among different domains and achieves knowledge transfer. Meanwhile, compared with MML trained without [Task 3](#), ISPL trained without [Task 3](#) suffers the larger reduction rate of the recommendation performance in most cases. It is because [Task 3](#) for ISPL is the only method to transfer knowledge across different domains.
- For MML and HSPL based on different backbone models, the recommendation performance decreases in nearly all cases without leveraging contrastive learning. Therefore, unbiased and robust hybrid-domain sequential preferences are important for improving the recommendation performance in all domains.
- For MML and HSPL based on different backbone models, removing the behavior regularizer degrades the recommendation performance in nearly all cases. It is because the behavior regularizer can alleviate the disconnection of the interacted items belonging to the same domain in hybrid interaction sequences. Meanwhile, compared with MML, the recommendation performance of HSPL decreases by a large margin in the situation without the behavior regularizer. The main reason is that the ISPL module in MML can fully capture the item-to-item correlations in the same domain. Therefore, the behavior regularizer plays a limited role in improving the recommendation performance for MML. HSPL only leverages hybrid interaction sequences as input. Therefore, the behavior regularizer of HSPL plays an important role in building connection for items belonging to the same domain and improving the recommendation performance.
- For MML without contrastive learning, the recommendation performance drops sharply and is even lower than MML without contrastive learning, the behavior regularizer and [Task 3](#) in some cases. It might be because the behavior regularizer becomes ineffective without contrastive learning. The behavior regularizer reduces the distance of item embeddings from different domains, therefore these item embeddings contain some similar knowledge. However, these similar knowledge might be biased, which might damage the recommendation performance. More unbiased hybrid-domain sequential preferences are obtained via contrastive learning, which indirectly helps these item embeddings contain more unbiased hybrid-domain knowledge. Therefore, in HSPL and MML, combining the behavior regularizer and contrastive learning achieves better recommendation performance.
- For MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec, removing the behavior regularizer, [Task 3](#) or contrastive learning, the recommendation performance will decrease. Therefore, our MML is a model-agnostic multi-domain sequential recommendation framework.
- Compared with HSPL based on GRU4Rec and SASRec, HSPL+Mamba4Rec achieves better performance. The same situation appears in ISPL and MML. Therefore, Mamba excels in modeling sequences ([Gu & Dao, 2023](#)).

4.5. Sparsity analyses (RQ4)

In this section, we will analyze the recommendation performance of our MML, C²DSR, TJAPL, GRU4Rec, SASRec and Mamba4Rec when the lengths of users' interaction sequences vary in each domain. In each domain, we divide users into six groups according to the lengths of the interaction sequences in the current domain. The results are shown in [Fig. 11](#). We can draw some conclusions as follows:

Table 7

Recommendation performance of our MML+GRU4Rec by removing one of the three domains.

Models	Book		CD		Movie	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
MML+GRU4Rec w/o Book	–	–	<u>0.2252</u>	<u>0.3926</u>	<u>0.1913</u>	<u>0.3358</u>
MML+GRU4Rec w/o CD	<u>0.1702</u>	<u>0.2993</u>	–	–	0.1894	<u>0.3358</u>
MML+GRU4Rec w/o Movie	0.1626	0.2878	0.2021	0.3580	–	–
MML+GRU4Rec	0.1790	0.3125	0.2404	0.4121	0.2044	0.3589

Table 8

Recommendation performance of our MML+SASRec by removing one of the three domains.

Models	Book		CD		Movie	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
MML+SASRec w/o Book	–	–	<u>0.2637</u>	<u>0.4408</u>	<u>0.2358</u>	<u>0.4001</u>
MML+SASRec w/o CD	<u>0.1853</u>	<u>0.3289</u>	–	–	0.2260	0.3924
MML+SASRec w/o Movie	0.1841	0.3242	0.2371	0.4111	–	–
MML+SASRec	0.1914	0.3362	0.2644	0.4455	0.2396	0.4079

Table 9

Recommendation performance of our MML+Mamba4Rec by removing one of the three domains.

Models	Book		CD		Movie	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
MML+Mamba4Rec w/o Book	–	–	<u>0.2680</u>	<u>0.4540</u>	<u>0.2457</u>	0.4151
MML+Mamba4Rec w/o CD	<u>0.2036</u>	<u>0.3463</u>	–	–	0.2456	<u>0.4160</u>
MML+Mamba4Rec w/o Movie	0.1975	0.3374	0.2588	0.4373	–	–
MML+Mamba4Rec	0.2074	0.3529	0.2801	0.4677	0.2558	0.4270

- In six groups, MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec demonstrate better recommendation performance than GRU4Rec, SASRec and Mamba4Rec, respectively. It is because our MML can transfer more knowledge from other domains for improving the recommendation performance.
- When the lengths of the interaction sequences are located between 1 and 6, our MML achieves better recommendation performance than C²DSR and TJAPL, which shows that our MML is able to alleviate the cold-start and data sparsity problems better by transferring more useful knowledge.
- With the lengths of the interaction sequences increasing, the recommendation performance of all models (i.e., GRU4Rec, SASRec, Mamba4Rec, C²DSR, TJAPL and our MML) improves in most cases in Book and Movie domains. It is because these models can extract more valuable knowledge from longer interaction sequences. When the lengths of users' interaction sequences are less than 10 in CD domain, the recommendation performance of all models improves. However, as the lengths of users' interaction sequences increase, the recommendation performance of all models does not improve and even decreases in CD domain. The main reason is that longer interaction sequences might contain more noise (Pi et al., 2020), which causes difficulty in extracting valuable knowledge.
- When the interaction sequences are short, our MML achieves better performance by a large margin in all domains compared with the corresponding backbone model. When the interaction sequences are long, our MML achieves better performance by a small margin. It is because our MML mainly aims to alleviate the data sparsity and cold-start problems. We can thus see that the transferred knowledge plays an important role when there exist data sparsity and cold-start problems (Ni et al., 2023).

4.6. Domain analyses (RQ5)

In this section, we remove one domain from three domains to analyze the effect of domains on recommendation performance. Note that w/o Book, w/o CD and w/o Movie refer to removing Book domain, CD domain and Movie domain, respectively. From the results shown in Tables 7–9, we can draw some conclusions as follows:

- For MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec, the recommendation performance of all models will decrease without one of the three domains. Therefore, each domain makes contributions for improving the recommendation performance. It is because our MML can transfer more valuable knowledge from more domains for improving the recommendation performance.
- MML+Mamba4Rec still demonstrates better results compared with MML+GRU4Rec and MML+SASRec without one of the three domains in all cases. Therefore, the backbone model plays an important role in improving the recommendation performance.
- MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec all achieve the best recommendation performance in CD domain with the help of Movie domain. In the real world, CD domain and Movie domain are also strongly associated. MML+GRU4Rec,

Table 10

Recommendation performance using different augmentation methods for our MML+GRU4Rec.

Models	Book		CD		Movie	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
MML+GRU4Rec w/o \mathcal{L}_{cl}	0.1672	0.2969	0.2210	0.3838	0.1847	0.3246
MML+GRU4Rec (<i>mask & balance</i>)	0.1632	0.2905	0.2234	0.3863	0.1880	0.3312
MML+GRU4Rec (<i>resort</i>)	<u>0.1719</u>	<u>0.3032</u>	<u>0.2249</u>	<u>0.3910</u>	<u>0.1930</u>	<u>0.3437</u>
MML+GRU4Rec (<i>mask</i>)	0.1667	0.2959	0.2167	0.3780	0.1857	0.3302
MML+GRU4Rec (<i>crop</i>)	0.1790	0.3125	0.2404	0.4121	0.2044	0.3589

Table 11

Recommendation performance using different augmentation methods for our MML+SASRec.

Models	Book		CD		Movie	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
MML+SASRec w/o \mathcal{L}_{cl}	0.1728	0.3114	0.2430	0.4125	0.2009	0.3500
MML+SASRec (<i>mask & balance</i>)	<u>0.1885</u>	<u>0.3339</u>	<u>0.2595</u>	<u>0.4398</u>	<u>0.2328</u>	<u>0.3994</u>
MML+SASRec (<i>resort</i>)	0.1821	0.3222	0.2524	0.4284	0.2191	0.3805
MML+SASRec (<i>mask</i>)	0.1811	0.3233	0.2511	0.4288	0.2247	0.3878
MML+SASRec (<i>crop</i>)	0.1914	0.3362	0.2644	0.4455	0.2396	0.4079

Table 12

Recommendation performance using different augmentation methods for our MML+Mamba4Rec.

Models	Book		CD		Movie	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
MML+Mamba4Rec w/o \mathcal{L}_{cl}	0.1868	0.3226	0.2647	0.4413	0.2284	0.3839
MML+Mamba4Rec (<i>mask & balance</i>)	0.1945	0.3358	0.2653	0.4425	0.2349	0.3937
MML+Mamba4Rec (<i>resort</i>)	<u>0.2037</u>	<u>0.3486</u>	<u>0.2749</u>	<u>0.4627</u>	<u>0.2530</u>	<u>0.4262</u>
MML+Mamba4Rec (<i>mask</i>)	0.1991	0.3425	0.2709	0.4537	0.2431	0.4080
MML+Mamba4Rec (<i>crop</i>)	0.2074	0.3529	0.2801	0.4677	0.2558	0.4270

MML+SASRec and MML+Mamba4Rec all transfer more valuable knowledge from Movie domain to CD domain. Therefore, each domain plays a different role in improving the recommendation performance in other domains.

4.7. Data augmentation methods (RQ6)

In this section, we use three additional kinds of data augmentation methods to conduct contrastive learning. Note that *mask & balance* is a variant of *mask*, which masks previously interacted items. For an augmented sequence of each user, the number of interactions in each domain is the same, which equals to the minimum number of interactions in each domain in the user's original sequence. And, each original hybrid sequence only has a unique augmented sequence. The *mask* and *resort* are the same as those of CL4SRec (Xie, Sun et al., 2022). As shown in Tables 10–12, we have following observations:

- Other data augmentation methods like *mask* and *resort* achieve better recommendation performance compared with approaches without data augmentation in nearly all cases. It is because *mask* can only change the number of interacted items in each domain and *resort* can only change the order of the interacted items in hybrid interaction sequences. Therefore, MML using *mask* and *resort* can learn different distributions from the augmented sequences. By contrastive learning, more unbiased and robust hybrid-domain sequential preferences can be obtained to some extent. The unbiased and robust hybrid-domain sequential preferences are vital for improving the recommendation performance in all domains.
- MML using *mask & balance* can also achieve better performance than that without contrastive learning in most cases. For augmented interaction sequences generated via this data augmentation method, the number of interacted items in each domain is the same. Therefore, MML can learn more unbiased hybrid-domain sequential preferences from the augmented sequences. Contrastive learning can minimize the distance between biased hybrid-domain sequential preferences and unbiased hybrid-domain sequential preferences, so the biased hybrid-domain sequential preferences can be alleviated to some extent. Meanwhile, this phenomenon also shows that data imbalance is highly prevalent.
- Compared with *mask* and *resort*, *crop* achieves better performance. Using *crop*, we can obtain a continuous sub-sequence. This sub-sequence may have completely different distributions compared with the original hybrid sequence. Note that *mask* and *resort* cannot achieve this. It is because *crop* can change the number of the interacted items in each domain and the positions of interacted items in hybrid interaction sequences simultaneously. By leveraging contrastive learning, the biased hybrid-domain sequential preferences can be more effectively alleviated (Li, Xie et al., 2023; Xie, Sun et al., 2022).

Table 13
Efficiency of different models on the Amazon datasets.

Method	GPU memory	Training time	Inference time
GRU4Rec	1.85 GB	3.15 s/epoch	12 s
SASRec	1.76 GB	3.15 s/epoch	26 s
Mamba4Rec	1.74 GB	3.15 s/epoch	24 s
MGCL	2.56 GB	128.25 s/epoch	239 s
GRU4Rec-hybrid	2.12 GB	7.7 s/epoch	38 s
SASRec-hybrid	2.11 GB	7.9 s/epoch	81 s
Mamba4Rec-hybrid	2.13 GB	7.65 s/epoch	78 s
MML+GRU4Rec	3.37 GB	17.8 s/epoch	63 s
MML+SASRec	3.25 GB	19.8 s/epoch	161 s
MML+Mamba4Rec	3.82 GB	19.65 s/epoch	148 s

- Though *crop* might obtain the sub-sequences with fewer interactions in data-sparse domains than the original sequences, it outperforms *mask & balance* which does not delete interactions in the domains with the least number of interactions. There are three reasons: (1) The crop operation involves randomness. By the crop operation, the sub-sequences might contain more interactions in some data-sparse domains than in some data-rich domains. Therefore, the augmented hybrid-domain sequential preferences might focus on data-sparse domains. By leveraging contrastive learning, the biased sequential preferences can be alleviated. 2) However, the obtained sub-sequences might also obtain fewer interactions in some data-sparse domains. This situation gives rise to the augmented hybrid-domain sequential preferences containing fewer sequential preferences in data-sparse domains. By leveraging contrastive learning, sequential preferences in data-sparse domains can be recovered by transferring knowledge from other domains. By transferring knowledge from data-rich domains to data-sparse domains, hybrid-domain sequential preferences contain more sequential preferences in data-sparse domains, which indirectly alleviates the data imbalance problem. (3) The sub-sequences contain diverse sequential patterns, which contribute to increasing the recommendation performance.

4.8. Complexity analyses (RQ7)

For MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec, the time complexities are $O(L^h)$, $O((L^h)^2)$ and $O(L^h)$ in training and inference. Meanwhile, for GRU4Rec-hybrid, SASRec-hybrid and Mamba4Rec-hybrid, the time complexities are also $O(L^h)$, $O((L^h)^2)$ and $O(L^h)$ in training and inference. Therefore, using the same backbone model, the time complexity of our MML is the same as that of an MDSR model taking only hybrid interaction sequences as input. We can also see our MML+Mamba4Rec can achieve linear time complexity. If we extend C²DSR to MDSR, the time complexity during training is $O(|D| + 1)(L^h)^2$. The time complexity linearly increases with the number of domains. Therefore, our MML is more effective. The specific theoretical analyses are shown in the subsequent paragraphs.

During the training of MML+SASRec, the time complexity is $O(2(L^h)^2)$ for the HSPL module. Meanwhile, the time complexity is $O(\sum_{d \in D} (L^d)^2) \leq O((\sum_{d \in D} L^d)^2) = O((L^h)^2)$ for the ISPL module. Therefore, the time complexity of MML+SASRec during training is $O(2(L^h)^2 + (L^h)^2) = O(3(L^h)^2) = O((L^h)^2)$. During the inference of MML+SASRec for domain d , the time complexity is $O((L^h)^2)$ for the HSPL module. Meanwhile, the time complexity is $O((L^d)^2)$ for the ISPL module. Therefore, the time complexity of MML+SASRec during inference for domain d is $O((L^h)^2 + (L^d)^2) \leq O(2(L^h)^2) = O((L^h)^2)$. As for MML+SASRec, the time complexity of training and inference is $O((L^h)^2)$.

During the training of MML+GRU4Rec and MML+Mamba4Rec, the time complexity is $O(2L^h)$ for the HSPL module. Meanwhile, the time complexity is $O(\sum_{d \in D} L^d) = O(L^h)$ for the ISPL module. Therefore, the time complexity of MML+GRU4Rec and MML+Mamba4Rec during training is $O(2L^h + L^h) = O(3L^h) = O(L^h)$. During the inference of MML+GRU4Rec and MML+Mamba4Rec for domain d , the time complexity is $O(L^h)$ for the HSPL module. Meanwhile, the time complexity is $O(L^d)$ for the ISPL module. Therefore, the time complexity of MML+GRU4Rec and MML+Mamba4Rec during inference for domain d is $O(L^h + L^d) \leq O(2L^h) = O(L^h)$. As for, MML+GRU4Rec and MML+Mamba4Rec, the time complexity of training and inference is $O(L^h)$.

If we extend C²DSR to MDSR by using the same split operation, the lengths of both a hybrid interaction sequence from all domains and a separate interaction sequence in each domain are the same for each user. The time complexity is $O((L^h)^2)$ for modeling the hybrid interaction sequence. Meanwhile, the time complexity is $O(\sum_{d \in D} (L^h)^2) = O(|D|(L^h)^2)$ for modeling the separate interaction sequence in each domain. The time complexity is $O(|D| + 1)(L^h)^2$ in all.

Then, we evaluate the efficiency of our MML through extensive experiments. For MDSR models (i.e., GRU4Rec-hybrid, SASRec-hybrid, Mamba4Rec-hybrid, MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec), the GPU memory and the time of inference and training are measured in all domains. For single-domain sequential recommendation models (i.e., GRU4Rec, SASRec and Mamba4Rec), the GPU memory and the time of inference and training are measured in one single domain (i.e., Book). For a relatively complex CDSR model (i.e., MGCL), the GPU memory and the time of inference and training are measured in one single source domain and one single target domain. We measure the efficiency of all models in a single Nvidia Tesla V100 with 16 GB of memory. The inference time is measured for all users. The training time is measured in 20 epochs, for which the average value is reported. From Table 13, we can draw some conclusions as follows:

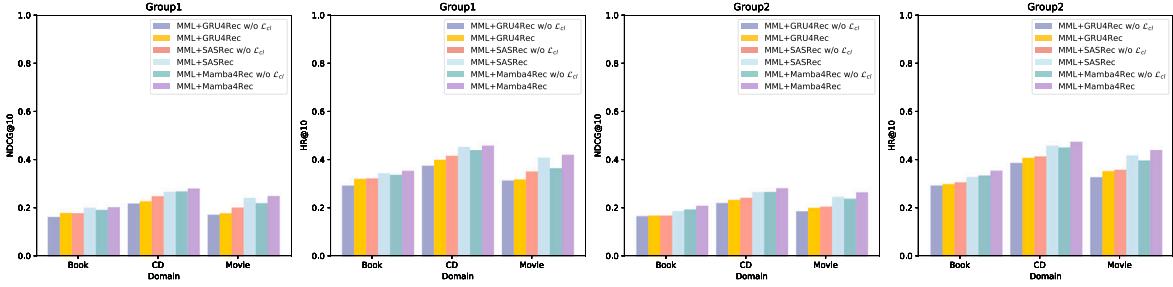


Fig. 12. Recommendation performance on two different groups.

- SASRec, GRU4Rec and Mamba4Rec consume comparable time while training, but the inference time of GRU4Rec is significantly less than that of SASRec and Mamba4Rec. The reason is that the process of preparing the training data consumes lots of time. However, this part of time is negligible in inference.
- GRU4Rec consumes less inference time than SASRec and Mamba4Rec. In the meantime, the inference time of Mamba4Rec is slightly less than that of SASRec. Though the time complexity of Mamba4Rec and GRU4Rec is linear, the actual inference time is affected by many factors. It is worth noting that the inference time of Mamba4Rec is significantly less than that of SASRec in the original paper introducing Mamba4Rec. It is because the number of Mamba layers is 1 in the original paper (Liu, Lin et al., 2024). In our experimental settings, both the number of self-attention blocks and the number of Mamba layers are 2.
- MML+GRU4Rec leverages $|D| + 2$ (i.e., 5) interaction sequences for each user in training. Compared with GRU4Rec-hybrid which only leverages a hybrid interaction sequence for each user, the training time of MML+GRU4Rec should be significantly less than $|D| + 2$ times of GRU4Rec-hybrid in theory. During actual training, the training time of MML+GRU4Rec is 2.3 times of GRU4Rec-hybrid. For MML+SASRec (or MML+Mamba4Rec) and SASRec-hybrid (or Mamba4Rec-hybrid), the results are similar. During inference, our MML+GRU4Rec only leverages two interaction sequences to predict the next interacted items for each user in each domain. During actual inference, the inference time of MML+GRU4Rec is less than two times of GRU4Rec-hybrid. For MML+SASRec (MML+Mamba4Rec) and SASRec-hybrid (Mamba4Rec-hybrid), we can also have some similar results. Compared with a relatively complex CDSR model (i.e., MGCL), the inference and training time of our MML is significantly less than that of MGCL. Therefore, our MML is efficient-yet-effective.
- MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec consume about 3 GB GPU memory. GRU4Rec-hybrid, SASRec-hybrid and Mamba4Rec-hybrid consume about 2 GB GPU memory. Therefore, the consumption of GPU memory is partially related to backbone models. Meanwhile, compared with models only using hybrid interaction sequences, our MML only consumes 1.5 times the GPU memory. Therefore, our MML is efficient in the usage of GPU memory.
- Mamba4Rec and SASRec consume comparable GPU memory. SASRec-hybrid and Mamba4Rec-hybrid, as well as MML+SASRec and MML+Mamba4Rec, show similar GPU memory consumption. However, some related works (Liu, Lin et al., 2024; Liu et al., 2025) show Mamba-based models have lower memory consumption than Transformer-based models. There are two reasons: (1) The number of Mamba layers is 2 in our experimental settings, which results in more GPU memory consumption. (2) The number of attention heads is set to 1 rather than 8,³ therefore the Transformer-based models consume less GPU memory (Kang & McAuley, 2018).

4.9. Group analyses (RQ8)

User u might interact with a few items in a domain, but a large number of items in other domains, which leads to the data imbalance problem. An MDSR model might not fully learn user u 's sequential preferences in the domain with sparse interactions. In this section, to analyze the effectiveness of our MML for tackling the data imbalance problem, we divide the users in each domain into two groups. If a user interacts with the minimum number of items in the current domain as compared with that in other domains, the user will be classified into Group 1, and Group 2 otherwise. Therefore, for users in Group 1 of a specific domain, modeling hybrid interaction sequences might not fully capture sequential preferences in the current domain. For users in Group 2 of a specific domain, the captured hybrid-domain sequential preferences might focus on the current domain. We conduct experiments on two groups with and without contrastive learning. The experimental results are shown in Fig. 12. We can draw some conclusions as follows:

- For Group 2, our MML achieves better recommendation performance than Group 1 in most cases without contrastive learning, especially in Movie domain. It is because the number of interacted items from the current domain is the smallest value in all domains for users in Group 1, sequential preferences in the current domain might fail to be captured. This

³ <https://github.com/Applied-Machine-Learning-Lab/SIMGA>.

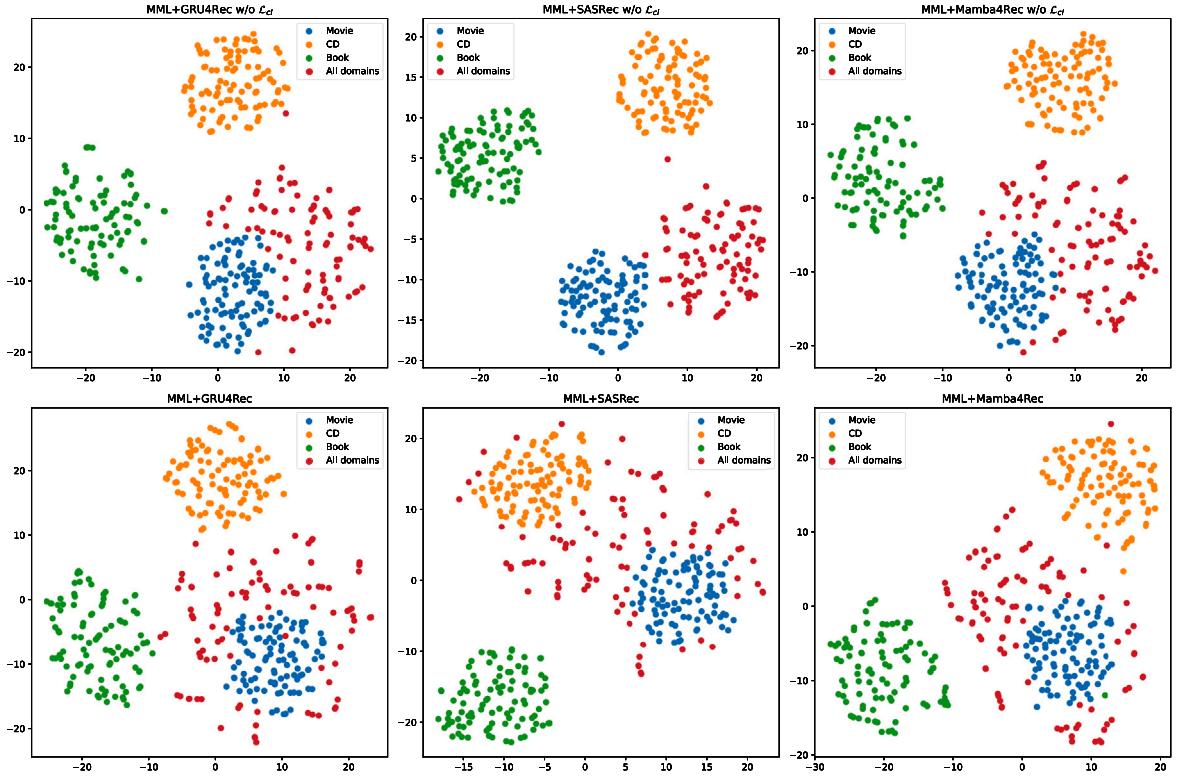


Fig. 13. Visualization of hybrid-domain and intra-domain sequential preferences. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

phenomenon demonstrates the data imbalance problem might affect the recommendation performance, especially in Movie domain. A similar conclusion can be drawn from Tables 10–12. By leveraging contrastive learning, MML demonstrates better recommendation performance by a larger margin in Movie domain than in other domains. It is because the data imbalance problem in Movie domain is alleviated to some extent.

- By leveraging contrastive learning, our MML achieves better recommendation performance across two groups in three domains. This phenomenon demonstrates that the unbiased and robust hybrid-domain sequential preferences can be fully captured by contrastive learning between the original hybrid interaction sequences and the augmented ones for alleviating the data imbalance problem.
- For Group 1, sequential preferences might fail to be captured in the current domain from the original hybrid interaction sequences. Sequential preferences in the current domain might be magnified by modeling the augmented ones. By contrastive learning, the more unbiased hybrid-domain sequential preferences can be captured, which might contain more knowledge in the current domain.
- For Group 2, the recommendation performance across all domains is also improved in all cases. It is because our MML can capture diverse sequential patterns and learn more robust hybrid-domain sequential preferences via data augmentation and contrastive learning. Meanwhile, hybrid-domain sequential preferences containing more sequential preferences in data-sparse domains also contribute to improving the recommendation performance in the current domain.

4.10. Analyses of hybrid-domain sequential preferences (RQ9)

In this section, we study whether our MML can learn more unbiased hybrid-domain sequential preferences by leveraging contrastive learning. To answer this question, we randomly select 100 users. Then, we draw hybrid-domain sequential preferences in all domains and intra-domain sequential preferences in each domain for the 100 users by a common dimension reduction method (i.e., t-SNE) (Van der Maaten & Hinton, 2008). As shown in Fig. 13, red dots represent hybrid-domain sequential preferences in all domains. Meanwhile, green, orange and blue dots represent intra-domain sequential preferences in Book, CD and Movie domains, respectively. The first row refers to the captured hybrid-domain sequential preferences without contrastive learning, and the second row refers to the captured hybrid-domain sequential preferences by leveraging contrastive learning. We can draw some conclusions as follows:

Table 14

The quantitative analyses of hybrid-domain and intra-domain sequential preferences.

Models	MML ^G w/o \mathcal{L}_{cl}	MML ^G	MML ^S w/o \mathcal{L}_{cl}	MML ^S	MML ^M w/o \mathcal{L}_{cl}	MML ^M
Metric	7415.72	6591.25	7517.92	6097.77	7165.90	6146.70

- Intra-domain sequential preferences can be classified into three different clusters according to Book, CD and Movie domains. It is because intra-domain sequential preferences are fully captured by [Task 2](#). Therefore, intra-domain sequential preferences have different distributions in different domains.
- Under the situation without using contrastive learning, hybrid-domain sequential preferences have different correlations with intra-domain sequential preferences in different domains. It is because the data imbalance problem affects the full learning of hybrid-domain sequential preferences. Though users interact with more items in Book domain than in other domains as shown in [Table 2](#), the number of interacted items in CD and Movie domains is more than in Book domain. Meanwhile, CD and Movie domains are closely related intuitively. Therefore, CD and Movie domains can be regarded as one domain to some extent. Therefore, as shown in [Fig. 13](#), hybrid-domain sequential preferences are more relevant to intra-domain sequential preferences in CD and Movie domains than in Book domain. As shown in [Table 2](#), users often interact with more items in Movie domain than CD domain. Therefore, hybrid-domain sequential preferences have stronger relationships with intra-domain sequential preferences in Movie domain than those in CD domain.
- The hybrid-domain sequential preferences become more diverse and unbiased by leveraging contrastive learning. From the second row in [Fig. 13](#), we can see more red dots appearing in the center of green, blue and yellow clusters. Therefore, the captured hybrid-domain sequential preferences have stronger correlations with intra-domain sequential preferences in all domains than those without leveraging contrastive learning. Meanwhile, the biased hybrid-domain sequential preferences are alleviated.

It is difficult to have some quantitative evidence to demonstrate that our MML learns more unbiased hybrid-domain sequential preferences. We propose one relatively reasonable metric to measure the unbiasedness of the hybrid-domain sequential preferences. For each user, we need to calculate $|D|$ different kinds of distances. Dis_u^d denotes the distance between the hybrid-domain sequential preferences and the intra-domain sequential preferences in domain d for user u . In [Table 14](#), we report the sum of distances between red dots and others (i.e., blue, yellow and green dots), which represent the sequential preferences of 100 randomly selected users in [Fig. 13](#). If the sum of distances is smaller, the biased hybrid-domain sequential preferences are usually alleviated more effectively. We adopt such a metric with the following considerations.

- The learned hybrid-domain sequential preferences are usually biased toward data-rich domains rather than data-sparse ones. We aim to capture more sequential preferences in the data-sparse domains from hybrid interaction sequences. Meanwhile, the sequential preferences in the data-rich domains are retained as much as possible. In other words, the distance between the hybrid-domain sequential preferences and the intra-domain sequential preferences in the data-sparse domains should become shorter. Meanwhile, the distance between the hybrid-domain sequential preferences and the intra-domain sequential preferences in the data-rich domains should keep unchanged to the greatest extent. In all, the sum of distances should become smaller.
- If the sum of distances is smaller, the hybrid-domain sequential preferences should contain more sequential preferences from all domains in a whole. If the hybrid-domain sequential preferences contain little knowledge from data-sparse domains, the sum of distances cannot be very small. It is noted that when the condition (i.e., $Dis_u^1 \approx \dots \approx Dis_u^d \approx \dots \approx Dis_u^{|D|}$) satisfies, we do not think unbiased hybrid-domain sequential preferences for user u are obtained. If Dis_u^d is very large, the hybrid-domain sequential preferences cannot contain useful knowledge from domain d . The biased hybrid-domain sequential preferences cannot be alleviated.

We leverage MML^G, MML^S and MML^M to briefly denote MML+GRU4Rec, MML+SASRec and MML+Mamba4Rec in [Tables 14](#) and [15](#). As shown in [Table 14](#), the sum of distances becomes smaller by leveraging contrastive learning. This phenomenon demonstrates that the learned more unbiased hybrid-domain sequential preferences contain more vital knowledge from all domains in a whole.

4.11. Generalization analysis (RQ10)

To evaluate the generalization ability of our MML more comprehensively, we include an additional metric (i.e., MRR@N with $N = 10$) and conduct experiments on the aforementioned three Amazon datasets ([Boka, Niu, & Neupane, 2024](#)).

We further use three Douban datasets (i.e., Book, Movie, Music).⁴ We leverage the same method that processes the Amazon datasets to process the Douban datasets. The statistics of the Douban datasets are shown in [Table 16](#). We select three representative baselines for each class of methods according to the recommendation performance in [Table 3](#). We conduct experiments following the same implementation details in Section [4.1.4](#).

⁴ <https://www.kaggle.com/datasets/fengzhujoey/douban-datasetratingreviewside-information>.

Table 15

Recommendation performance of some representative baselines and our MML on Amazon datasets in terms of MRR@10.

	GRU4Rec	SASRec	Mamba4Rec	CD-SASRec	MGCL	C ² DSR	DS-TF	DS-Mamba	TJAPL	MML ^G	MML ^S	MML ^M
Book	0.1038	0.0989	0.1295	0.1128	0.1050	0.1138	0.1192	0.1217	0.1234	0.1383	<u>0.1473</u>	0.1629
CD	0.1266	0.1449	0.1718	0.1532	0.1593	0.1782	0.1774	0.1805	0.1716	0.1879	<u>0.2090</u>	0.2226
Movie	0.1071	0.1324	0.1571	0.1349	0.1504	0.1428	0.1686	0.1753	0.1536	0.1573	<u>0.1881</u>	0.2032

Table 16

The statistics of the processed Douban datasets.

Domain	Users	Items	Actions	Avg.Len.	Density
All domains	1037	36 314	788 403	760.27	0.0209
Book	1037	8644	76 089	73.37	0.0085
Music	1037	7146	67 805	65.39	0.0091
Movie	1037	20 524	644 509	621.51	0.0303

Table 17

Recommendation performance of some representative baselines and our MML on three Douban datasets.

Models	Book			Music			Movie		
	NDCG@10	HR@10	MRR@10	NDCG@10	HR@10	MRR@10	NDCG@10	HR@10	MRR@10
GRU4Rec	0.1273	0.2555	0.0888	0.1547	0.2919	0.1132	0.3378	0.5223	0.2802
SASRec	0.1662	0.3115	0.1221	0.2047	0.3745	0.1529	0.4203	<u>0.5959</u>	0.3656
Mamba4Rec	0.1588	0.2707	0.1247	0.1932	0.3401	0.1483	0.3629	0.5468	0.3055
CD-SASRec	0.1767	0.3195	0.1334	0.2050	0.3738	0.1536	0.3796	0.5606	0.3228
MGCL	0.1565	0.2993	0.1135	0.1562	0.3047	0.1114	0.2722	0.4767	0.2090
C ² DSR	0.1638	0.3025	0.1218	0.2056	0.3693	0.1558	0.3677	0.5632	0.3066
DS-TF	0.1468	0.2851	0.1050	0.1813	0.3417	0.1328	0.3378	0.5632	0.2669
DS-Mamba	0.1314	0.2617	0.0922	0.1604	0.3218	0.1117	0.2789	0.4937	0.2119
TJAPL	0.1746	0.3266	0.1285	0.2009	0.3835	0.1458	0.3955	0.5985	0.3312
MML+SASRec ^P	0.1925	0.3475	0.1456	<u>0.2193</u>	<u>0.3976</u>	<u>0.1648</u>	<u>0.4033</u>	0.5956	<u>0.3432</u>
MML+GRU4Rec	0.1435	0.2833	0.1015	0.1786	0.3350	0.1316	0.3632	0.5410	0.3078
MML+SASRec	0.1775	0.3353	0.1299	0.2126	0.3877	0.1593	0.3517	0.5432	0.2916
MML+Mamba4Rec	<u>0.1871</u>	<u>0.3359</u>	<u>0.1421</u>	0.2276	0.4076	<u>0.1734</u>	0.3669	0.5471	0.3105

As shown in [Table 15](#), our MML again achieves the best in terms of MRR@10 on the Amazon datasets. From [Table 17](#), we can conclude that our MML achieves the best performance in the Book and Music domains, and exhibits remarkable generalization ability. Specifically, we have the following observations:

- MML performs better than all the representative baselines in two of the three domains. Meanwhile, our MML is a more generic framework. MML achieves better recommendation performance than the corresponding backbone model in most cases, especially for MML+GRU4Rec and MML+Mamba4Rec. From the experimental results, our MML is friendly to Mamba. Though Mamba4Rec is worse than SASRec on the Douban datasets, MML+Mamba4Rec surpasses MML+SASRec in some cases.
- Most of CDSR and MDSR models (e.g., C²DSR and TJAPL) perform worse than the corresponding single-domain models in Movie domain. For the Movie domain, users are associated with abundant interaction records. Therefore, the transferred knowledge plays an insignificant role in improving the recommendation performance for Movie domain. It is thus reasonable that MML+SASRec does not beat SASRec. However, MML+GRU4Rec and MML+Mamba4Rec achieve better recommendation performance than GRU4Rec and Mamba4Rec in all cases, respectively. We can thus see that our MML is able to alleviate negative transfer to some extent.
- Different from the recommendation performance in [Table 3](#), MML+SASRec^P demonstrates superior performance than MML+SASRec in three Douban domains. It is because that all Transformer-based encoders share the same position embedding matrix in MML+SASRec. When domains are not closely related, only one position embedding matrix might not capture distinct characteristics in each domain. The hard parameter sharing approach may cause negative transfer ([Ruder, 2017](#)).

5. Conclusions and future work

In this paper, we study an important problem that has been rarely explored in previous works, i.e., multi-domain sequential recommendation (MDSR). Specifically, we adopt a new perspective, i.e., multiple sequences and multiple tasks, which leads to a novel recommendation model called multi-sequence multi-task learning (MML). Our MML consists of three modules, including HSPL, ISPL and MLP. HSPL captures unbiased hybrid-domain sequential preferences by leveraging contrastive learning. ISPL captures intra-domain sequential preferences in each domain. For the MLP module, we design three tasks and a behavior regularizer in order to train the model more effectively and efficiently. The main merit of our MML is its ability to achieve a good balance between preference learning in each single domain and knowledge transfer across different domains, which makes it perform exceptionally

well in multi-domain recommendation. Meanwhile, our MML is efficient in modeling multiple interaction sequences during training and inference. Extensive experiments on some public datasets show that our MML based on three different backbone models is able to perform significantly better than the state-of-the-art single-domain, cross-domain and multi-domain recommendation methods in most cases.

For future works, we are interested in leveraging graph neural networks to capture collaborative signals. Combining sequential preferences and collaborative signals can further improve the recommendation performance. Moreover, in order to improve the recommendation performance for MDSR models in a multi-domain scenario in an e-commerce, video streaming, or social media platform, we are also interested in leveraging some side information such as multi-modal features to extract rich semantic information and help learn users' preferences more accurately. We also plan to study and deploy our MML in a large-scale real-world recommendation system.

CRediT authorship contribution statement

Liwei Pan: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Data curation. **Weike Pan:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Zhong Ming:** Supervision, Funding acquisition.

Acknowledgments

We thank the support of Guangdong Basic and Applied Basic Research Foundation (No. 2024A1515010122), and National Natural Science Foundation of China (Nos. 62461160311, 62172283 and 62272315).

Data availability

Data will be made available on request.

References

- Alharbi, N., & Caragea, D. (2022). Cross-domain self-attentive sequential recommendations. In *Proceedings of international conference on data science and applications* (pp. 601–614).
- Ariza-Casabona, A., Twardowski, B., & Wijaya, T. K. (2023). Exploiting graph structured cross-domain representation for multi-domain recommendation. In *Proceedings of the 45th European conference on information retrieval* (pp. 49–65).
- Boka, T. F., Niu, Z., & Neupane, R. B. (2024). A survey of sequential recommendation systems: Techniques, evaluation, and future directions. *Information Systems*, Article 102427.
- Cao, J., Cong, X., Sheng, J., Liu, T., & Wang, B. (2022). Contrastive cross-domain sequential recommendation. In *Proceedings of the 31st ACM international conference on information and knowledge management* (pp. 138–147).
- Chen, S., Qian, Y., Li, H., & Lin, C. (2021). Sequential recommendation in online games with multiple sequences, tasks and user levels. In *Proceedings of the 17th international symposium on spatial and temporal databases* (pp. 130–140).
- Chen, S., Xu, Z., Pan, W., Yang, Q., & Ming, Z. (2024). A survey on cross-domain sequential recommendation. arXiv preprint [arXiv:2401.04971](https://arxiv.org/abs/2401.04971).
- Ding, Y., Li, H., Chen, K., & Shou, L. (2023). TPUF: Enhancing cross-domain sequential recommendation via transferring pre-trained user features. In *Proceedings of the 32nd ACM international conference on information and knowledge management* (pp. 410–419).
- Fu, Z., Li, X., Wu, C., Wang, Y., Dong, K., Zhao, X., et al. (2023). A unified framework for multi-domain CTR prediction via large language models. arXiv preprint [arXiv:2312.10743](https://arxiv.org/abs/2312.10743).
- Gu, A., & Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint [arXiv:2312.00752](https://arxiv.org/abs/2312.00752).
- Guo, L., Wang, C., Wang, X., Zhu, L., & Yin, H. (2023). Automated prompting for non-overlapping cross-domain sequential recommendation. arXiv preprint [arXiv:2304.04218](https://arxiv.org/abs/2304.04218).
- Hao, X., Liu, Y., Xie, R., Ge, K., Tang, L., Zhang, X., et al. (2021). Adversarial feature translation for multi-domain recommendation. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 2964–2973).
- Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2016). Session-based recommendations with recurrent neural networks. In *Proceedings of the 4th international conference on learning representations*.
- Hwang, J., Ju, H., Kang, S., Jang, S., & Yu, H. (2024). Multi-domain sequential recommendation via domain space learning. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval* (pp. 2134–2144).
- Jia, P., Wang, Y., Lin, S., Li, X., Zhao, X., Guo, H., et al. (2024). D3: A methodological exploration of domain division, modeling, and balance in multi-domain recommendations. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 8553–8561).
- Jiang, Y., Li, Q., Zhu, H., Yu, J., Li, J., Xu, Z., et al. (2022). Adaptive domain interest network for multi-domain recommendation. In *Proceedings of the 31st ACM international conference on information and knowledge management* (pp. 3212–3221).
- Ju, H., Kang, S., Lee, D., Hwang, J., Jang, S., & Yu, H. (2024). Multi-domain recommendation to attract users via domain preference modeling. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 8582–8590).
- Kang, W.-C., & McAuley, J. (2018). Self-attentive sequential recommendation. In *Proceedings of the 18th IEEE international conference on data mining* (pp. 197–206).
- Kim, D., Kim, S., Zhao, H., Li, S., Rossi, R. A., & Koh, E. (2019). Domain switch-aware holistic recurrent neural network for modeling multi-domain user behavior. In *Proceedings of the 12th ACM international conference on web search and data mining* (pp. 663–671).
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30–37.
- Li, J., Li, J., Li, J., Zheng, H., Liu, Y., Lu, M., et al. (2023). ADL: Adaptive distribution learning framework for multi-scenario CTR prediction. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval* (pp. 1786–1790).
- Li, C., Xie, Y., Yu, C., Hu, B., Li, Z., Shu, G., et al. (2023). One for all, all for one: Learning and transferring user embeddings for cross-domain recommendation. In *Proceedings of the 16th ACM international conference on web search and data mining* (pp. 366–374).
- Li, X., Yan, F., Zhao, X., Wang, Y., Chen, B., Guo, H., et al. (2023). HAMUR: Hyper adapter for multi-domain recommendation. In *Proceedings of the 32nd ACM international conference on information and knowledge management* (pp. 1268–1277).

- Lin, G., Gao, C., Zheng, Y., Chang, J., Niu, Y., Song, Y., et al. (2024). Mixed attention network for cross-domain sequential recommendation. In *Proceedings of the 17th ACM international conference on web search and data mining* (pp. 405–413).
- Lin, J., Pan, W., & 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* (pp. 130–139).
- Liu, C., Lin, J., Wang, J., Liu, H., & Caverlee, J. (2024). Mamba4Rec: Towards efficient sequential recommendation with selective state space models. arXiv preprint [arXiv:2403.03900](https://arxiv.org/abs/2403.03900).
- Liu, Z., Liu, Q., Wang, Y., Wang, W., Jia, P., Wang, M., et al. (2025). SIGMA: Selective gated Mamba for sequential recommendation. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 12264–12272).
- Liu, X., Zhang, J., Ren, C., Xu, S., Pan, Z., & Zhang, Z. (2024). Heterogeneous graph-based framework with disentangled representations learning for multi-target cross domain recommendation. arXiv preprint [arXiv:2407.00909](https://arxiv.org/abs/2407.00909).
- Luo, L., Li, Y., Gao, B., Tang, S., Wang, S., Li, J., et al. (2023). MAMDR: A model agnostic learning framework for multi-domain recommendation. In *Proceedings of the 39th IEEE international conference on data engineering* (pp. 3079–3092).
- Ma, H., Xie, R., Meng, L., Chen, X., Zhang, X., Lin, L., et al. (2024). Triple sequence learning for cross-domain recommendation. *ACM Transactions on Information Systems*, 42(4), 1–29.
- Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., & Chi, E. H. (2018). Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1930–1939).
- Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11).
- Min, E., Luo, D., Lin, K., Huang, C., & Liu, Y. (2023). Scenario-adaptive feature interaction for click-through rate prediction. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 4661–4672).
- Ni, X., Nie, J., Zuo, Z., Xie, H., Liang, X., Jiang, M., et al. (2023). DITN: User's indirect side-information involved domain-invariant feature transfer network for cross-domain recommendation. *Information Processing & Management*, 60(6), Article 103494.
- Ning, W., Yan, X., Liu, W., Cheng, R., Zhang, R., & Tang, B. (2023). Multi-domain recommendation with embedding disentangling and domain alignment. In *Proceedings of the 32nd ACM international conference on information and knowledge management* (pp. 1917–1927).
- Park, C., Kim, T., Yoon, H., Hong, J., Yu, Y., Cho, M., et al. (2024). Pacer and runner: Cooperative learning framework between single-and cross-domain sequential recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval* (pp. 2071–2080).
- Pi, Q., Zhou, G., Zhang, Y., Wang, Z., Ren, L., Fan, Y., et al. (2020). Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM international conference on information and knowledge management* (pp. 2685–2692).
- Qin, Z., Cheng, Y., Zhao, Z., Chen, Z., Metzler, D., & Qin, J. (2020). Multitask mixture of sequential experts for user activity streams. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 3083–3091).
- Qu, Z., Xie, R., Xiao, C., Yao, Y., Liu, Z., Lian, F., et al. (2023). Thoroughly modeling multi-domain pre-trained recommendation as language. arXiv preprint [arXiv:2310.13540](https://arxiv.org/abs/2310.13540).
- Ruder, S. (2017). An overview of multi-task learning in deep neural networks. arXiv preprint [arXiv:1706.05098](https://arxiv.org/abs/1706.05098).
- Sheng, X.-R., Zhao, L., Zhou, G., Ding, X., Dai, B., Luo, Q., et al. (2021). One model to serve all: Star topology adaptive recommender for multi-domain CTR prediction. In *Proceedings of the 30th ACM international conference on information and knowledge management* (pp. 4104–4113).
- Sun, W., Xie, R., Zhang, J., Zhao, W. X., Lin, L., & Wen, J.-R. (2024). Curriculum-scheduled knowledge distillation from multiple pre-trained teachers for multi-domain sequential recommendation. arXiv preprint [arXiv:2401.00797](https://arxiv.org/abs/2401.00797).
- Tang, M., Cui, S., Jin, Z., Liang, S.-n., Li, C., & Zou, L. (2025). Sequential recommendation by reprogramming pretrained transformer. *Information Processing & Management*, 62(1), Article 103938.
- Tang, Z., Huan, Z., Li, Z., Zhang, X., Hu, J., Fu, C., et al. (2023). One model for all: Large language models are domain-agnostic recommendation systems. arXiv preprint [arXiv:2310.14304](https://arxiv.org/abs/2310.14304).
- Tang, H., Liu, J., Zhao, M., & Gong, X. (2020). Progressive layered extraction (PLE): A novel multi-task learning (MTL) model for personalized recommendations. In *Proceedings of the 14th ACM conference on recommender systems* (pp. 269–278).
- Tang, J., & 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* (pp. 565–573).
- Tian, Y., Li, B., Chen, S., Li, X., Deng, H., Xu, J., et al. (2023). Multi-scenario ranking with adaptive feature learning. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval* (pp. 517–526).
- Tian, Z., Zhang, C., Zhao, W. X., Zhao, X., Wen, J.-R., & Cao, Z. (2023). UFIN: Universal feature interaction network for multi-domain click-through rate prediction. arXiv preprint [arXiv:2311.15493](https://arxiv.org/abs/2311.15493).
- Vats, A., Jain, V., Raja, R., & Chadha, A. (2024). Exploring the impact of large language models on recommender systems: An extensive review. arXiv preprint [arXiv:2402.18590](https://arxiv.org/abs/2402.18590).
- Wang, Y., Guo, H., Chen, B., Liu, W., Liu, Z., Zhang, Q., et al. (2022). Causalint: Causal inspired intervention for multi-scenario recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 4090–4099).
- Wang, Y., Guo, C., Chu, Y., Hwang, J.-N., & Feng, C. (2020). A cross-domain hierarchical recurrent model for personalized session-based recommendations. *Neurocomputing*, 380, 271–284.
- Wang, Y., Liu, Z., Wang, Y., Zhao, X., Chen, B., Guo, H., et al. (2024). DIFF-MSR: A diffusion model enhanced paradigm for cold-start multi-scenario recommendation. In *Proceedings of the 17th ACM international conference on web search and data mining* (pp. 779–787).
- Xie, X., Sun, F., Liu, Z., Wu, S., Gao, J., Zhang, J., et al. (2022). Contrastive learning for sequential recommendation. In *Proceedings of the 38th IEEE international conference on data engineering* (pp. 1259–1273).
- Xie, Y., Zhou, P., & Kim, S. (2022). Decoupled side information fusion for sequential recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 1611–1621).
- Xu, W., Li, S., Ha, M., Guo, X., Ma, Q., Liu, X., et al. (2023). Neural node matching for multi-target cross domain recommendation. In *Proceedings of the 39th international conference on data engineering* (pp. 2154–2166).
- Xu, S., Li, L., Yao, Y., Chen, Z., Wu, H., Lu, Q., et al. (2023). MUSENET: Multi-scenario learning for repeat-aware personalized recommendation. In *Proceedings of the 16th ACM international conference on web search and data mining* (pp. 517–525).
- Xu, Z., Pan, W., & Ming, Z. (2023). A multi-view graph contrastive learning framework for cross-domain sequential recommendation. In *Proceedings of the 17th ACM conference on recommender systems* (pp. 491–501).
- Xu, Z., Pan, W., & Ming, Z. (2024). Transfer learning in cross-domain sequential recommendation. *Information Sciences*, 669, Article 120550.
- Xu, Z., Wei, P., Liu, S., Zhang, W., Wang, L., & Zheng, B. (2023). Correlative preference transfer with hierarchical hypergraph network for multi-domain recommendation. In *Proceedings of the ACM web conference 2023* (pp. 983–991).
- Ye, X., Li, Y., & Yao, L. (2023). DREAM: Decoupled representation via extraction attention module and supervised contrastive learning for cross-domain sequential recommender. In *Proceedings of the 17th ACM conference on recommender systems* (pp. 479–490).
- Yin, M., Wang, H., Guo, W., Liu, Y., Li, Z., Zhao, S., et al. (2024). Learning partially aligned item representation for cross-domain sequential recommendation. arXiv preprint [arXiv:2405.12473](https://arxiv.org/abs/2405.12473).

- Zhang, J., Duan, H., Guo, L., Xu, L., & Wang, X. (2023). Towards lightweight cross-domain sequential recommendation via external attention-enhanced graph convolution network. In *Proceedings of the 28th international conference on database systems for advanced applications* (pp. 205–220).
- Zhao, P., Gao, X., Xu, C., & Chen, L. (2023). M5: Multi-modal multi-interest multi-scenario matching for over-the-top recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 5650–5659).
- Zhou, K., Yu, H., Zhao, W. X., & Wen, J.-R. (2022). Filter-enhanced MLP is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022* (pp. 2388–2399).
- Zhu, T., Shi, Y., Zhang, Y., Wu, Y., Mo, F., & Nie, J.-Y. (2024). Collaboration and transition: Distilling item transitions into multi-query self-attention for sequential recommendation. In *Proceedings of the 17th ACM international conference on web search and data mining* (pp. 1003–1011).