Cracking the Code of Negative Transfer: A Cooperative Game Theoretic Approach for Cross-Domain **Sequential Recommendation**

Chung Park cpark88kr@gmail.com SK Telecom Korea Advanced Institute of Science and Technology

> Yelim Yu velim.yu@sk.com SK Telecom

Mincheol Cho skt.mccho@sk.com SK Telecom

Taesan Kim

ktmountain@sk.com

SK Telecom

Hyungjun Yoon hjyoon@sk.com SK Telecom

kyunam@sk.com

Minsung Choi ms.choi@sk.com SK Telecom

Taekyoon Choi tgchoi03@gmail.com NAVER

Kyunam Lee

SK Telecom

Junui Hong skt.juhong@sk.com SK Telecom Korea Advanced Institute of Science and Technology

> Sungil Ryu ryu0121@sk.com SK Telecom

Jaegul Choo* jchoo@kaist.ac.kr Korea Advanced Institute of Science and Technology

ABSTRACT

This paper investigates Cross-Domain Sequential Recommendation (CDSR), a promising method that uses information from multiple domains (more than three) to generate accurate and diverse recommendations, and takes into account the sequential nature of user interactions. The effectiveness of these systems often depends on the complex interplay among the multiple domains. In this dynamic landscape, the problem of negative transfer arises, where heterogeneous knowledge between dissimilar domains leads to performance degradation due to differences in user preferences across these domains. As a remedy, we propose a new CDSR framework that addresses the problem of negative transfer by assessing the extent of negative transfer from one domain to another and adaptively assigning low weight values to the corresponding prediction losses. To this end, the amount of negative transfer is estimated by measuring the marginal contribution of each domain to model performance based on a cooperative game theory. In addition, a hierarchical contrastive learning approach that incorporates information from the sequence of coarse-level categories into that of fine-level categories (e.g., item level) when implementing contrastive learning was developed to mitigate negative transfer. Despite the potentially low relevance between domains at the fine-level, there may be higher relevance at the category level due to its generalised and broader preferences. We show that our model is superior to prior works in

CIKM '23, October 21-25, 2023, Birmingham, United Kingdom

terms of model performance on two real-world datasets across ten different domains.

CCS CONCEPTS

• Information systems → Recommender systems; • Computing methodologies \rightarrow Neural networks.

KEYWORDS

Cross-Domain Sequential Recommendation; Negative Transfer; **Cooperative Game**

ACM Reference Format:

Chung Park, Taesan Kim, Taekyoon Choi, Junui Hong, Yelim Yu, Mincheol Cho, Kyunam Lee, Sungil Ryu, Hyungjun Yoon, Minsung Choi, and Jaegul Choo. 2023. Cracking the Code of Negative Transfer: A Cooperative Game Theoretic Approach for Cross-Domain Sequential Recommendation. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21-25, 2023, Birmingham, United Kingdom. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/ 3583780.3614828

INTRODUCTION 1

In recommendation systems, user preferences typically reflect partial or incomplete engagement within a specific domain. This tendency results in biased recommendation outcomes based on the historical interactions observed within a single domain [1]. Cross-Domain Sequential Recommendation (CDSR), which aims to enhance the performance of multiple domain recommendations simultaneously by utilizing information from other domains, has been introduced to mitigate this concern. Another key aspect of the CDSR is capturing the sequential characteristics of the user interaction, which enables a model to better understand users' dynamic interest over time [1, 7].

In the dynamic landscape of the CDSR task, a phenomenon known as negative transfer arises when dissimilar knowledge across domains results in a performance decline due to discrepancies in

^{*}Corresponding Author (jchoo@kaist.ac.kr)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

^{© 2023} Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 979-8-4007-0124-5/23/10...\$15.00 https://doi.org/10.1145/3583780.3614828



Figure 1: Negative transfer cases between domain-domain pairs in the Amazon dataset is shown. The figure on the left compares the single-domain recommendation results for the Books domain with the cross-domain recommendation results for the Books domain paired with four other domains. The Books+Sports case shows worse performance than the Books case alone, due to the irrelevance between the Books and Sports domains (i.e., negative transfer). On the right is about Clothing domain with the same scheme as the left side. The performance of both the single- and cross-domain recommendation cases was evaluated using the C²DSR [1].

user preferences among these domains [2, 7, 22]. For instance, as shown in Figure 1, the *Books* domain enjoys a collaborative relationship with the *Movies* domain, yet maintains conflicting relationships with the *Sports* domain. This is due to the different domain interests of a same user. Note that the *Books+Movies* case shows higher performance than the *Books* only case. From this perspective, Li et al. [7] proposed the cross-domain recommendation method, which alleviates the negative transfer problem by training both a global and a domain-specific user representation, but their approach does not consider the sequential dynamics of user interaction. As users' preferences change over time, the dynamics of negative transfer further intensify and this has not been fully addressed in previous studies.

In this paper, we propose the model **CGRec**, which stands for <u>Cooperative Game</u> Theoretic Approach for Cross-Domain Sequential <u>Rec</u>ommendation. This model mitigates the negative transfer issue by minimizing its detrimental impact on the overall performance of each domain. In particular, we focus on the case of more than three domains for real-world applications, which has not been considered much in the previous CDSR studies. For this purpose, our model has been trained using hybrid sequences from more than three domains, conceptualizing each domain as players in a cooperative game [11]. Then, Shapley values [16] were derived for all domains and were used to approximate the degree of the negative transfer of each domain. This value is used to assign low weights to items in domains that show high negative transfer during the training process.

In addition, we develop a hierarchical contrastive learning method that sequentially learns user behavior patterns from the coarsest to the finest (i.e., item-level) categories of user interaction sequences, thereby discerning more general user preferences in each domain. These category-level sequences potentially provide more generalized user preferences and are more cross-domain relevant compared to the item-level preferences, as shown in Figure 2. These generalized user preferences play a pivotal role in diminishing the negative transfer effect caused by heavily biased user preferences in a specific domain. The proposed hierarchical contrastive learning method progresses from learning coarse-level categories down to fine-level ones, culminating at the item level.

As a result, our proposed CGRec allows the performance of the CDSR task to be improved with two real-world datasets for multiple domains. In particular, some of the domains that are performing poorly due to the negative transfer show large gains in performance in the model.



Figure 2: Illustration of user interaction sequence for two domains in *Amazon* dataset. The user sequences at the category level are more general and domain-relevant than those at the item level for the same user.

2 RELATED WORK

We focus on the CDSR task, which is a mixture of the CDR task for three or more domains and the SR task reflecting the temporal order of user sequences. Since most of the CDSR studies are for two domains, our study is the first CDSR study for multiple domains to our knowledge.

2.1 Sequential Recommendation

The sequential recommendation framework models the temporal dynamics of user-item interactions, by capturing user evolving preferences. GRU4Rec [4] is a session-based recommendation model using Gated Recurrent Units (GRU), capturing the complex dependencies within sessions. Also, attention mechanisms have been employed to identify and encapsulate sequential patterns, taking into account short- and long-term dependencies [6, 12, 17, 25]. For instance, SASRec [6] employed self-attention mechanisms to model user-item interactions while BERT4Rec [17] integrated the *Cloze* objective to construct a pre-trained model tailored for sequential recommendations. CORE [5] unified the self-attention based encoding and decoding representation space by using a robust distance measure to prevent overfitting.

Meanwhile, there have been studies using item features together with item sequences [12, 21, 25]. A feature-level self-attention layer was proposed to utilize the attribute information such as the category of the item in FDSA [21]. S³Rec [25], in their pretraining phase, used four self-supervised objectives to train the relationships among attributes, items, subsequences, and sequences. Adding extra information of items leads to significant performance improvement of S³Rec, and this improvement is also observed in CARCA [12]. CARCA utilized not only categorical attributes, but also contextual information such as buying time and image of items. The dynamic nature of abundant context and attribute is captured by self-attention blocks, but CARCA uses encoder-decoder architectures instead of encoder-only or decoder-only model. However, both models still have the negative transfer issues in the cross-domain recommendation.

2.2 Cross-Domain Recommendation

The aim of Cross-Domain Recommendation (CDR) is to enhance the quality of recommendations in the target domain through the utilization of complementary knowledge from other domains. For example, CMF [15] modeled the collective matrix factorization using the pairwise relational data while CLFM [2] introduced a clusterlevel latent factor model, which captures domain-specific patterns and common patterns across domains. Another approach, DTCDR [26], provided user and item embeddings using rating and multisource content, and applied a multi-task learning strategy to share these embeddings across domains, to improve the recommendation performance in both data-rich and sparse domains simultaneously. BiTGCF [8] extracted user-item relationships in each domain and facilitated their transfer using the graph convolutional networks. DeepAPF [20] approximated complex user-video interactions, capturing cross-domain interests and single domain interests using an attention layer. However, these methods approached the CDR task by modeling a pairwise domain-domain relationship, which loses effectiveness when dealing with a large number of domains. In order to reconcile this research deficit, CAT-ART [7] introduced a CDR method designed to enhance recommendations across $N \ge 3$ participating domains. The method generated a global user representation and a domain-specific user embedding based on the matrix factorization. Yet, these studies overlooked the sequential characteristics inherent in user interactions.

2.3 Cross-Domain Sequential Recommendation

The CDSR approach aims to improve the sequential recommendation task where items belong to multiple domains. This approach takes into account the sequential dynamics of user interactions. π -Net [9] devised gating mechanisms to transfer single-domain information to other paired domain. C²DSR [1] modeled singleand cross-domain representation utilizing the self-attention based encoder and graph neural network with proposed cross-domain infomax objective. However, these studies modeled a pairwise domaindomain relationship only. Applying the models to a situation involving $N \ge 3$ domains would necessitate managing an excessive number of domain pairs, which may not be realistic in real-world applications [7]. Therefore, we focus on extending CDSR to handle more than three domains as well as to consider the sequential dynamics, which has not been proposed in previous studies.

3 PRELIMINARY

In this work, we focus on a expanding CDSR scenario, where each interaction sequence involves $D \ge 3$ domains. The domain $d \in [1, 2, ..., D]$ can be shopping categories, such as books and movies, or the different service platforms (e.g., short video platforms) [7, 27].

Definition 1. Hierarchical Categories of Items: Each item corresponds to various levels of categories. We posit an *H*-level hierarchical categorization within each domain. In this hierarchy, level-1 signifies the coarsest-grained category, while level *H* (i.e., item level), represents the finest category. Note that the item level is treated as the level *H* category. For instance, if the *Books* domain is designated as the coarsest category level (level-1), the succeeding category level (level-2) may include various sub-categories such as *Economics* or *Psychology* in the *Amazon* dataset.

Thus, we introduce the notation $I^{d,h}$ to denote a set of interactions within the level-*h* category for domain *d*. Additionally, the cumulative set of interactions across all category levels within domain *d* is represented as $I^{d,\cdot} = \{I^{d,1}, ..., I^{d,H}\}$. Moreover, the cumulative set of interactions can be defined across all domains within the level-*h* category as $I^{\cdot,h} = \{I^{1,h}, ..., I^{D,h}\}$.

Definition 2. Domain-hybrid Sequence: The sequential interactions within the domain *d* can be encapsulated as a sequence $S^d = \{s_1^d, s_2^d, ..., s_{|S^d|}^d\}$, where each $s_t^d \in S^d$ represents a tuple of interactions across all levels of categories $(c_t^1, c_t^2, ..., c_t^H)^d$ interacted with at time *t*. Subsequently, we denote $S = (S^1, S^2, ..., S^D)$ as the domain-hybrid sequence that has been shuffled according to chronological order. As an illustration, it is assumed that a user interacted with three domains [1, 2, 3], with the interaction sequences for each domain as $S^1 = [s_1^1, s_2^1, s_1^3], S^2 = [s_1^2, s_2^2]$, and $S^3 = [s_1^3, s_2^3]$. The sequence $S = [s_1^1, s_2^1, s_2^1, s_2^2, s_3^1, s_1^3, s_2^3]$ then symbolizes the domain-hybrid interaction sequence, which merges S^1 , S^2 , and S^3 in chronological order¹. If domains 1 and 2 have two levels of categories, and domain 3 has three levels categories, then $s_1^1 = (c_1^1, c_1^2)^1, s_1^2 = (c_1^1, c_1^2)^2$, and $s_2^3 = (c_2^1, c_2^2, c_2^2)^3$. **Problem Statement:** Given the observed domain-hybrid sequences

Problem Statement: Given the observed domain-hybrid sequences until *t* time step $S_{1:t} = (S^1, S^2, ..., S^D)_{1:t}$, the goal of CDSR is to predict the tuple of next interactions $s_{t+1}^n = (c_{t+1}^1, c_{t+1}^2, ..., c_{t+1}^H)^n$:

$$\underset{s_{t+1}^n \in I^{n, \cdot}}{\operatorname{argmax}} P(s_{t+1}^n | S_{1:t}), \text{ if domain of the next item} = n, \quad (1)$$

where $P(s_{t+1}^n|S_{1:t})$ is the joint probability of the candidate interactions of all H levels categories in domain n. Consequently, our model is trained to predict the next interactions, considering all the categories at different hierarchical levels $(c_{t+1}^1, ..., c_{t+1}^H)^n$ concurrently. In CDSR, the interaction from the last hierarchy level (i.e., item level; c_{t+1}^H) with the highest score is selected as the next recommended item.

4 MODEL

4.1 Item Embedding Layer

The category sequence implies a more general user preference than the item sequence, which serves to mitigate negative transfers between irrelevant domains [21]. Derived from this hierarchy of items, we suggest considering the sequences of categories corresponding to these items as additional input features, and training them together with the proposed objective discussed in Section 4.4.

First, each domain-hybrid sequence is converted to a fixed length (= m) sequence by truncating or padding interactions, as in previous studies [1, 6, 25]. Then, the domain-hybrid sequence *S* is

¹This sequence can be described as $[s_1^1, s_2^2, s_3^1, s_4^2, s_5^1, s_6^3, s_7^3]$.



Figure 3: We illustrate our model with three domains (represented by the colors blue, yellow, and pink in the left side of the figure) and two hierarchical categories. \triangle indicates the loss per items. The domain hybrid sequence $S = [s_1^1, s_2^2, s_3^1, s_4^2, s_5^1, s_6^3, s_7^3]$ is input into our model. The first feeding process is expressed as the bold-red lines, and the sequence S is represented as tuple of two categories sequences. Note that the level 2 category indicates the item level in this figure.

transformed into a *m*-length embedding vector sequence. The level*h* category embedding matrix of domain *d*, $B^{d,h}$, is represented by a matrix $\mathbb{R}^{|I^{d,h}| \times r}$, where $|I^{d,h}|$ is the size of the level-*h* category sets in the domain *d*, and *r* is the embedding dimension. Then, all embedding matrix of *D* domains are concatenated to form the comprehensive level-*h* category embedding matrix $B^h \in \mathbb{R}^{|I^{-,h}| \times r}$. Specifically, a look-up operation using B^h was used in order to build up the input embedding matrix $\mathbf{E}^h \in \mathbb{R}^{m \times r}$. In addition, a trainable position embedding matrix $\mathbf{P} \in \mathbb{R}^{m \times r}$ is integrated to consider the sequential characteristic of the user interaction. Consequently, the final sequence representation $\mathbf{E} \in \mathbb{R}^{m \times r}$ is acquired by summing H+1 embedding matrices: $\mathbf{E}^1 + \mathbf{E}^2 + \ldots + \mathbf{E}^H + P$. The embedding of the level-*h* category in the *t*-th step is denoted as \mathbf{E}_t^h . This process results in an input sequence embedding for the self-attention encoder, which is described in the following section.

4.2 Self-Attention Encoder

The self-attention module[19] is employed to encode the relationships between items within sequences. This module typically incorporates two sub-layers as described in the following subsection. *4.2.1 Multi-head Self-Attention.* This mechanism has shown its effectiveness in selectively garnering information from various representation subspaces. The operation of this mechanism involves taking the embedding E as an input (refer to Section 4.1), transforming E into three matrices via linear projections, and subsequently feeding the three matrices into an attention function. The multihead self-attention (MHA) is defined as follows:

$$MHA(Z^{l}) = [head_{1}^{\frown} head_{2}^{\frown} \dots^{\frown} head_{p}]W^{F},$$

$$head_{i} = Attention(Z^{l}W_{i}^{Q}, Z^{l}W_{i}^{K}, Z^{l}W_{i}^{V}),$$
(2)

where the projection matrices $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{r \times r/p}$ and $W^F \in \mathbb{R}^{r \times r}$ are the trainable parameters, $Z^l \in \mathbb{R}^{|Z^l| \times d}$ is sequential input for the *l*-th self-attention layer, *p* is the number of heads, and $\widehat{}$ is the concatenate operation. When *l*=0, we set $Z^0 = \mathbb{E}$. Z_t^l is the *t*-th step embedding of the Z^l . The scaled dot-product operation adopts

the attention mechanism as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^{I}}{\sqrt{r/p}})V,$$
(3)

where $Q = Z^{l}W_{i}^{Q}$, $K = Z^{l}W_{i}^{K}$, and $V = Z^{l}W_{i}^{V}$ denote the output of linear projections of the embedding sequence, while r/p serves as a scale factor. In our model, *L* multi-head self-attention layers are stacked in the self-attention encoder.

4.2.2 Point-Wise Feed-Forward Network. We adopt a point-wise feed-forward network to all Z_t^l to incorporate nonlinearity into the model and facilitate interactions among different latent subspaces as follows:

$$Z^{l} = [FFN(Z_{1}^{l})^{\top} \widehat{FFN}(Z_{2}^{l})^{\top} \widehat{\cdots} \widehat{FFN}(Z_{m}^{l})^{\top}],$$

$$FFN(Z_{t}) = GELU(Z_{t}W_{1} + b_{1})W_{2} + b_{2}.$$
(4)

where W₁, b_1 , W₂, and b_2 represent trainable parameters, *GELU* is the gelu activation [3], and $\widehat{}$ symbolizes the concatenation. In the sequential recommendation task, only use the information prior to the current time step can be used. Therefore, a masking operation is employed in the output sequences of the multi-head self-attention layer to remove all links between Q_i and K_j when j > i.

4.3 Hierarchical Prediction Layer

In our model, sequences of categories from various hierarchies (*H*) are trained simultaneously. During the training process, the interdependencies among these sequences of categories are integrated by transferring information from coarse-grained to fine-grained categories. This strategy ensures an unbiased representation at the finest category level, i.e., the item level, by incorporating the patterns of coarser categories that reflect broader user preferences. The purpose of this approach is to prevent the negative transfer issue - a scenario in which a user preference in one domain excessively impacts other unrelated domains, subsequently reducing the overall performance. CGRec

Upon applying L multi-head self-attention layers, the next interaction is predicted using the final representation from the selfattention encoder given the first t items. Specifically, the user's preference score is initially computed for the level-1 category (i.e., the coarsest category) in step t+1, given the user's historical context as follows:

$$P(c_{t+1}^{1} = c^{1}|S_{1:t}) = e_{c^{1}}^{\top} \cdot U_{t}^{L,1},$$

$$U_{T}^{L,1} = FFN^{1}(Z_{t}^{L}),$$

$$FFN^{1}(Z_{t}) = GELU(Z_{t}W_{3}^{1} + b_{3}^{1})W_{4}^{1} + b_{4}^{1},$$
(5)

where e_{c^1} is the level-1 representation of positive item c^1 from the embedding matrix E^1 , and $U_t^{L,1}$ is the output of feed-forward network for level-1 prediction. In addition, Z_t^L is the output of the *L*-th self-attention layer at step *t*, and W_3^1 , b_3^1 , W_4^1 , b_4^1 are trainable parameters. Then, the preference score for the succeeding level 2 category in the step *t* + 1 is sequentially calculated as:

$$P(c_{t+1}^2 = c^2 | S_{1:t}) = e_{c^2}^\top \cdot (U_t^{L,2} \oplus U_t^{L,1}),$$
(6)

where e_{c^2} is the level-2 embedding of positive item c^2 from category embedding matrix E^2 and \oplus indicates the element-wise sum operation. This operation incorporates information from the coarsegrained categories into the fine-grained categories when calculating the preference score, and therefore the predictions of fine-grained categories depend on the predicted outcomes of subsequent coarsegrained categories. This interdependence between hierarchical levels highlights the significance of integrating information across multiple categories to gain a comprehensive understanding of the user preferences. From this process, the preference score for the level-*H* category (i.e., item level) in the step t + 1 is extracted as follows:

$$P(c_{t+1}^{H} = c^{H}|S_{1:t}) = e_{c^{H}}^{\top} \cdot (U_{t}^{L,H} \oplus U_{t}^{L,(H-1)}).$$
(7)

4.4 Training Objective

Previous studies [12, 21] have used these categories as auxiliary inputs but have overlooked the interdependency between items and their categories during the training process. We therefore adopt a hierarchical approach to contrastive learning, first predicting the coarsest categories, and then using the learned representations to predict finer-grained categories.

To simultaneously maximize all preference scores across all category levels and domains, the pairwise rank loss \mathcal{L}^{hcross} is incorporated as follows:

$$\mathcal{L}^{hcross} = \sum_{h=1}^{H} \sum_{t=1}^{m} \log\sigma \left(P(c_{t+1}^{h} = c^{h} | S_{1:t}) - P(c_{t+1}^{h} = c^{h-} | S_{1:t}) \right),$$
(8)

where σ denotes the sigmoid function, and the ground-truth item for the level-*h* category c^h is associated with a negative item c^{h-} that is chosen through random sampling.

4.5 Loss Re-balancing Layer

We focus on quantifying the negative transfer of a specific domain by measuring its marginal contribution for the training loss. We consider negative transfer and marginal contribution to be opposite concepts. Therefore, the larger the marginal contribution, the



Figure 4: The process of obtaining the negative transfer value for the domain 1.

smaller the degree of negative transfer. Low weight values are assigned for interacted items of high negative transfer (i.e., low marginal contribution) domains in the training process. Our model was trained using the domain hybrid sequences, which considers each domain as players of a cooperative game [11]. Then, Shapley values [16] were derived for all domains and were used to approximate the degree of the negative transfer of each domain. Before exploring the solution methodology, it is pertinent to briefly discuss some foundational concepts from the cooperative game theory.

4.5.1 Cooperative Games. In the realm of game theory, a cooperative game [11] is distinguished by the emphasis on cooperative behavior among clusters of players, referred to as coalitions. More formally, a cooperative game is often represented by a pair (N, v), where N is a finite set of players denoted by $N = \{1, 2, ..., n\}$, and vis a characteristic function. This characteristic function $v : 2^N \to \mathbb{R}$ allocates to each coalition $S \subseteq N$ a real number v(S), signifying the maximum cumulative payoff that players in S can attain through cooperation. For any subset $S \subseteq N$, v(S) quantifies the worth of coalition S. By default, we set $v(\emptyset) = 0$ and v(N) = 0.

4.5.2 Shapley Value. The Shapley value [16] offers an approach to equitably distribute the cumulative payoff of the grand coalition, i.e., the coalition that includes all players within the game. Shapley suggested a method that evaluates each player's role within the game by assessing its marginal contributions across all potential coalitions to which it could be a member. Formally, $\pi \in \Pi(N)$ is denoted as a permutation of players in N, and $C_{\pi}(i)$ as any coalition that exclude player *i*. Then the Shapley value of player *i*, $\phi_i(v)$, is defined as follows:

$$\phi_i(v) = \frac{1}{|N|!} \sum_{\pi \in \Pi(N)} [v(C_{\pi}(i) \cup \{i\}) - v(C_{\pi}(i))], \qquad (9)$$

where v is the characteristic function discussed in the next section. This value is the average marginal contribution of the player *i*.

4.5.3 Quantifying negative transfer of each Domain. In our model, each player *i* is a specific domain *d* in a sequence. S_{π} is defined as a sequence coalitions. As shown in Figure 4, a specific sequence is assumed to have three domains with seven interactions, which indicates $S = [s_1^1, s_2^2, s_3^1, s_4^2, s_5^1, s_6^3, s_7^3]$. Then $2^3 = 8$ coalitions can be derived based on the combinations of three domains, and one example sequence with domains 2 and 3 is $S_{\{2,3\}} = [(pad), s_2^2, (pad), s_4^2, (pad), s_4^2, (pad), s_4^2)$.

 s_6^3, s_7^3]. After, the degree of the negative transfer of each domain is computed by making use of the following cooperative game scheme. Note that the negative transfer and marginal contribution are opposite concepts. We define a cooperative game, (D, v), wherein the set of domains is the set of players and v is a characteristic function that attaches a value for each domain subset $\pi \in \Pi(D)$ in a specific sequence. In the example above, the domain hybrid sequence $S = [s_1^1, s_2^2, s_3^1, s_4^2, s_5^1, s_6^3, s_7^3]$ has 8 sequence coalitions, i.e., $\pi \in \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$. The value v of each $\pi \in \Pi(D)$ is defined as the training loss (See Section 4.4) of the model built using updates only from sequence S_{π} :

$$v(S_{\pi}) = \sum_{h=1}^{H} \sum_{t \in (S_{\pi})_{T}} \log \sigma \left(P(c_{t+1}^{h} = c^{h} | S_{1:t}) - P(c_{t+1}^{h} = c^{h-} | S_{1:t}) \right)$$

$$\phi_{i}(v) = \frac{1}{|D|!} \sum_{\pi \in \Pi(N)} [v(S_{C_{\pi}(i) \cup \{i\}}) - v(S_{C_{\pi}(i)})],$$

(10)

where $\phi_i(v)$ is the marginal contribution of the domain *i*, and $(S_\pi)_T$ is the set of time steps of S_π . Note that this module shares the embedding layers (Section 4.1), self-attention encoder (Section 4.2), and hierarchical prediction layers (Section 4.3) to calculate $P(c_{t+1}^h = c^h | S_{1:t}) - P(c_{t+1}^h = c^{h-} | S_{1:t})$. Then, the relative value of the negative transfer in each domain

Then, the relative value of the negative transfer in each domain is calculated. Let $\gamma = (\gamma_1, \gamma_2, ..., \gamma_D)$ be the vector of the degree of the negative transfer in each domain wherein γ_d represents the quantity of the negative transfer of domain *d*. In this case, γ_d is called *negative transfer value* of domain *d*. We posit that the more the negative transfer value for a domain, the less its contribution to the objective of our model. The negative transfer vector is initialized to be uniform $\gamma_d = 1/D$. Then at each mini-batch in the training, the Shapley value is obtained for all domains from the cooperative game (D, v) mentioned above. The computed Shapley values is denoted as $\phi_d(v)$ for each $d \in D$ and the negative transfer vector γ is updated as follows:

$$\gamma_d \leftarrow \alpha * \gamma_d + \beta * \phi_d(v); \forall d \in D, \tag{11}$$

where α and β are trainable parameters. The relative negative transfer value of each domain is determined by marginalizing γ as $\gamma \leftarrow \text{softmax}(\gamma; \lambda)$, where $\text{softmax}(; \lambda)$ is the softmax function with the temperature λ [18]. When $\lambda \to \infty$, then γ_d is uniform (= 1/D). Shapley values are typically approximated using Monte Carlo simulations due to their exponential computational complexity, which grows with the number of players involved. In all of our experimental settings, there are at most five domains participating in the game. With this number of players, calculating exact Shapley values involves computations over $2^5 = 32$ permutations, which is considered manageable. Therefore, Monte Carlo simulation was not applied to the Shapley value calculation in this paper.

4.5.4 Loss Re-balancing. The relative negative transfer value γ is used as a weight for the loss calculated in Eq. 8. A domain hybrid sequence comprises items ranging from 1 to a fixed length *m*, each of which belongs to a different domain. Consequently, the loss in Eq. 8 is aggregated separately depending on the item's domain, and the relative negative transfer value γ_d is multiplied by the domain as follows:

Table 1: Statistics of datasets

Dataset	Domain	#Users	#Items	#Interactions	#Sparsity
Amazon	Books		87,758	582,541	99.98%
	Clothing		69,480	370,169	99.98%
	Movies	34,771	36,070	343,056	99.97%
	Toys		47,940	279,011	99.98%
	Sports		57,387	242,169	99.99%
T-Seq	App-Use		9,192	36,716,069	96.00%
	Call-Use		3,301	3,038,385	99.07%
	PoI	99,936	549	1,892,868	96.33%
	E-comm		714	230,233	99.55%
	Membership		72	401,065	91.97%

$$\mathcal{L}^{hcross'} = \sum_{h=1}^{H} \sum_{t=1}^{m} \sum_{d=1}^{D} \gamma_d \log\sigma \left(P(c_{t+1}^h = c^h | S_{1:t}) - P(c_{t+1}^h = c^{h-} | S_{1:t}) \right)$$
(12)

Then, the re-balanced loss $\mathcal{L}^{hcross'}$ is minimized with the gradient descent algorithm.

5 EXPERIMENTS

The experiments aim to address the following research questions: **(RQ1)**: Can our model surpass the performance of state-of-the-art CD(S)R and SR models that involve multiple domains ($D \ge 3$)? **(RQ2)**: Is our model capable of mitigating the issue of the negative transfer in the context of the CDSR task?

(RQ3): How do the components of our model affect performance on the CDSR tasks?

5.1 Datasets

Table 1 describes the details of datasets used in this study.

5.1.1 **Amazon**. We obtained Amazon review datasets [10] with five domains, i.e., *Books, Clothing Shoes and Jewelry, Movies and TV, Toys and Games,* and *Sports and Outdoors.* They are represented by the abbreviations *Books, Clothing, Movies, Toys,* and *Sports* respectively. Users who have interactions across five domains were identified, and then retained domain-hybrid sequences that contain at least two items from each domain.

5.1.2 **T-Seq.** User logs were collected from five domains using various real-world applications owned by a global telecommunications company. The dataset, which included 99,936 customers who had agreed to have their information be collected and analyzed, was randomly sampled. The information included Application usage (*APP-Use*), Call detailed record (*Call-Use*), PoI preference (*PoI*), e-commerce purchase (*e-comm*), and membership usage (*MBR*).

5.2 Experimental Setup

5.2.1 Evaluation Metrics. Consistent with previous studies [6, 12, 25], the *leave-one-out* method for measuring recommendation performance was used. In particular, every sequence of user interactions is segmented into three segments: the last item is designated as test data, the item preceding the last item is set aside for validation data, and all remaining data is used for training purposes. The performance of each domain was reported separately based on the domain of the last item in the test data. Therefore, performance per domain was measured with different users. Given the

expansive item set, using all items as testing candidates would be prohibitively time-consuming. As a result, a widely accepted strategy that pair the ground truth item (i.e., positive sample) with 99 randomly selected items (i.e., negative samples) that the user had not previously interacted with was employed. Evaluation metrics were then extracted based on the ranking of these 100 items, and an average score across all users is presented. Following this, the Top-*k* recommendation performance is reported, derived from a ranked list of 100 items. Performance was assessed in terms of Hit Ratio (HR@{5, 10}), Normalized Discounted Cumulative Gain (NDCG@{5, 10}), and Mean Reciprocal Rank (MRR).

5.2.2 Compared Baselines. We compared the performance of our model to three categories of baselines: (1) traditional recommendation (TR), (2) sequential recommendation (SR), and (3) cross-domain (sequential) recommendation (CD(S)R) as shown in Table 2. Traditional recommendation includes BPRMF and FPMC. BPRMF [13] is a classic method, designed for developing recommendations from implicit feedback data, that introduces a pairwise loss function to model users' relative preferences. FPMC [14] uses personalized transition graphs over the markov chains to learn unique user transition matrices. See Section 2 for a description of SR and CD(S)R baselines. All the CDR baselines and some SR models were implemented using the RecBole framework [23, 24]. CARCA, S³Rec, C²DSR, and CAT-ART were implemented using their official code.

5.2.3 Hyperparameter Setting. For all models, certain hyperparameters were set. The embedding size *r* and mini-batch size was set to 128, and the dropout rate was fixed at 0.2. The training epoch was set at 100 to achieve optimal results and the Adam optimizer was employed to update all parameters. The best evaluation results were chosen based on the highest NDCG@5 performance on the validation set. In our model, we applied the number of category levels as two and three in *Amazon* and *T-Seq*, respectively. Moreover, we utilized two single-head attention blocks with four head size.

5.3 Performance Comparisons (RQ1)

Table 2 shows the performance of our model compared to most of the CD(S)R and SR baseline methods within the cross-domain scenario, which comprises five domains. Since all CD(S)R models only modeled pairwise domain-domain relationships, two domains out of five domains were selected to train these models. Each domain was paired with four other domains, and the average of the performance of these four cases per domain is reported in the Table 2. Due to space limitations, a more detailed domain pair-wise performance of the CD(S)R models cannot be reported and only the results for BiTGCF and C²DSR are shown in Figure 5.

Firstly, the CDR models performed poorly overall. There are two main reasons: (1) These models did not reflect the latest interests of users, as they failed to account for the dynamic sequential nature of user behavior; (2) Due to the negative transfer issue, these models tended to perform significantly worse for certain domain pairs in the experiments. As shown in Figure 5a, the pairs of *Sports* vs *Books*, *Movies, Clothing, Toys* domains in BiTGCF have performance deviations from 0.1287 to 0.1761 based on NDCG@5. In other words, the *Sports* domain is being distorted by the *Clothing* domain, resulting in a degradation of performance. On the other hand, CAT-ART, which is the only CDR framework to model relationships between



Figure 5: NDCG@5 of Domain-Domain pairs in BiTGCF [8] and C²DSR [1].

four or more domains, showed relatively reasonable performance. The reason is that the model did not reflect the temporal dynamics of user interaction, but considered the negative transfer of multiple domains in their model. This result suggests that expanding to three or more domains enhances the ability to reflect user preferences.

Secondly, the CDSR model, C²DSR, performed relatively well, but not as well as CAT-ART, because of its inability to resolve the negative transfer between domains. In fact, C²DSR, like the BiTGCF model, showed significant variation in performance between domain pairs. In C²DSR, pairs of Sports vs Books, Movies, Clothing, Toys domains have performance deviations from 0.1513 to 0.1839 based on NDCG@5, as shown in Figure 5c. Our model outperformed the CD(S)R baseline for most of the domains in both datasets (RQ1), because it minimized the negative transfer between dissimilar domains, modelling the relationships of all five domains together. Our model also outperformed CAT-ART, which attempted to reduce the negative transfer across three or more domains, because our model incorporated the temporal dynamics of users into the model. In the CDSR task, it is crucial to incorporate temporal dynamics into the model. This is due to the ever-evolving nature of users' preferences, which consequently amplifies the dynamics of negative transfers.

Lastly, the performance of SR models was also measured using domain-hybrid sequences. Overall, the existing SR models outperformed the CDR models. This indicates that the ability to capture the sequential nature of interactions provides valuable insights for making accurate recommendations. However, the performance of SR models was still lower than our model because of the negative transfer issue between different domains. In particular, in the case of CORE, it is clear the performance of a specific domain (e.g., *Sports*) is significantly lower. However, models that used category information as an additional input, such as FDSA, S³Rec, and CARCA, showed relatively high performance compared to other SR models. This suggests that the more general user preferences, such as category information, are strongly recommended to be incorporated in the model to mitigate the negative transfer issue. In comparison

Table 2: Comparison of the Proposed Method and Baselines. The top two methods are highlighted in bold and underlined.

		Dataset			Amazon					T-Sea		
Methods	Models	Domains	Book	Clothing	Movie	Toys	Sports	App-Use	Call-Use	Pol	MBR	e-comm
	BPRMF	HR@5	0.2382	0.2377	0.2402	0.2386	0.2398	0.9394	0.6387	0.8516	0.8657	0.1075
		HR@10	0.3458	0.3454	0.3468	0.3466	0.3445	0.9697	0.7993	0.9422	0.9281	0.3306
		NDCG@5	0.1028	0.1650	0.1042	0.1034	0.1035	0.8578	0.4488	0.6589	0.0880	0.049
TR		MRR	0.1524	0.1527	0.1534	0.1529	0.1523	0.8343	0.4075	0.6069	0.6363	0.0595
		HR@5	0.2763	0.1848	0.5994	0.2310	0.1926	0.9418	0.6728	0.8996	0.8689	0.3319
	FPMC	HR@10	0.3939	0.2762	0.6782	0.3414	0.2853	0.9709	0.8101	0.9608	0.9254	0.4606
		NDCG@5	0.1932	0.1253	0.4910	0.1595	0.1302	0.8521	0.5265	0.7487	0.7169	0.2492
		NDCG@10	0.2309	0.1547	0.5164	0.1950	0.1600	0.8616	0.5712	0.7687	0.7355	0.2902
		MKK	0.1813	0.1179	0.4654	0.1506	0.122	0.8257	0.4967	0.7065	0.6738	0.2384
		HR@5	0.2899	0.2861	0.5281	0.2929	0.2769	0.9504	0.7016	0.9214	0.8807	0.3885
	CPI II Pag	HR@10	0.4129	0.39//	0.6235	0.4032	0.3843	0.9774	0.8496	$\frac{0.9/16}{0.7650}$	0.9495	0.5355
	GRU4Rec	NDCG@10	0.1980	0.1903	0.4131	0.2009	0.1923	0.8532	0.5172	0.7824	0.7029	0.2879
		MRR	0.1840	0.1816	0.3877	0.1853	0.1791	0.8239	0.4762	0.7205	0.6522	0.2733
		HR@5	0.2588	0.2623	0.4781	0.2724	0.2473	0.9331	0.6677	0.8961	0.8576	0.3333
	SASRec	HR@10	0.3659	0.3701	0.5730	0.3807	0.3520	0.9656	0.8154	0.9636	0.9497	0.4530
		NDCG@5	0.1819	0.1888	0.3813	0.1967	0.1761	0.8400	0.5107	0.7537	0.6528	0.2622
		NDCG@10 MPD	0.2163	0.2235	0.4118	0.2315	0.2098	0.8506	0.5585	0.7759	0.6831	0.3004
		HR@5	0.191	0.159	0.3708	0.2002	0.1804	0.9360	0.4885	0.8900	0.8627	0.2785
	FDSA	HR@10	0.4218	0.4406	0.5465	0.4146	0.4092	0.9719	0.8155	0.9686	0.9318	0.4569
		NDCG@5	0.2166	0.2210	0.3403	0.2105	0.2118	0.8246	0.4645	0.7119	0.6551	0.2187
		NDCG@10	0.2544	0.2614	0.3754	0.2474	0.2466	0.8364	0.5147	0.7378	0.6776	0.2663
		MRR	0.2033	0.2065	0.3224	0.1961	0.1967	0.7918	0.4199	0.6632	0.5947	0.2088
		HR@5 HR@10	0.2852	0.2683	0.4993	0.2772	0.2299	0.9385	0.6544	0.8404	0.8357	0.2218
SR	BERT4Rec	NDCG@5	0.2031	0.1888	0.3950	0.1923	0.1583	0.8480	0.4740	0.6567	0.6406	0.1315
U II	Dining	NDCG@10	0.2369	0.2247	0.4288	0.2299	0.1909	0.8585	0.5225	0.6877	0.6708	0.1943
		MRR	0.1900	0.1775	0.3744	0.1800	0.1482	0.8217	0.4345	0.6084	0.5878	0.1277
		HR@5	0.3205	0.3086	0.3215	0.2989	0.2657	0.9465	0.7615	0.9289	0.9240	0.4603
	6 ³ D	HR@10	0.4253	0.4189	0.4278	0.4076	0.3616	0.9734	$\frac{0.8776}{0.6024}$	0.9787	0.9751	$\frac{0.5757}{0.2064}$
	5"Rec	NDCG@5	$\frac{0.2377}{0.2715}$	0.228	0.2505	0.2158	0.1966	0.8891	0.6024	0.7899	$\frac{0.7842}{0.7983}$	0.2964
		MRR	0.2432	0.2359	0.2591	0.2340	0.2058	0.8602	0.5721	0.7829	0.7676	0.3031
F	CORE	HR@5	0.2673	0.1949	0.4334	0.2331	0.2003	0.9382	0.7002	0.8527	0.8408	0.3242
		HR@10	0.3537	0.2699	0.5123	0.3111	0.2795	0.9712	0.8122	0.9406	0.9254	0.4451
		NDCG@5	0.2004	0.1491	0.3572	0.1753	0.1515	0.8398	0.5727	0.6886	0.6683	0.2550
		NDCG@10	0.2282	0.1731	0.3826	0.2003	0.1769	0.8506	0.6090	0.7174	0.6963	0.2935
		HR@5	0.1898	0.1439	0.3423	0.1003	0.1438	0.8582	0.5450	0.8281	0.0228	0.2470
		HR@10	0.3977	0.4112	0.5709	0.3663	0.3775	0.9052	0.7593	0.9142	0.9130	0.4943
	CARCA	NDCG@5	0.2065	0.1844	0.3702	0.2027	0.1835	0.7598	0.4326	0.6411	0.6782	0.1347
		NDCG@10	0.2398	0.2293	0.4008	0.2313	0.2193	0.7751	0.4783	0.6696	0.6918	0.2133
		MRR	0.2093	0.1970	0.3592	0.2117	0.1919	0.7370	0.3994	0.5963	0.6201	0.1498
		HR@5	0.3112	0.1857	0.4487	0.2121	0.1997	0.9557	0.6039	0.6023	0.8543	0.3907
	D'TOOL	HR@10	0.4287	0.2651	0.5640	0.3017	0.2807	0.9786	0.7449	0.7432	0.9168	0.5075
	BITGCF	NDCG@5	0.2196	0.1548	0.3410	0.1542	0.1451	0.8037	0.4464	0.4541	0.7557	0.3025
		MRR	0.2051	0.1284	0.3209	0.1471	0.1394	0.8733	0.4133	0.4239	0.7288	0.2890
		HR@5	0.2486	0.1881	0.4376	0.2005	0.2314	0.9435	0.6294	0.6076	0.7809	0.2556
		HR@10	0.3385	0.2875	0.5693	0.2972	0.3382	0.9745	0.7730	0.7623	0.9034	0.4341
	DTCDR	NDCG@5	0.1784	0.1224	0.3099	0.1406	0.1601	0.8414	0.4750	0.4680	0.5257	0.1591
		NDCG@10	0.2072	0.1544	0.3525	0.1716	0.1944	0.8516	0.5216	0.5181	0.5671	0.2163
		HR@5	0.2020	0.1353	0.3764	0.1550	0.1507	0.9632	0.6198	0.6164	0.4590	0.3154
		HR@10	0.2918	0.1972	0.4990	0.2263	0.2089	0.9819	0.7675	0.7589	0.9151	0.4327
	CMF	NDCG@5	0.1442	0.0982	0.2796	0.1170	0.1106	0.8832	0.4743	0.4944	0.7650	0.2496
		NDCG@10	0.1731	0.1181	0.3193	0.1387	0.1295	0.8894	0.5222	0.5404	0.7859	0.2872
		MKK HR@5	0.1371	0.0942	0.2642	0.1122	0.1054	0.8585	0.4461	0.4/31	0.7451	0.2435
		HR@10	0.1994	0.1752	0.2677	0.2130	0.1319	$\frac{0.9007}{0.9801}$	0.7950	0.7265	0.8885	0.3684
	CLFM	NDCG@5	0.1430	0.0855	0.2677	0.1054	0.0968	$\frac{0.9001}{0.8765}$	0.4982	0.4664	0.7320	0.2173
		NDCG@10	0.1698	0.1037	0.3106	0.1274	0.1147	0.8829	0.5428	0.5124	0.7531	0.2481
CD(S)R		MRR	0.1371	0.0942	0.2642	0.1122	0.1054	0.8585	0.4461	0.4731	0.7451	0.2435
(-)		HR@5	0.3376	0.2817	0.5128	0.2209	0.2994	0.9190	0.6376	0.6269	0.8568	0.3881
-	CAT-ART	NDCC@5	$\frac{0.4594}{0.2265}$	0.2817	0.5919	0.5185	0.3904	0.9710	0.8085	0.7769	0.9272	0.5110
		NDCG@10	0.2783	0.2268	0.4237	0.2297	0.2371	0.8354	0.5301	0.5499	0.7727	0.3362
		MRR	0.2604	0.2098	0.3966	0.1958	0.2205	0.7996	0.4787	0.4545	0.7227	0.2918
	C ² DSR	HR@5	0.2661	0.2022	0.4433	0.2256	0.2230	0.8582	0.6192	0.8281	0.8722	0.2510
		HR@10	0.3758	0.2877	0.5475	0.3186	0.3156	0.9052	0.7593	0.9142	0.9130	0.4943
		NDCG@5	0.1961	0.1497	0.3542	0.1618	0.1618	0.7598	0.4326	0.6411	0.6782	0.1347
		MRR	0.2314	0.1658	0.3458	0.1748	0.1915	0.7370	0.3994	0.5963	0.6201	0.1498
	DeepAFP	HR@5	0.1362	0.0956	0.2391	0.1239	0.1188	0.9592	0.5919	0.6109	0.8346	0.2770
		HR@10	0.2068	0.1500	0.3124	0.1790	0.1713	0.9767	0.7463	0.7539	0.8954	0.3719
		NDCG@5	0.0946	0.0664	0.1790	0.0925	0.0871	0.8829	0.4529	0.4803	0.7508	0.2245
		NDCG@10	0.1173	0.0838	0.2026	0.1102	0.1040	0.8878	0.5029	0.5199	0.7708	0.2540
	L	IVIKK	0.0903	0.0639	0.1088	0.0894	0.0837	0.8672	0.42/8	0.4/03	0./315	0.2199
		HR@5	0.3476	0.3167	0.5546	0.3137	0.3162	0.9537	0.7839	0.9151	0.9262	0.4911
	CGRee	NDCC@5	0.4607	0.4303	$\frac{0.0423}{0.4451}$	0.4228	0.4217	0.9784	0.8905	0.9/12	0.9/30	0.0566
	CGRec	NDCG@10	0.2930	0.2677	0.4732	0.2672	0.2671	0.8900	0.6627	0.8034	0.8039	0.4256
		MRR	0.2595	0.2373	0.4330	0.2392	0.2380	0.8622	0.5963	0.7508	0.7500	0.3732

*In the Methods column, TR, SR, and CD(S)R indicate the traditional recommendation, sequential recommendation, and Cross-Domain (Sequential) Recommendation. Note that BiTGCF, DTCDR, CMF, CLFM, CAT-ART, DeepAPF belong to the CDR method, and C²DSR and our proposed model CGRec is the CDSR method.

CGRee

CIKM '23, October 21-25, 2023, Birmingham, United Kingdom



Figure 6: Comparison of performance for different combinations of components.

Table 3: Ratio of our model performance to the average performance of all baselines. The top two domains per dataset are highlighted in bold and underlined.

Metric	Books	Clothing	Movies	Toys	Sports
HR@5	×1.31	×1.44	×1.32	×1.36	×1.51
HR@10	×1.25	×1.39	$\times 1.25$	$\times 1.30$	×1.43
NDCG@5	×1.36	×1.49	×1.35	×1.39	×1.58
NDCG@10	×1.32	$\times 1.44$	×1.31	×1.35	$\times 1.52$
MRR	×1.40	<u>×1.53</u>	×1.37	$\times 1.45$	×1.62
Metric	App-Use	Call-Use	PoI	MBR	e-comm
HR@5	×1.02	×1.20	×1.18	×1.08	×1.61
HR@10	×1.01	<u>×1.12</u>	×1.11	$\times 1.05$	×1.44
NDCG@5	×1.05	×1.28	$\times 1.27$	×1.13	$\times 1.72$
NDCG@10	×1.04	$\times 1.24$	×1.23	$\times 1.12$	×1.60
MRR	×1.05	<u>×1.31</u>	×1.27	$\times 1.14$	×1.72

to these SR models, our model exhibited high performance across most of domains in both datasets (RQ1). This superior performance is attributed to taking into account the negative transfer between multiple domains.

5.4 Discussion of the negative transfer (RQ2)

All CD(S)R and SR models examined in our study were significantly affected by the negative transfer issue, resulting in the performance degradation in several domains. This phenomenon was particularly evident in the Clothing and Sports domains of the Amazon dataset, and in the Call-Use and e-comm domains of the T-Seq dataset. Our model showed higher performance gains for these domains compared to other domains as shown in Table 3. The model outperformed the averaged NDCG@5 of all baselines of the Clothing and Sports domains in the Amazon dataset, with gains of ×1.49 and ×1.58 times respectively. Similarly, our model excelled in the Call-Use and e-comm domains in the T-Seq dataset, with NDCG@5 gains of $\times 1.28$ and $\times 1.77$ times compared to the average of all baselines. In essence, our model significantly improved the performance of domains that were negatively impacted by other domains. These results demonstrated that our model effectively solves the negative transfer issue, which contributes to its superior performance (RQ2).

5.5 Discussion of Model Variants (RQ3)

We carried out the following ablation studies to highlight the efficacy of each proposed module: (1) Basic Self-Attention model (BSA): The self-attention based model for the sequential recommendation, identical to SASRec except for the temporal encoding.

(2) +Hierarchical Contrastive Learning (HCL): We changed the objective of the BSA model into the hierarchical contrastive loss.
(3) +Loss Re-balancing Layer (LRL): We added loss re-balancing layer on the BSA.

Each of our modules improved the performance of the BSA model in all domains, and the model with both modules (+HCL+LRL) performed best in most domains, as shown in Figure 6 (RQ3). Specifically, the HCL module addressed the the negative transfer issue. The performance disparity between BSA and +HCL was most apparent in domains marked by the low performance domains, such as *Sports* and *e-comm* domain. In addition, the LRL also helped mitigate the negative transfer issue, as it outperformed the BSA model across all domains. This phenomenon is particularly pronounced in the *T-Seq* dataset, where there is a significant disparity in performance between domains.

6 CONCLUSION

This study focuses on the CDSR, a method using data from multiple domains for precise recommendations while recognizing users' dynamic interests. Negative transfer, a key challenge, is addressed by our proposed model, which quantifies and adaptively weights negative transfer in each domain. Furthermore, the model employed a hierarchical contrastive learning approach, capturing generalized user preferences. As a result, our model outperformed other CD(S)R and SR models on real-world datasets.

ACKNOWLEDGMENTS

This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1A2B 5B02001913). The authors would like to thank the AI Service Business Division of SK Telecom for providing GPU cluster support.

REFERENCES

- Jiangxia Cao, Xin Cong, Jiawei Sheng, Tingwen Liu, and Bin Wang. 2022. Contrastive Cross-Domain Sequential Recommendation. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management. 138–147.
- [2] Sheng Gao, Hao Luo, Da Chen, Shantao Li, Patrick Gallinari, and Jun Guo. 2013. Cross-domain recommendation via cluster-level latent factor model. In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part II 13. Springer, 161–176.
- [3] Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016).
- [4] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015).
- [5] Yupeng Hou, Binbin Hu, Zhiqiang Zhang, and Wayne Xin Zhao. 2022. Core: simple and effective session-based recommendation within consistent representation space. In Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval. 1796–1801.
- [6] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In 2018 IEEE international conference on data mining (ICDM). IEEE, 197–206.
- [7] Chenglin Li, Yuanzhen Xie, Chenyun Yu, Bo Hu, Zang Li, Guoqiang Shu, Xiaohu Qie, and Di Niu. 2023. One for All, All for One: Learning and Transferring User Embeddings for Cross-Domain Recommendation. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. 366–374.
- [8] Meng Liu, Jianjun Li, Guohui Li, and Peng Pan. 2020. Cross domain recommendation via bi-directional transfer graph collaborative filtering networks. In Proceedings of the 29th ACM international conference on information & knowledge management. 885–894.
- [9] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. π-net: A parallel information-sharing network for shared-account crossdomain sequential recommendations. In Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. 685– 694.
- [10] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. 43–52.
- [11] Roger B Myerson. 1991. Game theory: analysis of conflict. Harvard university press.
- [12] Ahmed Rashed, Shereen Elsayed, and Lars Schmidt-Thieme. 2022. Context and attribute-aware sequential recommendation via cross-attention. In Proceedings of the 16th ACM Conference on Recommender Systems. 71–80.
- [13] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012).

- [14] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In Proceedings of the 19th international conference on World wide web. 811–820.
- [15] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 650–658.
- [16] Philip D Straffin. 1993. Game theory and strategy. Vol. 36. MAA.
- [17] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In Proceedings of the 28th ACM international conference on information and knowledge management. 1441–1450.
- [18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2818–2826.
- [19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems. 5998–6008.
- [20] Huan Yan, Xiangning Chen, Chen Gao, Yong Li, and Depeng Jin. 2019. Deepapf: Deep attentive probabilistic factorization for multi-site video recommendation. TC 2, 130 (2019), 17–883.
- [21] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, Xiaofang Zhou, et al. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation.. In IJCAI. 4320–4326.
- [22] Wen Zhang, Lingfei Deng, and Dongrui Wu. 2020. Overcoming negative transfer: A survey. arXiv preprint arXiv:2009.00909 (2020), 36.
- [23] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, et al. 2022. RecBole 2.0: Towards a More Up-to-Date Recommendation Library. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management. 4722–4726.
- [24] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Kaiyuan Li, Yushuo Chen, Yujie Lu, Hui Wang, Changxin Tian, Xingyu Pan, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In CIKM.
- [25] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In Proceedings of the 29th ACM international conference on information & knowledge management. 1893–1902.
- [26] Feng Zhu, Chaochao Chen, Yan Wang, Guanfeng Liu, and Xiaolin Zheng. 2019. Dtcdr: A framework for dual-target cross-domain recommendation. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 1533–1542.
- [27] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. 2021. Cross-domain recommendation: challenges, progress, and prospects. arXiv preprint arXiv:2103.01696 (2021).