DREAM: Decoupled Representation via Extraction Attention Module and Supervised Contrastive Learning for Cross-Domain Sequential Recommender

Xiaoxin Ye
z5090071@ad.unsw.edu.au
School of Computer Science and Engineering, UNSW
Sydney, NSW, Australia

Yun Li
y.li@csiro.au
CSIRO's Data61
Sydney, NSW, Australia

Lina Yao
lina.yao@unsw.edu.au
CSIRO's Data61 and School of Computer Science and Engineering, UNSW
Sydney, NSW, Australia

ABSTRACT

Cross-Domain Sequential Recommendation(CDSR) aims to generate accurate predictions for future interactions by leveraging users' cross-domain historical interactions. One major challenge of CDSR is how to jointly learn the single- and cross-domain user preferences efficiently. To enhance the target domain's performance, most existing solutions start by learning the single-domain user preferences within each domain and then transferring the acquired knowledge from the rich domain to the target domain. However, this approach ignores the inter-sequence item relationship and also limits the opportunities for target domain knowledge to enhance the rich domain performance. Moreover, it also ignores the information within the cross-domain sequence. Despite cross-domain sequences being generally noisy and hard to learn directly, they contain valuable user behavior patterns with great potential to enhance performance. Another key challenge of CDSR is data sparsity, which also exists in other recommendation system problems. In the real world, the data distribution of the recommendation system is highly skewed to the popular products, especially on the large-scale dataset with millions of users and items. One more challenge is the class imbalance problem, inherited by the sequential recommendation problem. Generally, each sample only has one positive and thousands of negative samples. To address the above problems together, an innovative **D**ecoupled **R**epresentation via **E**xtraction **A**ttention **M**odule (DREAM) is proposed for CDSR to simultaneously learn single- and cross-domain user preference via decoupled representations. A novel Supervised Contrastive Learning framework is introduced to model the inter-sequence relationship as well as address the data sparsity via data augmentations. DREAM also leverages Focal Loss to put more weight on misclassified samples to address the class-imbalance problem, with another uplift on the overall model performance. Extensive experiments had been conducted on two cross-domain recommendation datasets, demonstrating DREAM outperforms various SOTA cross-domain recommendation algorithms achieving up to a 75% uplift in Movie-Book Scenarios.

CCS CONCEPTS

• **Information systems**;

KEYWORDS

Cross-Domain, Sequential Recommendation System, Contrastive Learning, Decouple Representation

ACM Reference Format:

Xiaoxin Ye, Yun Li, and Lina Yao. 2023. DREAM: Decoupled Representation via Extraction Attention Module and Supervised Contrastive Learning for Cross-Domain Sequential Recommender. In *Seventeenth ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3604915.3608780

1 INTRODUCTION

In recent years, recommendation systems have become an integral part of online platforms, such as e-commerce websites and social media platforms. These systems provide personalized recommendations to users, helping them discover relevant content and improve their overall experience. Traditional recommendation approaches typically focus on recommending items or content within a single domain. However, the increasing availability of data from multiple domains raises a growing demand for exploring recommendations across different domains.

To tackle the problem, share-account Cross-Domain Sequential Recommendation Systems(CDSR) utilize cross-domain in-session information to model the users' sequential behaviors to predict users' next-item events for both domains. There are three types of sequence relationships within Cross-Domain Sequential Recommendations, illustrated in Fig.1, including **intra-sequence**, **cross-domain**, and **inter-sequence relationships**.

Intra-Sequence Relationships. One good example is user 1 in Fig. 1. Her first purchased item is a romantic book, and her next purchased book is romantic again. The intrinsic relationships within the user sequence are the intra-sequence relationships we are trying to capture. Recommenders should be able to identify 'relevant' items from a user's action history and use them to predict the next item[15]. In CDSR settings, the model ideally should be able to capture the sequential relationship for both single- and cross-domains.

RecSys '23, September 18–22, 2023, Singapore, Singapore

ACM ISBN 979-8-4007-0241-9/23/09…$15.00
https://doi.org/10.1145/3604915.3608780

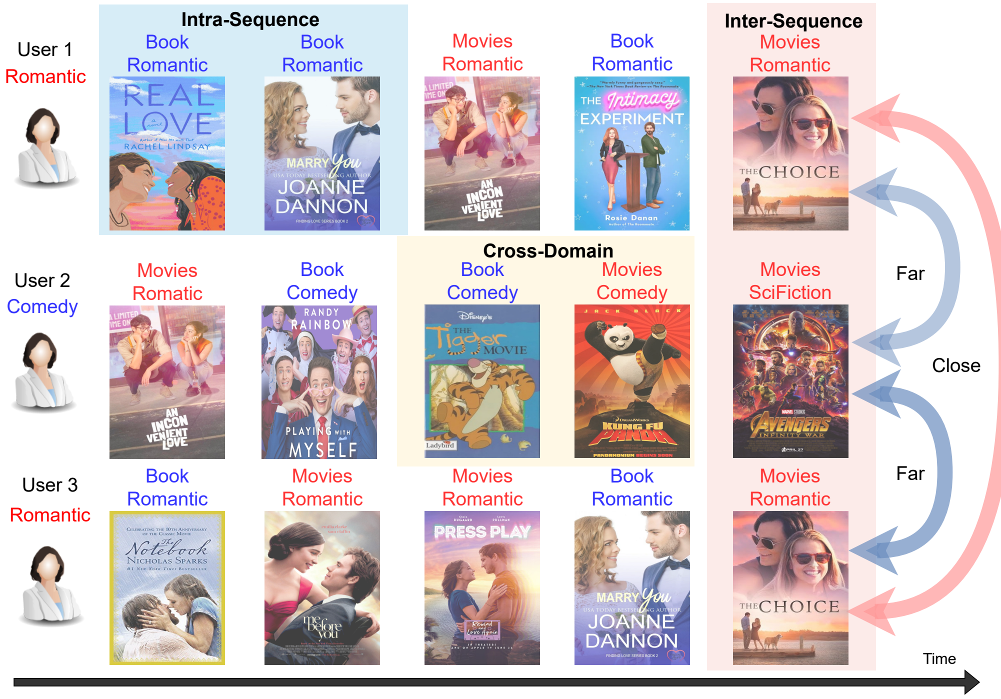


Figure 1: Toy illustrations of CDSR, including intra-sequence, cross-domain, and inter-sequence relationships.

Cross-Domain Relationships. User 2 in Fig. 1 is a good example. She has interactions in both book and movie domains. She purchases a romantic movie and two comedy books at the beginning. The fourth item she bought is 'KongFu Panda'(a comedy movie). Given her past purchase history of a romantic movie, a recommender is unlikely to recommend 'KongFu Panda' without considering the interactions from the book domain. Therefore, to achieve a good model performance in CDSR, models should be able to learn the knowledge from more than one domain and use them to provide more accurate and diverse recommendations to users. However, Cross-domain sequences are generally hard to learn directly, because of the accumulated noise from single-domains.

Inter-Sequence Relationships. As Fig. 1 illustrates, except for intra-sequence and cross-domain relationships, the inter-sequence relationship should also be considered. For example, in Fig. 1, users 1 and 3 share the same interest in the Romantic Movie – The Choice, but have different interests from user 2 in Scientific Fiction Movies. Ideally, the representations of users 1 and 3 should be closed together, while their representation should be far from user 2's since they had different interests/target items from user 2. The acquired knowledge from Inter-sequence can help us to guide a better knowledge transfer among both single and cross-domain[36] as well as allow the user to from their heterogeneous neighbours. It is trivial in the example, but inter-sequence relationships have rarely been studied in Cross-Domain Recommendation Problems. C2DSR[1] attempted to study the inter-sequence relationships in CDSR, but the problem is partially addressed since it purely considered the inter-sequence relationship within the same users without considering the inter-user sequence relationship.

To simultaneously capture the above three relationships for CDSR, we propose an innovative **D**ecoupled **R**epresentation **E**xtraction and **A**ttention **M**odule (**DREAM**) to model user preferences for both single- and cross-domains, as well as a **S**upervised **C**ontrastive **L**earning(**SCL**) framework to model inter-sequence relationships. A uni-directional transformer is leveraged to capture the casualty of item relationships within single- and cross-domain sequences. DREAM builds a decoupled learning mechanism to extract knowledge from single domains and leverage them to improve cross-domain representations without downgrading single-domain representations via gradient stop operation. An innovative Supervised Contrastive Learning mechanism is designed to capture inter-sequence (e.g. inter-domain and inter-user) relationships, aiming at maximizing the relevance between single- and cross-domain representations and the relevance among inter-sequence with the same preferences. SCL also leverages data augmentation to address the data sparsity problem and use semantic information to remove false negative samples(compared to Unsupervised Contrastive learning). Inherited by the Sequential Recommendation problem, CDSR suffers from an extremely class-imbalanced problem. Each sample only has one positive and thousands of negative samples. To tackle this problem, this work uses focal loss instead of the general recommendation loss (e.g. CE, BPR). Focal Loss down-weights well-classified samples and focuses on misclassified samples to prevent the model from being overwhelmed by a large number of easily classified samples [21]. Overall, our main contributions include:

(1) To generate decoupled representations for both single- and cross-domain, DREAM models the single-domain representation and refines the cross-domain representation by extracting the knowledge from single domains.
(2) To model inter-sequence relationships and address the data sparsity problem, we design an innovative Supervised Contrastive Learning framework via leveraging sequence augmentation to (i) maximize the relevance between single- and cross-domain representations within the same user, (ii) maximize the relevance among inter-sequence with the same preferences, (iii) minimize the relevance among inter-sequence with different preferences.
(3) To address the class imbalance problem, we extended focal loss to CDSR and conducted an in-deep theoretical and experimental analysis to demonstrate how focal loss addresses class imbalance problems.
(4) We conduct extensive experiments on the corrected CDSR data set[1] [1] to demonstrate that DREAM significantly outperforms previous SOTA baselines(with up to 75% performance uplift on MRR@10).

2 RELATED WORK

2.1 Sequential Recommendation(SR)

To capture the sequential relationship, many algorithms were developed to model intra-sequence relationships for single-domain. Early works on SR utilized Markov Chains[7] to model the item-to-item relationship. The recent development of deep learning inspired many deep sequential recommendations[9, 15], such as CNN[33], RNN[9]. SRGNN[35] introduced a GNN architecture for recommendation models to capture the dynamic changes in users' preferences. AttRec[39], SASRec[15], and BERT4Rec[31] leveraged self-attention and transformer to extend the capability of modelling the long sequences. By adopting a novel twin-attention sequential framework, LSAN[20] was able to capture users' both long- and short-term preference signals. By leveraging a pre-trained Transformer, ASReP[25] alleviated the data-sparsity issue on the revised user behaviour sequences to augment short sequences. Recently, IDNP[6] utilized generative neural processes to model user interests from a functional perspective, further enhancing the model performance on the short sequences. Rec-denoiser[4] was developed to prune noisy attention, resulting in sparse and clean attention distributions, leading to better training with much less distortion from the noisy information. Also, SDIL[13] comprehensively models dynamic interest based on temporal positive and negative excitation learning as well as capture both static and dynamic interest. Despite their promising performance in the single-domain sequential recommendation, those original designs ignored the cross-domain and inter-sequence relationship. This paper studies the opportunities of extending Transformer architecture to jointly learn the users' single- and cross-domain sequential patterns.

[1]Thanks to C2DSR[1] for addressing the risk of information leakage and sharing the correct data set. (leveraging the future information for past prediction)

2.2 Cross-Domain Recommendation(CDR)

Cross-Domain Recommendation models leveraged the historical interactions across domains to provide better item recommendations for several domains. Many existing algorithms started with learning single-domain knowledge and then built a cross-domain transfer to bridge the knowledge gap for each domain. CoNet [11] and BiTGCF[22] were examples of this category. MiNet[27] and Dual transfer[19] were also proposed to further enhance the CDR performance. SAVAE[30], VDEA[23], and CDRIB[2] utilized the variational auto-encoder (VAE) framework to exploit user domain-invariant embedding across different domains. DARec [38] leveraged domain adaptation techniques allowing knowledge transferred from a source domain to a different but related target domain. A few meta networks [41, 42] were also proposed to handle the cold-start problem in CDR. To tackle the Cross-Domain Sequential Recommendation problem, PSJNet[32], and π-Net[26] developed some gating mechanisms to transfer the single-domain knowledge to another domain. DA-GCN [3] and C2DSR[1] extended the GNN architecture to capture sequential relationships for both single- and cross-domain. One recent work RecGURU[18] utilized transformers to model sequential relationships for multiple domains and generalize the knowledge to other domains. This work focuses on modelling decoupled representations for both single- and cross-domain via DREAM as well as maximizing the relevance between single- and cross-domain via Supervised Contrastive Learning.

2.3 Contrastive Learning

Contrastive Learning (CL) aims to learn models by contrasting positive pairs against negative pairs to generate better representations for each sample and the downstream tasks[5]. Inspired by those successes in Computer Vision[5], CL4Rec[37] and CoSeRec [24] attempted to use contrastive learning to capture inter-sequence relationships in the single-domain sequential recommendation. Inspired by DIM[10], C2DSR leveraged a noise contrastive objective(InfoMax) to maximize the mutual information between single- and cross-domain, but it failed to learn the inter-sequence relationships among users. Similarly, CCDR[36] designed intra-domain CL and three types of inter-domain CL to achieve performance uplift. Additionally, HMP[12] devised a novel semantics-enhanced context embedding module to generate more informative context embedding for further improving the recommendation performance. DuoRec[29] pointed out that Unsupervised Contrastive Learning(UCL) generates false-negative samples since UCL treats all other sequences in the batch as negative. In Fig. 1, user 1 and 3 are mutually negative samples for each other within UCL. DuoRec[29] employed Supervised Contrastive Learning(SCL) to reduce the choice of generating false negative samples by leveraging the semantic information. If two sequences share the same target item, SCL treats them as positive pairs rather than negative pairs. Therefore, user 1 and 3 are defined as positive pairs in the same batch. Inspired by DuoRec, DREAM extends SCL to model inter-sequence relationships in cross-domain scenarios.

3 PROBLEM STATEMENT

This work focuses on share-account Cross-Domain Sequential Recommendation where each sequence involves two domains, namely domain X and Y. Let S denote the set of sessions. For each user u, we have $(s^x, s^y, s^c)_u \in S$, where $s^c = [x_1, y_2, x_3, x_4, y_5 ... y_{|s^y|} ... x_{|s^x|}]$ represents the cross-domain sessions, containing all the items user u has interacted with in chronological order. $s^x = [x_1, x_3, x_4, ... x_{|s^x|}]$and $s^y = [y_2, y_5 ... y_{|s^y|}]$ are subsets of s^c only containing the items from every single domain. CDSR model aims at predicting the next item by modelling the session representation for both single- and cross-domain sessions. The next item prediction(domain X as an example. It also holds for domain Y) is

$$\underset{x_{t+1} \in X}{\text{argmax}} \ P(x_{t+1} | s^x, s^y, s^c)$$

where $P(x_{t+1} | s^x, s^y, s^c)$ is denoted as the estimated probability of the next item in domain X, similar to $P(y_{t+1} | s^x, s^y, s^c)$.

4 METHODOLOGY

The core idea of DREAM is to capture users' dynamic preferences via decoupled representations. This session explains the details of **DREAM** blocks, including a **D**ecoupled **R**epresentation layer and domain **E**xtraction and **A**ttention **M**odule. Moreover, it also describes the Supervised Contrastive Learning framework (sec 4.3), Focal Loss (sec 4.4), and the training objective and inference strategy of DREAM(sec 4.5), followed by the Complexity Analysis (sec 4.6).

4.1 Disentangled Representation

As Fig. 2 illustrated, DREAM employs Transformer architecture to generate user representations. Hence, this part will describe the details of the embedding layer and how transformers are applied to generate decoupled representations.

4.1.1 Embedding Layer. DREAM initializes item embedding layers $E^x \in \mathbb{R}^{|X|*d}$ and $E^y \in \mathbb{R}^{|Y|*d}$ for domains X and Y. $|X|$and $|Y|$ represents the total item space for domain X and Y. DREAM also initializes positional embedding layer $Pos \in \mathbb{R}^{T*d}$ to preserve the time order of sequences, where T stands for the max length of the interaction sequence [2]. Formally, if we have a user cross-domain sequence $s_u = [x_1, y_2, x_3 ..., x_{t-k}, .., y_t]$, the complete item representation of y_t is $h_t = e_t^y + pos_t$, where h_t is the complete input vector at time t for the next layer, $e_t^y \in E^y$ is item representation of y_t, and $pos_t \in Pos$ is the positional embedding of y_t at time t.

4.1.2 Self-Attention Sequential Encoder. With the input embedding from the previous layers, a uni-directional Transformer is utilized to model the user's sequential representation and the causality of users' preferences by the multi-head attention mechanism [15]. Assuming the hidden representation H from the embedding layer or previous layer, a multi-head Transformer encoder is applied to model the next layer. Formally, $H_{l+1} = Transformer(H_l)$, where $H_{l+1} = [h_0, h_1, h_2 ... h_t]$, and h_t is the user representation at time t.

Similar to the C2DSR[1], DREAM first constructs three input sequences, including s^c, s^x, and s^y. s^c contains all the items users had interacted with in chronological order. s^x and s^y are subsets of s^c containing the item from domain X/ Y. Apart from these three original input sequences(s^c, s^x, and s^y), we also generate one augment sequence from the original sequence, and we will describe how to augment sequences in sec 4.3. DREAM consists of three separate transformers ($Transformer^c$, $Transformer^x$, and $Transformer^y$) which are responsible for capturing single- and cross-domain information and generating decoupled representations as follows,

$$H_{l+1}^c = Transformer^c(H_l^c)$$

$$H_{l+1}^x = Transformer^x(H_l^x)$$

$$H_{l+1}^y = Transformer^y(H_l^y)$$

Where H_l^c, H_l^x, and H_l^y are embeddings from the previous layer or the initial embedding layer for cross, X, and Y domains, while H_{l+1}^c, H_{l+1}^x, and H_{l+1}^y are the outputs of this layer.

4.2 Domain Extraction and Attention Module

Instead of directly using H^c for recommendation tasks, since the cross-domain sequences are generally hard to learn directly, DREAM builds a domain extraction and attention module to refine the cross-domain representation. This module contains domain extraction, single-domain attention, and cross-domain attention layers.

Domain Extraction(DE) is a Feed-Forward Network(FFN) with a stop-gradient operation. The single-domain representations are the input for the DE but a stop-gradient operator is applied before inputting them into DE:

$$H_e^x = DE(H^x) = FFN^x(SG(H^x)) = ReLU(SG(H^x)W^1 + b^1)W^2 + b^2$$

Where $SG(H^x)$ is a stop-gradient operation applied to H^x, FFN is a Feed-Forward Network, $FFN(H) = ReLU(W^1 H + b^1)W^2 + b^2$, W^1 and W^2 are both learnable d*d matrices, b^1 and b^2 are both learnable d-dimensional vectors.

The benefit of stop-gradient operation is allowing the single-domain transformer to learn the behaviour pattern purely from each single-domain without getting inherited noises incurred from cross-domain or another domain sequence, achieving the goal of decoupling knowledge for each domain. Also, based on our experiment, the model is prone to learn trivial patterns or overfitting the training dataset if the stop-gradient operation is not applied.

Attention Module for Single-Domain (AM^s) The extracted single-domain representations H_e^x and H_e^y are sent into a single-domain attention module to self-select the information to share with the cross-domain representation:

$$H^s = AM^s(H_e^x, H_e^y) = H_e^x * Att^s + H_e^y * (1 - Att^s)$$

$$Att^s = sigmoid(H_e^x * H_e^y)$$

Where H_e^x and H_e^y are the outputs of the previous domain extraction layer for domains X and Y.

Attention Module for Cross-Domain(AM^c) is a gating layer to refine the cross-domain representation:

$$H^{c\prime} = AM^c(H^s, H^c) = H^s * Att^c + H^c * (1 - Att^c)$$

$$Att^c = sigmoid(H^s * H^c)$$

Where H^s is a single-domain representation from the previous single-domain attention layer and H^c is a cross-domain sequence representation from the transformer encoder layer, and $H^{c\prime}$ is the final cross-domain sequence representation.

[2]T=15 in our study

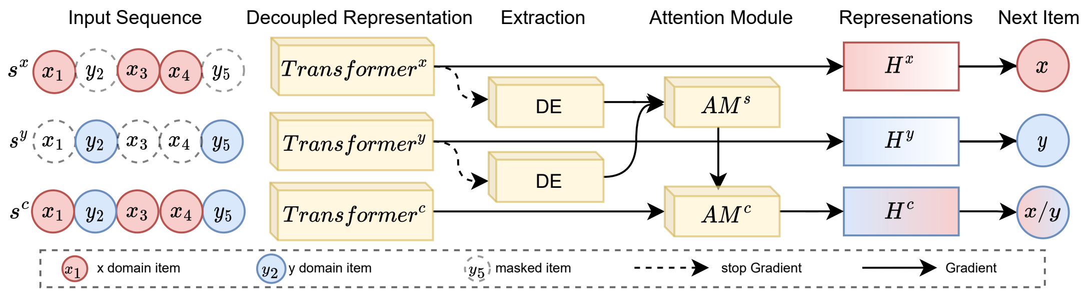


Figure 2: Illustration of our Decoupled Representation, domain Extraction Layer, and Attention Module

4.3 Supervised Contrastive Learning

DREAM develops a novel supervised contrastive learning mechanism to model the inter-sequence relationships between cross-domain and single-domain within one user and among inter-users. This session explains the structure of our supervised contrastive learning mechanism. We start with how to construct the augmented sequence and how to definite the positive and negative pairs, followed by the loss function SupCon.

4.3.1 Sequence Augmentation. Inspired by CoSeRec[24] and CL4Rec[37], DREAM generates the augmented sequence via five methods, including Crop, Mask, Reorder, Substitute, and Insert. To provide more opportunities for the non-popular products to learn, we do not leverage the informative augmentation method suggested by CoSeRec[24]. If we follow CoSeRec's setting, the products close to the popular products are more likely to get selected, reducing the chance for non-popular products to be selected. Therefore, we are randomly selecting the products to be substituted and inserted. The augmentation operation is only applied to the cross-domain sequence and not to the single-domain sequence. Given a cross-domain sequence $s^c = [y_1, x_2, y_3, x_4, x_5]$, we randomly apply one of the augmentation methods (Crop, Mask, Reorder, Substitute, and Insert) on s^c to generate the augmented sequence $s' = [y_1, x_2, y_3, y_6, x_5]$ (where x_4 is substituted by y_6). It is worth mentioning that we do not set a limited product pool for Substitute and Insert. In C2DSR[1], given products from domain X, the Substituted and Inserted products must come from domain Y. To provide more flexibility for the products to be learned, this constraint is removed.

4.3.2 Positive and Negative Sampling. DREAM utilizes the two levels of semantic similarity within the batch, including semantic similarities within the same user and among users, to determine whether they are positive or negative pairs. For the semantic similarity within the same user, similar to the original Unsupervised Contrastive Learning(UCL), the augmented sequences from the same source sequence are trusted as positive. Since this work focuses on the CDSR, we extended its definition. One user's cross/single-domain, and augmented sequence representations are mutually positive since they represent one user's preference. The key difference between UCL and Supervised Contrastive Learning(SCL) sits at the second level of semantic similarity between users. UCL generally treats all other samples within the same batch as negative pairs. In DREAM's SCL framework, when users 1 and 3 share the same next item, all their sequences (single/cross-domain and augment sequences) are treated as mutually positive rather than negative. On the other hand, if user 2 and user 3 had different next-item, their representations are treated as negative pairs pushing between each other to generate a decent margin. Therefore, in this case, the sequence in $\{h_1^c, h_1', h_1^s, h_3^c, h_3', h_3^s\}$ are mutually positive pairs, and the sequences in $\{h_2^c, h_2', h_2^s\}$ are mutually positive pairs. Other pairs are negative pairs due to their different preference, where h_1', h_1^s, and h_1^c represent the augmented, single-domain, and cross-domain representation of user 1. When user 1's last item is from domain X, $h_1^s = h_1^x$. When user 1's last item is from domain Y, $h_1^s = h_1^y$.

4.3.3 Supervised Contrastive Learning Loss. Considering the training batch of size $|B|$, after augmentation, there are $3|B|$ hidden vectors $\mathcal{H} = \{h_1^c, h_1', h_1^s, h_2^c, h_2', h_2^s ... h_{|B|}^c, h_{|B|}', h_{|B|}^s\}$ where h_1', h_1^s, h_1^c represent the augmented, single, and cross-domain representation of user 1, respectively. Accordingly, our Supervised Contrastive Learning Loss for each positive sequence pair i and j is as follows,

$$\mathcal{L}_{SupCon} = -log\left(\frac{exp(h_i * h_j/\tau)}{\sum_{h^- \in \mathcal{H}_i^-} exp(h_i * h^-/\tau) + \sum_{h^+ \in \mathcal{H}_i^+} exp(h_i * h^+/\tau)}\right) \tag{1}$$

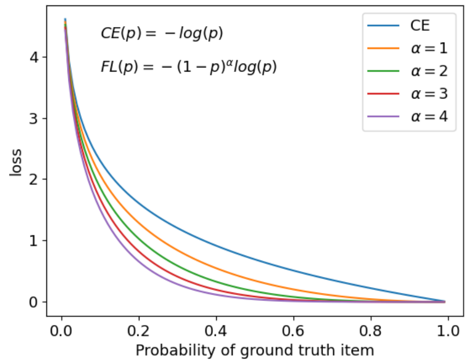
where h_i and h^- are vectors from $\mathcal{H}$ and $\mathcal{H}_i^-$(negative pairs set). h^+ and h_j are vectors from $\mathcal{H}_i^+$(positive pairs set), Taking h_1^s as an example, its full positive set ($\mathcal{H}_i^+$) is $\{h_1^c, h_1', h_3^c, h_3', h_3^s\}$ and its negative set ($\mathcal{H}_i^-$) is $\{h_2^c, h_2', h_2^s ... h_{|B|}^c, h_{|B|}', h_{|B|}^s\}$ if only user 3 shares the same interests with user 1 within the batch.

4.4 Focal Loss

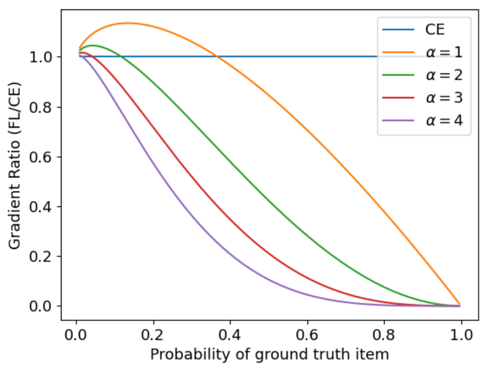
DREAM employs the Focal Loss(FL) function instead of commonly-used loss functions(e.g. Cross-Entropy or BPR). FL addresses the class-imbalance problems by reducing the relative loss for the well-classified samples and putting more weight on misclassified examples [21]. We start with what is FL and then provide an experimental and theoretical analysis of how FL puts more weight on misclassified examples. Focal Loss is a revised version of Cross-Entropy(CE), with a weighting factor $(1-p)^\alpha$, where p represents the probabilities of the ground-true item,

$$FL(p) = -(1-p)^\alpha log(p) \tag{2}$$

Fig. 3a illustrates the relationship between the probabilities and loss for CE and FL. Clearly, when the p is larger than 0.5, FL has a much lower loss than CE. Fig. 3b show that when the probability



(a) Focal Loss and Cross-Entropy Loss



(b) Gradient Ratio $\frac{\partial FL}{\partial p} / \frac{\partial CE}{\partial p}$

Figure 3: Illustration of Focal Loss

is small, $\frac{\partial FL}{\partial p} / \frac{\partial CE}{\partial p}$ is larger than 1, meaning that FL has a relatively larger gradient than CE. When the predicted probability is large, $\frac{\partial FL}{\partial p} / \frac{\partial CE}{\partial p}$ is lower than 1, meaning that FL has a relatively lower gradient than CE. For instance, when $p = 0.95$ and $\alpha = 1$, $\frac{\partial FL}{\partial p} / \frac{\partial CE}{\partial p} \approx 0.10$, meaning the gradient of FL is about 10% of CE. To theoretically analyze how FL puts more weight on hard samples, we calculate the gradient of both losses and then analyze when $\frac{\partial FL}{\partial p} / \frac{\partial CE}{\partial p} > 1$. Take $\alpha = 1$ as an example,

$$\frac{\partial FL}{\partial p} / \frac{\partial CE}{\partial p} = \frac{-(\frac{1}{p} - log(p) - 1)}{-\frac{1}{p}} = 1 - p(log(p)+1) > 1 \Rightarrow p < e^{-1}$$

Therefore, given $\alpha = 1$, when $p < e^{-1}$, FL has larger gradients than CE, allowing FL to learn faster than CE. For well-classified samples ($p > 0.5$), FL has much lower gradients than CE, slowing down the learning speed and preventing the model from being overwhelmed by the easy examples.

4.5 Training Objective and Inference

4.5.1 Joint Training Objective. DREAM includes $\mathcal{L}^x_{single}$ and $\mathcal{L}^y_{single}$ for single-domains (X/Y) recommendation tasks, $\mathcal{L}^x_{cross}$ and $\mathcal{L}^y_{cross}$ for cross-domain recommendation tasks, and $\mathcal{L}_{SupCon}$ for Supervised Contrastive Learning. Following the most common multi-task training strategy, we had the overall loss function as follows

$$\mathcal{L}_{total} = \underbrace{\mathcal{L}^x_{single} + \mathcal{L}^y_{single}}_{Single-Domain\ Loss} + \underbrace{\mathcal{L}^x_{cross} + \mathcal{L}^y_{cross}}_{Cross-Domain\ Loss} + \underbrace{\mathcal{L}_{SupCon}}_{SCL\ Loss} \tag{3}$$

$$\mathcal{L}_{rec} = \sum FL(P) = \sum -(1-P)^{\alpha} log(P) \tag{4}$$

$$P = \begin{cases} softmax(H^x * E^{|X|}/\tau) & for\ \mathcal{L}^x_{single} \\ softmax(H^y * E^{|Y|}/\tau) & for\ \mathcal{L}^y_{single} \\ softmax(H^c * E^{|X|}/\tau) & for\ \mathcal{L}^x_{cross} \\ softmax(H^c * E^{|Y|}/\tau) & for\ \mathcal{L}^y_{cross} \end{cases} \tag{5}$$

where $\mathcal{L}_{SupCon}$ is calculated via Eq.1, $\mathcal{L}^x_{single}$, $\mathcal{L}^y_{single}$, $\mathcal{L}^x_{cross}$ and $\mathcal{L}^y_{cross}$ are calculated via Eq. 4, and P represents the probabilities of the next item (calculated via Eq. 5), H^x, H^y, and H^c are the X, Y, and cross-domain sequence representation, $E^{|X|}$ and $E^{|Y|}$ are the item embedding of domain X and Y, and τ is a temperature hyper-parameter. Algorithm 1 and 2 summarize the overall steps within DREAM architecture and training procedure.

Algorithm 1: DREAM Architecture

Function DREAM(s^x, s^y, s^c, s', L)
  $H^x \leftarrow E(s^x) + Pos$;
  $H^y \leftarrow E(s^y) + Pos$;
  $H^c \leftarrow E(s^c) + Pos$;
  $H' \leftarrow E(s') + Pos$;
  for $i \leftarrow 1$ *to* L **do**
    $H^x \leftarrow Transformer^x(H^x)$;
    $H^y \leftarrow Transformer^y(H^y)$;
    $H^c \leftarrow Transformer^c(H^c)$;
    $H' \leftarrow Transformer^c(H')$;
    $H^x_e \leftarrow Extraction(H^x)$;
    $H^y_e \leftarrow Extraction(H^y)$;
    $H^s \leftarrow AM^s(H^x_e, H^y_e)$;
    $H^c \leftarrow AM^c(H^s, H^c)$;
  return H^x, H^y, H^c, H'

Algorithm 2: Overall Training Procedure for DREAM

Input: Training Sequence $\mathcal{S}$, Number of DREAM Layers L
for *each mini-batch sequences* $s \in \mathcal{S}$ **do**
  $s^x, s^y, s^c \leftarrow$ Generate Sequence(s);
  $s' \leftarrow$ Data Augmentation(s^c);
  $H^x, H^y, H^c, H' \leftarrow$ DREAM(s^x, s^y, s^c, s', L) from Alg. 1 ;
  Calculate P^x_s, P^y_s,P^x_c, P^y_c via Eq.5;
  Calculate $\mathcal{L}^x_{single}$, $\mathcal{L}^y_{single}$, $\mathcal{L}^x_{cross}$, $\mathcal{L}^y_{cross}$ via Eq.4;
  Calculate $\mathcal{L}_{SupCon}$;
  $\mathcal{L}_{total} \leftarrow \mathcal{L}^x_{single} + \mathcal{L}^y_{single} + \mathcal{L}^x_{cross} + \mathcal{L}^y_{cross} + \mathcal{L}_{SupCon}$;

Lastly, it is worth pointing out that the augmented representations and cross-domain representations share the same transformer since augmented sequences are augmented via cross-domain. Moreover, H^c is the output of the Domain Extraction and Attention Module rather than the transformer directly. If the representation from the transformer is used directly, the architecture will be downgraded to a training strategy without refining the cross-domain representations. Also, Focal Loss is used here since it is able to address the class imbalance problems, but it can be replaced by any other recommendation loss like CE. $\mathcal{L}_{SupCon}$ can also be replaced by $\mathcal{L}_{InfoNec}$ but it will be downgraded to UCL.

4.5.2 Inference. During the inference stage, both the single- and cross-domain representations are utilized for inference. If we want to make an inference for domain X, given the latest representation h_t^c and h_t^x, generated by DREAM, the item with the highest scores is selected as the recommended item (also holds for domain Y),

$$\underset{x_{t+1}\in X}{\operatorname{argmax}} P(x_{t+1}|s^x, s^y, s^c) = softmax(h_t^c * E^{|X|} + h_t^x * E^{|X|})$$

Where h_t^c and h_t^x are the cross- and single-domain representation at time t, and $E^{|X|}$ is item embedding of domain X.

4.6 Complexity Analysis

This session compares the model complexity with the existing SOTA algorithm C2DSR[1]. Table 1 summarizes the major steps and time complexities of both algorithms. Since C2DSR and DREAM leverage the same Attention architecture, they share the same time and memory complexity at this step. Although the time complexity of the two algorithms is different at the Contrastive Learning step, D and N are set to 256 in the experiment, so they share a similar time complexity here. Compared with C2DSR, DREAM has no preprocessing step($O(|V|)$) and GNN encoder($O(|V|D)$) but adds a light-wise step of extracting and refining step for cross-domain representation with a linear time complexity $O(NTD)$. $|V|$ is the number of item interactions(generally a large value). N is the minibatch size. T is the maximum length of the sequence. D is the dimension of each item representation. Overall, DREAM had a lighter model architecture than C2DSR without GNN steps.

5 EXPERIMENT

To answer the following research questions: **Q1**: Does DREAM outperform other SOTA methods? **Q2**: Does DREAM generate the decoupled representation for single- and cross-domain recommendations? **Q3**: Does Focal Loss enhance performance? **Q4**: Does SCL outperform UCL in the CDSR setting? **Q5**: Can SCL maximize the relevance between single- and cross-domain representations? **Q6**: Does contrastive learning improve model performance? This session starts with explaining the experiment settings, including evaluation protocol, datasets, and implementation details. The results of the experiment are explained in the next section.

5.1 Experiment settings

5.1.1 Datasets. Thanks to C2DSR[1] for identifying the information leak problem and providing the corrected dataset, this work follows C2DSR's evaluation settings(dataset, evaluation protocol, and baselines). To have a fair and comprehensive experiment, this work uses two well-known public CDSR benchmarks from four domains, including 'Movie-Book' (Amazon) and 'Food-Kitchen' (Amazon). To mimic the cross-domain setting and have a fair comparison, items with fewer than 10 interactions are excluded, and the users who only interacted with one single domain are removed as well. Moreover, to satisfy CDSR's sequential constraints, sequences with fewer than three items from each domain within a specified period (e.g., a month for the 'Movie-Book' and a year for the 'Food-Kitchen') are excluded as well. For the training/validation/test partition, the users' latest interaction sequences are equally divided into the validation/test set and the other interactions for the training set[1]. Table 2 provides the statistics of the corrected datasets from C2DSR[1].

5.1.2 Evaluation Protocol. The leave-one-out method is leveraged to calculate the recommendation performance, following previous works' settings [1, 15, 40]. Aiming to guarantee unbiased and fair evaluation, this work follows Rendle's work[17] to calculate 1,000 scores for each validation/test case (including 999 negative items and 1 positive item). Then, the Top-K recommendation performance of the 1,000 ranking lists is reported in terms of MRR@10 (Mean Reciprocal Rank)[34], NDCG@5, 10 (Normalized Discounted Cumulative Gain)[14], and HR@1, 5, 10 (Hit Ratio).

5.1.3 Compared baselines. Four categories of baselines are used.

(1) Cross-Domain Collaborative Filtering(**CDCF**): (i) **NCF-MLP**[8] learns user and item representations via MLP networks. To adapt to two domain settings, user and item are learned by two separate base MLP networks with shared user embeddings. (ii) **CoNet** [11], a classic cross-domain recommendation, starts with modelling the two domain's interactions separately and then transferring the knowledge between two base networks via a cross-domain network. The sequential constraints are ignored here.
(2) Single-Domain Sequential Recommendation(**SDSR**) (i) **GRU4Rec** [9] models intra-sequence pattern via a GRU architecture, (ii) **SASRec**[15], a well-known SOTA baseline for SDSR, uses a self-attention model to capture sequence pattern. (iii) **SRGNN** uses GNN to capture the sequential relationships.
(3) Contrastive Learning for Single-Domain Sequential Recommendation(**CL**) (i) **CL4Rec** [37], a strong CL baseline, utilizes item cropping, masking, and reordering as augmentations for Contrastive Learning, (ii) **CoSeRec** [24], a following up work of CL4Rec, leverages informative augmentations.
(4) Cross-Domain Sequential Recommendation(**CDSR**) (i)π**-net**[26], pioneering work for CDSR, devises a novel gating recurrent module to transfer knowledge across domains. (ii) **PSJNet** [32] transfers the different user intentions across domains via a parallel split-join scheme. (iii) **C2DSR**[1] leverages GNN to capture sequential patterns and InfoMax loss to enhance the correlation between single- and cross-domain user representations. For this category, we use the self-attention module as sequence encoder.

5.1.4 Implementation and Hyperparameter Setting. Our algorithm is built on PyTorch[28], and we follow the same evaluation protocol, implementation, and parameter setting as in C2DSR[1] to mimic a fair comparison. Embedding size(D) and mini-batch size(N) are fixed as 256, while the training epoch and the dropout are

Table 1: Overall Model Complexity Analysis

		Preprocess	Graph Encoder	Attention	Extraction	CL
C2DSR	Module	Build Graph	GNN *3	Transformer *4	-	InfoMax
	Complexity	$O(\|V\|)$	$O(\|V\|D)$	$O(NT^2D)$	-	$O(TND^2)$
DREAM	Module	-	-	Transformer *4	Extraction	SupCon
	Complexity	-	-	$O(NT^2D)$	$O(NTD)$	$O(TN^2D)$

$|V|$ is the number of edges, N is minibatch size, T is the maximum sequence length, D is the dimension of item representation.

Table 2: Statistics of two CDSR scenarios

Scenarios	#Items		Avg.Length	#Users	#Valid		#Test	
	Domain X	Domain Y			Domain X	Domain Y	Domain X	Domain Y
Food-Kitchen	29207	34886	9.91	34117	2722	5451	2747	5659
Movie-Book	36845	63937	11.98	58515	2032	5612	1978	5730

#Items represent each domain's total item size. Avg.Length represents the average length of the training sequence.
#Users represents the number of training sequences/users and all of them are overlapped for two domains.
#Valid/Test domain X represents the number of validation/Test sequence's last target value from domain X

fixed as 100 and 0.3. Adam[16] is selected as the optimizer for updating the parameters. $L2$ regularizer coefficient is selected from $\{0.0001, 0.00005, 0.00001\}$, and the learning rate lr is selected from $\{0.001, 0.0005, 0.0001\}$. For C2DSR, we select the depth of GNN L from $\{1, 2, 3, 4\}$, and the harmonic factor λ is selected from 0.1 to 0.9 with a step length of 0.1. For SASRec-based algorithms (such as SASRrec, CL4Rec, and CoSeRec) two single-head attention blocks and the learnable position embedding are adapted here. We follow the suggested parameters for CL4Rec and CoSeRec. The channel number is set to 5 for π -net, as the original paper[26]. For our algorithm, α for Focal Loss is selected from $\{1, 2, 3, 4\}$, and temperature τ is selected from $\{0.3, 0.5, 1, 2, 3, 4, 16\}$. The parameters for sequence augmentation are set, as CoSeRec[24] suggested. The best-performing models are selected based on the highest MRR performance on the validation set and their performances are reported on the test set.

6 EXPERIMENTAL RESULTS

This session leverages the experimental results to answer the six research questions(sec. 5). It starts with the overall model performance to answer question 1, followed by an ablation study to answer questions 2-6.

6.1 Overall Model Performance

6.1.1 Q1: Does DREAM outperform other existing SOTA methods? Table 3 and 4 demonstrate the model performance on two CDSR scenarios, including "Movie-Book" and "Food-Kitchen". Several insights are found via the above tables, (1) The SDSR baselines like SASRec, GRU4Rec, and SR-GNN consistently out-performance the CDCF (such as NCF-MLP and CoNet), validating that the intra-sequence relationship provides a strong signal for us to understand user's dynamic interest. SASRec and SR-GNN consistently perform better than GRU4Rec, demonstrating that Self-Attention blocks and Graph Neural Networks have more robust capabilities for capturing intra-sequence relationships than Gated Recurrent Units(GRU). (2) Contrastive-learning models generally perform better than the SDSR, except for MRR@10 for Book Domain. Benefiting from leveraging the augmented sequence, CL models can capture the inter-sequence relationship and address the sparsity problem. (3) By leveraging the cross-domain knowledge, CDSR baselines further extend the performance. Within CDSR, π-net and PSJNet demonstrate great capabilities that not only capture the item sequential relationships but also be able to learn from the cross-domain information. One interesting result is that π-net and PSJNet have poorer performances than the Contrastive Learning algorithms. One reason may be that both data sets are highly sparse, leading to a lot of non-popular products having limited learning opportunities. However, contrastive learning is leveraging the augmented sequence to alleviate the sparsity problem. Therefore, the introduction of contrastive learning in C2DSR further enhances the performance of the CDSR problem. (4) DREAM largely outperforms all baselines, demonstrating the robust effectiveness of our model. Compared with the pioneering method(C2DSR), decoupled representations show a promising direction for modelling single- and cross-domain representation, and also Supervised Contrastive Learning shows a more efficient way of modelling inter-sequence relationships compared with InfoMax.

6.2 Ablation study

To answer research questions 2-6, this subsection conducts an ablation study on the "Movie-Book" scenario to validate the effectiveness of DREAM architecture and Supervised Contrastive Learning for the cross-domain Sequential Recommendation problem. This experiment trained several model variants and conducted inference in three approaches(including the single, cross, and combine). The model variants are as follows,

- **Cross-SAS** trains one transformer on cross-domain sequences
- **Single-SAS** trains two separate transformers on the corresponding single-domain sequences
- **C2DSR** trains a C2DSR model

Table 3: Performance comparison of different methods on Movie-Book. Bold scores are the best in the method group, while underlined scores are the second best. Impr% represents the model improvement compared to the second-best model

		Movie						Book					
		MRR	NDCG		HR			MRR	NDCG		HR		
		10	5	10	1	5	10	10	5	10	1	5	10
CDCF	NCF-MLP	3.05	2.26	2.96	1.41	3.13	5.3	1.43	1.06	1.26	0.62	1.39	2.18
	CoNet	3.07	2.42	3.01	1.31	3.48	5.35	1.45	1.04	1.28	0.64	1.44	2.19
SDS	SASRec	3.79	3.23	3.69	2.37	3.99	5.2	1.81	1.41	1.71	0.95	1.83	2.75
	GRU4Rec	3.83	3.14	3.73	2.27	3.39	5.4	1.68	1.34	1.52	0.91	1.81	2.37
	SR-GNN	3.85	3.27	3.78	2.22	4.19	5.81	1.78	1.4	1.66	0.89	1.9	2.72
CL	CL4Rec	4.94	5.19	6.20	3.03	7.23	10.31	2.14	2.18	2.61	1.54	2.83	4.19
	CoSeRec	4.99	5.33	6.42	2.98	7.74	11.12	2.19	2.26	2.79	1.41	3.12	4.78
CDS	Pi-Net	4.16	3.72	4.17	2.52	4.75	6.11	2.17	1.84	2.03	1.43	2.25	2.84
	PSJNet	4.63	4.06	4.76	2.78	5.3	7.53	2.44	2.07	2.35	1.66	2.58	3.28
	C2DSR	4.94	5.05	6.09	3.24	6.67	9.91	2.74	2.79	3.08	2.23	3.30	4.22
Our	DREAM	**8.77**	**9.29**	**11.11**	**5.29**	**13.18**	**18.79**	**4.06**	**4.26**	**4.83**	**2.87**	**5.58**	**7.34**
	Impr%	75.71	74.39	73.16	63.49	70.39	68.94	48.29	52.87	56.69	28.48	69.17	53.50

Table 4: Performance comparison of different methods on Food-Kitchen. Bold scores are the best in the method group, while underlined scores are the second best. Impr% represents the model improvement compared the second-best model

		Food						Kitchen					
		MRR	NDCG		HR			MRR	NDCG		HR		
		10	5	10	1	5	10	10	5	10	1	5	10
CDCF	NCF-MLP	4.49	3.94	4.51	2.68	5.1	6.86	2.18	1.57	2.03	0.91	2.23	3.65
	CoNet	4.13	3.61	4.14	2.42	4.77	6.35	2.17	1.5	2.11	0.95	2.07	3.71
SDS	SASRec	7.3	6.9	7.79	4.73	8.92	11.68	3.79	3.35	3.93	1.92	4.78	6.62
	GRU4Rec	5.79	5.48	6.13	3.63	7.12	9.11	3.06	2.55	3.1	1.61	3.5	5.22
	SR-GNN	7.84	7.58	8.35	5.03	9.88	12.27	4.01	3.47	4.13	2.07	4.8	6.84
CL	CL4Rec	7.93	8.49	9.29	5.45	11.20	13.67	3.91	4.18	4.87	2.39	5.89	8.01
	CoSeRec	8.05	8.62	9.40	5.71	11.27	13.71	3.59	3.76	4.59	2.14	5.30	7.90
CDS	Pi-Net	7.68	7.32	8.13	5.25	9.25	11.75	3.53	2.98	3.73	1.57	4.34	6.67
	PSJNet	8.33	8.07	8.77	5.73	10.28	12.45	4.1	3.68	4.32	2.14	5.17	7.15
	C2DSR	8.91	8.65	9.71	5.84	11.24	14.54	4.65	4.16	4.94	2.51	5.74	8.18
Our	DREAM	**9.33**	**10.05**	**11.25**	**6.08**	**13.75**	**17.45**	**4.82**	**5.19**	**6.15**	**2.74**	**7.52**	**10.51**
	Impr%	4.71	16.18	15.86	4.11	21.98	20.01	3.66	24.02	24.49	9.16	27.73	28.48

- **DR** is one variant of DREAM, only containing three separate transformers that encode the single- and cross-domain sequence into single- and cross-domain representation for next-item prediction
- **DREAM** follows our full training architecture with CE as a loss function without leveraging Contrastive Learning
- **-SG** stop-gradient operations are removed in DREAM.
- **+FL** replaces Cross-Entropy in DREAM by Focal Loss
- **+FL + UCL** added Unsupervised Contrastive Learning loss on the +FL model
- **Our** utilizes the full architecture of DREAM.

During inference, the trained model variants encode the single- and cross-domain sequence into sequence representations, including h_t^x, h_t^y, h_t^c for each variant. Taking domain X as an example, the estimated probability of the next item is estimated via the following three approaches(also holds for domain Y), and the top 10 items with the highest estimated probability will be selected for measurement.

$$P(h_t) = \begin{cases} softmax(h_t^x * E^{|X|}) & for\ Single \\ softmax(h_t^c * E^{|X|}) & for\ Cross \\ softmax(h_t^x * E^{|X|} + h_t^c * E^{|X|}) & for\ Combine \end{cases}$$

It is worth noting that τ is removed during inference since it is a constant value without impacting the item rankings.

6.2.1 Q2: Does DREAM generate the decoupled representation for single- and cross-domain recommendations? As a baseline reference, SAS-Single shows a stronger performance than SAS-Cross, demonstrating that the model is easier to capture the single-domain

Table 5: Ablation study on Movie-Book. Bold scores are the best performing, while underlined scores are the second best.

Data	Metric	Inference	SAS		C2DSR	DREAM					
	@10		Cross	Single		DR	DREAM	- SG	+ FL	+ FL + UCL	Our
Movie	MRR	Combine	-	-	4.94	6.06	6.31	5.95	6.67	8.04	**8.77**
		Cross	3.66	-	3.67	4.13	5.67	3.88	6.03	7.74	**7.96**
		Single	-	5.62	3.98	5.74	5.83	5.72	6	7.09	**7.54**
	NDCG	Combine	-	-	6.09	7.46	7.94	7.23	8.24	10.18	**11.11**
		Cross	4.67	-	5.12	5.16	7.09	5.01	7.5	10.07	**10.37**
		Single	-	6.72	4.70	7.01	7.17	6.93	7.37	8.84	**9.55**
	HR	Combine	-	-	9.91	12.11	13.32	11.45	13.45	17.25	**18.79**
		Cross	8.05	-	9.91	8.58	11.8	8.76	12.35	17.79	**18.28**
		Single	-	10.37	7.08	11.2	11.59	10.93	11.9	14.57	**16.17**
Book	MRR	Combine	-	-	2.74	2.72	2.92	2.73	3.01	3.87	**4.06**
		Cross	1.59	-	1.78	1.69	2.42	1.84	2.38	3.19	**3.45**
		Single	-	2.52	2.23	2.69	2.68	2.65	2.74	3.34	**3.41**
	NDCG	Combine	-	-	3.08	3.2	3.36	3.14	3.51	4.54	**4.83**
		Cross	1.99	-	2.25	2.1	2.88	2.26	2.87	3.90	**4.23**
		Single	-	2.78	2.51	3.07	3.05	2.97	3.15	3.90	**4.02**
	HR	Combine	-	-	4.22	4.8	4.81	4.5	5.17	6.74	**7.34**
		Cross	3.34	-	3.80	3.46	4.42	3.67	4.49	6.23	**6.79**
		Single	-	3.63	3.44	4.33	4.3	3.99	4.48	5.73	**6.04**

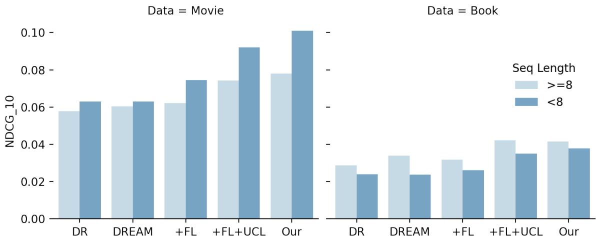
sequential relationship than the cross-domain sequential relationship. Despite that C2DSR shows a promising performance via the 'Combine' inference method, C2DSR shows a disappointing performance if we purely use a single-domain sequence to conduct inference. Therefore, C2DSR is able to generate a decent-quality cross-domain representation but is not able to generate a decoupled representation for each single-domain sequence. On the other side, three insights are found for all the variants of DREAM. (1)All the DREAM's variants not only improve overall performance but also improve the quality of single- and cross-domain representation since the performance of Single and Cross inference are consistently stronger than Single-SAS and Cross-SAS, achieving the goal of generating the decoupled representation for both single- and cross-domain (2) Combine method consistently outperforms other inference methods regardless of which DREAM's variants, showing a more robust and stronger performance, (3) By comparing DREAM and -SG, we found that without Stop Gradient operation, the performance of both single- and cross-domain is decreasing. The reason is that if the SG is not applied, the single-domain transformer does not only get gradients from the single domain but also gradients from the cross-domain. As we know, cross-domain sequence generally has more noise than single-domain increasing the difficulties of learning a decent representation.

6.2.2 Q3: Does Focal Loss help to enhance performance? By comparing the model performance DREAM and +FL, FL has stronger performances than CE, regardless of inference method. One more interesting finding is that the improvement of cross-domain representation is stronger than the one of single-domain. In the movie domain, FL boosted 6% MRR uplift (from 5.67 to 6.03) for cross-domain while boosting 3% NDCG uplift (from 5.83 to 6) for single-domain. Focal Loss puts more weight on misclassified samples and reduces the weight for well-classified samples, and cross-domain is generally easier to get misclassified than single-domain. Hence, FL generally has stronger performances here.

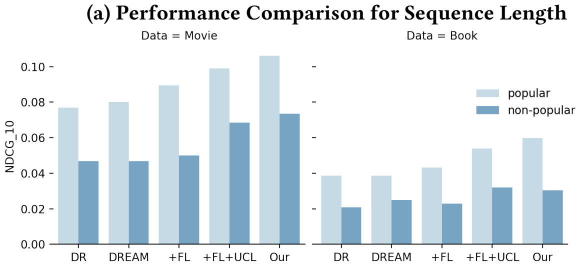
6.2.3 Q4: Does SCL outperform UCL in the CDSR setting? As Table 5 illustrates, +FL+UCL has a strong uplift compared with +FL, demonstrating that introducing Unsupervised Contrastive Learning can capture inter-sequential relationships to enhance model performances. Moreover, comparing our model performance with +FL+UCL, our model had around 10% uplifts for all the metrics, benefiting from the reduction of the false negative samples via semantic similarity.

6.2.4 Q5: Is SCL able to maximize the relevance between single- and cross-domain representations? To validate whether our model is able to maximize the relevance between single- and cross-domain representations. We calculate the single- and cross-domain representation of test sequences, and then we calculate the similarity between cross-domain and single-domain representation for the same user. Fig. 4c shows the distribution of the similarity, showing that the similarity is only 0.34 for DR. Introducing the Extraction and Attention Module(EAM) brings the similarity from 0.34 to 0.58 with a massive uplift. Focal Loss only increases the similarity to 0.59. Introducing UCL further increases the similarity to 0.69, while SCL gains another similarity uplift to 0.77. It shows that introducing EAM, UCL, and UCL is able to maximize the relevance between single- and cross-domain.

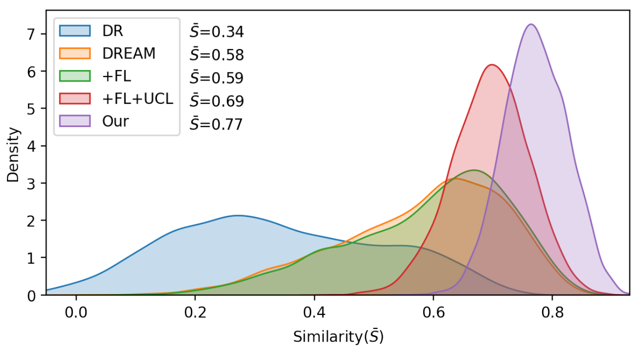
6.2.5 Q6: Does contrastive learning improve model performance? To understand how UCL and SCL enhance the model performances, two more analyses are conducted, including a performance at the popularity and sequence length. For popularity, the sequences are categorized into two groups of sequences (popular and non-popular



(a) Performance Comparison for Sequence Length



(b) Performance Comparison of target item popularity



(c) Distribution of Similarity between Cross- and Single-Domain

Figure 4: Performance Comparison

sequences) based on whether the last item is popular or not. For sequence length analysis, the sequences are categorized into two groups of sequences, including the sequence with more than 8 items and the sequence with less than 8 items.

Fig. 4a illustrates that both UCL and SCL have significant uplifts for long and short sequences. Another interesting finding is that both UCL and SCL have stronger improvements for short sequences than long sequences. One potential reason is that short sequences naturally have less information than long sequences, leading to lower performances. Introducing the Supervised Contrastive Learning framework allows short sequences to learn from the long sequences if they share the same target items/ interests. Fig. 4b illustrates that both UCL and SCL have significant uplifts for both popular and non-popular sequences. Moreover, SCL gains better performance than UCL for a popular product, but the uplift at non-popular products is not significant. One possible reason is that popular products are more likely to have the same target in the batch, further refining sequence representation.

7 CONCLUSION AND FUTURE DIRECTION

A novel model DREAM is proposed for a cross-domain sequential recommendation(CDSR). Particularly, DREAM includes a DREAM framework to simultaneously model the single- and cross-domain user preferences as well as a Supervised Contrastive Learning framework to capture and learn the inter-sequence relationship. Extensive experiments are conducted to validate the effectiveness of our model framework, reaching a new state-of-the-art performance. Moreover, the effectiveness of our model components and Supervised Contrastive Learning are analyzed in detail. One future extension of our algorithm is extending our model from two domains to multiple domains as well as seeking a simpler model architecture for CDSR.

REFERENCES

[1] 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 and Knowledge Management* (Atlanta, GA, USA) *(CIKM 22)*. Association for Computing Machinery, New York, NY, USA, 138–147. https://doi.org/10.1145/3511808.3557262

[2] Jiangxia Cao, Jiawei Sheng, Xin Cong, Tingwen Liu, and Bin Wang. 2022. Cross-Domain Recommendation to Cold-Start Users via Variational Information Bottleneck. arXiv:2203.16863 [cs.IR]

[3] Fengwen Chen, Shirui Pan, Jing Jiang, Huan Huo, and Guodong Long. 2019. DAGCN: Dual Attention Graph Convolutional Networks. https://doi.org/10.48550/ARXIV.1904.02278

[4] Huiyuan Chen, Yusan Lin, Menghai Pan, Lan Wang, Chin-Chia Michael Yeh, Xiaoting Li, Yan Zheng, Fei Wang, and Hao Yang. 2022. Denoising Self-Attentive Sequential Recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems* (Seattle, WA, USA) *(RecSys '22)*. Association for Computing Machinery, New York, NY, USA, 92–101. https://doi.org/10.1145/3523227.3546788

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. https://doi.org/10.48550/ARXIV.2002.05709

[6] Jing Du, Zesheng Ye, Bin Guo, Zhiwen Yu, and Lina Yao. 2023. IDNP: Interest Dynamics Modeling Using Generative Neural Processes for Sequential Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining* (Singapore, Singapore) *(WSDM '23)*. Association for Computing Machinery, New York, NY, USA, 481–489. https://doi.org/10.1145/3539597.3570373

[7] Ruining He and Julian McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. https://doi.org/10.48550/ARXIV.1609.09152

[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. https://doi.org/10.48550/ARXIV.1708.05031

[9] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based Recommendations with Recurrent Neural Networks. https://doi.org/10.48550/ARXIV.1511.06939

[10] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. https://doi.org/10.48550/ARXIV.1808.06670

[11] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 667–676. https://doi.org/10.1145/3269206.3271684

[12] Chengkai Huang, Shoujin Wang, Xianzhi Wang, and Lina Yao. 2023. Dual Contrastive Transformer for Hierarchical Preference Modeling in Sequential Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Tai Pei, Taiwan) *(SIGIR '23)*. Association for Computing Machinery, Tai Pei, Taiwan, 2825–2837. https://doi.org/10.1145/3539597.3570373

[13] Chengkai Huang, Shoujin Wang, Xianzhi Wang, and Lina Yao. 2023. Modeling Temporal Positive and Negative Excitation for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2023* (Austin, TX, USA) *(WWW '23)*. Association for Computing Machinery, New York, NY, USA, 1252–1263. https://doi.org/10.1145/3543507.3583463

[14] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[15] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. https://doi.org/10.48550/ARXIV.1808.09781

[16] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

[17] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *KDD 2020*. https://dl.acm.org/doi/10.1145/3394486.3403226

[18] Chenglin Li, Mingjun Zhao, Huanming Zhang, Chenyun Yu, Lei Cheng, Guoqiang Shu, BeiBei Kong, and Di Niu. 2022. RecGURU: Adversarial Learning of Generalized User Representations for Cross-Domain Recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. ACM. https://doi.org/10.1145/3488560.3498388

[19] Pan Li and Alexander Tuzhilin. 2019. DDTCDR: Deep Dual Transfer Cross-Domain Recommendation. https://doi.org/10.48550/ARXIV.1910.05189

[20] Yang Li, Tong Chen, Peng-Fei Zhang, and Hongzhi Yin. 2021. Lightweight Self-Attentive Sequential Recommendation. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. ACM. https://doi.org/10.1145/3459637.3482448

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal Loss for Dense Object Detection. https://doi.org/10.48550/ARXIV.1708.02002

[22] 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 and Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 885–894. https://doi.org/10.1145/3340531.3412012

[23] Weiming Liu, Xiaolin Zheng, Mengling Hu, and Chaochao Chen. 2022. Exploiting Variational Domain-Invariant User Embedding for Partially Overlapped Cross Domain Recommendation. arXiv:2205.06440 [cs.IR]

[24] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S. Yu, Julian McAuley, and Caiming Xiong. 2021. Contrastive Self-supervised Sequential Recommendation with Robust Augmentation. https://doi.org/10.48550/ARXIV.2108.06479

[25] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S. Yu. 2021. Augmenting Sequential Recommendation with Pseudo-Prior Items via Reversely Pre-training Transformer. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. https://doi.org/10.1145/3404835.3463036

[26] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. pi-Net: A Parallel Information-Sharing Network for Shared-Account Cross-Domain Sequential Recommendations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 685–694. https://doi.org/10.1145/3331184.3331200

[27] Wentao Ouyang, Xiuwu Zhang, Lei Zhao, Jinmei Luo, Yu Zhang, Heng Zou, Zhaojie Liu, and Yanlong Du. 2020. MiNet: Mixed Interest Network for Cross-Domain Click-Through Rate Prediction. https://doi.org/10.48550/ARXIV.2008.02974

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs.LG]

[29] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. ACM. https://doi.org/10.1145/3488560.3498433

[30] Aghiles Salah, Thanh Binh Tran, and Hady Lauw. 2021. Towards Source-Aligned Variational Models for Cross-Domain Recommendation. In *Proceedings of the 15th ACM Conference on Recommender Systems* (Amsterdam, Netherlands) *(RecSys '21)*. Association for Computing Machinery, New York, NY, USA, 176–186. https://doi.org/10.1145/3460231.3474265

[31] 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. https://doi.org/10.48550/ARXIV.1904.06690

[32] Wenchao Sun, Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. Parallel Split-Join Networks for Shared-account Cross-domain Sequential Recommendations. https://doi.org/10.48550/ARXIV.1910.02448

[33] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. https://doi.org/10.48550/ARXIV.1809.07426

[34] Ellen M Voorhees et al. 1999. The trec-8 question answering track report.. In *Trec*, Vol. 99. 77–82.

[35] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (jul 2019), 346–353. https://doi.org/10.1609/aaai.v33i01.3301346

[36] Ruobing Xie, Qi Liu, Liangdong Wang, Shukai Liu, Bo Zhang, and Leyu Lin. 2021. Contrastive Cross-domain Recommendation in Matching. (2021). https://doi.org/10.48550/ARXIV.2112.00999

[37] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive Learning for Sequential Recommendation. https://doi.org/10.48550/ARXIV.2010.14395

[38] Feng Yuan, Lina Yao, and Boualem Benatallah. 2019. DARec: Deep Domain Adaptation for Cross-Domain Recommendation via Transferring Rating Patterns. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (Macao, China) *(IJCAI'19)*. AAAI Press, 4227–4233.

[39] Shuai Zhang, Yi Tay, Lina Yao, and Aixin Sun. 2018. Next Item Recommendation with Self-Attention. https://doi.org/10.48550/ARXIV.1808.06414

[40] 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 and Knowledge Management*. ACM. https://doi.org/10.1145/3340531.3411954

[41] Yongchun Zhu, Kaikai Ge, Fuzhen Zhuang, Ruobing Xie, Dongbo Xi, Xu Zhang, Leyu Lin, and Qing He. 2021. Transfer-Meta Framework for Cross-domain Recommendation to Cold-Start Users. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. https://doi.org/10.1145/3404835.3463010

[42] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2021. Personalized Transfer of User Preferences for Cross-domain Recommendation. https://doi.org/10.48550/ARXIV.2110.11154