

One for All, All for One: Learning and Transferring User Embeddings for Cross-Domain Recommendation

Chenglin Li
University of Alberta
Edmonton, AB, Canada

Yuanzhen Xie
Platform and Content Group, Tencent
Shenzhen, China

Chenyun Yu
Sun Yat-sen University
Shenzhen, China

Bo Hu
Platform and Content Group, Tencent
Shenzhen, China

Zang Li
Platform and Content Group, Tencent
Shenzhen, China

Guoqiang Shu
Platform and Content Group, Tencent
Shenzhen, China

Xiaohu Qie
Platform and Content Group, Tencent
Shenzhen, China

Di Niu
University of Alberta
Edmonton, AB, Canada

ABSTRACT

Cross-domain recommendation is an important method to improve recommender system performance, especially when observations in target domains are sparse. However, most existing techniques focus on single-target or dual-target cross-domain recommendation (CDR) and are hard to be generalized to CDR with multiple target domains. In addition, the negative transfer problem is prevalent in CDR, where the recommendation performance in a target domain may not always be enhanced by knowledge learned from a source domain, especially when the source domain has sparse data. In this study, we propose CAT-ART, a multi-target CDR method that learns to improve recommendations in all participating domains through representation learning and embedding transfer. Our method consists of two parts: a self-supervised Contrastive AuToencoder (CAT) framework to generate global user embeddings based on information from all participating domains, and an Attention-based Representation Transfer (ART) framework which transfers domain-specific user embeddings from other domains to assist with target domain recommendation. CAT-ART boosts the recommendation performance in any target domain through the combined use of the learned global user representation and knowledge transferred from other domains, in addition to the original user embedding in the target domain. We conducted extensive experiments on a collected real-world CDR dataset spanning 5 domains and involving a million users. Experimental results demonstrate the superiority of the proposed method over a range of prior arts. We further conducted ablation studies to verify the effectiveness of the proposed components. Our collected dataset will be open-sourced to facilitate future research in the field of multi-domain recommender systems and user modeling.

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 ACM 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.

WSDM '23, February 27-March 3, 2023, Singapore, Singapore

© 2023 Association for Computing Machinery.

ACM ISBN 978-1-4503-9407-9/23/02...\$15.00

<https://doi.org/10.1145/3539597.3570379>

CCS CONCEPTS

• Information systems → Recommender systems; • Computing methodologies → Learning latent representations.

KEYWORDS

Cross-Domain Recommendation; User Modeling; Representation Learning; Contrastive Learning; Autoencoder

ACM Reference Format:

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 (WSDM '23)*, February 27-March 3, 2023, Singapore, Singapore. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3539597.3570379>

1 INTRODUCTION

Data sparsity is a long-standing issue hampering traditional single-domain recommendation systems. Cross-domain Recommendation (CDR) has proven to be a promising approach to alleviate the data sparsity issue, aiming to leverage auxiliary information learned from other domains to improve recommendation in the target domain. However, most prior research on cross-domain recommendation focuses on either the single-target CDR (STCDR) or dual-target CDR (DTCDR) [38] scenarios, with only two domains involved. STCDR aims to improve the recommendation accuracy in the target domain, while DTCDR tries to improve the performance in both domains simultaneously. Multi-target CDR (MTCDR) is a more general and challenging problem, which aims to improve the recommendation performance in multiple participating domains concurrently. Since previous methods solve the STCDR and DTCDR by modeling a pair-wise domain-domain relationship, intuitively speaking, extending them to the MTCDR scenario with n domains involves handling at least $\binom{n}{2}$ pairs of relations, which is not practical when the number of domains is large.

Relatively fewer efforts have been put onto MTCDR. Current state-of-the-art solutions usually generate a shared cross-domain user representation for each user, which, combined with domain-specific features, is used to boost recommendations in any given domains [3, 29]. HeroGRAPH [3] collects user behavior in all domains to build a heterogeneous graph. It then applies the graph

convolutional networks (GCN) [7] to generate the cross-domain user and item embeddings, which are directly transferred to target domains to boost recommendation. However, most recommender systems are built on users' sensitive data, e.g., check-in data, browsing records, which is held by different domains and cannot be shared directly to form a large heterogeneous graph. MPF [29] learns a global embedding for each user, which is directly shared among all domains and optimized by the recommendation losses in all participating domains via multi-task learning. However, the cross-domain user representation, extracted directly on the collected data from all domains, may be severely biased by the domains with richer data and may fail to model the user preferences in sparse domains. The biased global representation of a user may negatively affect recommendation performance when transferred to a target domain. This unbalanced data problem also exists in HeroGRAPH [3].

We observe that two types of user embeddings can both be helpful in cross-domain recommendation, including 1) the global user embedding, which represents the overall domain-invariant characteristics of a user, and 2) domain-specific user embeddings, which model the user behavior in various individual domains. Two questions naturally arise concerning the interaction of these embeddings. First, can we generate a global user representation only based on the user's embeddings obtained from individual domains? If this global user embedding is representative enough and not biased toward a single domain, it can be directly used to improve recommendation in all domains. We refer to the first goal by "One for All", as in One global user embedding for recommendations in All domains. To ensure better data isolation, ideally, the global user embedding should be synthesized only based on domain-specific user embeddings without accessing raw behavior data in each domain.

On the other hand, what is missing in prior work on MTCDR [3, 29] is the transfer of domain-specific user embeddings to assist with recommendation in the target domain. However, directly transferring these features may cause performance degradation in the target domain due to various reasons, e.g., low-quality embeddings transferred from irrelevant domains. This phenomenon is often referred to as negative transfer [33]. The second question we ask is—can we also transfer domain-specific user embeddings in one domain to help improve prediction in another domain while avoiding negative transfer? We refer to the second goal by "All for One", as in All domain-specific user embeddings for helping the recommendation in One domain.

To address the aforementioned challenges, we propose CAT-ART, a novel multi-target CDR framework, which starts from using traditional matrix factorization to pretrain domain-specific user embeddings in each domain. To achieve the concept of One for All, we propose a Contrastive AuToencoder (CAT) to learn a global user embedding solely based on the pretrained domain-specific user embeddings from all the domains, without directly accessing raw behavior data in each domain. To attain the goal of All for One, we build an Attention-based Representation Transfer (ART) unit in each target domain, which transfers and utilizes the pretrained domain-specific embeddings to boost its recommendation performance while minimizing the impact of negative transfer. Our contributions can be summarized as follows:

We introduce a contrastive autoencoder (CAT), which learns a general global embedding for each user by reconstructing the

concatenated sequence of domain-specific user embeddings. To further benefit from self-supervised representation learning and extract robust representations from domain-specific embeddings, we randomly mask some domain in the input sequence to the autoencoder and learn to reconstruct the original sequence, while a contrastive self-supervised loss is used to ensure the masked and unmasked domain-specific user embedding sequences can map to similar global user embeddings for the same user.

We propose Attention-based Representation Transfer (ART), which judiciously adapts the domain-specific user embeddings from other domains to the target domain according to an attention mechanism. ART then combines the target domain user embedding, the global user embedding, as well as the adapted domain-specific user embeddings to jointly improve recommendation in the target domain.

To evaluate our method in real scenarios, we have collected a large dataset involving over a million users spanning 5 domains, including App installation (App-install), Recent App usage (App-use), article viewing, short-video viewing, and long-video viewing. Each of these domains has its own user behavior history and independently pretrained user embeddings. We conduct extensive experiments on the collected data and demonstrate the superiority of the proposed CAT-ART method by comparing with a range of state-of-the-art MTCDR baselines. Experimental results suggest that CAT-ART significantly outperforms all baselines in most of the domains, e.g., App-install, App-use, article, long-video, on several evaluation metrics. Moreover, we show that while other state-of-the-art baselines are severely impacted by negative transfer, CAT-ART can effectively avoid the negative transfer issue.

2 RELATED WORK

Single-Target CDR. Previous single-target CDR (STCDR) works can be classified into content-based [4, 11, 31] and embedding-based approaches [12, 21, 28, 34]. In the former, Kanagawa et al. [11] propose a content-based domain adaptation model and a domain separation network for cross-domain recommendations. In the latter, EMCDCR [21] solves the cold-start problem in the target domain by learning a mapping function between the user embeddings of the source and the target domain. CDIE-C [28] enhances item embedding learning by cross-domain co-clustering for the sequential recommendation.

Dual-Target CDR. Given two domains, DTCDR is to improve the recommendation accuracy in both domains at the same time by leveraging their observed information [15–17, 19, 36, 37, 40]. Zhu et al. [36] first proposed the DTCDR problem and a DTCDR framework that learns more representative embeddings of users and items based on multi-sources. Liu et al. [19] combine the embeddings of common users based on hyper-parameters and data sparsity degrees of users. GA-DTCDR Zhu et al. employs graph embedding to generate more informative embeddings of users and items, and employs element-wise attention to combine the embeddings of common users/items across domains. The deep dual transfer CDR (DDTCDR) [16] considers the bi-directional latent relations between users and items and applies a latent orthogonal mapping to extract user preferences. CATN [34] learns aspect correlations across domains with an attention mechanism. Some work

focuses on extracting domain-invariant or domain-independent user attributes for CDR [18, 24, 41]. ACDN [18] models individual's propensity from the aesthetic perspective and captures users' domain-independent aesthetic preference for CDR.

Multi-Target CDR. The MTCDR methods [3, 13, 14, 29, 30, 32] have emerged to improve the recommendation performance of multiple domains simultaneously. Kim et al. [13] adopt recurrent neural networks (RNNs) to model the sequential behavior of users in multiple domains simultaneously. [30, 32] achieve knowledge transfer through parameter sharing across multiple domains. However, they all focus on sequential recommendation. Recent works try to extract domain-specific and cross-domain features simultaneously, e.g., MSDCR [35], HeroGRAPH [3], and MPF [29]. Specifically, HeroGRAPH [3] constructs heterogeneous graph from interactions between users and items from all domains and develops a graph embedding algorithm to extract common features for MTCDR. MPF [29] captures both the cross-site and site-specific preferences for multi-site video recommendations. GA-MTCDR [39], extended from GA-DTCDR [37], employs element-wise attention to combine embeddings of overlapped users/items from all domains. However, it requires side information for graph construction in each domain. However, most of the previous MTCDR methods ignore the data isolation constraint between domains in practice, and none of them has considered the negative transfer problem. In this study, we try to solve the MTCDR problem in a more realistic scenario where user and item data are held by each individual domain and cannot be shared across domains. Furthermore, our framework is also designed to avoid the negative transfer problem in MTCDR.

3 METHODOLOGY

We focus on the MTCDR problem with a global user set U , and item sets $\{V_1, \dots, V_n\}$ in $n \geq 3$ domains. User-item interactions in domain $d \in [1, \dots, n]$ are represented by a matrix R^d with the shape of $|U| \times |V_d|$ where $|U|$ is the number of users and $|V_d|$ denotes the number of items in domain d . Under the implicit feedback setting, all the elements in the matrix R^d are with the value of either 1 or 0, which indicates whether there is an interaction between a given user-item pair, i.e., for user i and item j in domain d , $r_{ij}^d \in [0, 1], \forall i \in |U|, \forall j \in |V_d|$. We further consider the scenario of data isolation in practical applications, that is, the interactive information between users and items in a specific domain is not observable by other domains. Our goal is to improve the recommendation accuracy in all n domains simultaneously based on the interaction matrices.

3.1 Architecture Overview of CAT-ART

We set two objectives when dealing with the multi-target CDR problem. That is, 1) One for All: extracting global user representation that is used for recommendations in all domains. And 2) All for One: transferring domain-specific embeddings from all available domains to assist the recommendation in a target domain without negative transfer. To achieve these two objectives and avoid the direct use of raw data across domains, we propose the CAT-ART model where the Contrastive Autoencoder (CAT) module and the Attention-based Representation Transfer (ART) unit are designed for the above two objectives, respectively.

Figure 1 provides an overview of the CAT-ART model. First, the domain-specific user embeddings are pretrained within each domain independently using BPRMF [23]. Then, the CAT module takes the domain-specific embeddings collected from all domains as input and generates global user representation. To create unbiased cross-domain user embeddings, we combine the reconstruction loss and contrastive self-supervised loss in model training, so that the CAT module is capable of extracting informative global user representation. Finally, the ART module transfers the domain-specific user embeddings from all the other domains to boost the recommendation performance in a single domain, e.g., domain 2 in Figure 1. By incorporating attention mechanisms to the ART module, the contribution of each domain can be judiciously adjusted according to their relevance so as to address the negative transfer issue.

3.2 Domain-specific User Embedding

Shown in Figure 1, in the Single-Domain User Modeling unit, we adopt the widely used Matrix Factorization (MF) model [20] with the Bayes Personalized Ranking (BPR) [23] loss to get the user and item embeddings in each domain. Formally, to factorize the interaction matrix R^d in domain d , we create two trainable embedding matrices $\mathbb{I}^d \subset \mathbb{R}^{(|V_d|+1) \times m}$ and $\mathbb{E}^d \subset \mathbb{R}^{(|U|+1) \times m}$ to represent item and user embeddings in domain d , respectively. Where m represents the number of dimensions in the latent space. For simplicity, we set the embedding dimensions of users and items in all domains to be m . In domain d , given a user u_i with embedding $\mathbf{e}_i^d \in \mathbb{E}^d$ and an item $v_j \in V^d$ with embedding $\mathbf{I}_j^d \in \mathbb{I}^d$, the preference score of the user to the item is computed as the inner product between their embeddings, $r_{ij}^d = \mathbf{e}_i^d \mathbf{I}_j^d$. Note that, we use bold font to denote vector variables. Then, the BPR loss in domain d is formulated as:

$$\mathcal{L}_{bpr}^d = - \sum_{i \in U} \sum_{j \in p_i^d} \sum_{l \notin p_i^d} \log \sigma(r_{ij}^d - r_{il}^d), \quad (1)$$

where p_i^d is the set of items that user i has interacted with in domain d , and $\sigma(\cdot)$ represents the sigmoid function.

By minimizing the BPR loss, we obtain domain-specific user embeddings suitable for recommendation in each domain. Instead of sharing raw user data, we only share the pretrained domain-specific user embeddings across domains to enable knowledge sharing across domains under the raw data isolation constraint.

3.3 Contrastive Autoencoder

The pretrained domain-specific user embeddings are collected and fed into the CAT module to get global user representations. To do this, we adopt an autoencoder framework which takes the domain-specific embeddings of a user as input, and generates latent user representation. Specifically, we first stitch all domain-specific user embeddings, which is a set of real-valued dense vectors, into a large one-dimensional vector in a predefined order, e.g., from domain 1 to domain n . Then, an encoder is used to extract the latent user representation which is further fed into a decoder to reconstruct the input domain-specific embeddings. We use Multi-Layer Perception (MLP) [5] to build both the encoder and decoder modules. Formally, for a given user u_i with its pretrained domain-specific embeddings $\{\mathbf{e}_i^1, \mathbf{e}_i^2, \dots, \mathbf{e}_i^n\}$, we apply the following procedures to get its latent

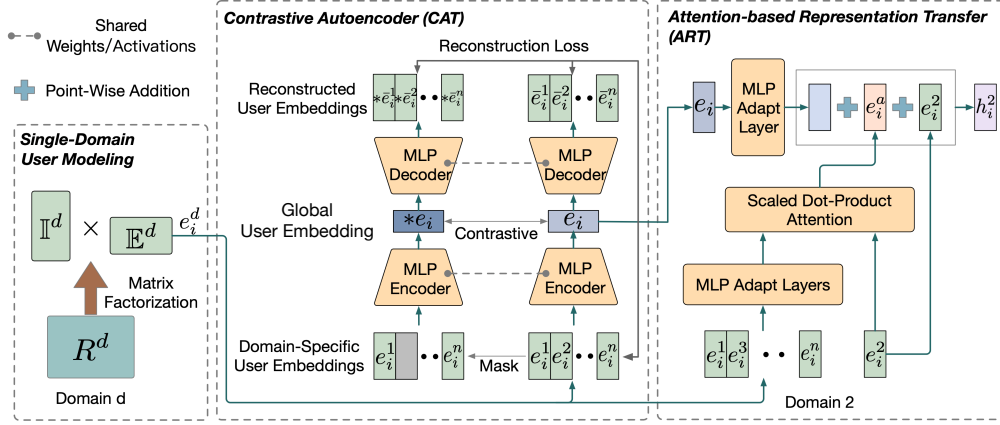


Figure 1: The architecture of the CAT-ART model. The CAT module takes domain-specific user embeddings as input and generates global user representation in a self-supervised manner. Then, the global user embedding e_i and the domain-specific embeddings from all the other domains are transferred to a target domain, e.g., domain 2, for boosted recommendations.

representation:

$$\begin{aligned} \mathbf{e}_i &= \text{MLP}_{\text{enc}}(\mathbf{e}_i^1 \frown \mathbf{e}_i^2 \frown \dots \frown \mathbf{e}_i^n) \\ \bar{\mathbf{e}}_i^1 \frown \bar{\mathbf{e}}_i^2 \frown \dots \frown \bar{\mathbf{e}}_i^n &= \text{MLP}_{\text{dec}}(\mathbf{e}_i), \end{aligned} \quad (2)$$

where \mathbf{e}_i is the latent representation of user u_i , and $\mathbf{e}_i^1 \frown \mathbf{e}_i^2$ denotes the long one-dimensional vector after concatenating the vectors \mathbf{e}_i^1 and \mathbf{e}_i^2 . Here $\bar{\mathbf{e}}_i^d$ represents the reconstruction of user embedding \mathbf{e}_i^d in domain d . Note that the latent vector \mathbf{e}_i has the same size as domain-specific embeddings in individual domains, i.e., $|\mathbf{e}_i| = |\mathbf{e}_i^d| = m, \forall d \in [1, n]$.

By optimizing the mean square reconstruction error (3) between the input and the reconstructed embeddings, the encoder can learn the most important global attributes to reconstruct the domain-specific user embeddings in each domain.

$$\mathcal{L}_{\text{rec}} = \frac{1}{|U|} \sum_{i \in U} \sum_{d=1}^n \|\mathbf{e}_i^d - \bar{\mathbf{e}}_i^d\|_2. \quad (3)$$

On the other hand, the effectiveness of the pretrained domain-specific user embeddings, which are the input of the autoencoder, is highly affected by the data quality and sparsity in each domain. For example, under-trained user embeddings from a sparse domain may introduce noise to the input of the autoencoder. Furthermore, the autoencoder may be biased towards domains with a higher quality of user embedding, as it is easier to reconstruct a well-trained embedding than an under-trained embedding with noise.

Therefore, we adopt the contrastive self-supervised learning [2] to further train the autoencoder for a more general and robust latent user representation that does not bias to any specific domain. The core idea of contrastive self-supervised learning is to make the representation of an input sample agree with that of an augmented sample, e.g., obtained by applying Gaussian noise or Cutout [2].

In our problem, contrastive self-supervised learning is integrated into the autoencoder framework to extract the global representations of users from their domain-specific embeddings. We adopt an *Mask* operation to generate the “augmentations” for an input, e.g., $\mathbf{e}_i^1 \frown \mathbf{e}_i^2 \frown \dots \frown \mathbf{e}_i^n$. Specifically, with *Mask*, we generate “augmented”

inputs by removing the domain-specific user embeddings from several randomly selected domains. Formally, the following procedure is applied:

$$\begin{aligned} \mathbf{e}_i^1 \frown \mathbf{e}_m \frown \dots \frown \mathbf{e}_i^n &= \text{Mask}(\mathbf{e}_i^1 \frown \mathbf{e}_i^2 \frown \dots \frown \mathbf{e}_i^n) \\ *e_i &= \text{MLP}_{\text{enc}}(\mathbf{e}_i^1 \frown \mathbf{e}_m \frown \dots \frown \mathbf{e}_i^n) \end{aligned} \quad (4)$$

where \mathbf{e}_m is a trainable vector used to replace the domain-specific embeddings of masked domains, e.g., domain 2 in Figure 1. After random masking and padding, the “augmented” sample, i.e., $\mathbf{e}_i^1 \frown \mathbf{e}_m \frown \dots \frown \mathbf{e}_i^n$, is fed into the encoder MLP_{enc} to get the latent representation $*e_i$. Then, a contrastive loss function is defined for the contrastive prediction task. Given a set of latent embeddings $\{\mathbf{e}_1, \dots, \mathbf{e}_k, *e_1, \dots, *e_k\}$ where $(\mathbf{e}_i, *e_i)$ forms a positive pair of representations, the contrastive prediction is to identify $*e_i$ in $\{\mathbf{e}_1, \dots, \mathbf{e}_k, *e_k\}$ for a given \mathbf{e}_i , and vice versa, to identify \mathbf{e}_i in $\{\mathbf{e}_1, \dots, \mathbf{e}_k, *e_k\}$ for a given $*e_i$.

We randomly sample a minibatch of N users and define the contrastive autoencoder task on the pairs of “augmented” and original user embeddings derived from the minibatch, resulting in $2N$ representations. Let $\phi(\mathbf{e}_i, *e_i) = \frac{\mathbf{e}_i \cdot *e_i}{\|\mathbf{e}_i\| \|*e_i\|}$ denotes the cosine similarity between \mathbf{e}_i and $*e_i$. Then, the contrastive loss of user i in the minibatch is computed as:

$$l_i = -\log \frac{\exp(\phi(\mathbf{e}_i, *e_i)/\tau)}{\sum_{k=1}^N \exp(\phi(\mathbf{e}_i, *e_k)/\tau)} - \log \frac{\exp(\phi(*e_i, \mathbf{e}_i)/\tau)}{\sum_{k=1}^N \exp(\phi(*e_i, \mathbf{e}_k)/\tau)}, \quad (5)$$

where τ is a temperature parameter. The first term defines the contrastive prediction loss when identifying $*e_i$ given \mathbf{e}_i and the second term is the loss for identifying \mathbf{e}_i given $*e_i$. By minimizing the contrastive loss, the encoder is trained to put the embeddings of a user and its “augmentations”, i.e., \mathbf{e}_i and $*e_i$, close to each other in the latent embedding space. By doing this, we can extract a more general latent user representation that is robust to noisy input introduced by low-quality domain-specific user embeddings from a sparse domain.

Apart from the contrastive loss, we further put the $*e_i$ into the decoder MLP_{dec} to reconstruct the original input. That is, instead of minimizing the reconstruction error between the “augmentation”, i.e., $e_i^1 \frown e_m \frown \dots \frown e_i^n$, and its reconstruction, i.e., $*e_i^1 \frown *e_i^2 \frown \dots \frown *e_i^n$, we optimize the reconstruction loss between the original domain-specific embeddings and the reconstructed embeddings from $\text{MLP}_{\text{dec}}(*e_i)$, that is,

$$\begin{aligned} *e_i^1 \frown *e_i^2 \frown \dots \frown *e_i^n &= \text{MLP}_{\text{dec}}(*e_i), \\ \mathcal{L}_{\text{rec}}^* &= \frac{1}{|U|} \sum_{i \in U} \sum_{d=1}^n \|e_i^d - *e_i^d\|_2. \end{aligned} \quad (6)$$

By minimizing $\mathcal{L}_{\text{rec}}^*$, the encoder can extract latent representation which is informative enough to reconstruct the masked domains even when the embeddings of masked domains are missing. This further encourages the encoder to extract unbiased global representations of users. Finally, the loss function to train the CAT module can be summarized as:

$$\mathcal{L}_{\text{cat}} = \alpha_1 \mathcal{L}_{\text{rec}} + \alpha_2 \mathcal{L}_{\text{rec}}^* + (1 - \alpha_1 - \alpha_2) \sum_{i=1}^{|U|} l_i, \quad (7)$$

where α_1 and α_2 are hyper-parameters to control the weights of each component. The third term represents the sum of the contrastive loss over all users.

The extracted global representation represents the general preferences and overall characteristics of users, that are beneficial to the recommendation task in each domain. In addition, because the global user representation is not biased to any domain, it can be directly transferred to any target domain without worrying about the negative transfer issue.

3.4 Attention-based Representation Transfer

The domain-specific embeddings represent the users’ preferences in different domains, which are useful features to boost recommendations in a specific domain. However, the direct transfer of domain-specific user embeddings is prone to negative transfers in the target domain. First, if two domains are unrelated, the transferred embeddings may have little or even negative impact on the target domain. Second, if a domain-specific user embedding is under-trained in its original domain due to data sparsity or insufficient data quality, directly transferring such an embedding may introduce noise to the target domain, leading to performance degradation.

Therefore, we build an attention-based representation transfer (ART) unit in each domain, to integrate the domain-specific embeddings from other domains for better recommendations in the current domain. Specifically, we first build an MLP-based domain adaptation layer for domain-specific embeddings from each domain. Then, we adopt the scaled-dot produce attention [26] which uses the user embedding in the target domain as query and attends to the domain-specific embeddings from other domains. Formally, in domain d , we have

$$\begin{aligned} Q &= e_i^d \\ K &= V = \text{MLP}_{\text{adapt}}(\{e_i^k\}, k \neq d) \\ e_i^a &= \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{m}}\right)V, \end{aligned} \quad (8)$$

Table 1: Statistics of the Collected Dataset with 5 Domains.

Domain	#Users	#Items	#Interactions	Density(%)
App-Ins		100,000	101,981,793	0.874
APP-Use		100,000	18,156,535	0.155
Articles	1,166,552	50,000	102,832,656	1.763
Video-S		50,000	74,911,020	1.284
Video-L		50,000	11,412,988	0.196

where m is the dimensionality of the user embedding e_i^d in domain d . With attention, the ART module can assign more weights to the embeddings from the most related domains and reduce the influence of noisy embeddings from sparse or unrelated domains, and thus can effectively alleviate negative transfers in MTCDR.

Final User Embedding. We build a MLP-based domain adaptation module, i.e., $\text{MLP}_{\text{ind}}(\cdot)$, which adapts the global embedding to current domain for recommendation. Then, we combine features from the global and domain-specific user embeddings with point-wise addition. For example, in domain d , we have

$$h_i^d = e_i^d + \text{MLP}_{\text{ind}}(e_i) + e_i^a, \quad (9)$$

where the first term is the user embedding in current domain, the second term represents the information transferred from the global user representation, and the last term denotes features from all the other domains. With user embedding h_i^d , the preference score of the user i to an item j in domain d is recalculated as: $r_{ij}^d = h_i^d I_j^d$.

3.5 Model Training

We adopt the following three steps to train the CAT-ART model. First, we apply the BPRMF model in individual domains to get the pretrained domain-specific user and item embeddings based on BPR loss (1). Then, we train the CAT module in a self-supervised manner with both the reconstruction loss and contrastive loss (7). Finally, we fix the domain-specific user embeddings and the CAT module, and optimize the parameters of the $\text{MLP}_{\text{ind}}(\cdot)$ and ART module according to the BPR loss in individual domains.

4 EXPERIMENTS

We conduct extensive experiments on a collected dataset and compare our approach with state-of-the-art MTCDR methods.

4.1 Datasets

We collected user logs from five domains through multiple real apps owned by Tencent. Specifically, they are Application installation preferences (“APP-Ins”), Application usage preferences (“APP-Use”), “Articles”, Short Video (“Video-S”), and Long Video (“Video-L”). Among them, the “APP-Ins” and “APP-Use” data is collected from January 2021 to August 20, 2021, and the data of “Articles”, “Video-S”, and “Video-L” are collected from June 2021 to August 2021. We use the tags that best describe the visited items to avoid a tremendous set of items. TF-IDF [22] is applied to select a fixed number of the most informative tags in each domain, resulting in 100,000 tags in the “APP-Ins” and “APP-Use” domains and 50,000 tags in the rest of domains. We filter out users with less than 5 visited tags. Furthermore, the number of tags a user has visited in each domain

is truncated to a fixed number, i.e., a user can have a maximum of 100 tags in the “Video-S” domain, and a maximum of 300 tags in the other domains. All the above data processing procedures are provided by the data owner (Tencent), and have proven to be effective in real-world tasks such as gender and age prediction. Table 1 provides detailed statistics of the collected dataset with 5 domains.

4.2 Experimental Setup

Our experiments aim at answering the following questions:

- **RQ1.** How does CAT-ART perform vs. state-of-the-art baselines in the MTCDR task?
- **RQ2.** Does our model handle the problem of negative transfer?
- **RQ3.** How does the sub-modules help the model succeed in solving the MTCDR problem?

To train and evaluate a model, we randomly split the interactions of a user in each domain into three parts for: training (70%), validation (10%), and testing (20%). The evaluation metrics are Precision@K, Recall@K, and NDCG@K [10] that are computed by the all-ranking protocol where all items/tags are ranked. We repeat all experiments three times and give the average and standard deviation of all metrics.

Compared Methods. We compare our model with single-domain recommendation methods(SMF), STCDR methods(CMF), and MTCDR methods(MPF, GA-MTCDR, HeroGRAPH).

- **SMF:** It factorizes the user-tag interaction matrix of each domain separately based on the BPR loss [23].
- **CMF** [25]: It collects interactions from all domains to form a single matrix which is further factorized for recommendation.
- **HeroGRAPH-L:** HeroGRAPH [3] learns the cross-domain and domain-specific representations. Since the HeroGRAPH does not open-source its code, we implement this method based on Lightgcn [9].
- **MPF** [29]: It captures the cross-domain preference with user’s behavior in all domain, and combine it with the user embedding in the target domain for recommendation.
- **GA-MTCDR** [39]: It adopts the node2vec [6] model to pretrain the user/item embeddings in each domain, and uses element-wise attention to transfer embeddings among multiple domains.

4.3 Model Implementation and Complexity

Environment. We implement our model using PyTorch [1] with python 3.6 and train the framework on Tesla P40 GPUs with a memory size of 22.38 GiB and a 1.53 GHz memory clock rate.

Proposed Method¹. We use the BPRMF [23] model with the user and item embedding size of $m = 64$ for single domain recommendation and pretraining of domain-specific user embeddings. We reformulate the contrastive loss function given in [2] for the contrastive autoencoder task in our model. For the CAT module, we build the autoencoder with Multi-Layer Perceptron [5] where the sizes of the hidden layers are set to be $[5 \times m, 3 \times m, m]$ and $[m, 3 \times m, 5 \times m]$, $m = 64$ for the encoder and decoder modules, respectively. We adopt the PReLU activation function introduced by He et al. [8] between layers. We use a minibatch size of $N = 4096$

in the training of the CAT module, and set $\tau = 0.1$ in the contrastive loss (5). For each user, we mask out the domain-specific user embedding of one randomly selected domain out of the five domains in our collected dataset. In the loss function \mathcal{L}_{cat} , defined in (7), we set $\alpha_1 = \alpha_2 = 0.4$. The ART units in each domain have an identical structure, in which all adaptation modules are MLP layers with only one hidden layer which has the same size as its input, and the attention module is with only one head.

Model Complexity. The time complexity of the model training is $\mathcal{O}(\max_{i \in [1, n]} I_i * d + |U|nd^2)$ where n is the number of domains, I_i is the number of interactions in domain i , $|U|$ is the number of users and d denotes the embedding dimension of user.

4.4 Experimental Results

Table 2 summarizes the performances of our proposed CAT-ART model and all the baseline methods on the collected multi-target CDR scenario with 5 domains. The proposed CAT-ART model outperforms all the other baselines in most of the domains. CAT-ART achieves the best performance in “APP-Ins”, “APP-Use”, “Article”, and “Video-L” domains (**RQ1**). CAT-ART also avoids negative transfer and outperforms most of the baselines in “Video-S”. HeroGRAPH-L achieves a slightly better performance on the “Video-S”. The reasons for this phenomenon are twofold. First, the user behaviors in the “Video-S” domain are richer and more diverse which makes it less likely to be affected by information from other domains, that is, it is hard to improve the recommendation performance in “Video-S” through cross-domain information, as is shown in Table 2. Therefore, the recommendation performance in “Video-S” mainly depends on how we model user and item embeddings from the diverse user behavior data in the single domain. Secondly, we use the state-of-the-art graph-based recommendation model, i.e., Lightgcn [9], in the HeroGRAPH-L method, which is a deep Graph Convolutional Networks (GCN) that can generate better user and item embeddings in single domain compared to the MF model [27]. In summary, HeroGRAPH-L can get a slightly better performance than our method in the “Video-S” domain due to the superiority of GCN over the MF model. However, HeroGRAPH-L still suffers from the negative transfer problem causing great performance reduction in other domains, such as “APP-Use”, “Video-L”. Furthermore, CAT-ART outperforms all the baselines in the other four domains.

Compared with the SMF, we can see that CAT-ART effectively handles the negative transfer problem in all domains (**RQ2**). We attribute the success of the CAT-ART framework to the architecture design of the CAT and ART, which are specifically structured to avoid the negative transfer issue, while try to integrate as much useful information as possible from other domains to boost performance. Specifically, the CAT module generates high-quality global user embeddings for recommendations in all domains. And the ART module adoption of the attention mechanism to integrate domain-specific embeddings from other domains. We give a detailed analysis of each module in ablation studies in the following subsection. On the other hand, all the baseline cross-domain methods are severely affected by the negative transfer problem, causing significant performance degradation in many domains, e.g., “APP-Use”, “Articles”, and “Video-L”. This is reasonable. First, as shown in Table 1, the “APP-Use” and “Video-L” domains are much sparser

¹<https://github.com/Chain123/CAT-ART>

Table 2: Results (in %) of the Proposed Method and Baselines. The ↓ represents negative transfer compared with SMF.

Model	Domain	Precision		Recall		NDCG	
		@10	@20	@10	@20	@10	@20
SMF	APP-Ins	33.82±0.70	25.46±0.88	21.51±0.39	31.91±1.22	32.56±0.43	32.53±0.89
	APP-Use	20.91±0.23	12.21±0.26	65.5±0.89	75±1.50	57.39±1.46	60.81±1.72
	Article	16.02±0.73	12.05±0.58	16.64±0.43	23.25±0.40	21.59±1.30	21.93±1.06
	Video-S	3.9±0.03	3.86±0.02	3.59±0.44	6.9±0.77	3.83±0.13	4.84±0.25
	Video-L	5.98±0.20	3.91±0.10	26.73±0.87	34.6±0.88	20.37±1.19	22.91±1.2
CMF	APP-Ins	33.57±0.37↓	25.19±0.37↓	21.8±0.19	32.05±0.43	32.39±0.29↓	32.45±0.27↓
	APP-Use	20.41±0.11↓	12.17±0.05↓	64.91±0.27↓	75.54±0.16	43.99±0.78↓	47.89±0.74↓
	Article	10.29±0.27↓	8.37±0.19↓	8.83±0.23↓	13.79±0.28↓	11.24±0.34↓	12.07±0.31↓
	Video-S	3.87±0.12	3.81±0.12↓	4.08±0.17	7.6±0.29	4.00±0.14	5.04±0.18
	Video-L	4.74±0.03↓	3.26±0.01↓	21.44±0.12↓	29.14±0.06↓	12.67±0.07↓	15.14±0.05↓
MPF	APP-Ins	36.08±1.53	27.11±0.41	23.28±0.99	34.29±0.44	36.95±5.56	36.53±4.02
	APP-Use	20.95±0.12	12.26±0.16	65.55±0.44	75.18±0.84	55.67±2.71↓	59.14±2.52↓
	Article	14.55±0.16↓	11.14±0.11↓	15.35±0.07↓	21.72±0.12↓	20.96±0.63↓	21.29±0.52↓
	Video-S	3.63±0.29↓	3.67±0.13↓	3.71±0.30	7.16±0.68	3.85±0.40	4.91±0.11
	Video-L	2.74±0.95↓	2.09±0.52↓	11.96±4.31↓	18.2±4.66↓	8.03±3.65↓	10.01±3.79↓
GA-MTCRR	APP-Ins	16.77±0.05↓	10.35±0.02↓	11.7±0.01↓	14.37±0.03↓	17.81±0.08↓	16.01±0.03↓
	APP-Use	13.88±0.05↓	10.46±0.01↓	45.44±0.13↓	67.2±0.16↓	32.35±0.13↓	40.16±0.1↓
	Article	4.62±0.13↓	3.73±0.03↓	4.12±0.14↓	6.37±0.11↓	6.22±0.18↓	6.36±0.13↓
	Video-S	3.44±0.03↓	3.1±0.02↓	3.48±0.08↓	6.03±0.06↓	4.22±0.05	4.69±0.04
	Video-L	3.18±0.15↓	2.22±0.07↓	14.21±0.74↓	19.76±0.54↓	10.46±0.63↓	12.23±0.49↓
HeroGRAPH-L	APP-Ins	34.05±2.01	24.47±1.16↓	22.34±1.14	31.61±1.35↓	40.5±1.91	38.12±1.51
	APP-Use	20.68±0.36↓	11.98±0.15↓	66.11±0.83	74.96±0.61↓	59.51±1.08	62.74±0.98
	Article	11.27±0.12↓	8.61±0.12↓	15.01±0.2↓	20.68±0.33↓	18.19±0.16↓	18.86±0.23↓
	Video-S	3.99±0.14	3.7±0.15	5.29±0.21	8.97±0.34	5.31±0.18	6.2±0.23
	Video-L	5.42±0.29↓	3.65±0.15↓	24.62±1.22↓	32.84±1.29↓	18.71±1.21↓	21.35±1.24↓
CAT-ART	APP-Ins	38.36±0.58	27.96±0.31	24.86±0.34	35.46±0.39	43.47±1.23	41.55±0.94
	APP-Use	21.23±0.18	12.33±0.18	66.53±0.65	75.66±1.02	59.98±0.86	63.27±1.02
	Article	16.82±0.21	12.4±0.13	18.76±0.56	25.47±0.6	25.97±0.61	25.79±0.58
	Video-S	3.93±0.08	3.93±0.06	3.83±0.50	7.35±0.82	3.93±0.14	5.05±0.24
	Video-L	6.08±0.09	3.96±0.08	27.18±0.39	35.01±0.67	21.03±0.38	23.54±0.86

than the other domains. Furthermore, user behavior in “App-Ins”, “APP-Use” and “Article” are monotonous in nature compared with “Video-S” (therefore, the SMF achieves a much higher precision score in these domains). Obviously, when data from all domains are collected and trained together, domains with more sparsity tend to be overwhelmed by other domains causing biased cross-domain representation and negative transfer, e.g., in domains “APP-Use” and “Video-L”. In addition, domains with simple and monotonous data prone to be affected by too much information in other domains, e.g., for domains “App-Ins”, “APP-Use”, and “Articles”. Apparently, according to the experimental results, none of the previous work can handle these situations(**RQ2**).

4.5 Ablation Study and Analysis

We conduct ablation studies to show the effectiveness of each proposed module and to demonstrate how negative transfer is addressed through model design (**RQ3**). We incrementally accommodate different modules into the single-domain matrix factorization model (SMF), until we incorporate all the proposed sub-modules and features. Specifically, the following models are evaluated:

- **SMF**: The single-domain Matrix Factorization (MF) model.

- **+Autoencoder**: We add the original autoencoder to extract global representations for CDR.
- **+Contrastive**: We further add the contrastive loss for the training of the autoencoder, i.e., the CAT module.
- **+ART**: The ART module is further incorporated to integrate domain-specific user embedding.

Furthermore, to show the impact of negative transfer when domain-specific embeddings are directly transferred without attention, we further remove the attention layer within the ART module. Specifically, the following model is evaluated:

- **-Attention**: We remove the attention from the ART and only use MLP layers to integrate domain-specific features.

Table 3 summarizes the results of ablation studies. The values of metrics Precision@10, Recall@10, and NDCG@10 are given. We can see, each time we add a new sub-module or feature incrementally on top of the previous model, we can observe an improvement on the overall recommendation performance, which illustrates the effectiveness of autoencoder, contrastive self-supervised learning, and the ART modules. Specifically, with the original autoencoder framework, we already get rid of the negative transfer in all domains of our dataset. This can be attribute to the natural advantage

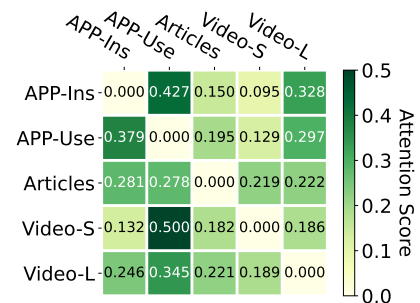
Table 3: Results (in %) of ablation studies. The ↓ represents negative transfer compared with the SMF model.

Domain	Metric	SMF	+Autoencoder	+Contrastive	+ART	-Attention
App-Ins	Precision@10	33.82±0.70	37.64±1.17	37.95±0.45	38.36±0.58	36.24±0.26
	Recall@10	21.51±0.39	24.35±0.76	24.58±0.35	24.86±0.34	23.35±0.23
	NDCG@10	32.56±0.43	41.34±3.75	42.56±2.02	43.47±1.23	36.08±1.54
APP-Use	Precision@10	20.91±0.23	21.00±0.11	21.08±0.23	21.23±0.18	21.01±0.07
	Recall@10	65.50±0.89	65.77±0.33	66.01±0.88	66.53±0.65	65.92±0.41
	NDCG@10	57.39±1.46	59.09±0.37	58.61±0.40	59.98±0.86	59.28±0.24
Article	Precision@10	16.02±0.73	16.54±0.46	16.46±0.34	16.82±0.21	15.88 ± 0.15↓
	Recall@10	16.64±0.43	17.48±1.21	17.19±1.13	18.76±0.56	15.89±0.38↓
	NDCG@10	21.59±1.30	23.98±2.28	23.54±2.75	25.97±0.61	22.25±1.71
Video-S	Precision@10	3.89±0.025	3.91±0.08	3.97±0.13	3.93±0.08	3.82±0.28↓
	Recall@10	3.59±0.44	3.71±0.40	3.72±0.37	3.83±0.50	3.46±0.25↓
	NDCG@10	3.83±0.13	3.87±0.08	3.91±0.05	3.93±0.14	3.73±0.18↓
Video-L	Precision@10	5.98±0.20	6.04±0.01	6.07±0.04	6.08±0.09	5.86±0.03↓
	Recall@10	26.73±0.87	27.00±0.08	27.17±0.20	27.18±0.39	26.27±0.09↓
	NDCG@10	20.37±1.19	21.00±0.14	21.12±0.21	21.03±0.38	20.26±0.15↓

of the autoencoder framework where the encoder is trained to extract the most important information to reconstruct the input, thus, is able to reduce the effect of noisy sample. Furthermore, by adding contrastive training (*+Contrastive*), we improve the performance in most of the domains via a more general user representation. However, the recommendation is not always improved, especially in the “Article” domain. The reasons are twofold. 1) In our work, the goal of contrastive learning is to make the user embedding more robust to noise and less dependent on domain-specific information. 2) The “Article” domain is less related to the rest of the domains, that’s why all the baselines methods encounter the negative transfer problem in this domain, shown in Table 2. Therefore, a more domain-independent global embedding given by the CAT module has less domain-specific information from the “Article” domain, resulting in slightly worse recommendation performance compared to the user embedding given by the autoencoder. Finally, domain-specific embeddings are incorporated through the *ART* module to further boost recommendation performance, while avoiding negative transfer through attention. Note that, due to the special data characteristics in the “Video-S” domain, we can only get a relative small improvement. Thus, all variants have close performance. However, it is clearly shown in Table 3 that we get the best performance when incorporating all the modules, i.e., *+ART*, in the rest of the domains.

Furthermore, without attention (*-Attention*), the performance is greatly reduced in all domains and the negative transfer problem also prevails, which demonstrate the effectiveness of the attention module in avoiding the negative transfer problem.

Figure 2 shows the averaged attention scores on the test set, given by the *ART* module in each individual domain. Each row represents the attention weights assigned to the source domains by the corresponding target domain, therefore the sum of weights in a row equals 1. We can see, the weights between related domains are high, e.g., domains “APP-Ins” and “APP-Use”, while weights between unrelated domains are low, e.g., “APP-Ins” and “Video-L”. Furthermore,

**Figure 2: Averaged attention scores on the test set.**

the asymmetry of the weight matrix shows the inequality in the use of shared domain embedding between two domains. For example, “APP-Use” assigns a weight of 0.13 to the “Video-S” domain, while the “Video-S” gives an attention score of 0.5 to the “APP-Use”. This is reasonable, it’s helpful for the recommendation of videos if we know what Application a user is more likely to use, but the other way around is much harder. These phenomena further demonstrate the need for a mechanism to select the most important and helpful information among the features provided by multiple domains.

5 CONCLUSION

In this paper, we focus on the MTCDR problem and propose the CAT-ART model. We build a CAT module to extract robust unbiased global user representation in a self-supervised manner via contrastive learning and an autoencoder framework based on pre-trained domain-specific user embeddings. Then, the *ART* module is built in each domain, which transfers domain-specific user embeddings from other domains with the attention mechanism. Combining these two modules, CAT-ART boosts recommendation in all participating domains and avoiding negative transfer at the same

time. We believe CAT-ART has made a valuable contribution in exploring the MTCDR and the negative transfer issue, approaching the objectives of One for All and All for One.

REFERENCES

- [1] Paszke Adam, Gross Sam, Chintala Soumith, and Chanan Gregory. 2017. *Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration*. <https://pytorch.org/>.
- [2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.
- [3] Qiang Cui, Tao Wei, Yafeng Zhang, and Qing Zhang. 2020. HeroGRAPH: A Heterogeneous Graph Framework for Multi-Target Cross-Domain Recommendation. In *ORSUM@ RecSys*.
- [4] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th international conference on world wide web*. 278–288.
- [5] Matt W Gardner and SR Doring. 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment* 32, 14–15 (1998), 2627–2636.
- [6] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.
- [7] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 1025–1035.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.
- [9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.
- [10] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.
- [11] Heishiro Kanagawa, Hayato Kobayashi, Nobuyuki Shimizu, Yukihiro Tagami, and Taiji Suzuki. 2019. Cross-domain recommendation via deep domain adaptation. In *European Conference on Information Retrieval*. Springer, 20–29.
- [12] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-supervised learning for cross-domain recommendation to cold-start users. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1563–1572.
- [13] Donghyun Kim, Sungchul Kim, Handong Zhao, Sheng Li, Ryan A Rossi, and Eunye Koh. 2019. Domain switch-aware holistic recurrent neural network for modeling multi-domain user behavior. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 663–671.
- [14] Adit Krishnan, Mahashweta Das, Mangesh Bendre, Hao Yang, and Hari Sundaram. 2020. Transfer Learning via Contextual Invariants for One-to-Many Cross-Domain Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1081–1090.
- [15] 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 (Virtual Event, AZ, USA) (WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 571–581. <https://doi.org/10.1145/3488560.3498388>
- [16] Pan Li and Alexander Tuzhilin. 2020. Dtdcdr: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 331–339.
- [17] Pan Li and Alexander Tuzhilin. 2021. Dual Metric Learning for Effective and Efficient Cross-Domain Recommendations. *IEEE Transactions on Knowledge and Data Engineering* (2021).
- [18] Jian Liu, Pengpeng Zhao, Fuzhen Zhuang, Yanchi Liu, Victor S Sheng, Jiajie Xu, Xiaofang Zhou, and Hui Xiong. 2020. Exploiting aesthetic preference in deep cross networks for cross-domain recommendation. In *Proceedings of The Web Conference 2020*. 2768–2774.
- [19] 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.
- [20] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*. 931–940.
- [21] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach.. In *IJCAI*, Vol. 17. 2464–2470.
- [22] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, Vol. 242. Citeseer, 29–48.
- [23] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
- [24] Ashish Kumar Sahu and Pragma Dwivedi. 2020. Knowledge transfer by domain-independent user latent factor for cross-domain recommender systems. *Future Generation Computer Systems* 108 (2020), 320–333.
- [25] 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.
- [26] 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.
- [27] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.
- [28] Yaqing Wang, Chunyan Feng, Caili Guo, Yunfei Chu, and Jenq-Neng Hwang. 2019. Solving the sparsity problem in recommendations via cross-domain item embedding based on co-clustering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 717–725.
- [29] Huan Yan, Chunfeng Yang, Donghan Yu, Yong Li, Depeng Jin, and Dah Ming Chiu. 2021. Multi-Site User Behavior Modeling and Its Application in Video Recommendation. *IEEE Transactions on Knowledge & Data Engineering* 33, 01 (2021), 180–193.
- [30] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1469–1478.
- [31] Feng Yuan, Lina Yao, and Boualem Benatallah. 2019. DARec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns. *arXiv preprint arXiv:1905.10760* (2019).
- [32] Fajie Yuan, Guoxiao Zhang, Alexandros Karatzoglou, Joemon Jose, Beibei Kong, and Yudong Li. 2021. One person, one model, one world: Learning continual user representation without forgetting. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 696–705.
- [33] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. 2020. Overcoming negative transfer: A survey. *arXiv preprint arXiv:2009.00909* (2020).
- [34] Cheng Zhao, Chenliang Li, Rong Xiao, Hongbo Deng, and Aixin Sun. 2020. Catn: Cross-domain recommendation for cold-start users via aspect transfer network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 229–238.
- [35] Xiaoyun Zhao, Ning Yang, and Philip S Yu. 2022. Multi-Sparse-Domain Collaborative Recommendation via Enhanced Comprehensive Aspect Preference Learning. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1452–1460.
- [36] Feng Zhu, Chaochao Chen, Yan Wang, Guanfang Liu, and Xiaolin Zheng. 2019. Dtdcr: A framework for dual-target cross-domain recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1533–1542.
- [37] Feng Zhu, Yan Wang, Chaochao Chen, Guanfang Liu, and Xiaolin Zheng. 2020. A Graphical and Attentional Framework for Dual-Target Cross-Domain Recommendation.. In *IJCAI*. 3001–3008.
- [38] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfang Liu. 2021. Cross-domain recommendation: challenges, progress, and prospects. *arXiv preprint arXiv:2103.01696* (2021).
- [39] Feng Zhu, Yan Wang, Jun Zhou, Chaochao Chen, Longfei Li, and Guanfang Liu. 2021. A unified framework for cross-domain and cross-system recommendations. *IEEE Transactions on Knowledge and Data Engineering* (2021).
- [40] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized transfer of user preferences for cross-domain recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1507–1515.
- [41] Fuzhen Zhuang, Yingmin Zhou, Fuzheng Zhang, Xiang Ao, Xing Xie, and Qing He. 2018. Cross-domain novelty seeking trait mining for sequential recommendation. *arXiv preprint arXiv:1803.01542* (2018).