Exploring False Hard Negative Sample in Cross-Domain Recommendation

Haokai Ma•
School of Software, Shandong University
Jinan, China
mahaokai@mail.sdu.edu.cn

Ruobing Xie
WeChat, Tencent
Beijing, China
ruobingxie@tencent.com

Lei Meng*
School of Software, Shandong University;
Shandong Research Institute of Industrial Technology
Jinan, China
lmeng@sdu.edu.cn

Xin Chen
WeChat, Tencent
Beijing, China
andrewxchen@tencent.com

Xu Zhang
WeChat, Tencent
Beijing, China
xuonezhang@tencent.com

Leyu Lin
WeChat, Tencent
Beijing, China
goshawklin@tencent.com

Jie Zhou
WeChat, Tencent
Beijing, China
withtomzhou@tencent.com

ABSTRACT

Negative Sampling in recommendation aims to capture informative negative instances for the sparse user-item interactions to improve the performance. Conventional negative sampling methods tend to select informative hard negative samples (HNS) besides the default random samples. However, these hard negative sampling methods usually struggle with false hard negative samples (FHNS), which happens when a user-item interaction has not been observed yet and is picked as a negative sample, while the user will actually interact with this item once exposed to it. Such FHNS issues may seriously confuse the model training, while most conventional hard negative sampling methods do not systematically explore and distinguish FHNS from HNS. To address this issue, we propose a novel model-agnostic Real Hard Negative Sampling (RealHNS) framework specially for cross-domain recommendation (CDR), which aims to discover the false and refine the real from all HNS via both general and cross-domain real hard negative sample selectors. For the general part, we conduct the coarse- and fine-grained real HNS selectors sequentially, armed with a dynamic item-based FHNS filter to find high-quality HNS. For the cross-domain part, we further design a new cross-domain HNS for alleviating negative transfer in CDR and discover its corresponding FHNS via a dynamic user-based FHNS filter to keep its power. We conduct experiments on four datasets based on three representative hard negative sampling methods, along with extensive model analyses, ablation studies, and universality analyses. The consistent improvements indicate the effectiveness, robustness, and universality of RealHNS, which is also easy-to-deploy in real-world systems as a plug-and-play strategy. The source code is avaliable in https://github.com/hulkima/RealHNS.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommender System, Cross-domain Recommendation, Negative Sampling

ACM Reference Format:
Haokai Ma•, Ruobing Xie, Lei Meng*, Xin Chen, Xu Zhang, Leyu Lin, and Jie Zhou. 2023. Exploring False Hard Negative Sample in Cross-Domain Recommendation. In *Seventeenth ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3604915.3608791

1 INTRODUCTION

Personalized recommendation aims to provide appropriate items for users. Like other classical supervised learning tasks, recommendation models also need positive and negative samples for training [21, 22, 25]. However, real-world recommender usually have more than millions of users and items. The personalized demands and data sparsity make it impossible for all users to interact with all items to get the comprehensive golden positive/negative feedback matrix [34]. In practical recommender systems, cross-domain recommendation (CDR) is a straightforward but effective technique to transfer useful positive signals from the source domain to the target domain [17, 41]. Lots of strategies (e.g., data augmentation and multi-behavior recommendation [38, 49]) are also proposed

* indicates corresponding author.
• This work has been done when the author was at Tencent for internship.

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

ACM ISBN 979-8-4007-0241-9/23/09…$15.00
https://doi.org/10.1145/3604915.3608791

to bring in additional positive samples. Unlike the widespread attention paid to positive samples, most works simply consider the randomly sampled items in the overall corpus as negative samples (NS) [9, 30, 48]. However, this mainstream strategy often selects NS that are too easy to distinguish for the model, which makes the training less challenging and informative [2, 7, 50].

To address the issue of negative samples being too simple, some efforts start to focus on **hard negative sample (HNS)** in recommendation. There are roughly two types of HNS based on their constructions. The first HNS comes from real user behaviors. A typical hard negative sampling strategy widely used in practice is regarding the *exposed-but-unclicked* items of a user as his/her hard negative feedback. However, such implicit negative feedback usually have a large amount of noise (sometimes even containing certain positive preferences [40]). Moreover, all exposed items strongly depend on the online recommendation algorithm, which may be biased. The second HNS derives from **hard negative sampling**. For example, DNS [44] selects the hardest item (i.e., having the largest predicted user-item click probability) among randomly-sampled candidate items as HNS. MixGCF [13] uses the interpolation of random samples and positive samples as HNS. AFT [11] adopts adversarial training to build HNS. In general, most existing hard negative sampling methods attempt to choose items that are close to the user or the positive sample for challenging and informative training.

However, in many cases, simply selecting HNS for training may even degrade the model performance. This is mainly because there is an undeniable proportion of **false hard negative samples (FHNS)** in the HNS generated by either natural feedback or negative sampling [1, 40]. FHNS indicates a positive sample (but not observed yet) that is mistakenly selected as a HNS. There are two typical reasons for FHNS: (a) some clicks occur spontaneously and unpredictably that are beyond users' existing preferences, and (b) some similar items of the clicked one may also be welcomed by the user. Carelessly training models with FHNS (positive) viewed as HNS (negative) will largely confuse the overall optimization. The root of the FHNS issue lay in the incompleteness of the user-item interaction matrix, since we cannot be 100% certain that a user will definitely click or not click on an item before it is exposed to the user, and there is no general golden standard for all users. Moreover, the harder NS are, the more likely they are false NS. Therefore, it is extremely challenging (and even impossible) to perfectly discover all false NS from HNS. Unfortunately, most conventional hard negative sampling strategies do not contrapuntally face the challenges of FHNS. Some works [12] straightforwardly discard the top hardest negative items to reduce FHNS. Other works [13, 44] indirectly alleviate the effects of FHNS by only selecting the hardest items from a random item subset (rather than the overall corpus, hence the selected HNS are not absolutely "hard"). Besides, the definitions of false and hard NS should be more relative based on the current model's capability. It is also essential to dynamically balance the random, hard, and false NS in training.

In this work, we attempt to explore the FHNS that widely existed in CDR, understand their characteristics, and improve the quality of HNS. We propose an effective, universal, and simple **Real hard negative sampling (RealHNS)** framework to deal with different types of FHNS in CDR, which is supposed to enhance different hard negative sampling methods. Specifically, RealHNS aims to discover and refine two types of HNS, namely the general HNS and the cross-domain HNS, providing more informative and challenging training while preventing the model from being affected by false HNS. (1) In the *general real hard negative sample selector*, we design both coarse- and fine-grained HNS selectors to efficiently find HNS. We propose a **dynamic item-based FHNS filter**, which could smartly discard HNS that is too similar to the positive item. (2) In the *cross-domain real hard negative sample selector*, we creatively propose a novel *cross-domain HNS* to fight against the negative transfer issue in CDR. Correspondingly, we find that some users are less affected by such negative transfer. Hence, we design a **dynamic user-based FHNS filter** cooperating with the coarse- and fine-grained HNS selectors specially for refining cross-domain HNS from the user aspect. (3) To smartly balance the random, hard, and false NS in training, we further design a *curriculum learning* framework for real HNS selection.

In experiments, we systematically evaluate the effectiveness and universality of our proposed RealHNS on four domains adopted with three representative hard negative sampling methods, where RealHNS achieves significant improvements. We conduct extensive ablation study and parameter analyses on multiple datasets to generate a solid and comprehensive understanding of false and hard NS in CDR. RealHNS is also verified on classical CDR and even single-domain recommendation models. The contributions of this work are concluded as follows:

- To the best of our knowledge, We are the first to systematically explore the false hard negative sample and propose both general and cross-domain real HNS selectors in CDR.
- We design a new dynamic item-based FHNS filter along with a curriculum learning framework, which could be adopted with different hard negative sampling methods and different base cross-domain/single-domain models. We also propose a new type of cross-domain HNS and highlight its corresponding FHNS specially for CDR.
- Our RealHNS achieves significant improvement based on different hard negative sampling methods and base models. It could be used as a plug-and-play effective and robust negative sampling strategy in practice. Extensive ablation study, universality analyses, and model analyses enable a comprehensive understanding of FHNS.

2 RELATED WORK

Cross-domain Recommendation. Cross-domain recommendation (CDR) is one of the representative methods to alleviate the data sparsity problem with auxiliary user behaviors from other domains [24, 53]. Classical CDR methods generally model the cross-domain knowledge transfer with the multi-task learning [52], alignment constraint [20, 35], and contrastive learning [41]. Cross-domain sequential recommendation (CDSR) focuses more on users' multi-domain chronological behavior sequences in CDR [2, 3, 10, 17, 39, 46]. DASL [17] designs a dual attention strategy to emphasize the correlation of users' multi-domain behavior. DDGHM [46] builds a global dynamic graph to jointly leverage the local and global information. C^2DSR [2] explores the user's single- and cross-domain preferences via the mutual information maximization mechanism.

However, most existing CDR methods merely focus on the feature-level cross-domain correlations with the negative samples randomly selected from the target domain, ignoring the cross-domain differences at the sample level. These samples, to a certain extent, disregard users' source-domain preference, which may lead to sub-optimal performance. To the best of our knowledge, RealHNS is the first negative sampling framework in CDR.

Negative Sampling in Recommendation. Negative Sampling has been widely used in the field of Computer Vision [14, 33, 43], Natural Language Processing [18, 37, 45], Information Retrieval [8, 26, 27] and Recommendation System [6, 13, 31, 44]. In recommendation, existing negative sampling methods are usually classified into static negative sampling strategies and hard negative sampling strategies due to the fixed sampling probability distribution or not.

Static negative sampling strategy generally samples negative instances based on a pre-defined probability distribution, including uniform probability [4, 29] and item popularity [28]. UNS [29] randomly samples items with equal probability; NNCF [28] assigns the sampling weights to items based on their popularity; ENMF [4] designs a non-sampling training strategy to include all the corpus in training. However, these strategies perform the negative sampling according to a fixed distribution probability, which makes it fail to capture the preference variation between users and items.

Hard negative sampling is one of the basic training methods to improve the models' accuracy and training efficiency, which is proposed to select more informative negative samples. The early study, DNS [44], generally selected the hardest item as the HNS from the randomly selected item candidates. However, recent studies leverage the item similarity[19], adversarial learning[11, 51], heuristic statistical features[6], interpolation[13] and random noise[42] to select HNS. SRNS [6] proposed a variance-based sampling function with the observed statistical features to distinguish HNS; MixGCF [13] designs the hop mixing and positive mixing strategies to synthesize the informative HNS; AugNS [42] achieves NS augmentation by introducing uniform noise to the representation, which preserves most of the original information while bringing semantic differences; DNS+ [31] enhances the DNS [44] by adjusting the sampling difficulty through the utilization of additional parameters to accommodate different metrics. Nevertheless, these aforementioned strategies are primarily designed for collaborative filtering and may not be directly applicable to CDR tasks. Additionally, these methods are designed to avoid the false negative problem by the selection of parameters, which is unstable and non-interpretable for different datasets, making it challenging to effectively explore and utilize HNS.

3 MOTIVATION ANALYSIS

3.1 Early Uniform Negative Sampling Methods

Given the item sets $\mathcal{I}_u^+$ that user u has interacted with, traditional recommenders generally select items from $\mathcal{I}_u^- = \mathcal{I} \backslash \mathcal{I}_u^+$ as the NS uniformly, which are too random and too noisy to obtain idealized performance. These NS prove excessively facile for the recommender, thereby dominate its global optimization direction. Therefore, recent works improved its accuracy and robustness by increasing the number of NS or designing complex methods to select hard and informative NS. Drawing from the foundational definition of FHNS, their gradient direction tends to vary from the gradient directions of other easy NS. Theoretically, it is intuitive that more NS can reduce the impact of the FHNS on the global gradient, thus benefiting the model's optimization. Therefore, due to the imbalance of the FHNS and the easy NS, **the more the number of randomly sampled triplets is, the smaller the impact of FHNS on the global gradient**.

We also conduct simulation experiments on the number of random NS on two different datasets. Fig. 1 shows that: (1) Most metrics first increase and then decrease as the number of NS gets large. (2) The desirable computational cost and performance are generally achieved on all datasets when the number of NS is 20. These findings demonstrate that the appropriate number of NS does achieve comprehensive optimization and alleviates the impact of false NS.

3.2 Recent Hard Negative Sampling Strategies

As mentioned in Sec. 3.1, recent works generally design different methods to select HNS in recommendation. We detail the effective mechanisms and the existing problems of these hard negative sampling strategies in this section.

The existing hard negative sampling methods seldom directly address the false negative problems. Instead, they tend to select informative NS based on dynamic score, interpolation, and other selection criteria [13, 31, 44]. DNS [44] oversamples the top-ranked items from the randomly selected candidates and adjusts the number of the candidates to avoid sampling too hard NS (perhaps the FHNS). DNS+ [31] further reduces the probability of selecting overly hard NS by expanding both the number of item candidates and the selection range on the basis of DNS [44]. MixGCF interpolates different proportions of positive embeddings into item candidates, which not only optimizes the feature representation of the selected samples but also translates the potential FHNS into HNS to avoid training bias.

However, these methods have a certain randomness in sampling HNS, making it difficult to stably solve the false negative problem. Additionally, there is a lack of interpretability in parameter selection for different datasets, which cannot utilize HNS in a universal manner. Furthermore, they have failed to reveal the essence of the false negative problem: **Which items can be considered as false HNS?** In the CDR scenario there is another problem : **Is there the false negative problem in CDR, and if so, how should we find the false HNS under the CDR setting?**

4 METHOD

4.1 Problem Formulation

In this section, we first outline the definition of negative samples, then we introduce some challenges unique to the CDR setting and further propose three types of Real HNS in CDR, and finally give a formal depiction of the CDR task.

Due to the requirement of positive and negative samples and the absence of negative feedback in the implicit feedback, traditional recommenders randomly select items that users have not interacted with as ***negative samples (NS)***. Some algorithms attempt to capture ***hard negative samples (HNS)*** with the pre-defined distribution or other dynamic sampling methods. HNS refers to the samples with more information than uniform NS, and their incorporation can aid the model's optimization. ***False hard negative samples***

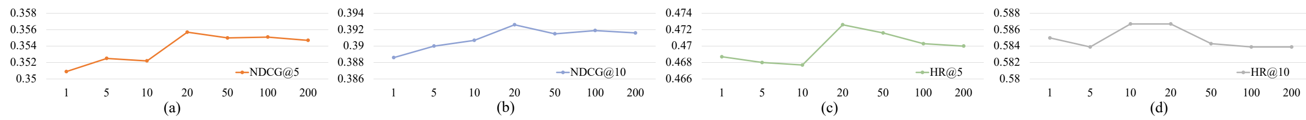

Figure 1: Performance comparison over different number of random negative samples in CDR.

(FHNS) are items wrongly selected as HNS but will be clicked by users, which lead to training bias but are still under-explored. We argue that there is a certain overlap in the existing definitions of HNS and FHNS, and it is impossible to explicitly distinguish which the selected NS belongs to. Therefore, we propose a concept of ***real hard negative samples (RHNS)***, that is, the selected HNS are neither too easy (uninformative) nor too hard (potential FHNS). Intuitively, we have $\{RHNS\} = \{HNS\} - \{FHNS\}$.

It is natural that there is always exist a bias between a user's multi-domain behavior, and the general RHNS cannot be applied to CDR tasks that involve both multi-domain behavior and temporal information. Especially for those users who demonstrate significant differences among their multi-domain preferences (called outliers), this bias impacts their transfer preferences due to the mainstream transfer of all users. As a result, the general RHNS may harm the optimization specific to these outliers. Meanwhile, there is often a situation in which users sporadically exhibit preferences for some items that are contrary to their global preferences in CDR or even in conventional recommendation scenarios. These non-standard users are unpredictable and represent a small proportion of all users, thus are not within the scope of our current discussion. To this end, we define three types of HNS in CDR: general HNS, cross-domain HNS, and occasional HNS. And then we propose the following three hypotheses based on the above empirical analysis:

- Hypothesis 1: **Items in close proximity to the positive sample are more likely to be FHNS in CDR.**
- Hypothesis 2: **Samples that bear a strong resemblance to the user's source domain representation indicate their transfer preferences and are more likely to be RHNS for outliers.**
- Hypothesis 3: **Introducing all HNS at the beginning of the training process may lead to computational wastage and sub-optimal performance.**

We define the *source behavior sequence* $S^S = \{v_1^S, v_2^S, \cdots, v_p^S\}$ and the *target behavior sequence* $S^T = \{v_1^T, v_2^T, \cdots, v_q^T\}$ in the source domain S and target domain T for each user, where v_i^S and v_j^T are the items that the user has interacted with in the source/target domains and p, q denote the source/target historical behavior lengths respectively. RealHNS tries to recommend the target item v_{q+1}^T that will be interacted by this user in the target domain.

4.2 Overall Framework

In this section, we aim to provide a simple and universal Real Hard Negative Sampling (RealHNS) framework, which employs the general real HNS selector and the cross-domain real HNS selector during the sampling process to improve CDR. The overall structure of RealHNS is illustrated in Fig.2. Specifically, given the source/target sequence representations, we first propose a general real HNS selector to sample the informative general real HNS and eliminate the potential false HNS from them with the item-based FHNS filter. To further alleviate the negative transfer in CDR, we further design a cross-domain real HNS selector, which is armed with the dynamic user-based and item-based FHNS filter to differentiate the outliers among the entire user population and the false HNS within the cross-domain item candidates in a dynamic sampling manner. It is noteworthy that the above sampling methods in both domains are symmetric and model-agnostic, which allows such methods can be easily migrated to Collaborative Filtering (CF) and Sequential Recommendation (SR) tasks, and the analysis of scenario universality is shown in Sec.5.6.

4.3 Base Sequence Encoder

We adopt SASRec[15] as the sequential encoder to capture the user's domain-specific preferences. Taking the target domain sequence S^T as an example, we first build the input matrix $\boldsymbol{D}^T = [\boldsymbol{v}_1^T + \boldsymbol{p}_1^T, \boldsymbol{v}_2^T + \boldsymbol{p}_2^T, \cdots, \boldsymbol{v}_q^T + \boldsymbol{p}_q^T] \in \mathbb{R}^{q\times d}$, where d is the embedding size, $\boldsymbol{v}_i^T$ and $\boldsymbol{p}_i^T$ denote the learnable item embedding and sequence position embedding respectively. We project $\boldsymbol{D}^T$ into three matrices as query $\boldsymbol{Q}^T$, key $\boldsymbol{K}^T$, and value $\boldsymbol{V}^T$, and then apply the self-attention as follows:

$$\hat{\boldsymbol{H}}^T = \text{Attention}(\boldsymbol{Q}^T, \boldsymbol{K}^T, \boldsymbol{V}^T) = \text{Softmax}\left(\boldsymbol{Q}^T(\boldsymbol{K}^T)^\top/\sqrt{d}\right)\boldsymbol{V},$$
$$\hat{\boldsymbol{H}}^T \in \mathbb{R}^{q\times d}. \tag{1}$$

After that, we apply a point-wise feed-forward network to endow the model with non-linearity, which is defined as:

$$\boldsymbol{H}^T = \text{ReLU}\left(\hat{\boldsymbol{H}}^T\boldsymbol{w}_1 + \boldsymbol{b}_1\right)\boldsymbol{w}_2 + \boldsymbol{b}_2, \quad \boldsymbol{H}^T \in \mathbb{R}^{q\times d}. \tag{2}$$

where $\boldsymbol{H}^T$ is the behavior matrix, $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{b}_1, \boldsymbol{b}_2$ denote the weight and bias respectively. The behavior matrix $\boldsymbol{H}^S$ of source domain are similarly constructed with the user's source-domain behavior sequence S^S.

4.4 General Real Hard Negative Sample Selector

The prime task for solving the false negative problem is leveraging the user's historical behavior information to distinguish the false HNS and real HNS from the corpus. In this section, we propose a general real HNS selector to select general real HNS in CDR. Precisely, we design a general coarse-grained real HNS selector to generate item candidates related to the user's general preference and propose the general fine-grained real HNS selector to eliminate items from the candidate which are too similar to the positive sample and dynamically select general real HNS (See Hypothesis 1).

4.4.1 General Coarse-grained RHNS Selector. Previous hard negative sampling works generally sample a fixed number of items candidates uniformly, and then dynamically select the item which scored highest by the recommender from the randomly selected candidates as the HNS. However, such methods excessively rely on the appropriate number of candidates, that is, too few candidates lead to the randomness and unstable quality of HNS, while too many may significantly increase the hardness of the selected HNS,

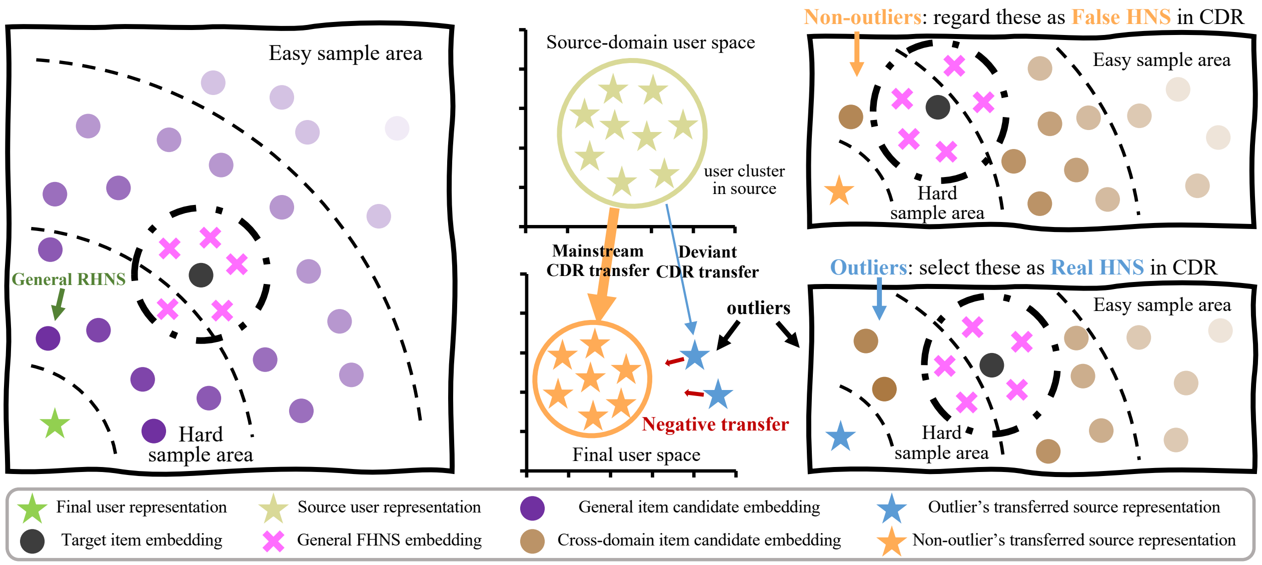

Figure 2: An illustration of two false hard negative samples in CDR, where each star and circle denote the user and item representation and the brightness of color indicates the item's sampling probability. (i) General FHNS are close to the positive sample (Left). (ii) Negative transfer in CDR. Users having deviant transfer patterns with the mainstream transfer are viewed as outliers, who will be wrongly impacted by non-outliers as the red arrows (Middle). (iii) Cross-domain FHNS are related to non-outliers in CDR (Right).

which in turn leads to bias in model optimization. Meanwhile, these methods exhibit significant variability in their parameters across different datasets, and they fail to deliver consistent improvement over static negative sampling methods in all scenarios (See the main results in Sec.5.4).

To this end, we propose a general coarse-grained selector to sample item candidates related to the user's general preference in CDR. Specifically, RealHNS first calculates the source/target prior behavior matrices $\bar{\boldsymbol{H}}^S$ and $\bar{\boldsymbol{H}}^T$ via the base sequence encoder at the beginning of each epoch. Assuming that "the last behavior in a user's behavior sequence contains his/her overall preference", RealHNS generates the prior final user representation $\bar{\boldsymbol{u}}_u^T$ of user u as follows:

$$\bar{\boldsymbol{u}}_u^T = \text{MLP}^f(\bar{\boldsymbol{h}}_{u,p}^S \parallel \bar{\boldsymbol{h}}_{u,q}^T). \tag{3}$$

where $\bar{\boldsymbol{h}}_{u,p}^S$ and $\bar{\boldsymbol{h}}_{u,q}^T$ denote the prior last behavior embeddings of user u in source and target domains respectively, $\text{MLP}^f(.)$ denotes a two-layer fully-connected network with the LeakyReLU activation. Finally, RealHNS computes and sorts the score $s_u^T = [\bar{\boldsymbol{u}}_u^T \boldsymbol{v}_1^T, \cdots, \bar{\boldsymbol{u}}_u^T \boldsymbol{v}_n^T]$ with the prior final user representation $\bar{\boldsymbol{u}}_u^T$ and the learnable item embeddings $\boldsymbol{V}^T = [\boldsymbol{v}_1^T, \boldsymbol{v}_2^T, \cdots, \boldsymbol{v}_n^T] \in \mathbb{R}^{n \times d}$, and samples 1,000 top-ranked items to form the general item candidates $\mathcal{R}_u^T$. This means that RealHNS is able to sample general item candidates based on the prior knowledge of the user's general preferences, alleviating the problem of excessive randomness of existing methods in selecting item candidates. It is worth noting that RealHNS is only implemented at the beginning of each epoch without introducing excessive computational cost, and the $O(n)$ time complexity can be achieved by KD-Tree when serving online.

4.4.2 General Fine-grained RHNS Selector. In contrast to the classical hard negative sampling works which assumes that the higher score between users and items is more likely to be FHNS, RealHNS additionally proposes a hypothesis that items in close proximity to the positive sample are more likely to be false HNS in CDR. It is intuitive that if a user likes a certain item (Iron Man), it is probable that he/she will also like other items that are similar to it (Captain America and other superheroes). For this hypothesis, RealHNS designs a fine-grained real HNS selector based on unsupervised clustering to adaptively filter the item candidates. Precisely, RealHNS applies the K-means algorithm on the item embeddings $\boldsymbol{V}^T$ to generate the item cluster C^I and the corresponding cluster centers $\boldsymbol{\Phi}^I = [\boldsymbol{\phi}_1^I, \boldsymbol{\phi}_2^I, \cdots, \boldsymbol{\phi}_{k_i}^I] \in \mathbb{R}^{k_i \times d}$ (k_i is the number of item clusters). RealHNS then computes the average item scores $S^I = [s_1^I, s_2^I, \cdots, s_{k_i}^I]$, where s_k^I denotes the score between the embeddings $\boldsymbol{V}_k^I$ of $\mathcal{R}_k^I$ in cluster k with their corresponding cluster centers $\boldsymbol{\phi}_k^I$, can be defined as:

$$s_{k_i}^I = \sum \text{CalScore}(\boldsymbol{V}_k^I, \boldsymbol{\phi}_k^I)/\text{len}(C_k^I) = \textstyle\sum_{v \in C_k^I} (\boldsymbol{v}^T)^\top \boldsymbol{\phi}_k^I / \text{len}(C_k^I) \tag{4}$$

And then RealHNS calculates the average cluster score $s_c = \sum_{s_t^I \in S^I} s_t^I / k_i$ and dynamically set the size s_d of the proposed item-based filter and determines whether to exclude an item v from the general item candidates $\mathcal{R}_u^T$ by checking if the score between the item v and the positive sample is less than s_d. The dynamic calculation of s_d is described in Sec.4.6.

Following the classical hard negative sampling works [13, 31, 44], RealHNS ranks the general filtered item candidates $\hat{\mathcal{R}}_u^T$ and oversamples the high-ranked items. As shown in the left part of Fig.2, RealHNS indicates the hardness for each item with the brightness of purple, where the closer the item is to the final user representation (brighter), the more likely it is to be considered as the HNS. Given the source/target behavior matrix $\boldsymbol{H}^S$ and $\boldsymbol{H}^T$ of user u, RealHNS sequentially generate the final user representation $\boldsymbol{u}_u^T = \mathrm{MLP}^f(\boldsymbol{h}_u^S \parallel \boldsymbol{h}_u^T)$ with the latest behavior representation $\boldsymbol{h}_u^S$ and $\boldsymbol{h}_u^T$ in source/target domain respectively. Then, RealHNS computes the score between the final user representation $\boldsymbol{u}_u^T$ and the item embeddings in $\hat{\mathcal{R}}_u^T$, and then selects the item in the top-ranked item list as the general real HNS.

Referring to the specific concept of Hypothesis 1, we believe that items which are closer to the positive sample are more likely to be the FHNS. Therefore, the proposed general real HNS selector aids in eliminating items in proximity to the positive sample, which is a tailored solution to the false negative problem in general HNS.

4.5 Cross-domain Real Hard Negative Sample Selector

In order to alleviate the potential false negative problem in cross-domain transfer, RealHNS proposes a cross-domain real hard negative sample selector which employs a cross-domain coarse-grained real HNS selector to sample the cross-domain item candidates related to the user's source-domain preference, designs a cross-domain fine-grained real HNS selector with dynamic user- and item-based filters to eliminate potential false HNS in cross-domain transfer and cross-domain item candidates respectively, and further samples cross-domain real HNS in CDR.

4.5.1 Cross-domain Coarse-grained real HNS Selector. Most of the existing hard negative sampling methods are designed in CF and are difficult to be directly migrated into the CDR scenario. This is mainly due to the fact that the CDR task introduces additional information from the source domain to facilitate the accurate modeling of the user's comprehensive preferences. Therefore, these methods can only avoid the false negative problem in a single domain but fail to solve it in cross-domain positive transfer. RealHNS assumes that users with consistent behavior in the source domain may share similar preferences in the target domain and propose a coarse-grained real HNS selector to precisely model the user's cross-domain preference in the target domain and its related cross-domain item candidates.

Specifically, given the source/target prior behavior matrices $\bar{\boldsymbol{H}}^S$ and $\bar{\boldsymbol{H}}^T$, RealHNS implements the K-means algorithm on $\bar{\boldsymbol{H}}^S$ to generate the source-domain user cluster C^U. After that, RealHNS calculates the target-domain centroidal representation $\boldsymbol{\Phi}^U = [\boldsymbol{\phi}_1^U, \boldsymbol{\phi}_2^U, \cdots, \boldsymbol{\phi}_{k_u}^U] \in \mathbb{R}^{k_u \times d}$ based on the source-domain user cluster C^U and the target prior behavior matrix $\bar{\boldsymbol{H}}^T$ (k_u is the number of source-domain user clusters), in where $\boldsymbol{\phi}_k^U$ is measured as follows:

$$\boldsymbol{\phi}_k^U = \textstyle\sum_{u \in C_k^U} (\bar{\boldsymbol{h}}_{u,q}^T)/\mathrm{len}(C_k^U) \tag{5}$$

where $\bar{\boldsymbol{h}}_{u,q}^T$ denotes the last behavior embedding of user u in the target domain. Then we generate the prior transferred source representation $\bar{\boldsymbol{u}}_u^S = \mathrm{MLP}^f(\bar{\boldsymbol{h}}_{u,p}^S \parallel \boldsymbol{\phi}_k^U)$ in the final space, where k denotes the user cluster index in which u belongs. The subsequent operation is similar to Sec.4.4.2, RealHNS calculates and sorts the cross-domain score $s_u^S = [\bar{\boldsymbol{u}}_u^S \boldsymbol{v}_1^T, \cdots, \bar{\boldsymbol{u}}_u^S \boldsymbol{v}_n^T]$ with $\bar{\boldsymbol{u}}_u^S$ and the learnable item embeddings $\boldsymbol{V}^T$, and samples 1,000 items from the top-range to form the cross-domain item candidates $\mathcal{R}_u^S$. Those are composed of items that related to the user's source domain preferences, then we can accurately model his/her transferred preferences by fine-grained processing.

4.5.2 Cross-domain Fine-grained real HNS Selector. CDR aims to transfer informative knowledge from the source domain to the target domain for performance gains in the target domain. Owing to the disparate domains in which the items belong, there persists an inherent data bias of user preferences in multi-domains, thereby making it challenging to achieve their uniformity across diverse domains. The proposed cross-domain RHNS primarily aims to model the mainstream target domain preference with similar source domain preferences by unsupervised clustering. It functions well in broad terms, while over-optimizing the cross-domain RHNS can actually exacerbate the bias and introduce additional negative information for these users with consistent source- and target-domain preferences. As illustrated in the middle part of Fig.2, certain users that are aggregated into a cluster within the source user space display significant distribution patterns during cross-domain transfer. This is attributed to the dissimilar mapping functions of various users, as there is a profusion of source domain information in the real world, making certain users more susceptible to being dominated by the mainstream transfer (See the red arrows in the middle part of Fig.2).

To do this, we first define the outlier as the **users who exhibit similar source domain preferences but display significantly different preferences in the target domain compared to the mainstream target domain preferences**. Then we propose a dynamic user-based filter to incorporate outliers within the optimization scope of cross-domain RHNS. Given the user cluster index k, we calculate the score list S_k^U between the prior target-domain last behavior matrix $\boldsymbol{U}_k^U \in \mathbb{R}^{len(C_k^U) \times d}$ of user cluster C_k^U and the target-domain centroidal representation $\boldsymbol{\phi}_k^U$, defined as:

$$S_k^U = CalScore(\boldsymbol{U}_k^U, \boldsymbol{\phi}_k^U) = (\boldsymbol{U}_k^U)^\top \boldsymbol{\phi}_k^U \tag{6}$$

We then sort the score list S_k^U, and select $\mathrm{len}(C_k^U) * w_o$ users of the user cluster C_k^U with the lowest scores based on the pre-defined weight w_o as the outliers. Finally, we optimize these outliers with both general RHNS and cross-domain RHNS, while only applying general RHNS optimization to other users, as illustrated in the right part of Fig.2.

In addition to the inherent multi-domain preference bias, there exists an issue of negative transfer during the cross-domain preference modeling, whereby excessive reliance on cross-domain item candidates may introduce bias in the selection of cross-domain RHNS (See Hypothesis 2). To eliminate this bias , RealHNS designs a dynamic item-based filter to filter out parts of the cross-domain item candidates that are excessively similar to the positive sample. The item-based filter is designed based on unsupervised item similarity, and the cross-domain setting does not alter its inter-item relationships. Therefore, we employ the same size of the general item-based filter mentioned in Sec. 4.4.2 to eliminate cross-domain item candidates. Specifically, RealHNS assesses whether to eliminate an item v from the cross-domain item candidates $\mathcal{R}_u^S$ by comparing the score between the item and the positive sample with the size s_d of dynamic item-based filter, which is calculated in Sec.4.4.2.

After that, RealHNS models the transferred source representation $\boldsymbol{u}_u^S = \mathrm{MLP}^f(\boldsymbol{h}_u^S \parallel \boldsymbol{\phi}_k^U)$ with the latest behavior embedding $\boldsymbol{h}_u^S$ in the source domain and the target-domain centroidal representation $\boldsymbol{\phi}_k^U$ (k denotes the source-domain user cluster index in which u belongs). Similarly, RealHNS finally computes the score between $\boldsymbol{u}_u^S$ and the item embeddings in the filtered cross-domain item candidates $\hat{\mathcal{R}}_u^S$, and then selects the fixed number of items in the top-ranked items as the cross-domain hard negative samples.

4.6 Curriculum learning

As demonstrate in Hypothesis 3 and Sec.3, we assume that the inclusion of all HNS at the initial stage of training may result in computational wastage, inferior performances and excessively high gradient magnitudes, which may further hinder the model's convergence towards the global minima. To this end, we leverage a Curriculum learning (CL) scheme to improve the generalization capacity and convergence rate of CDR models. CL is one of a universal training strategy that trains from easier samples to harder samples, which imitates the learning order in human curricula [5, 36].

Precisely, we design two CL tasks, including the optimization-based CL and the filter-based CL. The former CL involves dynamically adjusting the proportion of RHNS in the NS to start with a smoothed objective, enabling the facile discovery of the global minima[47]. Here, the hyper-parameter μ controls the start epoch in CL, ψ is the number of epoch intervals in CL, η denotes the number of extra RHNS, we dynamically set the number n_r of RHNS as follows:

$$n_r = \begin{cases} 0, & e \leq \mu \\ \min(\eta * \lceil \frac{(e-\mu)}{\psi} \rceil, \frac{n_n}{2}), & e > \mu, \end{cases} \tag{7}$$

where e is the current epoch, and n_n denotes the number of NS. The above parameters are identical in each dataset, that is, $\mu = 5$, $\psi = 2$, $\eta = 1$. In contrast, the latter CL task is adopted to dynamically regulate the scope of general and cross-domain item candidates to be filtered to mitigate the inclusion of harder NS (potential FHNS) into the training. With the pre-defined $\chi = 5$ and the $\tau = 1.15$ denote the initial-scale factor and epoch-decay factor respectively, the dynamic size s_d of the proposed item-based filter is defines as:

$$s_d = \begin{cases} +\infty, & e \leq \mu \\ \min(s_c, \frac{s_c}{\chi} * \tau^{\lceil \frac{(e-\mu)}{\psi} \rceil}), & e > \mu, \end{cases} \tag{8}$$

It is worth noting that these two CL tasks share the same pre-defined parameters and the performance of RealHNS are not sensitive to those above parameters, and we further conduct a parameter analysis in Sec.5.7.

4.7 Optimization Objectives

We calculate the predicted probability $\hat{y}^T = \mathrm{CalScore}(\boldsymbol{u}^T, \boldsymbol{v}_{q+1}^T) = (\boldsymbol{u}^T)^\top \boldsymbol{v}_{q+1}^T$ with the final user representation $\boldsymbol{u}^T$ and the target item embedding $\boldsymbol{v}_{q+1}^T$. And then, we formulate the final loss $\mathcal{L}$ as follows:

$$\mathcal{L} = -\sum_{(u,d)\in R^T} \left[y_{u,d}^T \log \hat{y}_{u,d}^T + \left(1 - y_{u,d}^T\right) \log\left(1 - \hat{y}_{u,d}^T\right)\right] \tag{9}$$

where R^T is the target-domain training set, $y_{u,d}^T = 1$ and $y_{u,d}^T = 0$ denote the positive and negative samples respectively, and $\hat{y}_{u,d}^T$ denotes the predicted probability of (u, d).

5 EXPERIMENTS

5.1 Datasets

We conduct extensive experiments on two real-world cross-domain datasets with four domains to verify the effectiveness and universality of RealHNS. We select "Toys and Games" and "Video Games" to generate the **Amazon Toy & Game** dataset, "Books" and "Movies and TV" to form the **Amazon Book & Movie** dataset. Following classical CDR studies [17, 23], we construct users' behavioral sequence in each domain in chronological order and apply the Leave-one-out splitting method [16, 32] (set the last interacted item of each user for testing and the penultimate item for validation). To achieve this, we first select overlapping users who have interacted in both domains, filter them with the three-core setting and treat all the interaction records as positive feedback. The detailed statistics are shown in Table 1.

5.2 Baselines

To demonstrate the effectiveness of RealHNS in CDR, we compare it with the following hard negative sampling methods:

- **NNCF** [28] is a classical negative sampling method with a fixed popularity-based distribution.
- **AugNS** [42] adds uniform noises to the embedding space to smoothly adjust the representations' uniformity.
- **SRNS** [6] proposes an effective and robust hard negative sampling approach with the score-based memory update and variance-based sampling to sample high-quality negative samples.
- **DNS** [44] is one of the most widely-used dynamic hard negative sampling methods which ranks the randomly selected item candidates and selects the top-scoring item as HNS.
- **DNS*** [31] expands the number of item candidates and selection range synchronously based on DNS to accommodate different metrics and reduce the probability of overly hard negative samples.
- **MixGCF** [13] designs the hop mixing technique to synthesize hard negatives by leveraging both the user-item graph structure and GNNs' aggregation process.

5.3 Experimental Settings

We take Adam as our optimizing method and initialize parameters with the Xavier method. The batch size and the dimension of embedding size are set as 64, with a sequence length of 200 for each dataset. Based on detailed experimental analyses on negative samples (e.g., Fig. 1), we choose the negative sample number $n_n = 20$ (10 random and 10 hard HS) for all datasets. For a fair comparison, we set the same number of NS, RHNS, and UNS for all models. RealHNS shares the majority of parameters across all datasets to verify its robustness. We set the number of general and cross-domain HNS as 8 and 2, and define the numbers of item clusters k_i and source-domain user clusters k_u as 100 and 20, respectively. Moreover, we select the top 100 items for the general and cross-domain fine-grained RHNS selectors and the top 30% items for the cross-domain coarse-grained RHNS. According to the distribution of each dataset, we set the coarse-grained filter range of general RHNS as [10%, 20%] in Amazon Game, and [30%, 40%] in the other three datasets. The outlier weight w_o is set as 0.1 for

Table 1: Statistics of four classical CDR settings. Amazon Toy and Book are relatively sparse datasets in CDR.

Dataset	**Amazon Toy & Game**		**Amazon Book & Movie**	
Domain	Toy	Game	Book	Movie
Users	7,996	7,996	28,531	28,531
Items	37,868	11,735	239,042	38,185
Records	114,487	82,871	625,692	349,918
Density	0.0378%	0.0883%	0.0092%	0.0321%

relatively sparse Amazon Toy and Book, and 0.8 for relatively dense Amazon Game and Movie datasets. Sec. 5.7 gives comprehensive parameter analyses for a deeper understanding of RealHNS. Note that all baselines have been sufficiently tuned on different datasets to achieve their optimal performances. We conduct three runs and report the average results for all models.

5.4 Main results

We choose three typical evaluation metrics including NDCG@k (N@k), Hit Rate@k (HR@k), and AUC with different $k = 5, 10, 20, 50$. Following [15, 23], we randomly sample 99 negative items for each positive instance. We highlight the best results in bold and the best baselines with underline. Table.2 shows the results, we have the following observations:

(1) Generally, RealHNS significantly outperforms all baselines on four datasets, with the significance level $p < 0.05$ and the average error range ≤ 0.003. The improvements are larger with smaller N and rank-sensitive metrics (NDCG), which is natural that RealHNS focuses on distinguishing hard NS that is more beneficial in top positions. The superiority is consistent across the four cross-domain settings based on two hard negative sampling methods, indicating that RealHNS is beneficial to various hard negative sampling methods. Moreover, it also demonstrates the necessity of capturing the specific cross-domain informative RHNS to improve CDR. Note that we focus on a more realistic and challenging sampling setting (containing 10 random NS and 10 hard NS, more HNS will lower the impact of sampling). The current consistent improvements ($1\%-5\%$) brought by RealHNS are impressive compared to classical methods.

(2) Based on the challenging setting of considering 20 negative samples, no hard negative sampling baseline can consistently outperform other baselines on all datasets (sometimes even worse than only using random NS). Most existing HNS methods (e.g., DNS, MixGCF) only rely on selecting the hardest items from certain item subsets to alleviate the impact of false HNS, thereby resulting in excessive dependence on the quality of the sampled subset. Hence, they exhibit worse performance on datasets with larger item corpora or sparser user behaviors (e.g., Game→Toy and Movie→Book). The improvements of RealHNS over existing hard negative sampling methods confirm the significance of: (1) the item-based filter for general RHNS to provide the unbiased yet informative gradient to recommender, and (2) explicit cross-domain RHNS that could model the variations among multi-domain preferences and incorporate them into the training process through the curriculum learning scheme. We also conduct ablation studies to verify the effectiveness of different components of RealHNS in Sec. 5.5.

(3) Comparing the improvements among different datasets, we find that RealHNS is more beneficial on Game→Toy and Movie→Book settings. It reflects that the proposed RealHNS functions well on relatively sparser target domains by transferring the informative knowledge from denser source domains (similar to the conventional CDR methods). Meanwhile, it implies the practical usage of the cross-domain RHNS in RealHNS. The relative improvements of RealHNS over MixGCF and DNS* are also significant on all datasets, demonstrating RealHNS's ability to bring further consistency improvements on different base hard negative sampling algorithms. Benefiting from the novel cross-domain RHNS in CDR and the dynamic user- and item-based filter, RealHNS significantly outperforms other negative sampling methods in all datasets. We further conduct a universality analysis on RealHNS with different hard negative sampling methods, other conventional cross-domain recommendation models, and even single-domain methods in Sec. 5.6.

5.5 Ablation Study

In this section, we conduct ablation studies to verify the effectiveness of different components in RealHNS. Here, "GS" and "CS" denote the general RHNS and the cross-domain RHNS respectively, and "UF" denotes the dynamic user-based FHNS filter. Note that DNS*+GS+CS+UF equals RealHNS (DNS*) in Table. 2. Moreover, the effectiveness of the curriculum learning module will be demonstrated through parameter experiments in Sec. 5.7. Table. 3 reveals that:

(1) DNS* does not always outperform UNS across all cross-domain datasets, even with the sufficient tuning of parameters specific to each dataset. It demonstrates the instability of existing hard negative sampling methods in dealing with FHNS (only high-quality HNS are beneficial), which reconfirms the advantage of RealHNS in different datasets.

(2) With the GS, DNS*+GS achieves consistent improvement over DNS* and significantly outperforms UNS. It is primarily due to the dynamic item-based FHNS filter's ability to mitigate the false negative problem in general RHNS, thereby including more informative but moderately challenging samples in the training process.

(3) In general, DNS*+GS+CS (i.e., only considering cross-domain HNS without UF) outperforms DNS*+GS, which indicates the potential power of our proposed cross-domain HNS for alleviating negative transfer in CDR. However, such improvement is not that stable in different datasets and metrics, which implies the importance of FHNS detection.

(4) Comparing DNS*+GS+CS and DNS*+GS+CS+UF on four cross-domain settings, we further demonstrate that the dynamic user-based FHNS filter in cross-domain RHNS is indispensable. It not only helps RealHNS to accurately model the comprehensive preference of users with varying degrees of multi-domain preference differences but also dynamically regulates the hardness of cross-domain RHNS. Sec. 5.7 further gives more analyses on the

Table 2: Results on hard negative sampling methods in CDR. All improvements are significant (p<0.05 with paired t-tests).

Datasets	Algorithms	N@5	N@10	N@20	N@50	HR@5	HR@10	HR@20	HR@50	AUC
Game ↓ Toy	NNCF	0.1991	0.2234	0.2440	0.2742	0.2662	0.3417	0.4235	0.5762	0.5753
	AugNS	0.2079	0.2317	0.2530	0.2832	0.2764	0.3504	0.4348	0.5879	0.5758
	SRNS	0.2076	0.2314	0.2524	0.2827	0.2737	0.3472	0.4308	0.5840	0.5816
	DNS	0.2096	0.2325	0.2537	0.2834	0.2776	0.3487	0.4330	0.5835	0.5780
	DNS*	0.2056	0.2264	0.2481	0.2791	0.2655	0.3303	0.4164	0.5737	0.5691
	MixGCF	0.2093	0.2325	0.2543	0.2840	0.2769	0.3485	0.4349	0.5860	0.5782
	RealHNS(MixGCF)	0.2178	0.2414	0.2620	0.2903	0.2901	0.3632	0.4450	0.5889	0.5867
	RealHNS(DNS*)	**0.2198**	**0.2435**	**0.2636**	**0.2914**	**0.2952**	**0.3686**	**0.4484**	**0.5895**	**0.5904**
	Improvment	4.87%	4.73%	3.66%	2.61%	6.34%	5.19%	3.10%	0.27%	1.51%
Toy ↓ Game	NNCF	0.3469	0.3843	0.4120	0.4390	0.4629	0.5785	0.6881	0.8234	0.7870
	AugNS	0.3464	0.3862	0.4143	0.4402	0.4647	0.5879	0.6988	0.8288	0.7884
	SRNS	0.3545	0.3921	0.4202	0.4467	0.469	0.5852	0.6961	0.8286	0.7926
	DNS	0.3637	0.3995	0.4260	0.4529	0.4784	0.5889	0.6934	0.8285	0.7888
	DNS*	0.3625	0.3992	0.4256	0.4519	0.4778	0.5912	0.6953	0.8270	0.7899
	MixGCF	0.3636	0.4002	0.4261	0.4529	0.4766	0.5893	0.6919	0.8265	0.7877
	RealHNS(MixGCF)	0.3690	0.4044	0.4300	0.4559	**0.4891**	**0.5983**	0.6995	**0.8293**	0.7919
	RealHNS(DNS*)	**0.3697**	**0.4047**	**0.4309**	**0.4565**	0.4883	0.5965	**0.6999**	**0.8283**	**0.7924**
	Improvment	1.65%	1.12%	1.13%	0.79%	2.24%	1.20%	0.16%	0.06%	-0.03%
Movie ↓ Book	NNCF	0.2722	0.3044	0.3344	0.3744	0.3583	0.4580	0.5769	0.7791	0.7403
	AugNS	0.2918	0.3228	0.3516	0.3902	0.3848	0.4808	0.5950	0.7907	0.7515
	SRNS	0.3167	0.3487	0.3772	0.4136	0.4055	0.5048	0.6178	0.8020	0.7637
	DNS	0.3398	0.3682	0.3939	0.4297	0.4276	0.5155	0.6175	0.7990	0.7655
	DNS*	0.3402	0.3690	0.3951	0.4308	0.4278	0.5172	0.6205	0.8015	0.7675
	MixGCF	0.3293	0.3588	0.3856	0.4220	0.4168	0.5085	0.6149	0.7991	0.7639
	RealHNS(MixGCF)	0.3550	0.3843	0.4097	0.4433	0.4488	0.5395	0.6405	0.8105	0.7785
	RealHNS(DNS*)	**0.3584**	**0.3874**	**0.4130**	**0.4464**	**0.4527**	**0.5423**	**0.6435**	**0.8131**	**0.7807**
	Improvment	5.35%	4.99%	4.53%	3.62%	5.82%	4.85%	3.71%	1.38%	1.72%
Book ↓ Movie	NNCF	0.4458	0.4798	0.5062	0.5323	0.5606	0.6661	0.7704	0.9017	0.8592
	AugNS	0.4368	0.4738	0.5003	0.5247	0.5707	0.6849	0.7896	0.9117	0.8677
	SRNS	0.4689	0.5031	0.5282	0.5519	0.5901	0.6956	0.7944	0.9136	0.8727
	DNS	0.4729	0.5052	0.5298	0.5536	0.5888	0.6883	0.7854	0.9053	0.8668
	DNS*	0.4719	0.5049	0.5290	0.5530	0.5869	0.6888	0.7838	0.9047	0.8661
	MixGCF	0.4715	0.5044	0.5290	0.5528	0.5892	0.6910	0.7878	0.9078	0.8684
	RealHNS(MixGCF)	0.4752	0.5094	0.5335	0.5565	**0.5965**	**0.7017**	**0.7967**	**0.9123**	**0.8732**
	RealHNS(DNS*)	**0.4766**	**0.5105**	**0.5345**	**0.5577**	0.5954	0.7001	0.7950	0.9116	0.8720
	Improvment	0.78%	1.05%	0.89%	0.74%	1.08%	0.88%	0.29%	-0.14%	0.06%

quantitative effects of other components (e.g., the coarse-grained and fine-grained parts) related to hyper-parameters.

5.6 Universality of RealHNS

5.6.1 Universality Analyses on Different Hard Negative Sampling Methods. We evaluate RealHNS's universality on DNS [44] and MixGCF [13] besides DNS* on four cross-domain settings. Fig. 3 shows the results, and we observe that:

(1) RealHNS achieves consistent and significant improvements over the base DNS and MixGCF on all datasets and metrics, which confirms its universality on different representative hard negative sampling methods.

(2) In comparison to DNS and MixGCF, RealHNS accomplishes incremental performance improvement of each component, that is, both the general RHNS and cross-domain RHNS achieve a consistent enhancement relative to the basic negative sampling methods. This highlights the practical application and significance of the two types of RHNS.

(3) The relative improvements of RealHNS on MixGCF are more significant than those on DNS. This may be attributed to the fact that MixGCF could optimize the distribution of synthetic negative samples in the feature space through interpolation from positive samples. Consequently, RealHNS on MixGCF can benefit more from informative NS.

5.6.2 Universality Analyses on Different Recommendation Models. To verify the universality of RealHNS in different recommendation scenarios, we adopt RealHNS with (a) a classical CDR model DASL [17], (b) a classical CF-based model BPR-MF [29], and (c) a classical single-domain sequential model SASRec [15] on the Amazon Game (only the general RHNS selector is used for single-domain tasks). For a fair comparison, we fix the number of negative samples and hard negative samples to 20 and 10 for all models. Based on Table.4, we have the following observations:

(1) RealHNS performs better than the classical HNS method DNS in all metrics of both single-domain and cross-domain tasks, which demonstrates the universality of RealHNS in real-world recommendation tasks. It is impressive that our proposed general RHNS selector is effective even for single-domain recommendation models.

(2) RealHNS outperforms UNS largely on all NDCG metrics and Hit Rate metrics with smaller k. It is natural since hard negative

Table 3: Results on ablation study of RealHNS(DNS*). Generally, all components are effective.

Datasets	Algorithms	N@5	N@10	N@20	N@50	HR@5	HR@10	HR@20	HR@50	AUC
Game ↓ Toy	UNS	0.2084	0.2327	0.2536	0.2837	0.2776	0.3525	0.4358	0.5884	0.5829
	DNS*	0.2016	0.2241	0.2454	0.2769	0.2622	0.3322	0.4169	0.5769	0.5686
	DNS*+GS	0.2158	0.2399	0.2600	0.2886	0.2919	0.3664	0.4464	**0.5912**	0.5900
	DNS*+GS+CS	0.2168	0.2402	0.2609	0.2895	0.2859	0.3585	0.4409	0.5854	0.5877
	DNS*+GS+CS+UF	**0.2198**	**0.2435**	**0.2636**	**0.2914**	**0.2952**	**0.3686**	**0.4484**	0.5895	**0.5904**
Toy ↓ Game	UNS	0.3557	0.3926	0.4212	0.4474	0.4726	0.5867	0.7000	0.8307	0.7944
	DNS*	0.3625	0.3992	0.4256	0.4519	0.4778	0.5912	0.6953	0.8270	0.7899
	DNS*+GS	0.3638	0.4004	0.4262	0.452	0.4845	**0.5974**	0.6992	0.8291	0.7913
	DNS*+GS+CS	0.3674	0.404	0.4304	0.4561	0.4828	0.5961	**0.7004**	**0.8294**	**0.7924**
	DNS*+GS+CS+UF	**0.3697**	**0.4047**	**0.4309**	**0.4565**	**0.4883**	0.5965	0.6999	0.8283	**0.7924**
Movie ↓ Book	UNS	0.3471	0.3771	0.4030	0.4378	0.4385	0.5315	0.6343	0.8101	0.7754
	DNS*	0.3402	0.3690	0.3951	0.4308	0.4278	0.5172	0.6205	0.8015	0.7675
	DNS*+GS	0.3553	0.3845	0.4099	0.4437	0.4483	0.5388	0.6396	0.8109	0.7784
	DNS*+GS+CS	0.3575	0.3861	0.4115	0.4453	0.4503	0.5388	0.6398	0.8109	0.7786
	DNS*+GS+CS+UF	**0.3584**	**0.3874**	**0.4130**	**0.4464**	**0.4527**	**0.5423**	**0.6435**	**0.8131**	**0.7807**
Book ↓ Movie	UNS	0.4729	0.5060	0.5304	0.5541	0.5928	0.6950	0.7914	0.9103	0.8710
	DNS*	0.4719	0.5049	0.5290	0.5530	0.5869	0.6888	0.7838	0.9047	0.8661
	DNS*+GS	0.4759	0.5096	0.5336	0.5567	**0.6000**	**0.7041**	**0.7991**	**0.9150**	**0.8750**
	DNS*+GS+CS	0.4763	0.5103	**0.5345**	**0.5577**	0.5940	0.6990	0.7945	0.9109	0.8718
	DNS*+GS+CS+UF	**0.4766**	**0.5105**	**0.5345**	**0.5577**	0.5954	0.7001	0.7950	0.9116	0.8720

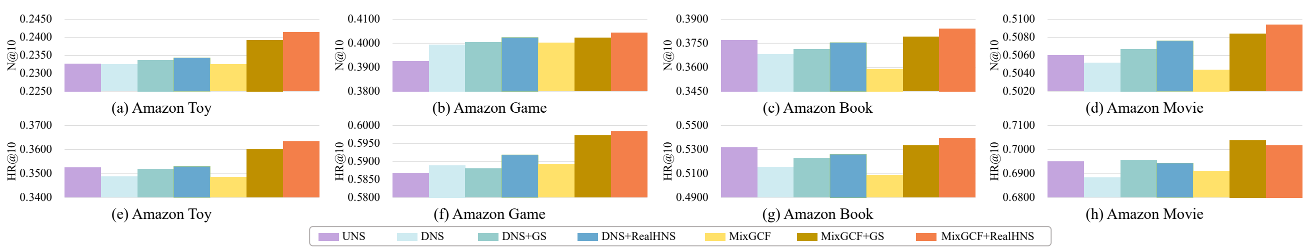

Figure 3: Universality analyses on RealHNS. We show the results of different versions of RealHNS on DNS and MixGCF.

Table 4: Results on universality analyses of RealHNS in CD, SR and CDR.

Scenarios	Algorithms	N@5	N@10	N@20	N@50	HR@5	HR@10	HR@20	HR@50	AUC
CF	BPR-MF(UNS)	0.2436	0.2843	0.3175	0.3552	0.3457	0.4715	0.6034	**0.7924**	**0.7507**
	BPR-MF(DNS)	0.2713	0.3069	0.3359	0.3720	0.3722	0.4825	0.5974	0.7799	0.7428
	BPR-MF(RealHNS)	**0.2762**	**0.3129**	**0.3423**	**0.3772**	**0.3832**	**0.4964**	**0.6130**	0.7888	0.7506
SR	SASRec(UNS)	0.3465	0.3841	0.4106	0.4391	0.4627	0.5789	**0.6840**	**0.8260**	**0.7871**
	SASRec(DNS)	0.3538	0.3860	0.4117	0.4408	0.4573	0.5563	0.6583	0.8047	0.7694
	SASRec(RealHNS)	**0.3617**	**0.3974**	**0.4221**	**0.4502**	**0.4719**	**0.5821**	0.6799	0.8207	0.7832
CDR	DASL(UNS)	0.3540	0.3918	0.4201	0.4465	0.4708	0.5877	0.6995	**0.8309**	**0.7948**
	DASL(DNS)	0.3612	0.3982	0.4254	0.4516	0.4766	0.5911	0.6985	0.8295	0.7919
	DASL(RealHNS)	**0.3664**	**0.4030**	**0.4289**	**0.4545**	**0.4852**	**0.5982**	**0.7009**	0.8287	0.7929

sampling is specially designed for distinguishing positive from negative among top-ranked items.

5.7 Comprehensive Parameter Analyses for Better FHNS Understanding

In order to demonstrate the robustness of RealHNS, we conducted four groups of parameter analyses in this section. Note that the odd-numbered rows in the Fig. 4 to 7 indicate experiments on Toy→Game, and the even-numbered rows indicate experiments on Game→Toy. The primary discovery is that ***RealHNS is generally insensitive to most of its hyper-parameters*** *except for the coarse-fined filter range and the number of HNS (which are essential for all recommendation models).* The idealized performance can be obtained by empirically selecting reasonable values of these parameters. Hence, ***developers can get relatively good improvements without trivial parameter selections***.

Impact of the filter range in single- and cross-domain. To verify the impact of the filter range in single- and cross-domain, we conduct experiments on the coarse-grained filter range and the fine-grained filter range. Fig. 4 shows the results of RealHNS (solid line) and UNS (dotted line) on both Toy→Game and Game→Toy settings. We observe that: (1) The optimal selection for the coarse-grained filter range of the general RHNS differs across datasets with different sparsity, and its impact on RealHNS is substantial. It is

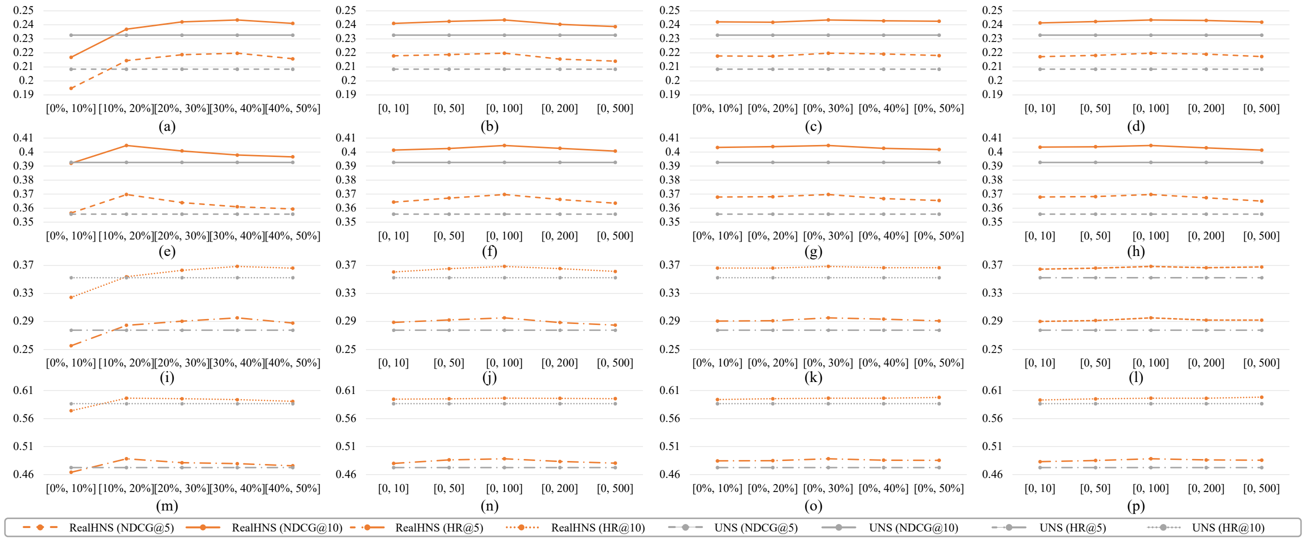

Figure 4: Parameter analyses on filter range in single- and cross-domain.

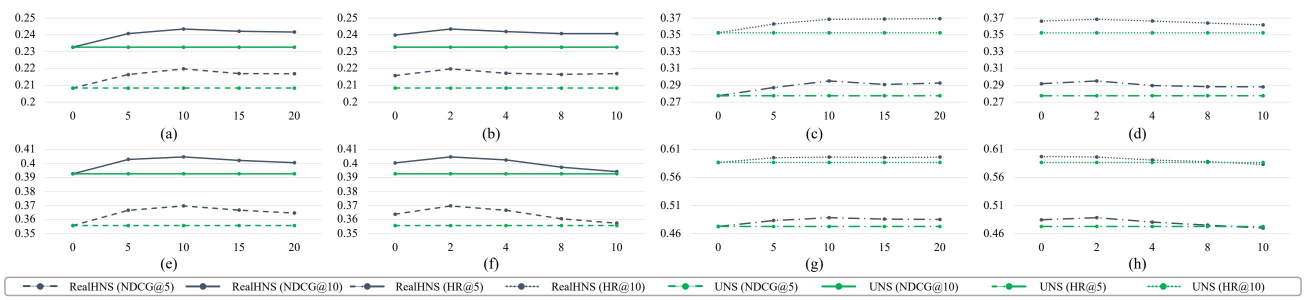

Figure 5: Parameter analyses on the number of RHNS and cross-domain RHNS.

worth noting that RealHNS shares the same coarse-grained filter range in Toy, Book, and Movie due to their lower density; (2) RHNS is insensitive to the fine-grained filter range of general RHNS and the fine-grained filter range of both the general and cross-domain RHNS, and we set the identical parameters across the four datasets to ensure the simplicity and practical deployability.

Impact of the number of NS and cross-domain RHNS. Building on Sec. 3, we also conduct experiments to analyze the impact of the number of RHNS and cross-domain RHNS on RealHNS in Fig. 5. Note that we continue the setting in Sec. 5.3 to set the ratio between the number of general RHNS and the number of cross-domain RHNS as 4:1 when experimenting with the number of RHNS, and set the number of RHNS as 10 when analyzing the number of cross-domain RHNS. The observations are as follows: (1) RHNS with various numbers can bring more significant improvements to RealHNS than the case where the number of RHNS is 0 (equals UNS), which further highlights the effectiveness of the proposed general and cross-domain RHNS; (2) Comparing the performance trend between Fig.5(b) and Fig.5(f), we further demonstrate that the performance on Game deteriorates more rapidly as the number of cross-domain RHNS increases, indicating that the negative transfer is inconsistent across different cross-domain settings.

Impact of the initial scale χ and the reduction rate τ of the item-based filter. We further conduct an experiment to investigate the influence of the initial scale χ and the reduction rate τ of the proposed item-based filter. The results are shown in Fig.6. As shown in Fig.6(b) and Fig.6(d), we find that excessively small τ may cause RealHNS to maintain the initial scale χ as the scale of item-based filter throughout the training process, making it difficult to effectively filter out False NS in the selected items. Conversely, it can be observed that reasonable values of chi can achieve a significant improvement relative to $chi = 1.01$, which proves the practical application of the proposed item-based filter.

Impact of the hardness of the source-domain transfer preference. To analyze the impact of the hardness of the source-domain transfer preference, we vary the weight w_o of outliers in [0, 0.1, 0.5, 0.8, 1] and the cluster number k_u of users in [5, 10, 20, 50, 100]. Note that $w_o = 0$ equals DNS*+GS, and $w_o = 1$ equals DNS*+GS+CS. Fig.7 shows the results, and we can observe that: (1) According to the behavior distribution, the weight w_o of outlier optimization

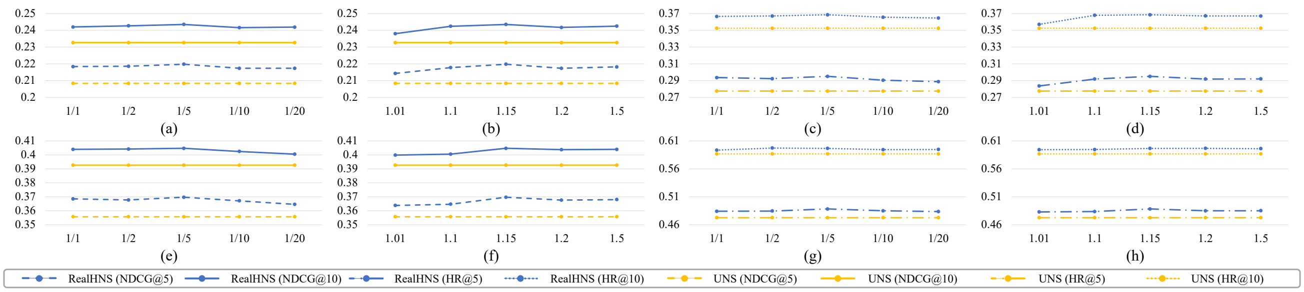

Figure 6: Parameter analyses on the initial scale χ and the reduction rate τ of the proposed filter.

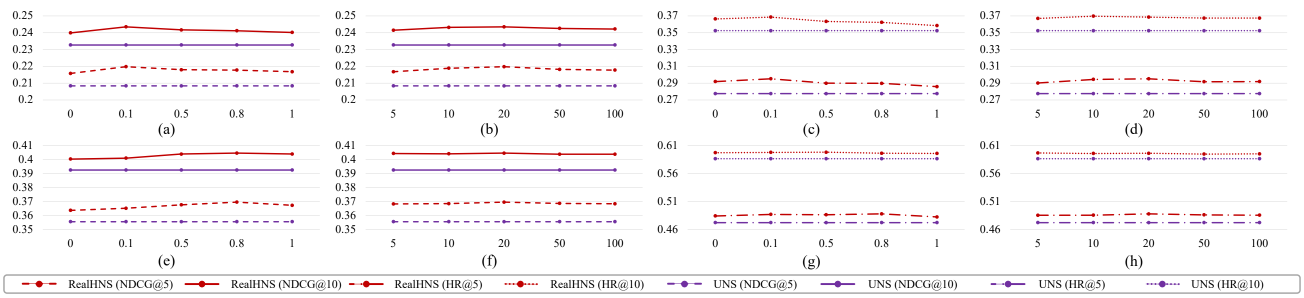

Figure 7: Parameter analyses on the weight w_o of outlier and the cluster number k_u of user.

differs for different datasets. $w_o = 0.1/0.8$ achieves the best performance on the Game→Toy and Toy→Game settings; (2) RealHNS is insensitive to the cluster number k_u of user, and we set k_u as 20 for the purpose of computational cost.

6 CONCLUSION

In this work, we propose a simple, effective, and model-agnostic Real Hard Negative Sampling framework (RealHNS). Instead of explicitly modeling the negative samples, RealHNS alleviates the false negative problem inherent in existing hard negative sampling methods with a novel dynamic item-based filter in the curriculum learning framework. Moreover, RealHNS introduces a unique cross-domain HNS that achieves accurate modeling of users' multi-domain preferences by filtering out potential noise in cross-domain transfer. We conduct extensive experiments to verify the effectiveness and universality of RealHNS and its components on various datasets. In the future, we plan to explore the occasional HNS with the users' multi-behavior and items' heterogeneous information in different recommendation scenarios.

ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China (Grant no. 62006141), the Excellent Youth Scholars Program of Shandong Province (Grant no. 2022HWYQ-048), the TaiShan Scholars Program (Grant no. tsqn202211289). ChatGPT and Grammarly were utilized to improve grammar and correct spelling.

REFERENCES

[1] Yinqiong Cai, Jiafeng Guo, Yixing Fan, Qingyao Ai, Ruqing Zhang, and Xueqi Cheng. 2022. Hard Negatives or False Negatives: Correcting Pooling Bias in Training Neural Ranking Models. In *Proceedings of CIKM.*

[2] Jiangxia Cao, Xin Cong, Jiawei Sheng, Tingwen Liu, and Bin Wang. 2022. Contrastive Cross-Domain Sequential Recommendation. In *Proceedings of CIKM.*

[3] Jiangxia Cao, Jiawei Sheng, Xin Cong, Tingwen Liu, and Bin Wang. 2022. Cross-domain recommendation to cold-start users via variational information bottleneck. In *Proceedings of ICDE.*

[4] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient neural matrix factorization without sampling for recommendation. *TOIS* (2020).

[5] Hong Chen, Yudong Chen, Xin Wang, Ruobing Xie, Rui Wang, Feng Xia, and Wenwu Zhu. 2021. Curriculum disentangled recommendation with noisy multi-feedback. *Proceedings of NeuIPS* (2021).

[6] Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and robustify negative sampling for implicit collaborative filtering. *Proceedings of NeuIPS.*

[7] Wenqi Fan, Xiaorui Liu, Wei Jin, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2022. Graph trend filtering networks for recommendation. In *Proceedings of SIGIR.*

[8] Luyu Gao and Jamie Callan. 2022. Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval. In *Proceedings of ACL.*

[9] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *Proceedings of IJCAI* (2017).

[10] Lei Guo, Li Tang, Tong Chen, Lei Zhu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2021. DA-GCN: a domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation. In *Proceedings of IJCAI.*

[11] Xiaobo Hao, Yudan Liu, Ruobing Xie, Kaikai Ge, Linyao Tang, Xu Zhang, and Leyu Lin. 2021. Adversarial feature translation for multi-domain recommendation. In *Proceedings of KDD.*

[12] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of KDD.*

[13] Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. 2021. Mixgcf: An improved training method for graph neural network-based recommender systems. In *Proceedings of KDD.*

[14] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. *Proceedings of NeurIPS* (2020).

[15] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of ICDM.*

[16] 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 WSDM.*

[17] Pan Li, Zhichao Jiang, Maofei Que, Yao Hu, and Alexander Tuzhilin. 2021. Dual Attentive Sequential Learning for Cross-Domain Click-Through Rate Prediction. In *Proceedings of KDD*.

[18] Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2019. Dice loss for data-imbalanced NLP tasks. *Proceedings of ACL* (2019).

[19] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-aware sequential location recommendation. In *Proceedings of KDD*.

[20] Weiming Liu, Jiajie Su, Chaochao Chen, and Xiaolin Zheng. 2021. Leveraging distribution alignment via stein path for cross-domain cold-start recommendation. *Proceedings of NeurIPS*.

[21] Haokai Ma, Xiangxian Li, Lei Meng, and Xiangxu Meng. 2021. Comparative study of adversarial training methods for cold-start recommendation. In *Proceedings of ADVM*.

[22] Haokai Ma, Zhuang Qi, Xinxin Dong, Xiangxian Li, Yuze Zheng, and Xiangxu Mengand Lei Meng. 2023. Cross-Modal Content Inference and Feature Enrichment for Cold-Start Recommendation. *Proceedings of IJCNN* (2023).

[23] Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Jie Zhou. 2023. Triple Sequence Learning for Cross-domain Recommendation. *arXiv preprint arXiv:2304.05027* (2023).

[24] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-domain recommendation: An embedding and mapping approach.. In *Proceedings of IJCAI*.

[25] Lei Meng, Fuli Feng, Xiangnan He, Xiaoyan Gao, and Tat-Seng Chua. 2020. Heterogeneous fusion of semantic and collaborative information for visually-aware food recommendation. In *Proceedings of MM*.

[26] Gustavo Penha and Claudia Hauff. 2023. Do the Findings of Document and Passage Retrieval Generalize to the Retrieval of Responses for Dialogues?. In *ECIR*.

[27] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of NAACL*.

[28] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of WSDM*.

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of UAI*.

[30] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. Fast context-aware recommendations with factorization machines. In *Proceedings of SIGIR*.

[31] Wentao Shi, Jiawei Chen, Fuli Feng, Jizhi Zhang, Junkang Wu, Chongming Gao, and Xiangnan He. 2023. On the Theories Behind Hard Negative Sampling for Recommendation. *arXiv* (2023).

[32] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of CIKM*.

[33] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. 2020. Equalization loss for long-tailed object recognition. In *Proceedings of CVPR*.

[34] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *Proceedings of KDD*.

[35] Tianxin Wang, Fuzhen Zhuang, Zhiqiang Zhang, Daixin Wang, Jun Zhou, and Qing He. 2021. Low-dimensional Alignment for Cross-Domain Recommendation. In *Proceedings of CIKM*.

[36] Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. *Proceedings of TPAMI* (2021).

[37] Taiqiang Wu, Xingyu Bai, and Yujiu Yang. 2022. Prompt-based model for acronym disambiguation via negative sampling. (2022).

[38] Yiqing Wu, Ruobing Xie, Yongchun Zhu, Xiang Ao, Xin Chen, Xu Zhang, Fuzhen Zhuang, Leyu Lin, and Qing He. 2022. Multi-view multi-behavior contrastive learning in recommendation. In *Proceedings of DASFAA*.

[39] Yiqing Wu, Ruobing Xie, Yongchun Zhu, Fuzhen Zhuang, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized prompts for sequential recommendation. *arXiv preprint arXiv:2205.09666* (2022).

[40] Ruobing Xie, Cheng Ling, Yalong Wang, Rui Wang, Feng Xia, and Leyu Lin. 2020. Deep feedback network for recommendation. In *Proceedings of IJCAI*.

[41] Ruobing Xie, Qi Liu, Liangdong Wang, Shukai Liu, Bo Zhang, and Leyu Lin. 2022. Contrastive cross-domain recommendation in matching. In *Proceedings of KDD*.

[42] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are Graph Augmentations Necessary? Simple Graph Contrastive Learning for Recommendation. In *Proceedings of SIGIR*.

[43] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. 2020. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of CVPR*.

[44] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Proceedings of SIGIR*.

[45] Yanzhao Zhang, Richong Zhang, Samuel Mensah, Xudong Liu, and Yongyi Mao. 2022. Unsupervised sentence representation via contrastive learning with mixing negatives. In *Proceedings of AAAI*.

[46] Xiaolin Zheng, Jiajie Su, Weiming Liu, and Chaochao Chen. 2022. DDGHM: Dual Dynamic Graph with Hybrid Metric Training for Cross-Domain Sequential Recommendation. In *Proceedings of MM*.

[47] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling user interest and conformity for recommendation with causal embedding. In *Proceedings of WWW*.

[48] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of KDD*.

[49] 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 CIKM*.

[50] 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 CIKM*.

[51] Yao Zhou, Jianpeng Xu, Jun Wu, Zeinab Taghavi, Evren Korpeoglu, Kannan Achan, and Jingrui He. 2021. PURE: Positive-unlabeled recommendation with generative adversarial network. In *Proceedings of KDD*.

[52] Feng Zhu, Chaochao Chen, Yan Wang, Guanfeng Liu, and Xiaolin Zheng. 2019. Dtcdr: A framework for dual-target cross-domain recommendation. In *Proceedings of CIKM*.

[53] 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 WSDM*.