Co-training Disentangled Domain Adaptation Network for Leveraging Popularity Bias in Recommenders

Zhihong Chen
jhon.czh@alibaba-inc.com
Alibaba Group, Hangzhou, China

Jiawei Wu
wujiawei@alibaba-inc.com
Alibaba Group, Hangzhou, China

Chenliang Li*
cllee@whu.edu.cn
School of Cyber Science and Engineering, Wuhan University
Wuhan, China

Jingxu Chen
xianlu@alibaba-inc.com
Alibaba Group, Hangzhou, China

Rong Xiao
xiaorong.xr@alibaba-inc.com
Alibaba Group, Hangzhou, China

Binqiang Zhao
binqiang.zhao@alibaba-inc.com
Alibaba Group, Hangzhou, China

ABSTRACT

Recommender system usually faces popularity bias. From the *popularity distribution shift* perspective, the normal paradigm trained on exposed items (most are hot items) identifies that recommending popular items more frequently can achieve lower loss, thus injecting popularity information into item property embedding, *e.g.,* id embedding. From the *long-tail distribution shift* perspective, the sparse interactions of long-tail items lead to insufficient learning of them. The resultant distribution discrepancy between hot and long-tail items would not only inherit the bias, but also amplify the bias. Existing work addresses this issue with inverse propensity scoring (IPS) or causal embeddings. However, we argue that not all popularity biases mean bad effects, *i.e.,* some items show higher popularity due to better quality or conform to current trends, which deserve more recommendations. Blindly seeking unbiased learning may inhibit high-quality or fashionable items. To make better use of the popularity bias, we propose a co-training disentangled domain adaptation network (CD^{2}AN), which can co-train both biased and unbiased models. Specifically, for popularity distribution shift, CD^{2}AN disentangles item property representation and popularity representation from item property embedding. For long-tail distribution shift, we introduce additional unexposed items (most are long-tail items) to align the distribution of hot and long-tail item property representations. Further, from the instances perspective, we carefully design the item similarity regularization to learn comprehensive item representation, which encourages item pairs with more effective co-occurrences patterns to have more similar item property representations. Based on offline evaluations and online A/B tests, we show that CD^{2}AN outperforms the existing debiased solutions. Currently, CD^{2}AN has been successfully deployed at Mobile Taobao App and handling major online traffic.

*Chenliang Li is the corresponding author.

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531952

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommender Systems, Popularity Bias, Domain Adaptation

ACM Reference Format:
Zhihong Chen, Jiawei Wu, Chenliang Li, Jingxu Chen, Rong Xiao, and Binqiang Zhao. 2022. Co-training Disentangled Domain Adaptation Network for Leveraging Popularity Bias in Recommenders. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477495.3531952

1 INTRODUCTION

Personalized recommendation plays an increasingly important role in a wide range of online applications, such as e-commerce, content-sharing platform, and search engines. However, the recommendation system is facing the popularity bias issue, which stands on the opposite of personalization. From the perspective of domain adaptation [4, 5, 8], we believe that there are two main reasons for this problem: (1) The current training paradigm, *i.e.,* pointwise or pairwise loss, encourages the model to recommend popular items more frequently to achieve lower loss, so the popularity information is injected into the item property embedding, *e.g.,* id embedding, to update the parameters in this direction. This phenomenon is prone to amplify the bias [36] by exacerbating the *popularity distribution shift* between hot item representation and long-tail item representation. As shown in Figure 1, even for positive samples, the model tends to score popular items high for every user. (2) Due to data sparsity of long-tail items on the exposed interactions, it is unable for them to obtain good representations [9]. This *long-tail distribution shift* means that it is difficult for these models to accurately predict long-tail items (hit ratio in Figure 1), because they always overfit hot items.

Unfortunately, this popularity bias will cause Matthew Effect [30], which means that popular items will be recommended more, while long-tail items are increasingly under-recommended, especially those that are new arrivals. Existing work eliminates this bias effect with inverse propensity scoring (IPS) or causal embeddings. (1) IPS, which employs balanced training by re-weighting the interactions [14, 33]. Although theoretically sound, IPS hardly works

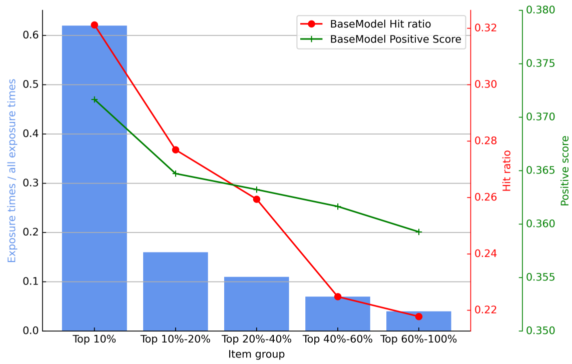


Figure 1: A description of the popularity bias in Taobao industrial dataset. The items are organized into groups w.r.t. click frequency. The histogram indicates the proportion of the exposed times of the item group to the all exposed times, the green line represents the average score of the positive item in the item group (popularity shift), and the red line indicates the hit ratio of the item group (long-tail shift).

well in practice due to its high sensitivity to the weighting strategy. (2) Causal embeddings, which formulate a causal graph to describe the important cause-effect relations in the recommendation process [36, 41]. As shown in Figure 2a, to avoid the popularity information being injected into item property embeddings, these methods generally cut off the path P $\longrightarrow$ I via do-calculus. However, such causal intervention only alleviates the popularity distribution shift, but ignores the insufficient learning of long-tail item representations caused by the long-tail distribution shift.

The recommendation system should make reasonable use of popularity bias, rather than eliminate it [41]. Compared with long-tail items, popular items are often of better quality or more in line with the current trend, so they deserve more recommendations. Blindly eliminating popularity bias may lose important information reflected in the distribution of interaction data, leading to suppression of high-quality or fashionable items. Therefore, this work aims to explore how to eliminate the bias amplification caused by popularity distribution shift and long-tail distribution shift in training, and further realize the co-training of biased and unbiased models.

In this work, we focus on the matching stage, since it is the fundamental part of the system and also the bottleneck to eliminate the popularity bias. To effectively leverage the popularity bias instead of eliminating it, we design a co-training disentangled domain adaptation network, which can train biased and unbiased models meantime. The key challenge is how to alleviate popularity and long-tail distribution shifts to obtain an unbiased and comprehensive item representations. Specifically, for the popularity shift, as shown in Figure 2b, we design a feature disentangling module with orthogonal regularization and popularity similarity regularization to separate the popularity representation and the item property representation from the item property embedding injected with the popularity information. For the long-tail shift caused by data sparsity, we additionally introduce unexposed items to realize the unsupervised distribution alignment of popular and long-tail item property representations. Further, encouraged by item-to-item based collaborative filtering [28, 32], we impose an

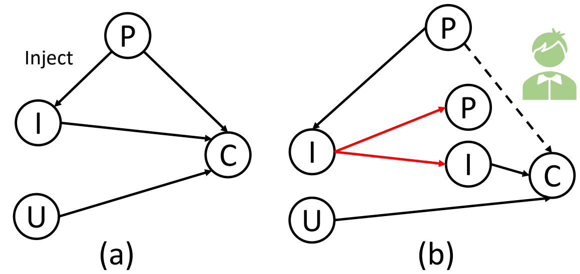


Figure 2: Causal graph for recommendation considering popularity information. (a) In the current training paradigm, the popularity information will not only directly affect the user's decision-making (P $\rightarrow$ C), but also be injected into the item property embedding to amplify the popularity bias (P $\rightarrow$ I). (b) CD^{2}AN, which performs debiasing by disentangling item property representation and popularity representation (read line). U: user property embedding, I: item property embedding, C: interaction probability, P: item popularity.

item similarity regularization to derive the more similar item property representations based on the user's historical interacted items. After obtaining unbiased item property representations, biased and unbiased models can be co-trained by introducing item popularity. When serving online, as shown by the dotted line in Figure 2b, the popularity bias can be reasonably utilized based on the fusion of the scores of two models. In summary, the main contributions of this paper are as follows: (1) To the best of our knowledge, this is the first work to attribute the popularity bias to the popularity distribution shift and long-tail distribution shift from the perspective of domain adaptation. (2) We propose a co-training disentangled domain adaptation network (CD^{2}AN), which can train biased and unbiased models meantime by separating the popularity representation and the item property representation with four regularizations. Furthermore, we carefully design a fusion strategy, so that the model can reasonably utilize the popularity bias for k-nearest neighbors (KNN) retrieval in online service. (3) We conduct extensive experiments on two real datasets and an online A/B test of Taobao, validating the effectiveness and rationality of our analyses and methods.

2 METHODOLOGY

2.1 Problem Formulation

Let z_u represents the user field, x_i represents the target item field, $\{x_{j_t}\}_{t=1}^{T}$ represents the historical interacted item fields of the user, x_k represents the unexposed item field, which randomly selected from the entire item pool, p_i, $\{p_{j_t}\}_{t=1}^{T}$ and p_k represent the popularity field of target item, interacted items of the user and unexposed item, respectively. The core task of the matching stage is to retrieve a small fraction of relevant items from the billion-scale item corpus. To satisfy the online real-time services, embedding-based retrieval (EBR), especially the deep learning network with double tower structure, has been widely adopted [20, 37, 39]. Generally, it can be formulated as follows:

$$s_{z_u,x_i} = v_{z_u} v_{x_i}^T, \tag{1}$$

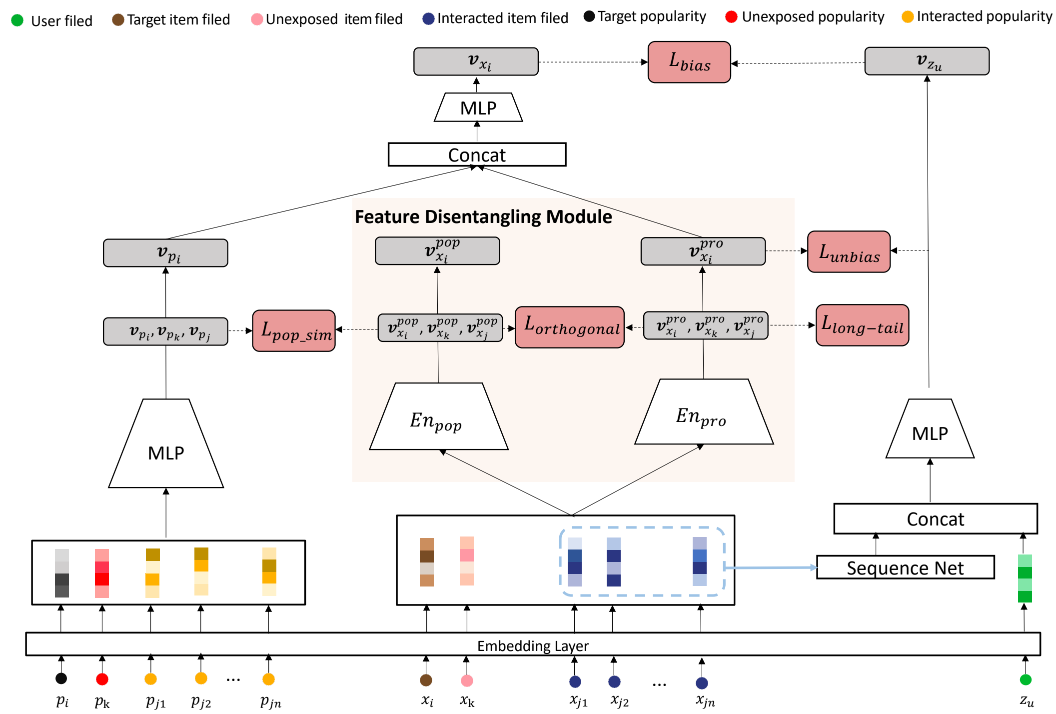


Figure 3: Overview of CD^{2}AN, including user and item towers. In the training stage, the feature disentangling module (FDM), which consists of a popularity encoder (En_{pop}) and a property encoder (En_{pro}), is used to learn to separate item property representation $v_x^{pro} \in \{v_{x_i}^{pro}, v_{x_k}^{pro}, v_{x_j}^{pro}\}$ and disentangled popularity representation $v_x^{pop} \in \{v_{x_i}^{pop}, v_{x_k}^{pop}, v_{x_j}^{pop}\}$. In addition, we introduce interacted items $\{x_{j_t}\}_{t=1}^{T}$ and unexposed items x_k in the item tower to regularize v_x^{pro} for long-tail distribution shift mitigation. CD^{2}AN can co-train an unbiased model based on target item property representation $v_{x_i}^{pro}$, and a biased model based on target popularity item representation v_{x_i} by introducing target real popularity representation v_{p_i} (*Best viewed in color*).

where $v_{z_u} \in \mathcal{R}^{1\times d}$ denotes the user representation extracted via user tower using z_u and $\{x_{j_t}\}_{t=1}^{T}$, $v_{x_i} \in \mathcal{R}^{1\times d}$ denotes the item representation extracted via item tower using x_i and p_i. s_{z_u,x_t} is the relevance score between u and target item x_i. In this sense, the online matching process can be transferred to perform the nearest neighbor (NN) search [23].

Based on this training paradigm, there are two main reasons for the popularity bias. (1) Popularity information is injected into the item property embedding, which increases the vector length of popular items, making inner product models score popular items high for every user [41]. (2) Data sparsity makes long-tail items unable to obtain good representations, so the representations of long-tail items have an inconsistent distribution compared to that of hot items whose records are adequate. Therefore, in this work, we mainly focus on the item tower to extract unbiased item property representations and biased popularity item representations. As shown in Figure 3, we design feature disentangling module (FDM) to separate popularity information from item property embedding. Furthermore, we additionally introduce user interacted items and unexposed items in the item tower to achieve better long-tail item representation learning. We will introduce each component of CD^{2}AN in the following.

2.2 Feature Composition and Embedding Layer

As shown in Figure 3, there are three feature fields in our recommender system: user field z_u; item property field x for target item, interacted items in user historical behaviors, and unexposed item; item popularity field p corresponding to each item. The user field includes many personal features like gender, province, age, etc. Item property field includes item id, item category, brand and so on. Item popularity field includes statistical features that can reflect the popularity of the item, such as exposure frequency, click frequency, and purchase frequency. At first, we transfer continuous features into one-hot schemes through discretization. Then, each one-hot feature is projected into a fixed-size dense embedding. After this, features within a field are concatenated into a field embedding representation. Formally, field embedding representations of user field, item property field, and popularity field can be written as $\boldsymbol{E}(z_u) \in \mathcal{R}^{d_u}$, $\boldsymbol{E}(\boldsymbol{x}) \in \mathcal{R}^{d_x}$, $\boldsymbol{E}(\boldsymbol{p}) \in \mathcal{R}^{d_p}$, respectively.

2.3 Feature Disentangling Module

To suppress the popularity distribution shift, we design FDM to separate item property representation $v_x^{pro} \in \mathcal{R}^{1\times d}$ and disentangled popularity representation $v_x^{pop} \in \mathcal{R}^{1\times d}$. Specifically, a property

Table 1: Distribution shift L_{MMD} calculated with BaseModel.

Domains	MovieLens-1M	Taobao industrial dataset
Hot↔Hot	0.0612	0.1322
Tail↔Tail	0.0537	0.1298
Hot↔Tail	**0.2954**	**0.4735**

encoder En_{pro} learns to capture property information, and a popularity encoder En_{pop} learns to capture popularity information, which can be formulated as:

$$v_x^{pro} = En_{pro}(E(x)), \tag{2}$$

$$v_x^{pop} = En_{pop}(E(x)), \tag{3}$$

where En_{pro} and En_{pop} are composed of multilayer perceptron (MLP), $v_x^* \in \{v_{x_i}^*, v_{x_k}^*, v_{x_j}^*\}$. Since item popularity does not affect item content, we argue that v_x^{pro} and v_x^{pop} are independent with each other. Hence, to ensure that FDM can disentangle in the expected direction, we add discrepancy tasks:

$$L_{disentangled} = L_{pop\_sim} + L_{orthogonal}, \tag{4}$$

$$L_{pop\_sim} = \sum_{b=1}^{B} 1 - \frac{v_{x_b}^{pop} v_{p_b}^T}{\|v_{x_b}^{pop}\| \cdot \|v_{p_b}\|}, \tag{5}$$

$$L_{orthogonal} = \sum_{b=1}^{B} \frac{\|v_{x_b}^{pop} (v_{x_b}^{pro})^T\|^2}{\|v_{x_b}^{pop}\| \cdot \|v_{x_b}^{pro}\|}, \tag{6}$$

where $\|\cdot\|$ is defined as L2-norm, $v_p \in \{v_{p_i}, v_{p_k}, v_{p_j}\} \in \mathcal{R}^{1\times d}$ is defined as the real popularity representation which can be generated by $MLP(E(p))$, and B is the batch size. L_{pop_sim} encourages the module to separate out the popularity information injected into $E(x)$ by constraining v_x^{pop} to be similar to v_p. $L_{orthogonal}$ encourages orthogonality between v_x^{pop} and v_x^{pro}, enabling the popularity and property encoders to encode different aspects of the inputs.

2.4 Regularizations for Long-tail Shift

Domain Alignment. Although item property representations v_x^{pro} are extracted through FDM, we believe that the popularity bias still exists due to data sparsity. Compared with popular items with sufficient interaction, long-tail items cannot be effectively learned. Inspired by domain adaptation, we additionally introduce unexposed items x_k (most are long-tail items) randomly sampled from the entire item pool, and adopt Maximum Mean Discrepancy (MMD) regularization [13] to align the distribution of hot and long-tail items:

$$L_{MMD} = \|\frac{1}{B}\sum_{b=1}^{B}\phi(v_{x_{i_b}}^{pro}) - \frac{1}{B}\sum_{b=1}^{B}\phi(v_{x_{k_b}}^{pro})\|_{\mathcal{H}}^2, \tag{7}$$

where ϕ is the kernel function, $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS). It is worth mentioning that the gradient of the labeled $v_{x_i}^{pro}$ is stopped to avoid the distribution alignment task from affecting the final retrieval accuracy, *i.e.,* stop the gradient of L_{MMD} from flowing back into $v_{x_{i_b}}^{pro}$. Intuitively, L_{MMD} encourages the cluster center of long-tail items to approach the cluster center of hot items. As shown in Table 1, L_{MMD} between different domains is much larger than that between the same domains, which shows that there indeed exists the long-tail distribution shift.

Instance Alignment. Due to the long-tail shift of the exposed data, the learned item representation space cannot well reflect the latent relationship [9]. For example, a hot item may be far away from a long-tail item of the same category, but close to another popular item of different category. We argue that item pairs with more effective co-occurrences should have more similar item representations. Therefore, to allow the model to explicitly capture this structured pattern, in the training stage, we additionally introduce the user's historical interacted items in the item tower. Based on contrastive learning [7, 38], we encourage the items clicked by the same user to be more similar. Specifically, for each user, we treat $\{x_{i_b}, x_{j_t}\}_{t=1}^T$ as positive pairs, and $\{x_{i_b}, x_{i_n}\}_{n=1,n\neq b}^B$ as negative pairs. Note that, negative pairs are collected by permuting the positive target item pair in a training batch. Let $r(v_{x_i}, v_{x_j}) = v_{x_i} v_{x_j}^T / (\|v_{x_i}\| \cdot \|v_{x_j}\|)$, the item similarity regularization can be formulated as:

$$L_{it\_sim} = -\sum_{b=1}^{B}\sum_{t=1}^{T} \lambda(v_{x_{i_b}}^{pro}, v_{x_{j_t}}^{pro}) * \frac{exp(r(v_{x_{i_b}}^{pro}, v_{x_{j_t}}^{pro})/\tau)}{\sum_{n=1}^{B} exp(r(v_{x_{i_n}}^{pro}, v_{x_{j_t}}^{pro})/\tau)} \tag{8}$$

$$\lambda(v_{x_{i_b}}^{pro}, v_{x_{j_t}}^{pro}) = \frac{max(0, r(E(x_{i_b}), E(x_{j_t})))}{log(c_{i_b}) * log(c_{j_t})} \tag{9}$$

where τ is a tunable hyper-parameter for the softmax temperature, c_* is the exposure frequency of the correponding item, and $\{x_{j_t}\}_{t=1}^T$ represents the user's historical behavior sequence in this record $\{z_{u_b}, x_{i_b}\}$. Note that since CD^{2}AN is a matching model, x_{i_b} in each record of the batch is a positive item that the user z_{u_b} clicked. L_{it_sim} w/o $\lambda(v_{x_{i_b}}^{pro}, v_{x_{j_t}}^{pro})$ may have two problems. (1) Users always have multiple interests, which means that the categories of historical interactive items are diverse. Hence, blindly aligning the target positive sample with all items in user behavior will lead to the collapse of the model. (2) Due to the existence of popularity bias, the exposure frequency of popular items is far greater than that of long-tail items. If the popularity factor is ignored, it will introduce the popularity bias again such that the property representations of popularity items will be close to other popular items with different properties. Based on the above analysis, we introduce the weight $\lambda(\cdot)$, which is composed of $max(0, r(E(x_{i_b}), E(x_{j_t})))$ and $log(c_{i_b}) * log(c_{j_t})$. Like DIN [43], $max(0, r(E(x_{i_b}), E(x_{j_t})))$ is used to preserve items with properties similar to the target item in user behavior. $log(c_{i_b}) * log(c_{j_t})$ is used to suppress the influence of hot items. It is worth mentioning that the gradient of $\lambda(\cdot)$ is excluded from the optimization. The above loss function learns a robust item representation space such that similar items are close to each other, and negative examples are pushed farther away.

Based on domain alignment and instance alignment, the loss function adopted to suppress long-tail distribution shift can be formulated as:

$$L_{long-tail} = L_{MMD} + L_{it\_sim}. \tag{10}$$

2.5 User Tower for User Representation

CD^{2}AN focuses on the improvement of item tower, and the study of user tower is beyond the scope of this work. Hence, the user tower used to extract the user representation can be any architecture of the existing two-tower model, which can be formulated as:

$$v_{z_u} = MLP(Cat(SeqNet(\{E(x_{j_t})\}_{t=1}^{T}), E(z_u))), \tag{11}$$

where $Cat(\cdot)$ represents concatenation, $SeqNet$ represents the sequence net, which can employ any existing sequential networks for user behavior, such as Recurrent Neural Network (RNN) [19], transformer-based models [6], or average pooling [10].

2.6 Co-training and Online Service

Co-training. After the unbiased property representation $v_{x_i}^{pro}$ of the target item is extracted from the item tower, we can train the unbiased matching model in terms of inner product $v_{x_i}^{pro} v_{z_u}^{\top}$. However, we argue that not all popularity biases mean bad effects. The reason why an item is popular maybe that it is of good quality or in line with the current trend, so it is worth more recommendation. Blindly removing this bias will prevent the exposure of good items, thus affecting the efficacy of the recommendation system. To make the model better utilize the popularity bias, we introduce the real popularity representation v_{p_i} to extract the biased popularity item representation v_{x_i} containing popularity information, which can be formulated as: $v_{x_i} = MLP(Cat(v_{x_i}^{pro}, v_{p_i}))$. Based on $r(v_{x_i}^{pro}, v_{z_u})$ and $r(v_{x_i}, v_{z_u})$, we co-train unbiased model and biased model to optimize top-k accuracy by adopting the batch softmax loss used in both recommenders [39] and NLP [12]:

$$L_{un\_bias} = -\sum_{b=1}^{B} \frac{exp(r(v_{x_{i_b}}^{pro}, v_{z_{u_b}})/\tau)}{\sum_{n=1}^{B} exp(r(v_{x_{i_n}}^{pro}, v_{z_{u_b}})/\tau)}, \tag{12}$$

$$L_{bias} = -\sum_{b=1}^{B} \frac{exp(r(v_{x_{i_b}}, v_{z_{u_b}})/\tau)}{\sum_{n=1}^{B} exp(r(v_{x_{i_n}}, v_{z_{u_b}})/\tau)}. \tag{13}$$

Note that, since the popularity information in $v_{x_{i_b}}^{pro}$ has been disentangled, the biased model we trained will only inherit the popularity bias without amplifying it. In summary, the total loss of CD^{2}AN can be formulated as:

$$L_{all} = L_{un\_bias} + L_{bias} + \beta_1 L_{disentangled} + \beta_2 L_{long-tail}, \tag{14}$$

where β_1 andβ_2 are hyper-parameters which control the impact of corresponding terms.

Online Service The online matching process can be transferred to perform the nearest neighbor with the inner product. In online retrieval, to consider both unbiased score for user real interest and biased score for user conformity, the item representation online $v_{x_i}^{online}$ and user representation online $v_{z_u}^{online}$ can be generated as:

$$v_{x_i}^{online} = Cat(\alpha * v_{x_i}^{pro}, (1-\alpha) * v_{x_i}), \tag{15}$$

$$v_{z_u}^{online} = Cat(v_{z_u}, v_{z_u}), \tag{16}$$

where $v_{x_i}^{online} \in \mathcal{R}^{1\times 2d}$, $v_{z_u}^{online} \in \mathcal{R}^{1\times 2d}$, and α is the hyper-parameter to control the strength of popularity drift. The inner product between $v_{x_i}^{online}$ and $v_{z_u}^{online}$ is equivalent to the sum of biased and unbiased scores: $\alpha * r(v_{x_i}^{pro}, v_{z_u}) + (1-\alpha) * r(v_{x_i}, v_{z_u})$.

Table 2: Statistics of the two datasets.

Dataset	#Users	#Items	#Interaction
MovieLens-1M	6,040	3,260	998,539
Taobao industrial dataset	603,885,504	1,757,787	251,276,912

3 EXPERIMENTS

In this section, we conduct experiments to evaluate the effectiveness of CD^{2}AN. Specifically, the experiments aim to answer the following research questions. **RQ1:** Does CD^{2}AN outperform existing debiasing methods and achieve the goal of removing the bad effect of popularity bias? **RQ2:** What is the role of each component in CD^{2}AN, and whether they can ensure interpretability to eliminate popularity bias? **RQ3:** How does CD^{2}AN leverage the popularity bias by fusing biased model and unbiased model?

3.1 Experimental Settings

3.1.1 Datasets. We conduct offline experiments on two real-world datasets: MovieLens-1M [1] and an industrial dataset collected from Mobile Taobao App. The statistics of the two datasets are summarized in Table 2. For MovieLens-1M, following the settings of [17], we filter these datasets in the same way that retained only users with at least 20 interactions. More details on this preprocessing for the dataset have been elaborated in [17], so we do not restate here. For the Taobao industrial dataset, we use traffic logs of eight weeks for training, and the samples in the following two days for validation and testing, respectively. In addition to user filed z_u, target item filed x_i, and user history behavior $\{x_{j_t}\}_{t=1}^{T}$ that most methods use, we additionally introduce item popularity filed p and an unexposed sample x_k randomly selected from the entire item pool for each record. The item popularity field in MovieLens-1M is composed of the number of interactions on the item, the percentage of interactions to the total number of interactions, and the average user score, while the item popularity field in the Taobao industrial dataset is composed of exposure frequency, click frequency, purchase frequency, etc.

3.1.2 Comparison Methods. We compare CD^{2}AN with the following state-of-the-art unbiased learning method. (1) **BaseModel**: Since this work focuses on the matching stage, the basic model is the same two-tower architecture as Youtube DNN [10], in which the sequence net for user history behavior adopts the transformer-based module [6]. (2) **IPS [22]**: IPS eliminates popularity bias by adding the standard inverse propensity weight to re-weight each instance according to item popularity. (3) **ESAM [9]**: To improve long-tail performance, ESAM designs the attribute correlation alignment to achieve the distribution alignment between long-tail and hot items. (4) **DICE [42]**: This is a method for learning causal embeddings to handle the popularity bias. Specifically, it proposes a general framework to disentangle interest and conformity. (5) **PDA [41]**: To model and leverage popularity bias, PDA performs deconfounded training with do-calculus and causally intervenes the popularity bias during inference. (6) **MACR [36]**: MACR formulates a causal graph for popularity bias and proposes a framework that trains the

[1] https://grouplens.org/datasets/movielens/1m/

Table 3: Overall performance on Movielens-1M dataset and the Taobao industrial dataset. "Hot" represents the hot test set, *i.e.,* $x_i \in \mathcal{I}_{hot}$, "Long-tail" represents the long-tail test set, *i.e.,* $x_i \in \mathcal{I}_{long}$, and "Entire" represents the whole test set. The best results are highlighted in boldface and best results in comparison methods are underlined. The improvements over comparison methods are statistically significance at 0.05 level.

Method	MovieLens-1M							Taobao industrial Dataset						
	HR@20			NDCG@20			C-Ratio	HR@300			NDCG@300			C-Ratio
	Hot	Long-tail	Entire	Hot	Long-tail	Entire	Entire	Hot	Long-tail	Entire	Hot	Long-tail	Entire	Entire
BaseModel	0.0908	0.0224	0.0783	0.0156	0.0043	0.0147	0.8153	0.3022	0.2378	0.2915	0.1634	0.1164	0.1487	0.8801
IPS [22]	0.0629	0.0608	0.0619	0.0084	0.0075	0.0079	0.6638	0.2750	0.2691	0.2734	0.1346	0.1304	0.1325	0.6733
ESAM [9]	0.1105	0.0488	0.0962	0.0278	0.0061	0.0258	0.7659	0.3159	0.2559	0.2948	0.1749	0.1233	0.1502	0.8244
MACR [36]	0.1332	0.1047	0.1207	0.0529	0.0251	0.0414	0.7018	0.3098	0.2705	0.3003	0.1697	0.1324	0.1549	0.7761
DICE[42]	0.0643	0.1201	0.0758	0.0114	0.0306	0.0135	0.6329	0.2813	0.2829	0.2820	0.1383	0.1415	0.1408	0.6479
PDA [41]	0.1459	0.0514	0.1231	0.0627	0.0069	0.0471	0.7476	0.3177	0.2545	0.3016	0.1763	0.1201	0.1576	0.8152
$\text{CD}^2\text{AN}_{unbias}$	0.1297	**0.1265**	0.1283	0.0503	**0.0466**	0.0492	**0.5952**	0.3072	**0.2984**	0.3021	0.1682	**0.1641**	0.1657	**0.6075**
$\text{CD}^2\text{AN}_{bias}$	**0.1548**	0.1230	**0.1455**	**0.0742**	0.0374	**0.0613**	0.6820	**0.3284**	0.2893	**0.3179**	**0.1854**	0.1553	**0.1729**	0.7361

recommender model according to the causal graph and performs counterfactual inference for better performance.

To guarantee fair comparison, the same frameworks are used for sequence net in all comparison methods.

3.1.3 Implementation Details and Metrics. All methods are implemented with TensorFlow [2] and trained with Adam optimizer [24]. All methods are trained five times and the average results are reported. For the hyperparameters β_1 and β_2, we search the best ones in the range of $\{0.01, 0.1, 0.3, 0.5, 0.7, 1, 10\}$ and set $\beta_1 = 0.5$ and $\beta_2 = 0.7$. For other hyperparameters, we set learning rate to be $1e-4$, batch size B to be 1024, temperature τ to be 0.07, representation dimension $d = 64$ in the Taobao dataset and $d = 16$ in MovieLens-1M. For the hyperparameters of comparison methods, we use a grid search based on performance on the validation set. For offline evaluation, we adopt Hit Ratio (HR) [11], Normalized Discounted Cumulative Gain (NDCG) [16] and Concentration Ratio (C-Ratio) we defined as the performance metrics. Specifically, we follow the all-ranking protocol [18], that is, all items that are not interacted by a user are the candidates. Here HR can be interpreted as a recall-based measure, NDCG is a ranking-based measure, and C-Ratio is an indicator of Matthew effect. We will divide items according to click frequency in the training set. For example, the top 20% of items are divided into hot group ($\mathcal{I}_{hot}$), and the rest into long-tail group ($\mathcal{I}_{long}$). Therefore, the C-Ratio, which is denoted as the proportion of hot items in retrieval items, can be formulated as:

$$C-Ratio = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{|\mathcal{I}_{hot}| \cap |\mathcal{P}_u|}{|\mathcal{P}_u|} \tag{17}$$

where $\mathcal{P}_u$ denotes the top-k retrieved items for the user.

3.2 Performance Comparison (RQ1)

3.2.1 Overall Offline Performance. As Eq. 17, we divide the entire item space into hot item set and long-tail item set by the click frequency. Table 3 shows the comparison of top-k recommendation performance. We make the following observations from the results.

Firstly, the C-Ratio of the BaseModel retrieval result in Taobao industrial dataset is 0.8801, while the C-Ratio in the training set [3] of Taobao industrial dataset is 0.7864, which means the existence of popularity distribution shift, *i.e.,* the method without considering the popularity bias will not only inherit the bias, but also amplify the bias. In addition, the performance of the BaseModel in $\mathcal{I}_{hot}$ is much better than that in $\mathcal{I}_{long}$, which indicates the existence of long-tail distribution shift caused by data sparsity.

Secondly, our proposed CD^2AN achieves the best performance in all sets. This shows the effectiveness of our method. Specifically, for the hot item set, the performance of $\mathcal{I}_{hot}$ has the HR increase, while the recommended frequency of hot items (C-Ratio) is reduced. This demonstrates that the feature disentangling module we designed can capture users' real interests by separating popularity information, rather than blindly recommending irrelevant popular items. For the improvement of the performance in the long-tail set, we owe it to the $L_{long-tail}$, which reduces the disparity between hot and long-tail representation distributions, and makes sufficient learning of long-tail item representation through the direct supervision of the co-occurrence pattern.

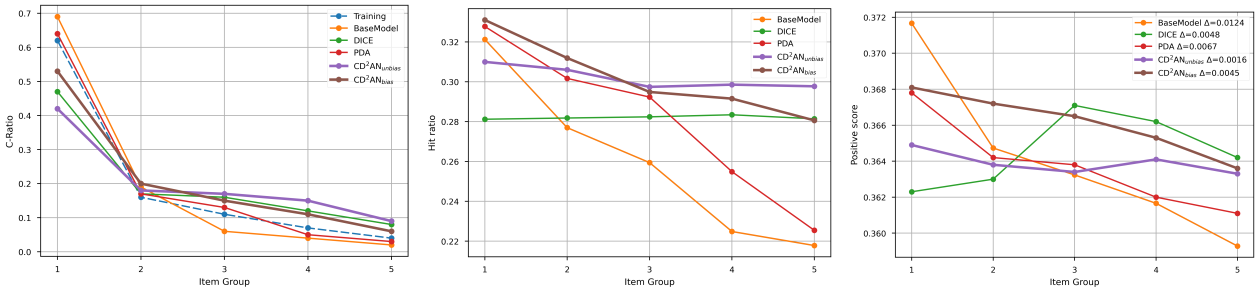
Thirdly, although IPS and DICE guarantee an increase in the proportion of long-tail items, they will under-estimate users' interest matching for popular items, which leads to the limited improvement of such methods in the entire item pool. This indicates the rationality of leveraging popularity bias instead of blindly eliminating the bias. PDA and MACR have a relatively small improvement in the long-tail item set. Since the sparse interactions of items in this set, it is difficult to obtain a comprehensive representation.

Finally, compared with $\text{CD}^2\text{AN}_{unbias}$, the performance gain of $\text{CD}^2\text{AN}_{bias}$ in the entire item set is obvious, especially in the hot item set. This verifies that the biased model of CD^2AN can effectively leverage popularity bias to improve the recommendation accuracy.

3.2.2 Online Experiments. Beyond offline studies, we conduct online A/B experiments by deploying $\text{CD}^2\text{AN}_{unbias}$ and $\text{CD}^2\text{AN}_{bias}$ respectively in the recommender system of Taobao for two weeks.

[2] https://www.tensorflow.org/

[3] $\mathcal{P}_u$ of training set is composed of target items in the training set



(a) **(b)** **(c)**

Figure 4: Retrieval analysis of different methods. (a) C-Ratio w.r.t. item group, where the blue line of dashes represents the statistical results of target items in the training set. (b) Hit ratio w.r.t. item group. (c) Positive score w.r.t. item group, where Δ represents the gap between the maximum score and the minimum score of each method.

Table 4: Online A/B test improvements in Taobao.

Method	pCTR	ANClk	ANPV	T10-Rat	T100-Rat	Exp-Rat
Baseline	+0.0%	+0.0%	+0.0%	+0.0%	+0.0%	+0.0%
CD^2AN_{unbias}	+0.28%	+0.15%	+0.21%	**−7.45%**	**−5.70%**	**+2.22%**
CD^2AN_{bias}	**+0.76%**	**+1.10%**	**+0.63%**	−3.80%	−3.58%	+1.33%

In the control setup (*i.e.,* Baseline), it includes all matching strategies in our current production system. In the variation experiment setup, the proposed CD^2AN is deployed which replaces the existing embedding-based retrieval methods (*i.e.,* BaseModel). The same ranking strategy and business logic are applied for a fair comparison. For evaluation, we select a range of core commercial metrics, including efficiency metrics (*i.e.,* the click-through rate per page view (pCTR), the average number of clicks (ANClk), and the average number of page view (ANPV)) and popularity bias metrics (*i.e.,* the exposure ratio of the top 10,000 hot items (T10-Rat), the exposure ratio of the top 100,000 hot items (T100-Rat), and the ratio of exposed items in the entire item pool (Exp-Rat)).

Table 4 summarizes the experimental results. Compared with Baseline and CD^2AN_{unbias}, CD^2AN_{unbias} alleviates the popularity bias by a large margin, in which the exposure ratio of hot items decreased significantly (T10-Rat and T100-Rat), and more long-tail items have exposure opportunities (Exp-Rat). However, the efficiency metrics increased slightly, indicating that blindly ignoring the popularity information would remove the beneficial patterns in the data. CD^2AN_{bias} improves the recommender system for all core metrics, especially efficiency metrics, which shows that CD^2AN_{bias} can reasonably leverage the popularity bias to satisfy both the real interests and conformity of users, rather than retrieve the popular items that are not relevant to users. Note that, the metrics online are reported with relative improvement.

3.2.3 Retrieval Analysis. To further verify that CD^2AN can inhibit the amplification of popularity bias, we divided the item pool into five groups based on click frequency in the training set, namely $\mathcal{I}_1$ (Top 10%), $\mathcal{I}_2$ (Top 10%-Top 20%), $\mathcal{I}_3$ (Top 20%-Top 40%), $\mathcal{I}_4$ (Top 40%-Top 60%) and $\mathcal{I}_5$ (Top 60%-Top 100%), and conduct an analysis from three aspects: (1) C-Ratio (Figure 4a), we show the retrieval rates of each group, as Eq. 17. If the more popular groups have higher retrieval rates, there is the amplification of popularity bias. (2) Hit ratio of each item group (Figure 4b), which is adopted to indicate the impact of eliminating the popularity bias on each item group. (3) Positive sample score (Figure 4c), which is used to show the impact of various methods to eliminate popularity bias on scoring. Note that, for ease of reading, in the comparison methods, we only select the BaseModel, the method with the best overall performance (PDA), and the method with the best long-tail performance (DICE) for display. The main findings are as follows.

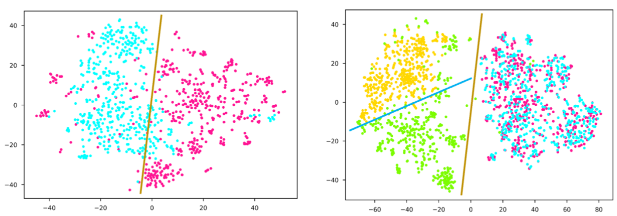
Firstly, it is clear that popularity bias will be amplified by BaseModel. As shown in Figure 4a, compared with the training set, the rate of popular items in the retrieval results is further aggravated. In addition, as shown in Figure 4c, even if they are positive samples, there is a gap ($\Delta = 0.0124$) between hot items and long-tail items. These phenomena show that models without considering popularity bias tend to recommend popular items, even if they do not match users' interests, resulting in poor long-tail performance.

Secondly, the C-Ratio of popularity group decreases for DICE. This is because DICE strategically increases the scores of unpopular items to suppress the problem of amplified interest for popular items. However, this suppression will under-estimate users' interest matching for popularity items. As such, blindly eliminating popularity bias would remove the beneficial patterns in the data, resulting in the degradation of the performance in the hot item group. The C-Ratio of PDA is similar to that of training set, which demonstrates PDA does not over-amplify or over-suppress interest for popular items. However, the unpopular item group has relatively small improvement due to the existence of long-tail distribution shift caused by data sparsity.

Thirdly, CD^2AN_{unbias} scores the positive samples most flat ($\Delta = 0.0016$) and can achieve the fairest retrieval in each item group. This shows that the FDM we design can separate the popularity information and realize the retrieval without popularity bias. However, completely ignoring the popularity bias will encounter the same problems as DICE mentioned above. Therefore, we argue that not all popularity bias is harmful. Compared with the baselines, CD^2AN_{bias} achieves the best hit ratio in all item groups, which proves that CD^2AN_{bias} can leverage the popularity pattern in the data to improve the retrieval accuracy. Specifically, for hot items, the retrieval ratio of the popular item group decreases, but the hit ratio increases, which shows that CD^2AN_{bias} can find popular

Table 5: Ablation study on Taobao industrial dataset. Here, CD^{2}AN represents CD^{2}AN$_{bias}$.

Method	HR@300			NDCG@300			C-Ratio
	Hot	Long-tail	Entire	Hot	Long-tail	Entire	Entire
CD^{2}AN w/o L_{pop_sim}	0.3224	0.2803	0.3085	0.1769	0.1392	0.1653	0.7869
CD^{2}AN w/o $L_{orthogonal}$	0.3187	0.2769	0.3033	0.1720	0.1375	0.1589	0.7748
CD^{2}AN w/o L_{MMD}	0.3195	0.2677	0.2964	0.1738	0.1259	0.1556	0.7541
CD^{2}AN w/o L_{it_sim}	0.3180	0.2639	0.2885	0.1692	0.1207	0.1478	0.7600
CD^{2}AN	**0.3284**	**0.2893**	**0.3179**	**0.1854**	**0.1553**	**0.1729**	**0.7361**
CD^{2}AN$_{unbias}$ w/o L_{bias}	0.3021	0.2933	0.2985	0.1640	0.1592	0.1609	0.6224
CD^{2}AN$_{unbias}$	**0.3072**	**0.2984**	**0.3021**	**0.1682**	**0.1641**	**0.1657**	**0.6075**



(a) BaseModel **(b) CD^{2}AN**

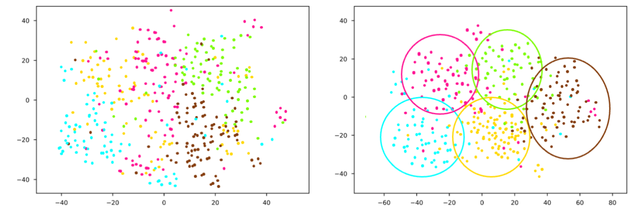
Figure 5: T-SNE visualization on Taobao industrial dataset. The pink points are hot item property representations, the blue points are long-tail item property representations, the yellow points are hot disentangled popularity representations, and the green point are long-tail disentangled popularity representations (*Best viewed in color*).

items that users are really interested in, rather than blindly over recommending. For long-tail items, the hit ratio of the long-tail item group is improved by a large margin, which shows that the $L_{long-tail}$ we design can encourage the model to obtain comprehensive representations of these items for the long-tail shift problem, which is ignored in most current de-bias methods.

3.3 Ablation Study and Interpretability (RQ2)

3.3.1 Ablation Study. To further investigate the impact of each component in our proposed CD^{2}AN, we conduct the ablation study on the Taobao industrial dataset. Several observations can be made based on Table 5: (1) Compared with CD^{2}AN, the C-Ratio of CD^{2}AN w/o L_{pop_sim} and CD^{2}AN w/o $L_{orthogonal}$ has increased. This phenomenon shows that the L_{pop_sim} and $L_{orthogonal}$ we design can guide the FDM to disentangle the real popularity information and realize the elimination of bias. (2) Compared with CD^{2}AN w/o L_{MMD} and CD^{2}AN w/o L_{it_sim}, CD^{2}AN can greatly improve the performance of the model in the long-tail space, which proves that comprehensive long-tail item representation can be learned via distribution alignment and instance alignment between long-tail and hot items. (3) The combination of these components (*i.e.,* CD^{2}AN) yields the best performance, which indicates the necessity of these regularization terms we design. (4) Compared with CD^{2}AN$_{unbias}$ w/o L_{bias}, CD^{2}AN$_{unbias}$ has a slight improvement in each metrics, which illustrates that co-training of biased and unbiased models can further improve the learning of item property representation.

3.3.2 Interpretability Based on Disentangled Representation. To illustrate the existence of distribution shift and the validity of the



(a) CD^{2}AN w/o L_{it_sim} **(b) CD^{2}AN**

Figure 6: Different categories are plotted in different colors, in which blue represents watch, pink represents digital, yellow represents apparel, green represents shoe, and brown represents travel (*Best viewed in color*).

proposed CD^{2}AN, we randomly select 500 items from $\mathcal{I}_{hot}$ and $\mathcal{I}_{long}$ respectively and visualize the learned item property representation v_x^{pro} and disentangled popularity representation v_x^{pop} using t-SNE [34]. As shown in Figure 5, we can make intuitive observations. (1) As shown in Figure 5a, there is a distribution gap (yellow line in Figure 5a) between hot and long-tail item representations learned by BaseModel. This gap indicates that BaseModel will inject popularity information into item property embedding, which makes the model that fits hot items over recommend popular items, thus amplifying the popularity bias. (2) As shown in Figure 5b, we find that there is no disparity in the distribution of v_x^{pro} learned by CD^{2}AN, which is attributed to two designs. On the one hand, the FDM we design separates the popularity information that causes the popularity distribution shift. On the other hand, L_{MMD} explicitly aligns the distribution of hot and long-tail item property representations and suppresses the long-tail distribution shift caused by data sparsity. (3) With direct supervision $L_{orthogonal}$ on disentanglement, item property representation and disentangled popularity representation are far from each other, separated by a linear classifier (yellow line in Figure 5b). Besides, due to the difference in popularity between hot items and long-tail items, hot and long-tail disentangled popularity representations are clustered into two sets (blue line in Figure 5b), which proves that L_{pop_sim} can constrain the separated popularity representation to contain semantic information about popularity. In a word, visualizations of the learned representations illustrate the high quality of disentangling in the proposed framework.

3.3.3 Effectiveness of Item Similarity Regularization. Besides eliminating popularity bias, we expect the item property representations regularized with L_{it_sim} to have better quality than the counterparts without L_{it_sim}. To verify our hypothesis, we take the item property representations trained on the Taobao industrial dataset and plot them using t-SNE. Compared to CD^{2}AN w/o L_{it_sim} (Figure 6a), item property representations learned with CD^{2}AN (Figure 6b) are better clustered according to their own categories, and the separation of different categories is much more clear. For example, we could find that the digital category (pink point) is mixed together with the apparel category (yellow point) in Figure 6a. While in Figure 6b, we clearly see the 5 categories of representations grouped among themselves. This demonstrates that the representations learned with L_{it_sim} based on co-occurrence pattern have

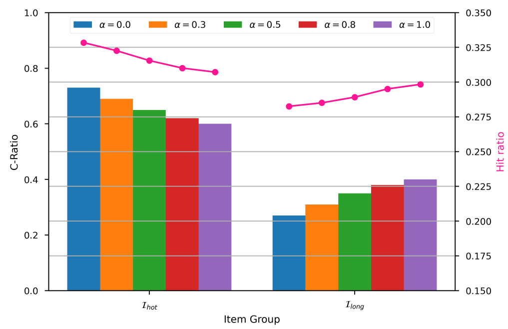


Figure 7: The impact of α on retrieval results in the Taobao industrial dataset. The histogram indicates C-Ratio, and the pink line indicates HR@300.

stronger semantic structures, and is also why CD^{2}AN leads to better model performance in our experiments.

3.4 Performance of Fusion Strategy (RQ3)

Popular items are often of better quality and deserve more recommendation. Blindly eliminating popularity bias may cause the model unable to learn the current popular trend, resulting in a reduction in the accuracy of the recommendation. As mentioned in [42], there are generally two kinds of items that users click on, one is really interested, and the other is conformity. Therefore, we believe that popularity bias is not all harmful. To encourage the model to make better use of popularity bias, as shown by the dotted line in Figure 2b and Eq. 15, we adjust the dependence of retrieval results on popularity by fusing the popularity item representation $\boldsymbol{v}_x$ of biased model CD^{2}AN$_{bias}$ and item property representation $\boldsymbol{v}_x^{pro}$ of the unbiased model CD^{2}AN$_{unbias}$. To verify the effectiveness of the fusion strategy, we change α to observe its correlation with the retrieval results. As shown in Figure 7, when $\alpha = 1$ (CD^{2}AN$_{unbias}$), it can be found that C-Ratio in $\mathcal{I}_{hot}$ decreases significantly, and the HR has a small gap between hot and long-tail item sets, which means that retrieval without considering popularity tends to find users’ real interests. However, due to ignoring users’ conformity, the HR in the hot item set and the entire item set decreases compared with some methods. With the decrease of α, we find that the HR and C-Ratio in $\mathcal{I}_{hot}$ gradually increase, and have the best performance in the entire set when $\alpha = 0$ (CD^{2}AN$_{bias}$), it shows that the designed fusion strategy can effectively leverage the popularity bias to meet the real interest and conformity of users at the same time. It is worth mentioning that since the model separates the popularity information from item property embedding, even if $\alpha = 0$, CD^{2}AN$_{bias}$ only inherits the popularity bias without amplifying it.

4 RELATED WORK

Researchers have explored many approaches to analyze and eliminate popularity bias. These works mainly consider three types of methods: IPS-based methods [3, 14, 26, 31], causal embedding methods [15, 27, 35, 36, 42] and regularization-based methods [1, 9, 40, 44]. IPS-based methods re-weight each instance as the inverse of corresponding item popularity, thus the more popular the item, the lower the weight. However, there is no guaranteed accuracy for estimating propensity scores, a series of variants have been proposed to attain more stable results based on IPS [3, 14]. Causal embedding methods focus on confounding effects. DCF [35] takes the de-confounding technique in linear models to learn real interest affected by unobserved confounders. ExpoMF [27] jointly models both users’ exposure to an item, and their resulting click decisions, resulting in a model which naturally down-weights the expected, but ultimately un-clicked items. The popularity of items can be added as an exposure covariate and thus be used to alleviate popularity bias. DICE [42] formulates the problem of disentangling user interest and conformity for recommender systems, which tackles the causal recommendation problem from the perspective of users. MACR [36] analyzes the causal relations in a fine-grained manner, modeling the influence of item popularity and user conformity on recommendations. Regularization-based methods aim to improve long-tail performance. XQuAD [1] introduces a personalized diversification re-ranking approach to increase the representation of less popular items in recommendations. Bayesian personalized ranking loss [44] investigates the potential of two approaches to reduce this bias: a post-processing approach to compensate for popularity in recommendation; and an in-processing approach that regularizes predicted scores and item popularity. ESAM [9] proposes a general entire space adaptation model for ranking models, which exploits the domain adaptation to improve long-tail performance.

One line of research leverages popularity information to improve the model performance. [25] tries to use temporary popularity in music recommendation. [2, 21, 29] try to utilize temporal popularity for the general recommendation. PDA [41] performs deconfounded training with do-calculus and causally intervenes the popularity bias during recommendation inference.

From the perspective of leveraging popularity bias, the most relevant work is PDA [41]. The difference is that CD^{2}AN we proposed can co-train both biased and unbiased models, and apply regularizations to obtain the comprehensive representation of long-tail items. From the perspective of model architecture, the most similar work is DICE [42], which disentangles user interest and conformity from user embedding. The difference is that CD^{2}AN disentangles popularity information from item property embedding, which has not been explored in the existing work.

5 CONCLUSION

In this paper, a co-training disentangled domain adaptation network (CD^{2}AN) is proposed to leverage popularity bias in recommenders. Specifically, for popularity distribution shift, FDM is designed to disentangle item property representation and disentangled popularity representation. For long-tail distribution shift, two regularizations (L_{MMD} and L_{it_sim}) are adopted to learn the comprehensive long-tail item property representation by introducing unexposed items and user behavior sequences into the item tower. To the best of our knowledge, this is the first work that can co-train both biased and unbiased models. Furthermore, based on the retrieval paradigm (NN search) of the matching stage when serving online, we design a fusion strategy that can balance retrieval results to meet users’ real interests and conformity by adjusting the weights. Offline and online experiments demonstrate the effectiveness and rationality of CD^{2}AN. Currently, CD^{2}AN has been successfully deployed in our online recommender system in Taobao.

REFERENCES

- [1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing popularity bias in recommender systems with personalized re-ranking. In *The thirty-second international flairs conference.*
- [2] Vito Walter Anelli, Tommaso Di Noia, Eugenio Di Sciascio, Azzurra Ragone, and Joseph Trotta. 2019. Local popularity and time in top-n recommendation. In *European Conference on Information Retrieval.* Springer, 861–868.
- [3] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising. *Journal of Machine Learning Research* 14, 11 (2013).
- [4] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. *Advances in neural information processing systems* 29 (2016), 343–351.
- [5] Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. 2019. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 3296–3303.
- [6] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data.* 1–4.
- [7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. 2020. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029* (2020).
- [8] Zhihong Chen, Chao Chen, Zhaowei Cheng, Boyuan Jiang, Ke Fang, and Xinyu Jin. 2020. Selective transfer with reinforced transfer network for partial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 12706–12714.
- [9] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. Esam: Discriminative domain adaptation with non-displayed items to improve long-tail performance. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 579–588.
- [10] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems.* 191–198.
- [11] Mukund Deshpande and George Karypis. 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 143–177.
- [12] Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *arXiv preprint arXiv:1811.08008* (2018).
- [13] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research* 13, 1 (2012), 723–773.
- [14] Alois Gruson, Praveen Chandar, Christophe Charbuillet, James McInerney, Samantha Hansen, Damien Tardieu, and Ben Carterette. 2019. Offline evaluation to make decisions about playlistrecommendation algorithms. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining.* 420–428.
- [15] Priyanka Gupta, Ankit Sharma, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. 2021. CauSeR: Causal Session-based Recommendations for Handling Popularity Bias. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management.* 3048–3052.
- [16] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management.* 1661–1670.
- [17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web.* 173–182.
- [18] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval.* 549–558.
- [19] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
- [20] 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 the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2553–2561.
- [21] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2020. A re-visit of the popularity baseline in recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.*
- [22] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining.* 781–789.
- [23] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data* (2019).
- [24] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [25] Elisabeth Lex, Dominik Kowald, and Markus Schedl. 2020. Modeling popularity and temporal drift of music genre preferences. *Transactions of the International Society for Music Information Retrieval* 3, 1 (2020).
- [26] Dawen Liang, Laurent Charlin, and David M Blei. 2016. Causal inference for recommendation. In *Causation: Foundation to Application, Workshop at UAI. AUAI.*
- [27] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of the 25th international conference on World Wide Web.* 951–961.
- [28] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.
- [29] Koki Nagatani and Masahiro Sato. 2017. Accurate and Diverse Recommendation based on Users' Tendencies toward Temporal Item Popularity.. In *RecTemp@ RecSys.* 35–39.
- [30] Matjaž Perc. 2014. The Matthew effect in empirical data. *Journal of The Royal Society Interface* 11, 98 (2014), 20140378.
- [31] Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70, 1 (1983), 41–55.
- [32] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web.* 285–295.
- [33] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning.* PMLR, 1670–1679.
- [34] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).
- [35] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. 2020. Causal inference for recommender systems. In *Fourteenth ACM Conference on Recommender Systems.* 426–431.
- [36] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.* 1791–1800.
- [37] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. 2020. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion Proceedings of the Web Conference 2020.* 441–447.
- [38] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management.* 4321–4330.
- [39] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems.* 269–277.
- [40] Yin Zhang, Derek Zhiyuan Cheng, Tiansheng Yao, Xinyang Yi, Lichan Hong, and Ed H Chi. 2021. A model of two tales: Dual transfer learning framework for improved long-tail item recommendation. In *Proceedings of the Web Conference 2021.* 2220–2231.
- [41] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. *arXiv preprint arXiv:2105.06067* (2021).
- [42] 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 the Web Conference 2021.* 2980–2991.
- [43] 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 the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 1059–1068.
- [44] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-Opportunity Bias in Collaborative Filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining.* 85–93.