Invariant Collaborative Filtering to Popularity Distribution Shift

An Zhang
National University of Singapore
Sea-NExT Joint Lab
anzhang@u.nus.edu

Jingnan Zheng
National University of Singapore
e0718957@u.nus.edu

Xiang Wang[*]
University of Science and Technology of China
xiangwang1223@gmail.com

Yancheng Yuan
The Hong Kong Polytechnic University
yanchengyuanmath@gmail.com

Tat-Seng Chua
National University of Singapore
Sea-NExT Joint Lab
dcscts@nus.edu.sg

ABSTRACT

Collaborative Filtering (CF) models, despite their great success, suffer from severe performance drops due to popularity distribution shifts, where these changes are ubiquitous and inevitable in real-world scenarios. Unfortunately, most leading popularity debiasing strategies, rather than tackling the vulnerability of CF models to varying popularity distributions, require prior knowledge of the test distribution to identify the degree of bias and further learn the popularity-entangled representations to mitigate the bias. Consequently, these models result in significant performance benefits in the target test set, while dramatically deviating the recommendation from users' true interests without knowing the popularity distribution in advance. In this work, we propose a novel learning framework, Invariant Collaborative Filtering (**InvCF**), to discover disentangled representations that faithfully reveal the latent preference and popularity semantics without making any assumption about the popularity distribution. At its core is the distillation of unbiased preference representations (*i.e.,* user preference on item property), which are invariant to the change of popularity semantics, while filtering out the popularity feature that is unstable or outdated. Extensive experiments on five benchmark datasets and four evaluation settings (*i.e.,* synthetic long-tail, unbiased, temporal split, and out-of-distribution evaluations) demonstrate that InvCF outperforms the state-of-the-art baselines in terms of popularity generalization ability on real recommendations. Visualization studies shed light on the advantages of InvCF for disentangled representation learning. Our codes are available at https://github.com/anzhang314/InvCF.

CCS CONCEPTS

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

KEYWORDS

Collaborative Filtering, Popularity Distribution Shift, Debiasing

ACM Reference Format:
An Zhang, **Jingnan Zheng**, **Xiang Wang**, **Yancheng Yuan**, and **Tat-Seng Chua**. 2023. Invariant Collaborative Filtering to Popularity Distribution Shift. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3543507.3583461

1 INTRODUCTION

Collaborative filtering (CF) is a keystone of personalized recommendation, hypothesizing that behaviorally similar users tend to have similar preferences on items. Inspecting leading CF models [29, 44, 45], we can systematize a dominant paradigm — view historical interactions between users and items as the training data, encode the collaborative signals as their representations, and then use these representations to predict future interactions in the test data. It simply assumes that the training and test data are drawn from the same distribution. However, this assumption hardly holds in the real-world scenarios, due to an inherent factor — **popularity distribution shift** between the training and test data (See real-world datasets statistics in Figures 3, 6-8). That is, the statistical popularity of items varies across historical and future interactions, which is usually caused by the demographic, regional, and chronological diversity of human behaviors [16, 24]. For example, the fashion trend and alteration of the season will influence the change in item popularity (See Figure 1); meanwhile, the sales of take-out food will fluctuate between weekdays and weekends [43]. Due to the prevalence of popularity distribution shifts, items and users representations are infused with unstable popularity correlations, which, under real-world recommendation scenarios, render CF models unrobust and severely degrade their performance.

To overcome the lack of robustness guarantees in real applications, popularity generalization over CF models is attracting a surge in interest [18, 40, 55, 58]. However, rather than addressing the vulnerability of CF model to popularity distribution shifts [2, 51], current prevalent studies mainly focus on popularity debiasing. That is, measuring the degree of popularity bias in the training data first, and then mitigating the bias using a variety of debiasing strategies, including regularization-based [1, 6, 11, 59], reweighting-based [9, 12, 19, 35, 47, 53], and causal embedding [14, 25, 27, 42, 56, 57]. Such the degree of bias is determined by the popularity distribution deviation of the training set from the target data. Inevitably, to precisely quantify the deviation, there is an implicit but impractical constraint for these debiasing methods, *i.e.,* the target popularity distribution must be known in advance.

[*] Xiang Wang is corresponding author, also with Institute of Artificial Intelligence, Hefei Comprehensive National Science Center.

WWW '23, April 30–May 04, 2023, Austin, TX, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9416-1/23/04.
https://doi.org/10.1145/3543507.3583461

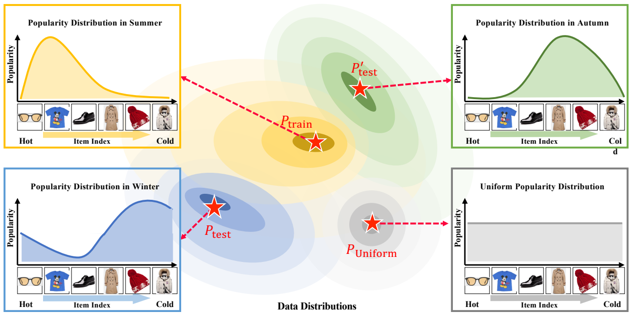


Figure 1: Illustration of real-world popularity distribution shifts. The alteration of the season naturally influences the item popularity distribution, where both P_{test} and P'_{test} are possible to occur but are unpredictable. More datasets statistics can be found in Figure 3 and Appendix A.1 .

- The reweighting-based approaches [7, 13, 36], also referred to as Inverse Propensity Score (IPS), inversely weight each item's prediction error with the propensity of its popularity. The propensities, once inverted, lead to an inherent assumption that the uniform popularity distribution with regard to items is unbiased and on target.
- Most cutting-edge debiasing techniques demand prior knowledge of the test popularity distributions, such as leveraging the validation set equipped with similar popularity distribution of test as a guide for hyperparameter adjustment [12, 50, 57], or leaking a small amount of unbiased data to strengthen unbiased recommendation learning [5, 9, 27].

These debiasing methods suffer from injecting popularity information into representations. As a result, without prior knowledge of the target distribution, highly popularity-entangled representations encounter a severe performance drop in practice, which further limits the applicability of these debiasing techniques.

We postulate that an ideal method to handle unknown popularity distribution shifts should learn disentangled representations that faithfully reflect the hidden invariant features and popularity semantics, rather than learning superficial entangled representations. Our core idea is, inspired by invariant learning [3, 23] and disentangled representations [17], to identify the invariant features (*i.e.,* a user's true preference, an item's real property) that causally determine the historical and future interactions, regardless of changes in variant popularity. Such popularity-invariant representations are able to yield a consistently plausible performance and enhanced generalization ability in real recommender systems (See Table 2).

Towards this end, we propose a new learning framework, Invariant Collaborative Filtering (**InvCF**), that integrates invariance and disentanglement principles. By disentanglement principle, we mean that the representations are decomposed into two independent components - popularity and preference representations, while changing the "popularity" semantic does not affect the preference representations. By invariance principle, we mean that the relations between preference representations and interactions are invariant throughout a variety of popularity distributions, and that preference representations are sufficient on their own to preserve the critical facts of interactions.

Guided by these two principles, our InvCF strategy incorporates four modules: a preference encoder, a popularity encoder, a representation augmentation module, and a representation disentangling module. First, the popularity encoder and preference encoder, respectively, learn inference mappings from popularity statistics and historical interactions to the latent representation space. Then the augmentation and disentangling modules are implemented based on the proposed principles to drive the representation learning. Specifically, towards the disentanglement principle, disentangling module utilizes distance correlation as a regularizer to encourage independence of popularity and preference representations (See additional results employing various discrepancies in Table 10). Towards the invariant principle, the augmentation module concatenates the target preference representation with other's popularity representations and enforces the prediction to be invariant. Jointly training under these two principles enables the CF models to disentangle the invariant/causal semantic features and variant/spurious popularity information, and further boosts the model's capability to popularity generalization. Our main contributions are summarized as follows:

- From a more realistic standpoint, we broaden our understanding of the current popularity debiasing problem and reformulate it as the problem of popularity distribution shift in CF.
- We propose a novel Invariant Collaborative Filtering (InvCF) method that hinges on the representation level disentanglement and augmentation to guarantee invariant feature learning.
- We conduct in-depth experiments with extensive test evaluations to justify the superiority of InvCF in diverse popularity distributions.

2 PRELIMINARY

We begin with the definition of popularity distribution shift in CF, and highlight its differences from popularity bias. Then, we formulate the problem of popularity generalization, and reveal the limitations of current debiasing approaches when facing this generalization problem. Throughout the paper, we represent the random variables and their deterministic values with the upper-cased (*e.g.,* X) and lower-cased (*e.g.,* x) letters, respectively.

2.1 Popularity Distribution Shift

Background. Here we focus on item recommendation from implicit feedback [34], where an interaction between a user and an item (*e.g.,* view, purchase, click) implicitly reflects the user's preference. The task is building a CF model to learn the user preference from historical interactions and predict future interactions. Let $\mathcal{D}_{\text{train}} = \{(x, y)|x = (u, i), y = 1\}$ be the training set that involves historical interactions between users and items, and $\mathcal{D}_{\text{test}} = \{(x, y)|x = (u, i), y = 1\}$ be the test set that contains future interactions, where $y = 1$ indicates that user u interacts with item i, otherwise $y = 0$.

Formally, the dominant paradigm of learning CF models [33, 34] optimizes the model parameters $\hat{\theta}$ via maximum log-likelihood estimation on the training data $\mathcal{D}_{\text{train}}$:

$$\hat{\theta} = \arg\max_{\theta} \log P_{\text{train}}(Y|X) = \arg\max_{\theta} \sum_{(x,y)\in\mathcal{D}_{\text{train}}} \log P(y|x), \quad (1)$$

where X and Y are the variables of user-item pair and interaction, respectively; $P(y|x)$ is the probability of $Y = y$ conditioned on $X = x$, indicating how likely user u interacts with item i. To approach the estimation, extensive model architectures [15, 26, 29, 34] have been designed to develop the CF idea — behaviorally similar users tend to have similar preferences on items. Regardless of diverse designs, at the core is distilling the CF signals as the representation of X and regressing it to Y. For example, MF [21] and LightGCN [15] learn a pair of user representation and item representation to depict X, and use the inner product of them to fit the interaction Y.

Definition. Following prior studies [10, 57], we reveal two parts inherent in X's representation: (1) information on users' pure preferences, Z_{pref}, which reflects user interest in item properties; (2) information on popularity, Z_{pop}, which describes user conformity influenced by item popularity. On closer inspection on these parts [16, 49], Z_{pref} is more stable to serve as the causation of interaction Y; in stark contrast, Z_{pop} more easily changes due to demographic, regional, and chronological diversity of human behaviors, thus holding the unstable correlation with interaction Y. Conventional CF models mostly assume that the training and test data are from the same distribution, thereby having the same popularity information Z_{pop}. However, this assumption is unrealistic in the real-world scenarios.

Formally, across the training and test data (*i.e.,* $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$), we define the underlying changes in popularity information Z_{pop} as popularity distribution shift:

$$P_{\text{train}}(Z_{\text{pop}}, Y) \neq P_{\text{test}}(Z_{\text{pop}}, Y). \tag{2}$$

Differences from Popularity Bias. There has been increasing interest in the popularity distribution difference between the training and test data, which is quantitatively measured as popularity bias [9, 36, 50, 57]. However, these studies on popularity bias inherently assume the popularity information of test distribution is known or assumed in advance during training. See Section 2.2 for more details. In sheer contrast, the popularity distribution shift focuses on the more general and practical scenario, which has no access to any popularity information about the test distribution. That is, we hardly quantify the distribution discrepancy *w.r.t.* popularity between the training and test data. Hence, it poses a major obstacle in enhancing the generalization ability of CF models to unknown popularity distribution shifts.

2.2 Popularity Generalization

Problem Formulation. Following previous work [37], we reformulate the recommendation problem with the focus on popularity generalization. Specifically, upon historical interactions drawn from the training data $\mathcal{D}_{\text{train}}$, a CF model is learned to generalize well on future interactions from the test data $\mathcal{D}_{\text{test}}$, considering underlying popularity distribution shift:

$$\theta^* = \arg\max_{\theta} \log P_{\text{test}}(Y|X), \tag{3}$$

where θ^* is the oracle parameters of model, which differs from the estimation $\hat{\theta}$ in Equation (1); $\mathcal{D}_{\text{test}}$ remains unknown during the training phase. Worse still, no access is available to quantify the distribution shift *w.r.t.* popularity information between $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$. Therefore, it is infeasible to solve this generalization problem, without any assumption.

To make reasonable assumptions on popularity generalization, we derive a fine-grained analysis of $P_{\text{test}}(Y|X)$ and reveal how it is distinct from $P_{\text{train}}(Y|X)$ *w.r.t.* popularity information. Specifically, we can decompose $P_{\text{test}}(Y|X)$ into the following terms via the Bayes theorem:

$$\begin{aligned} P_{\text{test}}(Y|X) &= P_{\text{test}}(Y|Z_{\text{pref}}, Z_{\text{pop}}) \\ &= \frac{P_{\text{test}}(Z_{\text{pref}}, Z_{\text{pop}}|Y) \cdot P_{\text{test}}(Y)}{P_{\text{test}}(Z_{\text{pref}}, Z_{\text{pop}})} \\ &\propto \underbrace{P_{\text{test}}(Z_{\text{pop}}|Y)}_{\text{Bias term}} \cdot \underbrace{P_{\text{test}}(Z_{\text{pref}}|Y, Z_{\text{pop}})}_{\text{Entanglement term}}. \end{aligned} \tag{4}$$

The bias term shows the direct effect of popularity information Z_{pop} on $P_{\text{test}}(Y|X)$, whose comparison with $P_{\text{train}}(Z_{\text{pop}}|Y)$ of $P_{\text{train}}(Y|X)$ frames the certain popularity bias. Meanwhile, the entanglement term depicts that Z_{pre} are entangled with Z_{pop}, making the distillation of preference information dependent inherently on the popularity information. Hence, the key to popularity generalization lies in mitigating the influence of popularity on these terms.

Limitations of Debiasing Approaches. Before introducing our assumptions, we first exhibit two deficiencies of current debiasing approaches as follows:

- For the bias term in Equation (4), most debiasing methods make an implicit but unrealistic assumption that the popularity information in test is available. For example, the reweighting methods [7, 13, 19, 35, 36, 53] (also known as IPS families) actually use the uniform distribution of popularity as the unbiased test data. Recent efforts require the test knowledge in advance to help the model training, such as leveraging the validation set conforming to the test data to guide the hyperparameter adjustment [12, 50, 57], or a small unbiased set to boost the unbiased learning [5, 9, 27]. Although these methods could achieve better performance on certain test distribution, they still suffer from other popularity distribution shifts and get degenerated performance.
- For the entanglement term in Equation (4), prior studies [42, 50, 56] simply hypothesize $P_{\text{test}}(Z_{\text{pref}}|Y, Z_{\text{pop}}) = P_{\text{train}}(Z_{\text{pref}}|Y, Z_{\text{pop}})$. That is, the popularity information Z_{pop} has a stable influence on the preference information Z_{pref} across different distributions. However, the correlation between Z_{pop} and Z_{pref} is naturally shifting. Hence, it is crucial to disentangle the preference information from the popularity information.

3 METHODOLOGY

To achieve the popularity generalization, we propose a new learning paradigm, Invariant Collaborative Filtering (InvCF). Specifically, we begin by introducing two principles: (1) invariance, which encourages the interaction prediction to be determined by the preference information solely, while invariant to the popularity change; (2) disentanglement, which enforces the preference and popularity information decomposable and independent from each other. We then describe our implementation of these principles.

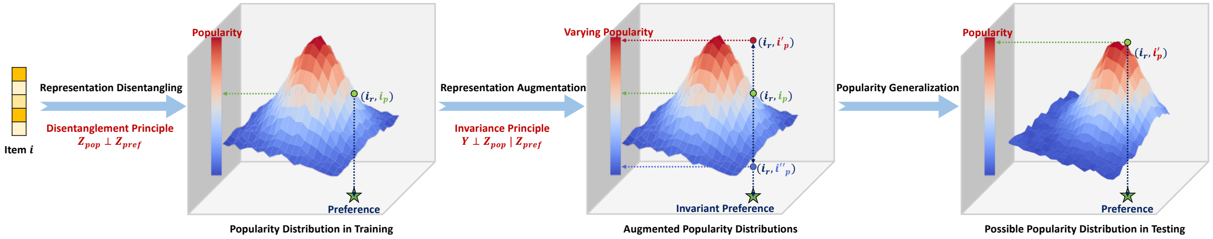

Item i
Representation Disentangling
Disentanglement Principle
$Z_{pop} \perp Z_{pref}$
Popularity
(i_r, i_p)
Preference
Popularity Distribution in Training
Representation Augmentation
Invariance Principle
$Y \perp Z_{pop} \mid Z_{pref}$
Varying Popularity
(i_r, i'_p)
(i_r, i_p)
(i_r, i''_p)
Invariant Preference
Augmented Popularity Distributions
Popularity Generalization
Popularity
(i_r, i'_p)
Preference
Possible Popularity Distribution in Testing

Figure 2: Illustration of item representation derived from InvCF and its popularity generalization.

3.1 Invariant Collaborative Filtering

Upon the inspection of the bias and entanglement terms in Equation (4), we draw inspiration from invariant learning [3, 23] and disentangled representation learning [17] to propose two principles. Next, we elaborate these principles towards popularity generation.

Invariance Principle. By "invariance", we conjecture that an ideal CF model should refine the invariant preference information (*e.g.,* user preference on item properties) that causally determines the interaction behaviors, regardless of changes in popularity information (*e.g.,* user conformity, item popularity). More formally, this principle is:

$$Y \perp Z_{\text{pop}} \mid Z_{\text{pref}}, \tag{5}$$

where $\perp$ denotes probabilistic independence. It delineates that Z_{pref} shields Y from the influence of Z_{pop}, making the prediction-making process from Z_{pref} to Y stable across different Z_{pop}. Taking movie recommendation as an example, the invariance principle is a lever for looking at users' stable incentives to watch a movie (*e.g.,* interest in the director and star aspects), rather than the spurious correlations caused by popularity factors (*e.g.,* box office).

Disentanglement Principle. By "disentanglement", we mean that the preference and popularity information should be decomposable and independent [17] from each other , so that changing the popularity does not affect the user preference on item properties. This principle can be formulated as:

$$Z_{\text{pop}} \perp Z_{\text{pref}}. \tag{6}$$

It stipulates that the learning of Z_{pref} is not susceptible to Z_{pop}. Considering movie recommendation again, the disentanglement principle makes the popularity of a movie hardly derail a user's interest in the director and star aspects.

Integration of Two Principles. These two principles collaborate with each other to guide the learning of CF models, so as to endow them with powerful prediction and generalization abilities.

3.2 Implementations of Two Principles

Here we present how to parameterize these two principles in InvCF. As illustrated in Figure 2, it comprises two additional modules on top of the CF backbone: representation augmentation and disentanglement. Specifically, the CF backbone is used to encode the preference and popularity information as the corresponding representations. The augmentation module couple a preference representation with diverse popularity representations to approach an invariant prediction, so as to achieve the invariance principle. Meanwhile, the disentanglement module pursues the independence of preference and popularity representations. Next, we will elaborate on these modules one by one.

Representation Learning. During the training phase, the ground truth of oracle (ideal) preference and popularity information is out of reach, while only the historical user-item interactions are available. Such an absence motivates us to estimate them in the CF modeling. Specifically, upon the historical interactions, we employ a CF backbone model to intensify the preference information Z_{pref} as the representations:

$$\mathbf{u}_r, \mathbf{i}_r = f_r(u, i), \tag{7}$$

where f_r is the CF backbone (*e.g.,* MF [34], LightGCN [15]), termed preference encoder. It takes the ID of user u and item i as the input, and then yields the d-dimensional preference representations $\mathbf{u}_r \in \mathbb{R}^d$ and $\mathbf{i}_r \in \mathbb{R}^d$, respectively.

Besides the preference encoder f_r, we hire another popularity encoder f_p, which has the same architecture to f_r but aims to embrace the popularity information Z_{pop} as the representations:

$$\mathbf{u}_p, \mathbf{i}_p = f_p(u, i), \tag{8}$$

where f_p takes the statistical metrics of popularity as the input (*i.e.,* the numbers of interactions that user u and item i are involved in historically) and outputs the d-dimensional preference representations $\mathbf{u}_p \in \mathbb{R}^d$ and $\mathbf{i}_p \in \mathbb{R}^d$. It is worth noting that these popularity statistics are treated as categorical features like ID.

As a result, the preference and popularity representations are to estimate Z_{pref} and Z_{pop}, respectively. Their combination can parameterize user u and item i as:

$$\mathbf{u} = \mathbf{u}_r || \mathbf{u}_p, \quad \mathbf{i} = \mathbf{i}_r || \mathbf{i}_p, \tag{9}$$

where $||$ denotes the concatenation operation. To optimize these two encoders, we can adopt the prevalent learning strategy of empirical risk minimization (ERM). Specifically, the risk function measures the quality of interaction predictions, which can be formulated as BPR loss [34] and Softmax loss [4, 33]. Here we apply Softmax loss l on an observed interaction between user u and item i as:

$$l(\mathbf{u}_r, \mathbf{i}_r) = -\sum_{(u,i)\in\mathcal{D}_{\text{train}}} \log \frac{\exp\left(s(\mathbf{u}_r, \mathbf{i}_r)/\tau\right)}{\sum_{i'\in\mathcal{N}_u\cup\{i\}} \exp\left(s(\mathbf{u}_r, \mathbf{i}'_r)/\tau\right)}, \tag{10}$$

where s is the cosine similarity function; $\mathcal{N}_u = \{i' | (u, i') \notin \mathcal{D}_{\text{train}}\}$ is the set of sampled items that u did adopt before, in which $\mathbf{i}'_r$ is the preference representation of i'; τ is the temperature hyperparameter in softmax [4]. Analogously, we can employ Softmax loss on the popularity representations $\mathbf{u}_p$ and $\mathbf{i}_p$. In a nutshell, the ERM framework minimizes these expected risks:

$$\mathcal{L}_{\text{rep}} = l(\mathbf{u}_r, \mathbf{i}_r) + \alpha \cdot l(\mathbf{u}_p, \mathbf{i}_p), \tag{11}$$

which essentially encourages these two representation groups to fit the training interaction and collect the signals relevant to it. Here, α is the hyperparameter to control the trade-off between preference and popularity representation learning. However, solely minimizing the risks over the empirical training distribution suffers from popularity distribution shift [51].

Representation Augmentation. To bring forth better generalization *w.r.t.* popularity distribution shift, we advocate for the invariance principle in Equation (5). Wherein, the relationship between the preference information and the interaction remains stable, regardless of changes in popularity information. To parameterize this principle, we first devise an augmentation operator on user and item representations, which preserves the estimated preference information, but intervenes the estimated popularity information. Formally, the operator first collects all popularity representations into two memory banks $\mathcal{U}_p = \{\mathbf{u}'_p|\forall u'\}$ and $\mathcal{I}_p = \{\mathbf{i}'_p|\forall i'\}$. Then it samples a memory $\mathbf{u}'_p \in \mathcal{U}_p$ to replace $\mathbf{u}_p$ of u, which is combined with $\mathbf{u}_r$ to create the augmented user $\mathbf{u}^*$. Similarly, the augmentation can be performed on i in parallel to construct the augmented item $\mathbf{i}^*$. More formally, the augmentation process is:

$$\mathbf{u}^* = \mathbf{u}_r||\mathbf{u}'_p, \quad \mathbf{i}^* = \mathbf{i}_r||\mathbf{i}'_p. \tag{12}$$

Having established the augmented representations, we enforce all (u, i)'s popularity-intervened versions to hold the consistent discriminative signals about interaction:

$$\mathcal{L}_{\text{aug}} = \mathbb{E}_{\mathbf{u}'_p \in \mathcal{U}_p} l(\mathbf{u}^*, \mathbf{i}) + \mathbb{E}_{\mathbf{i}'_p \in \mathcal{I}_p} l(\mathbf{u}, \mathbf{i}^*), \tag{13}$$

where l is Softmax loss in Equation (10) but with different inputs. As a result, it learns to rule out the influence of popularity information, so as to make the preference representations more robust against diversified popularity representations.

Representation Disentanglement. Moreover, we parameterize the disentanglement principle in Equation (6) to make the preference and popularity representations independent of each other, so as to assist the invariance principle. It can be achieved by minimizing a disentanglement regularizer, such as distance correlation [17, 46, 57], Pearson correlation coefficient [10], and Maximum Mean Discrepancy (MMD) [38] (See Section 5). Here we minimize the distance correlation $dCor$ between two representation groups:

$$\mathcal{L}_{\text{dis}} = dCor(\mathbf{u}_r, \mathbf{u}_p) + dCor(\mathbf{i}_r, \mathbf{i}_p), \tag{14}$$

which takes these representations apart in the latent space. In conjunction with the invariance principles, it deprives the preference information of mixing the popularity clues.

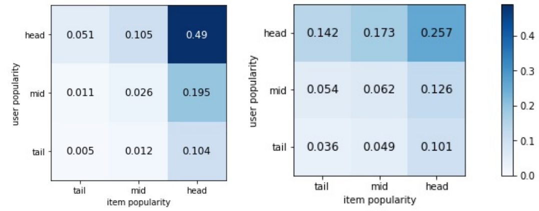
Joint Training. Overall, we can aggregate all foregoing risks and attain the final objective of InvCF:

$$\mathcal{L} = \mathbb{E}_{(x=(u,i),y) \in \mathcal{D}_{\text{train}}}(\mathcal{L}_{\text{rep}} + \lambda_1 \cdot \mathcal{L}_{\text{aug}} + \lambda_2 \cdot \mathcal{L}_{\text{dis}}), \tag{15}$$

where λ_1 and λ_2 are the hyperparameters to control the strengths of invariance and disentanglement principles. Jointly optimizing these risks with these two principles allows the CF backbone (*i.e.,* the preference encoder) to focus on the critical cues about users' stable interest in items, regardless of popularity changes. It endows the CF backbone with better popularity generalization. In the inference phase, we use the preference representations to make predictions, shielding them from the influence of popularity distribution shifts.

Table 1: Dataset statistics.

	Yahoo!R3	Coat	Douban Movie	Meituan	Tencent	Yelp2018
#Users	14,382	290	36,644	67,529	95,709	4886
#Items	1,000	295	22,226	29,785	41,602	4804
#Interactions	129,748	2,776	5,397,926	2,190,658	2,937,228	134, 031
Density	0.0090	0.0324	0.0066	0.0011	0.0007	0.0057
$D_{KL}(P_{train}\|\|P_{test})$	0.3561	0.1745	0.1601	0.0110	0.3639/0.7370/1.1282	-



(a) Training Set **(b) Unbiased Test Set**

Figure 3: An illustration of popularity distribution shifts among different subgroups on Yahoo!R3. Compared to the typical long-tail distribution in the training set, the popularity distribution in the unbiased test set is more evenly distributed. See more examples in Appendix A.1.

4 EXPERIMENTS

We aim to answer the following research questions:

- **RQ1:** How does InvCF perform compared with other debiasing strategies and popularity generalization baselines?
- **RQ2:** Does InvCF successfully learn popularity-disentangled representations?
- **RQ3:** What are the impacts of the components (*e.g.,* disentangling module, augmentation module) on InvCF?

Datasets. We conduct extensive experiments on five real-world benchmark datasets (*i.e.,* Yahoo!R3 [30], Coat [36], Douban Movie [39], Meituan [43], and Yelp2018 [15]) and one synthetic dataset (*i.e.,* Tencent [54]). Table 1 provides an overview of the statistics for all datasets, which differ in size, sparsity, domain, and the degree of popularity distribution variations. Specifically, the popularity shift degree is calculated by KL-divergence between the popularity distribution in the training set and test set, *i.e.,* $D_{KL}(P_{train}||P_{test})$, where a higher KL-divergence value indicates a larger shift. Moreover, to visually demonstrate the distribution varies, we partition the training and test sets into nine disjoint subgroups based on the number of interactions of each item/user: head (the top third), mid (the middle), and tail (the bottom third). Then we summarize the frequency of interactions across all subgroups. Figure 3 and Figures 6-8 clearly indicate that all datasets exhibit significant popularity distribution shifts between the training and test data.

Test Evaluations. For comprehensive comparisons, three standard test evaluations - unbiased test set [30, 36], temporal split test set [6, 42, 56], and out-of-distribution test set [43, 49]) - as well as one synthetic evaluation - various popularity distributions as test sets are covered in the experiments.

Baselines. Two high-performing Collaborative Filtering (CF) models - ID-based (MF [21]) and graph-based (LightGCN [15]), are selected as the backbone models being optimized. We thoroughly compare InvCF with two backbones and two categories of baselines:

- **Popularity debiasing baselines:** regularization-based frameworks (sam+reg [6]), reweighting-based methods (IPS-CN [13]), and causal embedding methods (CausE [5], MACR [50]).

Table 2: The performance comparison on Tencent dataset. The improvement achieved by InvCF is significant (p-value << 0.05).

	$\gamma = 200$			$\gamma = 10$			$\gamma = 2$			Validation		
	HR	Recall	NDCG	HR	Recall	NDCG	HR	Recall	NDCG	HR	Recall	NDCG
MF	0.0803	0.0292	0.0167	0.0504	0.0188	0.0106	0.0347	0.0132	0.0081	0.2537	0.0919	0.0542
+ sam+reg	0.0811	0.0295	0.0168	0.0525	0.0192	0.0110	0.0352	0.0133	0.0082	0.2539	0.0917	0.0543
+ IPS-CN	0.1299	0.0468	0.0273	0.0894	0.0328	0.0189	0.0656	0.0248	0.0147	0.1702	0.0613	0.0329
+ CausE	0.0936	0.0340	0.0192	0.0591	0.0212	0.0120	0.0407	0.0153	0.0091	0.2461	0.0878	0.0528
+ MACR	0.0846	0.0301	0.0173	0.0541	0.0203	0.0114	0.0386	0.0149	0.0089	0.2409	0.0862	0.0488
+ sDRO	0.1468	0.0533	0.0311	0.0941	0.0336	0.0196	0.0664	0.0242	0.0144	0.3386	0.1318	0.0810
+ CD^{2}AN	0.1409	0.0494	0.0286	0.0852	0.0300	0.0170	0.0569	0.0208	0.0119	0.2965	0.1108	0.0645
+ InvCF	**0.1580***	**0.0575***	**0.0342***	**0.1031***	**0.0374***	**0.0221***	**0.0734***	**0.0272***	**0.0165***	0.3230	0.1246	0.0748
Imp.%	7.63%	7.88%	9.97%	9.56%	11.31%	12.76%	10.54%	9.68%	12.24%	–	–	–
LightGCN	0.1167	0.0426	0.0240	0.0738	0.0272	0.0151	0.0514	0.0192	0.0114	0.3018	0.1137	0.0684
+ sam+reg	0.1522	0.0542	0.0307	0.1008	0.0356	0.0198	0.0707	0.0255	0.0141	0.3014	0.1130	0.0682
+ IPS-CN	0.1316	0.0472	0.0280	0.0874	0.0313	0.0182	0.0644	0.0242	0.0145	0.2496	0.0920	0.0545
+ CausE	0.1284	0.0469	0.0259	0.0795	0.0289	0.0157	0.0558	0.0210	0.0116	0.2870	0.1065	0.0659
+ MACR	0.1068	0.0387	0.0208	0.0663	0.0244	0.0131	0.0473	0.0182	0.0101	0.2969	0.1122	0.0666
+ sDRO	0.1455	0.0516	0.0286	0.0857	0.0304	0.0166	0.0552	0.0205	0.0110	0.3485	0.1374	0.0872
+ CD^{2}AN	0.1540	0.0559	0.0305	0.0960	0.0347	0.0185	0.0658	0.0247	0.0134	0.3594	0.1427	0.0897
+ InvCF	**0.1651***	**0.0605***	**0.0331***	**0.1061***	**0.0386***	**0.0204***	**0.0722***	**0.0272***	**0.0149***	0.3611	0.1443	0.0912
Imp.%	7.21%	8.23%	7.82%	5.26%	8.43%	3.03%	2.85%	7.94%	2.76 %	–	–	–

- **Popularity domain generalization baselines:** the latest competitive methods CD^{2}AN [10] which co-trains biased and unbiased models, and sDRO [51] which improves worst-case performance under distributionally robust optimization framework.

See detailed introductions of datasets and baselines in Appendix A.
Evaluation Metrics. We adopt the all-ranking strategy [22], where all items — aside from the positive ones in the training set — are ranked by the CF model for each user. Three commonly used metrics—Hit Ratio (HR@K), Recall@K, and Normalized Discounted Cumulative Gain (NDCG@K) — are used to assess the quality of the recommendations, with K being set by default at 20.

4.1 Performance Comparison (RQ1)

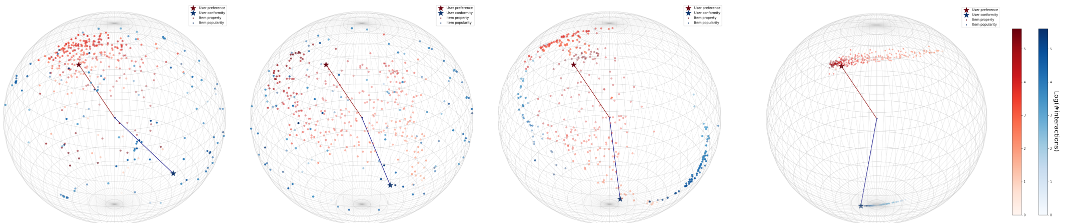
4.1.1 *Evaluations on Various Popularity Distributions*.

Motivation. Many prevalent popularity debiasing techniques concentrate on mitigating bias for single target test distribution [9, 11, 50]. However, the test popularity distributions in real-world recommendation scenarios may be diverse, unpredictable, and unknown. We argue that a good CF model is crucial to consistently yield a satisfactory performance when dealing with the unidentified popularity distribution shift. In our settings, the models are identical per method across multiple test sets, and no prior information about the test distribution is provided in advance.
Settings. To evaluate the robustness of InvCF and baselines over various popularity distributions, we randomly select three long-tailed subsets of interactions as the test sets to mimic the popularity distribution shift, with each subset containing 10% interactions. Concretely, we first rank items in descending order and divide them into 50 groups according to their popularity. Then, for the i-th group, $N_i = N_0 \cdot \gamma^{-\frac{i-1}{49}}$ interactions are sampled out to generate the test set. Here, N_0 is the maximum number of interactions among all the groups in the test, and γ describes the long-tail degree. A smaller γ indicates a stronger distribution shift and a more uniform popularity distribution compared to the training data. Besides, the test splits for the validation set are equally long-tailed as the train set, *i.e.,* randomly split the remaining interactions into training, and validation sets (60% : 10%). Figure 9 shows the popularity distributions of the training and three test sets on Tencent.
Results. Table 2 reports the comparison of performance on all the baselines with different levels of long-tail degree. The best-performing methods per test are bold and starred, while the strongest baselines are underlined; Imp.% measures the relative improvements of BC loss over the strongest baselines. We observe that:

- **InvCF consistently and significantly outperforms the state-of-the-art baselines in terms of all metrics across all popularity distributions.** Specifically, InvCF achieves remarkable improvements compared to the best baselines by 10.2% and 5.9% *w.r.t.* NDCG@20 on average over the MF and LightGCN backbones, respectively. We attribute the robustness of InvCF to distilling the invariant preference information, regardless of changes in popularity distributions.
- **Entangled models behave in an unstable manner when the popularity distribution shifts.** Compared to CD^{2}AN and sDRO, popularity debiasing methods show a limited enhancement of recommendation quality over backbone models, reflecting a lack of generalization capability when the target prior is unknown. When $\gamma = 2$, as expected, IPS-CN performs the second best as its unbiased test defaults to a uniform distribution. With a closer look at performance on the validation set, debiasing models tend to gain improvements in tests by sacrificing the fitness on training data. In contrast, benefiting from no implicit assumption on the test data, CD^{2}AN, sDRO and our InvCF can substantially boost the performance over backbone models by a large margin. These observations serve to corroborate our study in Section 2 that entangled representations and strict requirements of prior information are two deficiencies of current debiasing approaches.

4.1.2 *Evaluations on Unbiased Test Sets*.

Motivation. Offline evaluation on collaborative filtering is challenging because of the missing-not-at-random condition in real-world recommender systems. Unbiased evaluation, where its test



(a) BPR loss (b) Softmax loss (c) InvCF-i (d) InvCF

Figure 4: 3D Visualizations of item representations learned by MF backbone model on Yelp2018. Subfigures (a-d) showcase the preference and property representations of the identical head user as red and blue stars, respectively. In each subfigure, representations of the head user’s all historical items are projected on the unit sphere. The brightness of the color indicates the popularity degree, while red and blue dots denote preference and popularity representations of items, respectively. More visualization results can be found in Appendix A.4.

Table 3: The performance comparison on Yahoo!R3 and Coat.

	Yahoo!R3				Coat			
	MF		LightGCN		MF		LightGCN	
	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
Backbone	0.1063	0.0476	0.1478	0.0686	0.0741	0.0361	0.2658	0.1574
+ sam+reg	0.1198	0.0548	0.1498	0.0693	0.2303	0.1869	0.2659	0.3569
+ IPS-CN	0.1081	0.0487	0.1331	0.0612	0.1700	0.1377	0.2474	0.1771
+ CausE	0.1252	0.0573	0.1490	0.0693	0.2004	0.1713	0.2479	0.1689
+ MACR	0.1243	0.0539	0.1499	0.0691	0.0798	0.0358	0.0939	0.0584
+ sDRO	0.1390	0.0661	0.1426	0.0660	0.2012	0.1767	0.2415	0.1790
+ CD^{2}AN	0.1451	0.0690	0.1397	0.0638	0.2325	0.1885	0.2245	0.1708
+ InvCF	**0.1566***	**0.0732***	**0.1515***	**0.0718***	**0.2672***	**0.2059***	**0.2686***	**0.1819***
Imp.%	7.93%	6.09%	1.07%	3.61%	14.43%	9.23%	1.01%	1.62%

Table 4: The performance comparison on Douban Movie.

	MF			LightGCN		
	HR	Recall	NDCG	HR	Recall	NDCG
Backbone	0.3509	0.0289	0.0552	0.3569	0.0294	0.0558
+ sam+reg	0.3584	0.0303	0.0577	0.3569	0.0307	0.0575
+ IPS-CN	0.2844	0.0216	0.0403	0.3213	0.0268	0.0522
+ CausE	0.3587	0.0300	0.0579	0.3640	0.0310	0.0589
+ MACR	0.3559	0.0303	0.0584	0.3620	0.0310	0.0595
+ sDRO	0.3670	0.0303	0.0553	0.3707	0.0324	0.0618
+ CD^{2}AN	0.3602	0.0296	0.0553	0.3770	0.0327	0.0604
+ InvCF	**0.3757***	**0.0321***	**0.0604***	**0.3897***	**0.0343***	**0.0635***
Imp.%	2.37%	5.94%	3.42%	3.37%	5.86%	2.75%

Table 5: The performance comparison on Meituan.

	MF			LightGCN		
	HR	Recall	NDCG	HR	Recall	NDCG
Backbone	0.5490	0.2343	0.2250	0.5760	0.2545	0.2574
+ sam+reg	0.5518	0.2358	0.2272	0.5797	0.2590	0.2664
+ IPS-CN	0.5311	0.2207	0.2046	0.5592	0.2448	0.2449
+ CausE	0.5665	0.2456	0.2409	0.5849	0.2615	0.2619
+ MACR	0.5583	0.2368	0.2129	0.5779	0.2587	0.2522
+ sDRO	0.5922	0.2648	0.3002	0.5929	0.2279	0.2289
+ CD^{2}AN	0.5914	0.2466	0.2664	0.5751	0.2509	0.2634
+ InvCF	**0.5954***	**0.2780***	**0.3073***	**0.6132***	**0.2642***	**0.2778***
Imp.%	0.54%	4.98%	2.37%	3.42%	1.03 %	4.28%

set is composed of items selected at random rather than by users, is considered an ideal offline test for eliminating the impact of CF models. Here we conduct experiments on widely used missing-complete-at-random datasets: Yahoo!R3 and Coat.

Results. As Table 3 depicts that InvCF steadily superior over all baselines *w.r.t.* all metrics on Yahoo!R3 and Coat. Specifically, compared to the state-of-the-art baseline, it achieves substantial gains on the MF backbones over Yahoo!R3 and Coat in terms of Recall@20 by 7.93% and 14.43%, respectively. In contrast, baselines perform unstably across datasets. Consistent with our study, this validates that InvCF successfully parameterizes the invariance and disentanglement principle, resulting a popularity-disentangled representations.

4.1.3 *Evaluations on Temporal split and Out-of-distribution Test Sets*.

Motivation. In real-world applications, popularity distribution dynamically changes over time. For a comprehensive comparison, we take two time-related evaluations into consideration. On Douban Movie, we divide the historical interactions into the training, validation, and test sets according to the timestamps (7:1:2). On Meituan, following the settings in [43], the user interactions during weekdays are regarded as the training (60%) and validation (10%) sets, while user purchases during the weekend are used as the test set (30%). See popularity distributions of each sets in Appendix A.1.

Results. Tables 4 and 5 clearly show, InvCF yields a consistent boost compared to the SOTA baselines. This indicates that InvCF endows the backbone models with better robustness and generalization ability against changes in the popularity distribution that are caused by time. Considering the empirical success of InvCF on test evaluations, we believe that InvCF provides a promising research direction to cope with popularity distribution shifts.

4.2 Visualizations of Representations (RQ2)

To visualize the latent representation space and evaluate the effectiveness of two principles in InvCF, we train toy recommenders on Yelp2018 using the MF backbone whose embedding size is three. We compare InvCF with two variants: InvCF-i, which disables the augmentation module, and InvCF-d, which deactivates the disentangling module. In Figure 4, 3-dimensional normalized item preference and popularity representations under different CF models (*i.e.,* MF+BPR, MF+Softmax [52], MF+InvCF-i, MF+InvCF) are illustrated on a 3D unit sphere. In Figure 5, we further summarize the distributions of angles between the preference and popularity representations learned by different CF models. We observe that:

- **Item preference and popularity representations learned by BPR and SSM are chaotically distributed and difficult to distinguish**, as shown by Figures 4 and 10. This validates that BPR and SSM extract entangled representations, further resulting in learning a suboptimal latent space.
- **Item preference and popularity representations learned by InvCF are disentangled and discriminative.** Figure 5 reveals that angles learned by InvCF between preference and popularity

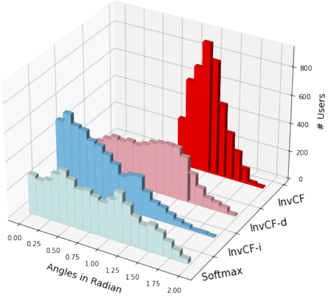


Figure 5: Distribution of angles between preference and popularity representations.

Table 6: Ablation Study on Yahoo!R3 dataset.

	HR	Recall	NDCG	#epoch
Softmax	0.2224	0.1470	0.0688	264.5
InvCF-i	0.2195$^{-1.30\%}$	0.1457$^{-0.88\%}$	0.0701$^{+1.89\%}$	304.8
InvCF-d	0.2241$^{+0.76\%}$	0.1464$^{-0.41\%}$	0.0696$^{+1.16\%}$	419.1
InvCF	**0.2333**$^{+4.90\%}$	**0.1566**$^{+6.53\%}$	**0.0732**$^{+6.40\%}$	164.2

representations tend to be more concentrated, leading to stable latent spaces. This clearly shows that InvCF not only effectively learns popularity-disentangled representations but also successfully distills the invariant features that causally determine the interactions. We attribute this breakthrough to representation-level augmentation and disentangling modules.

4.3 Study on InvCF (RQ3)

Ablation Study. Jointly analyzing the results in Figures 4, 5, 10, and Table 6, we observe that:

- **From a performance perspective, invariance and disentanglement principles are critical for InvCF and indispensable for one another.** In particular, InvCF-i and InvCF-d are susceptible to degrade the recommendation quality as shown in Table 6. Furthermore, with additional regularization terms, InvCF surprisingly converges on fewer training epochs, highlighting the fact that lacking any principle may undermine the model learning.
- **In terms of representations, InvCF incorporates disentangling and augmentation modules to decouple the invariant features and spurious popularity, further improving generalization ability.** Precisely, from Softmax to InvCF-i in Figure 4, the preference and popularity representations present a clear boundary, demonstrating the efficiency of the disentangling module. However, as Figure 5 shows, the overall angle distributions of InvCF-i, InvCF-d, and Softmax barely differ from one another. We ascribe the limited difference to employing one module alone, rather than two modules in cooperation, thus failing to acquire stable and high-quality feature spaces. Compared to InvCF-i and InvCF-d, InvCF drives an impressive breakthrough in representation learning, implying the cooperation of disentangling and augmentation modules is integral.

Effect of Disentangling/Augmentation Module. Assembling different discrepancy regularizers in the disentangling module will lead to performance fluctuations (See Table 10), while InvCF might be insensitive to various augmentation strategies (See Table 11).

5 RELATED WORK

Popularity Debiasing in recommender. Leading popularity debiasing approaches can roughly fall into three research lines:

- **Regularization-based frameworks** [1, 6, 11, 59] regulate the trade-off between accuracy and coverage with additional penalty terms. ESAM [11] introduces self-training regularizers to handle the missing of target labels. ALS+Reg [1] leverages intra-list diversity (ILD) as the regularization.
- **Re-weighting methods** [9, 12, 13, 36, 47] re-weight the loss of training instances by inverse propensity scores. Recent studies explore learning unbiased propensity estimators (AutoDebias [9], BRD [12]) instead of directly adopting the observed frequencies (IPS-CN [13]) to lower the variance of propensities.
- **Causal embedding learning** [5, 14, 27, 42, 50, 56, 57] utilizes counterfactual inference to mitigate the effect of item popularity. DecRS [42] uses backdoor adjustment to eliminate the effect of imbalanced item group distribution. PDA [56] adopts do-calculus [20] to remove the confounding popularity bias.

Out-of-distribution (OOD) Generalization. To devise stable models and address the problem of distribution shifts, s-DRO [51] adopts Distributionally Robust Optimization (DRO) framework, CD^{2}AN [10] disentangles item property representations from popularity under co-training networks, while BC Loss [55] incorporates bias-aware margins to achieve better generalization ability. Another line of research incorporates causal inference and discovery into OOD generalization (CausPref [16], COR [43]).

Disentangled Representation Learning in CF. To disentangle user or item intents into a finer granularity, current CF models assemble different discrepancy metrics including distance correlation (DGCF [46], DICE [57]), Pearson Correlation Coefficient (CD^{2}AN [10]), Maximum Mean Discrepancy (D2Rec [38]) and L_1, L_2 normalization (DICE [57]). Another line of research utilizes Variational Auto Encoders (VAE) to factorize latent features [28, 32, 48].

Data Augmentation in CF. Data augmentation is a popular strategy in recommender systems to deal with cold-start [8] or data sparsity problems [41]. However, semantic augmentation in CF remains stagnant. To the best of our knowledge, we are among the first to undertake representation level data augmentation to enhance the popularity generalization ability. We believe that this work provides a promising research line to deal with the OOD problem and will shed light on future work.

6 CONCLUSION

Leading popularity debiasing methods in collaborative filtering are still far from resolving the recommender's vulnerability to popularity distribution shift. In this work, we proposed a novel learning strategy, InvCF, that extracts invariant and popularity-disentangled features to enhance popularity generalization ability. Grounded by extensive test evaluations and real-world visualization studies, InvCF steadily outperforms the SOTA baselines by learning disentangled representation spaces. A worthwhile direction for future work is to extend InvCF to handle generic distribution shifts in recommender systems. We believe that InvCF will inspire research to incorporate latent space disentanglement and augmentation.

ACKNOWLEDGMENTS

This research is supported by the Sea-NExT Joint Lab, the National Natural Science Foundation of China (9227010114), the University Synergy Innovation Program of Anhui Province (GXXT-2022-040), and CCCD Key Lab of Ministry of Culture and Tourism.

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning-to-Rank Recommendation. In *RecSys*.

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing Popularity Bias in Recommender Systems with Personalized Re-Ranking. In *FLAIRS Conference*.

[3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

[4] Yoshua Bengio and Jean-Sébastien Senecal. 2003. Quick Training of Probabilistic Neural Nets by Importance Sampling. In *AISTATS*.

[5] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *RecSys*.

[6] Ludovico Boratto, Gianni Fenu, and Mirko Marras. 2021. Connecting user and item perspectives in popularity debiasing for collaborative recommendation. *Inf. Process. Manag.* 58, 1, 102387.

[7] Léon Bottou, Jonas Peters, Joaquin Quiñonero Candela, Denis Xavier Charles, Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Y. Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: the example of computational advertising. *J. Mach. Learn. Res.* 14, 1, 3207–3260.

[8] Dong-Kyu Chae, Jihoo Kim, Duen Horng Chau, and Sang-Wook Kim. 2020. AR-CF: Augmenting Virtual Users and Items in Collaborative Filtering for Addressing Cold-Start Problems. In *SIGIR*.

[9] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. AutoDebias: Learning to Debias for Recommendation. In *SIGIR*.

[10] 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 *SIGIR*.

[11] 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 *SIGIR*.

[12] Sihao Ding, Peng Wu, Fuli Feng, Yitong Wang, Xiangnan He, Yong Liao, and Yongdong Zhang. 2022. Addressing Unmeasured Confounder for Recommendation with Sensitivity Analysis. In *KDD*.

[13] Alois Gruson, Praveen Chandar, Christophe Charbuillet, James McInerney, Samantha Hansen, Damien Tardieu, and Ben Carterette. 2019. Offline Evaluation to Make Decisions About Playlist Recommendation Algorithms. In *WSDM*.

[14] Priyanka Gupta, Ankit Sharma, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. 2021. CauSeR: Causal Session-based Recommendations for Handling Popularity Bias. In *CIKM*. ACM, 3048–3052.

[15] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*.

[16] Yue He, Zimu Wang, Peng Cui, Hao Zou, Yafeng Zhang, Qiang Cui, and Yong Jiang. 2022. CausPref: Causal Preference Learning for Out-of-Distribution Recommendation. In *WWW*.

[17] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. 2018. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*.

[18] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2020. A Re-visit of the Popularity Baseline in Recommender Systems. In *SIGIR*.

[19] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2018. Unbiased Learning-to-Rank with Biased Feedback. In *IJCAI*.

[20] Henry E. Kyburg Jr. 2005. Judea Pearl, *Causality*, Cambridge University Press (2000). *Artif. Intell.* 169, 2, 174–179.

[21] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8, 30–37.

[22] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *KDD*.

[23] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. 2021. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*.

[24] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented Fairness in Recommendation. In *WWW*.

[25] Dawen Liang, Laurent Charlin, and David M Blei. 2016. Causal inference for recommendation. In *Causation: Foundation to Application, Workshop at UAI. AUAI*.

[26] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *WWW*.

[27] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A General Knowledge Distillation Framework for Counterfactual Recommendation via Uniform Data. In *SIGIR*.

[28] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *NeurIPS*.

[29] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation. In *CIKM*.

[30] Benjamin M. Marlin and Richard S. Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In *RecSys*.

[31] Zaiqiao Meng, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2020. Exploring Data Splitting Strategies for the Evaluation of Recommendation Models. In *RecSys*.

[32] Preksha Nema, Alexandros Karatzoglou, and Filip Radlinski. 2021. Disentangling Preference Representations for Recommendation Critiquing with ß-VAE. In *CIKM*.

[33] Steffen Rendle. 2021. Item Recommendation from Implicit Feedback. *CoRR* abs/2101.08769.

[34] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. *CoRR* abs/1205.2618.

[35] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased Recommender Learning from Missing-Not-At-Random Implicit Feedback. In *WSDM*.

[36] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. In *ICML*.

[37] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. 2021. Towards Out-Of-Distribution Generalization: A Survey. *CoRR* abs/2108.13624.

[38] Paras Sheth, Ruocheng Guo, Lu Cheng, Huan Liu, and K. Selçuk Candan. 2022. Causal Disentanglement with Network Information for Debiased Recommendations. *CoRR* abs/2204.07221.

[39] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-Based Social Recommendation via Dynamic Graph Attention Networks. In *WSDM*.

[40] Harald Steck. 2018. Calibrated recommendations. In *RecSys*.

[41] Qinyong Wang, Hongzhi Yin, Hao Wang, Quoc Viet Hung Nguyen, Zi Huang, and Lizhen Cui. 2019. Enhancing Collaborative Filtering with Generative Augmentation. In *KDD*. ACM, 548–556.

[42] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. In *KDD*.

[43] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. 2022. Causal Representation Learning for Out-of-Distribution Recommendation. In *WWW*.

[44] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *KDD*.

[45] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*.

[46] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *SIGIR*.

[47] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly Robust Joint Learning for Recommendation on Data Missing Not at Random. In *ICML*.

[48] Yifan Wang, Suyao Tang, Yuntong Lei, Weiping Song, Sheng Wang, and Ming Zhang. 2020. DisenHAN: Disentangled Heterogeneous Graph Attention Network for Recommendation. In *CIKM*.

[49] Zimu Wang, Yue He, Jiashuo Liu, Wenchao Zou, Philip S. Yu, and Peng Cui. 2022. Invariant Preference Learning for General Debiasing in Recommendation. In *KDD*.

[50] 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 *KDD*.

[51] Hongyi Wen, Xinyang Yi, Tiansheng Yao, Jiaxi Tang, Lichan Hong, and Ed H Chi. 2022. Distributionally-robust Recommendations for Improving Worst-case User Experience. In *WWW*.

[52] Jiancan Wu, Xiang Wang, Xingyu Gao, Jiawei Chen, Hongcheng Fu, Tianyu Qiu, and Xiangnan He. 2022. On the Effectiveness of Sampled Softmax Loss for Item Recommendation. *CoRR*.

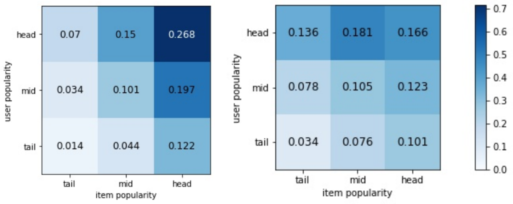
[53] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge J. Belongie, and Deborah Estrin. 2018. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *RecSys*.

[54] Fajie Yuan, Xiangnan He, Haochuan Jiang, Guibing Guo, Jian Xiong, Zhezhao Xu, and Yilin Xiong. 2020. Future Data Helps Training: Modeling Future Contexts for Session-based Recommendation. In *WWW*.

[55] An Zhang, Wenchang Ma, Xiang Wang, and Tat seng Chua. 2022. Incorporating Bias-aware Margins into Contrastive Loss for Collaborative Filtering. In *NeurIPS*.

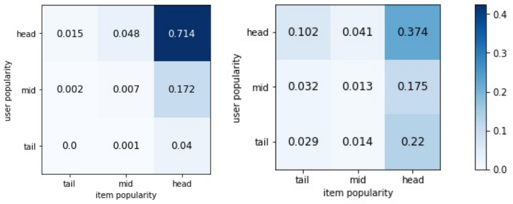
[56] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity

Bias in Recommendation. In *SIGIR*.
[57] 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 *WWW*.
[58] Ziwei Zhu, Yun He, Xing Zhao, and James Caverlee. 2021. Popularity Bias in Dynamic Recommendation. In *KDD*.
[59] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-Opportunity Bias in Collaborative Filtering. In *WSDM*.



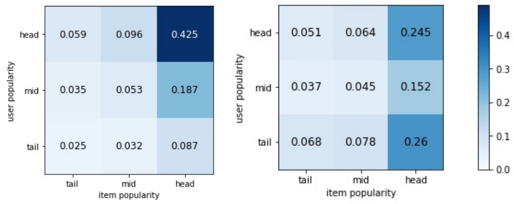
(a) Training Set (b) Unbiased Test Set

Figure 6: Popularity distribution on Coat.



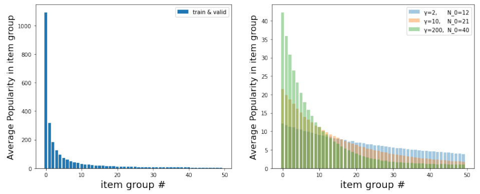
(a) Training Set (b) Temporal Split Test Set

Figure 7: Popularity distributions on Douban Movie.



(a) Training Set (Weekday) (b) Test Set (Weekend)

Figure 8: Popularity distributions on Meituan.



(a) Training Set (b) Synthetic Test Set

Figure 9: Popularity distributions on Tencent.

A EXPERIMENTAL SETTINGS

A.1 Datasets.

We conduct experiments on both real-world benchmark datasets and one synthetic dataset.

(1) **Yahoo!R3 [30]** & **Coat [36]**: These two datasets are obtained from the music and coat recommendation services. Both Yahoo!R3 and Coat are specially designed to evaluate on unbiased settings. The training data, considered as a normal biased dataset, contains ratings for items selected by users. The testing data is collected from an online survey, where each user has to rate on randomly selected items.
(2) **Douban Movie [39]**: This dataset is collected from a popular movie review website Douban in China. We split it based on the temporal splitting strategy [31].
(3) **Meituan [43]**: Meituan is a public food recommendation dataset. The shifts of consumption levels from weekdays to weekends causes popularity drift in items.
(4) **Yelp2018 [15]**: This dataset is a subset of Yelp's businesses, reviews, and user data. We randomly sample 10% of interactions from the original dataset to do the 3D visualization.
(5) **Tencent [54]**: The original dataset is collected from Tencent short-video platform. We generate three testing datasets to explore model performance on dataset with different distribution shifts. The popularity distributions are shown in Figure 9.

Figures 3, 6-9 show the popularity distribution shift from training to testing. In the stage of pre-processing data, explicit feedback (Yahoo!R3, Coat) are converted into implicit feedback. We treat the items rated with four or higher scores (five in total) as positive feedback and the remaining as negative feedback. Following the standard 10-core setting [44, 56], we filter out items and users with less than ten interactions for all five datasets.

A.2 Baselines

We compare with popular debiasing strategies in various research lines: sam+reg [6], IPS-CN [13], and CausE [5], MACR [50]. We also compare with domain generalization baselines: CD^{2}AN [10] and s-DRO [51].

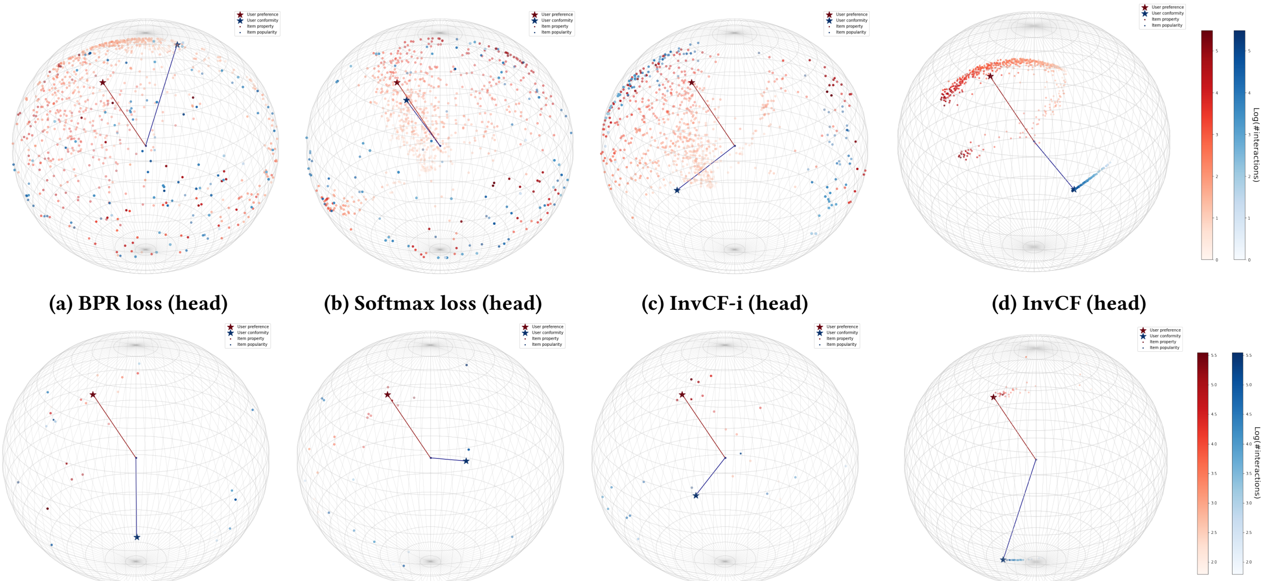
- **sam+reg** [6]: This method comprises two parts: training examples mining (sam) to balance the distribution of observed and unobserved items, and regularized optimization (reg) to minimize biased correlations between predicted user-item relevance and item popularity.
- **IPS-CN** [13]: IPS [36] re-weights each training instance with item popularity to eliminate popularity bias. IPC-CN adds normalization on plain IPS to achieve lower variance.
- **CausE** [5]: This method leverages a small unbiased dataset to simulate the training process under a fully random recommendation policy.
- **MACR** [50]: This method assigns popularity bias to the causal effects of item popularity on the prediction scores. To this end, it introduces two additional modules to capture the effects of item popularity and user conformity and injects the results into the final prediction scores.
- **CD^{2}AN** [10] This model uses Pearson coefficient correlation to disentangle item property representations from item popularity representation and introduces additional unexposed items to align item popularity distributions between hot and long-tail items.
- **s-DRO** [51]: This model adds streaming optimization improvement to the Distributionally Robust Optimization (DRO) framework to mitigate the amplification of Empirical Risk Minimization (ERM) on popularity bias.

A.3 Parameter Settings and Training Cost

Table 8 and 9 show the parameter settings for InvCF and baselines, respectively. We further record the training cost of InvCF compared to selected baselines in Table 7.

A.4 Additional Experiments

3D Visualizations We equally divide users into three subgroups: head, mid, and tail, according to their interaction frequencies. We then select the most popular one from the head group and another



(a) BPR loss (head) (b) Softmax loss (head) (c) InvCF-i (head) (d) InvCF (head)

(e) BPR loss (tail) (f) Softmax loss (tail) (g) InvCF-i (tail) (h) InvCF (tail)

Figure 10: Additional 3D Visualizations of item representations learned by MF backbone model on Yelp2018.

Table 9: Hyper-parameters search spaces for baselines.

	Hyper-paramete space

MF & LightGCN	lr = 5e-4, batch size ~ {64, 128, 256, 512, 1024, 2048}
	No. negative samples ~ {128, 256, 512, inbatch}

sam-reg	Log(rweight) ~ {-1,-2,-3,-4,-5,-6, -7, -8}
CausE	Log(cf_pen) ~ {-1,-2,-3,-4,-5,-6, -7, -8}
MACR	Log(c) ~ {-1,-2,-3,-4,-5,-6, -7, -8}
CD^{2}AN	τ ~ {0.03, 0.05, 0.07, 0.11, 0.13, 0.15, 0.17} Log(λ_1) ~ {-1,-2,-3,-4,-5}, Log(λ_2) ~ {-1,-2,-3,-4,-5}
sDRO	τ ~ {0.03, 0.05, 0.07, 0.11, 0.13, 0.15, 0.17} t_1, t_2 ~ {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} Log($dro\_temperature$) ~ {-1,-2,-3}, Log($streaming\_lr$) ~ {0, -1,-2}

Table 10: Ablation Study on Different Discrepancy Metrics.

	NDCG@20			
	$\gamma = 200$	$\gamma = 10$	$\gamma = 2$	Validation
dCor	0.0342	0.0221	0.0165	0.0748
MMD	$0.0355^{+3.80\%}$	$0.0241^{+9.05\%}$	$0.0182^{+10.30\%}$	0.0475
L_2	$0.0279^{+18.42\%}$	$0.0178^{-24.16\%}$	$0.0135^{-18.18\%}$	0.0863

Table 11: Ablation Study on Different Data Augmentation Approaches.

	NDCG@20		
	$\gamma = 200$	$\gamma = 10$	$\gamma = 2$
Random Permutation : randomly shuffle the orders of popularity embeddings inside the current batch.	0.0342	0.0221	0.0165
Head Group : for each user/item, randomly sample a popularity embedding from head group users/items.	0.0335	0.0215	0.0163
Tail Group : for each user/item, randomly sample a popularity embedding from tail group users/items.	0.0332	0.0214	0.0161
Different Groups : *i.e.* for a user in head group, randomly sample a popularity embedding from mid and tail groups.	0.0333	0.0217	0.0162

Table 7: Training cost on Yahoo!R3 (seconds per epoch/in total).

	Backbone	+sam+reg	+IPS-CN	+CausE	+MACR	sDRO	CD^{2}AN	InvCF
MF	1.5 / 230	1.5 / 74	1.3 / 538	1.6 / 86	1.2 / 119	2.0 / 918	2.5 / 960	3.3 / 541
LightGCN	1.8 / 232	1.8 / 232	1.7 / 66	2.0 / 328	1.9 / 369	4.7 /66	2.3 / 378	4.5 / 738

Table 8: Model architectures and hyperparameters for InvCF.

	InvCF hyper-parameters						
	α	λ_1	λ_2	τ	lr	batch size	No. negative samples
MF							
Yahoo!R3	1e-4	1e-5	1e-3	0.15	5e-4	1024	128
Coat	1e-3	1e-6	1e-2	0.09	5e-4	1024	64
Douban	1e-2	1e-5	1	0.13	5e-4	1024	128
Meituan	1	1e-4	0	0.03	5e-4	1024	128
Tencent	1	1	1e-8	0.09	5e-4	1024	128
LightGCN							
Yahoo!R3	1	1e-1	1e-7	0.18	5e-4	1024	64
Coat	1	1e-4	0	0.95	5e-4	64	inbatch
Douban	1e-4	1e-2	1e-2	0.13	5e-4	1024	128
Meituan	1e-1	1e-6	0	0.03	5e-4	1024	inbatch
Tencent	1e-2	1e-2	1e-6	0.17	5e-4	1024	inbatch

from the tail group to illustrate the feature representations of their interacted items in 3D space in Figure 10.

Effect of Disentangling Module. Table 10 shows the performance comparison among deploying different discrepancy metrics in the disentangling module, including Maximum Mean Discrepancy (MMD), distance correlation and L_2 normalization.

Effect of Augmentation Module. Table 11 illustrates the performance comparison among various augmentation strategies.