Adversarial Propensity Weighting for Debiasing in Collaborative Filtering

Kuiyu Zhu , **Tao Qin**[*] , **Pinghui Wang** and **Xin Wang**
MOE Key Laboratory for Intelligent Networks and Network Security, Xi'an Jiaotong University
{kyzhu, wx508810851}@stu.xjtu.edu.cn, {qin.tao, phwang}@mail.xjtu.edu.cn

Abstract

Debiased recommendation focuses on alleviating the negative impact of various biases on recommendation quality to achieve fairer personalized recommendations. Current research mainly relies on propensity score estimation or causal inference methods to alleviate selection bias; at the same time, research on prevalence bias has proposed a variety of methods based on causal graphs and contrastive learning. However, these methods have shortcomings in dealing with unstable propensity score estimates, bias interactions, and decoupling of interest and bias signals, which limits the performance improvement of recommender systems. To this end, this paper proposes APWCF, a collaborative filtering debiased method that combines dynamic propensity modeling and adversarial learning. APWCF solves the problem of high variance in propensity scores through the dynamic propensity factor, and decouples user interests and bias signals through the adversarial learning to effectively remove multiple biases. Experiments show that APWCF significantly outperforms existing methods across various benchmark datasets from different domains. Compared with the current optimal baseline PDA, Recall@10 and NDCG@10 improve by 0.10%-5.42% and 1.01%-8.60% respectively.

1 Introduction

Recommender systems (RS) are an effective tool to alleviate information overload because they can provide personalized content based on user preferences and significantly improve user experience [Bakhshizadeh, 2024]. However, user behavior data is usually observational data rather than experimental data [Chen *et al.*, 2023], which leads to common biases in RS, such as popularity bias, selection bias, exposure bias, and conformity bias [Yang *et al.*, 2024]. Among them, popularity bias and selection bias are the most common. For example, recommender systems often tend to prioritize items that are already popular, such as popular movies or best-selling books, a phenomenon known as popularity bias. This has led to a contradiction between the personalized recommendations pursued by recommender systems for a long time and the item popularity bias [Wei *et al.*, 2021; Ning *et al.*, 2024]. In addition, the existence of these biases not only limits the diversity of recommendations and damages user experience, but may also exacerbate the formation of information cocoons [Wu *et al.*, 2024]. Therefore, studying debiasing algorithms has become one of the important directions in the current field of RS [Yalcin and Bilge, 2022].

Existing recommendation methods mainly focus on solving popularity bias and selection bias. In response to selection bias, some works [Schnabel *et al.*, 2016; Bonner and Vasile, 2018] use the inverse propensity score (IPS) to mitigate the bias, or use causal inference methods to explore the potential causal mechanisms in the recommender systems to analyze the source of bias and intervene. For example, IPS [Schnabel *et al.*, 2016] improves the fairness of recommendation results by assigning inverse propensity score weights to user behavior data and adjusting the selection distribution of observed data. CausE [Bonner and Vasile, 2018], on the other hand, models user behavior as the generation process of observed variables by establishing a causal inference model to alleviate selection bias from its root. In addition, some other works [Zhang *et al.*, 2021; Wei *et al.*, 2021; Lee *et al.*, 2023; Ning *et al.*, 2024] focus on eliminating popularity bias. For example, MACR [Wei *et al.*, 2021] introduces causal graphs to describe the key causal relationships in recommendations, analyzes the fundamental mechanism of popularity bias, and proposes a model-independent counterfactual reasoning framework. MACR [Wei *et al.*, 2021] trains the recommendation model through multi-task learning and eliminates the direct impact of popularity in the reasoning stage, thereby effectively alleviating popularity bias. Unlike MACR [Wei *et al.*, 2021], PDA [Zhang *et al.*, 2021] believes that popularity bias is not always negative and can be leveraged. Therefore, PDA [Zhang *et al.*, 2021] applies a method of deconfounding and adjusting popularity bias to eliminate the negative impact of bias during training, and uses causal inference to control the new popularity bias. uCTRL [Lee *et al.*, 2023] optimizes user and item representations from the perspective of contrastive learning, proposes an unbiased alignment function and an improved inverse propensity weighting method to effectively eliminate the popularity bias of users and items, and improves the quality of representation learn-

[*] Corresponding author

ing through alignment and uniformity functions. PPAC [Ning *et al.*, 2024] introduces the concept of personal popularity by measuring the similarity of user-item interaction sets. In order to solve the problem that traditional global popularity bias cannot reflect the preferences of individual users, a counterfactual reasoning framework for personal popularity perception is designed. In addition, global popularity and personal popularity are jointly considered in the recommendation process to accurately control the impact of bias and significantly improve the personalization and accuracy of recommendations.

However, the existing debias methods still have the following problems. First, the existing work relies on the IPS method for accurate estimation of the propensity score, but because the calculation of the propensity score is affected by the imbalance of the observed data, its estimation is prone to biases or high variance, especially in long-tail distribution or non-random missing (MNAR) scenarios. This instability will directly affect the debiasing performance. Second, existing methods mostly focus on the removal of a single bias, such as selection bias or popularity bias, while ignoring the potential interactions among these biases. The sources of bias in the recommender systems are complex, and the processing of a single bias cannot fully solve the problem of bias amplification. Third, most methods entangle interest signals and bias signals during the learning process of user and item representations, and fail to decouple the two, making it difficult for the recommendation model to accurately capture the user's true interests and thus limiting the diversity and robustness of the recommendations.

To address these problems, in this paper, we propose APWCF, a collaborative filtering debiasing method that combines dynamic propensity modeling and adversarial learning, which aims to eliminate both popularity bias and selection bias. Specifically, APWCF captures the interactive propensity characteristics of users and items through the Dynamic Propensity Factor (DPF) module, and combines it with the Bias Adversarial Learning (BAL) module to effectively achieve joint modeling and removal of multiple biases. The DPF module dynamically adjusts the propensity score estimation to address the issues of high variance and instability in traditional propensity weighting methods; the BAL module separates user interest and item bias signals through bias discriminators and gradient reversal techniques, ensuring that the recommendation model focuses on the user's true preferences and improves the personalization and diversity of recommendations. Our main contributions can be summarized as follows:

(1) We propose Dynamic Propensity Factor (DPF) to model dynamic propensity scores of users and items to alleviate issues of high variance and inaccurate propensity score estimation in traditional IPS methods and improve the robustness of the recommendation performance.

(2) We design Bias Adversarial Learning (BAL) that applies a bias discriminator and a gradient reversal mechanism to decouple user interests from item bias signals and mitigate the negative impact of bias signals.

(3) We evaluate APWCF on five benchmark datasets. Experimental results show that compared with existing debias methods, APWCF improves Recall@10 and NDCG@10 by 0.10%-5.42% and 1.01%-8.60% over the current best baseline PDA.

2 Related Work

In this section, we briefly introduce the research related to our work, including GNN-based collaborative filtering and debiased collaborative filtering.

2.1 GNN-based Collaborative Filtering

Recently, given the advantages of GNN in collaborative filtering for mining high-order interaction patterns, GNN-based recommendations have become the mainstream of research [Gao *et al.*, 2023], and researchers have proposed various GNN-based recommendation models [Wang *et al.*, 2019; He *et al.*, 2020]. Among them, the most typical works are NGCF [Wang *et al.*, 2019] and LightGCN [He *et al.*, 2020]. Specially, NGCF [Wang *et al.*, 2019] can achieve higher-order information aggregation by stacking multiple layers of graph neural networks, extracting higher-order information from bipartite graphs to fully explore the relationship between user-item behavior data. LightGCN [He *et al.*, 2020] learns user and item embedding through linear propagation on the user-item interaction graph, and performs weighted summation of the learned user and item embeddings to complete the final representation of users and items. Based on NGCF [Wang *et al.*, 2019], it removes feature transformation and nonlinear activation, and only retains neighborhood aggregation and propagation. This linear propagation method eliminates complex feature transformation and nonlinear activation, making the model more efficient and lightweight.

2.2 Debiased Collaborative Filtering

The existence of various biases in recommender systems (e.g., selection bias, popularity bias, unfairness) can easily diminish user satisfaction and may even further accelerate the formation of information cocoons [Zhao *et al.*, 2020; Li *et al.*, 2022]. In recent years, to alleviate or eliminate these biases, the research community has proposed various methods for specific biases.

For example, IPS [Schnabel *et al.*, 2016] utilizes inverse propensity scores to eliminate selection bias during model evaluation in recommender systems . CausE [Bonner and Vasile, 2018] leverages causal inference techniques to address selection bias, and introduces a domain-adaptive algorithm that learns from biased data to enhance recommendation performance. Different from the above two methods, REL [Saito *et al.*, 2020] aims to focus on exposure bias. REL [Saito *et al.*, 2020] proposes an ideal loss function tailored for exposure bias and introduces an unbiased estimator to optimize recommendations for predicting highly relevant items. It effectively addresses critical challenges, including the positive-unlabeled problem and the missing-not-at-random issue. Additionally, improves the bias-variance trade-off through a clipped estimator, enhancing recommendation effectiveness. In addition, PDA [Zhang *et al.*, 2021] argues that popularity bias is not always negative, and completely unbiased learning may remove beneficial patterns in the data. PDA [Zhang *et al.*, 2021] proposes a new framework that uses decontamination and causal intervention methods to deal with popularity bias. It eliminates the negative impact of popularity bias during training, and adjusts the recommendation results through causal inference to include moderate popularity bias, thereby improving the accuracy of recommendations. Unlike the traditional method of using inverse propensity weighting to solve popularity bias, MACR [Wei *et al.*, 2021] introduces causality, analyzes the source of popularity bias through causal graphs, and eliminates the bias through counterfactual inference in the reasoning stage, ensures recommendation results remain unaffected by errors in item attributes. In addition, from the perspective of contrastive representation learning, uCTRL [Lee *et al.*, 2023] designs an unbiased alignment function and an improved inverse propensity weighting method to eliminate popularity bias.

3 Methodology

3.1 Problem Definition

Collaborative filtering (CF) is a widely adopted recommendation technique that predicts a user's potential preference for unseen items based on their historical user-item interactions. Formally, we use $U = \{u_1, u_2, u_3, ..., u_m\}$ to represent the set of users and $I = \{i_1, i_2, i_3, ..., i_n\}$ to represent the set of items. $O = \{(u, i) \mid u \in U, i \in I\}$ represents the set of interactions between users and items, where (u, i) represents the interaction record between user u and item i. The goal of CF is to learn a prediction function $\hat{y} : U \times I \to \mathbb{R}$ such that $\hat{y}_{u,i}$ represents the preference score of user u for item i. The recommender systems generates a user's recommendation list by sorting items based on $\hat{y}_{u,i}$ values.

3.2 Overall Framework

The proposed APWCF framework is illustrated in Fig. 1. APWCF supports various backbones, including MF or LightGCN. LightGCN is used as an example to explain the framework, and consists of a primary task and an auxiliary task. Among them, the main task is the recommendation task, whose goal is to improve the recommendation performance by optimizing the embedding match between users and items. Auxiliary tasks include propensity score estimation and bias adversarial learning. Specifically, the former dynamically calculates the interaction propensity score between users and items to ensure that low-propensity samples are fully paid attention to during training. The latter separates user interest and item bias signals through the gradient reversal layer and bias discriminator to ensure that the recommendation model focuses on the user's real interest.

3.3 Dynamic Propensity Factor

Selection bias in recommender systems causes the model to prefer frequent interaction samples (such as popular items) and ignore low-frequency interaction samples (such as long-tail items). This tendency not only limits the diversity of recommendations, but also may amplify the negative impact of data bias on model training. To address this problem, we introduce propensity scores, which aim to measure the likelihood of interaction between users and items, and use them to adjust the weighted loss in recommendation tasks to mitigate the impact of selection bias.

Algorithm 1 APWCF Algorithm

```
Input: Interaction graph $G$, training epochs $T$, hyperparameters: $\lambda$, $\alpha$
Output: User/item embeddings $E^{(0)} = (E_u^{(0)}, E_i^{(0)})$
Initialize user/item embeddings $E_u^{(0)}$ and $E_i^{(0)}$ using Xavier initialization Construct the normalized adjacency matrix $A$
for $t = 1$ to $T$ do
    for each batch $(u, i, j)$ in training data do
        // Step 1: Main Task - Recommendation
        Perform message passing on graph $G$ using $A$ to obtain updated embeddings for users and items
        Calculate propensity scores $P(u, i)$ for user-item pairs using Eq. (1)
        Clip $P_{\text{clip}}(u, i)$ using Eq. (2)
        Compute the main recommendation loss $\mathcal{L}_{\text{main}}$ using Eq. (6)
        // Step 2: Auxiliary Task - Propensity Score Estimation
        Estimate propensity scores $P(u, i)$ and compute the propensity score estimation loss $\mathcal{L}_{\text{ps}}$ using Eq. (7)
        // Step 3: Auxiliary Task - Bias Adversarial Learning
        Generate $y_{\text{bias}}$ for items using Eq. (3)
        Apply the gradient reversal layer to item embeddings using Eq. (5)
        Use the bias discriminator to compute $\mathcal{L}_{\text{adv}}$ using Eq. (4)
        // Step 4: Total Loss Computation and Optimization
        Combine losses into the total loss $\mathcal{L}$ using Eq. (8)
        Backpropagate and update parameters
    end for
end for
return Final embeddings $E = (E_u, E_i)$
```

Propensity score calculation: The dynamic propensity factor calculates the propensity score based on the user embedding e_u and the item embedding e_i, as shown in Eq. (1).

$$P(u, i) = \sigma(\mathbf{e}_u^{\top} \mathbf{e}_i) \tag{1}$$

Clipping for stability: In addition, to avoid extreme values of the propensity score (such as close to 0 or 1) from interfering with the stability of the model training process and to ensure that the weighting effect is effectively played in the recommendation task, we clip the propensity score, as shown in Eq. (2).

$$P_{\text{clip}}(u, i) = \text{clip}(P(u, i), \text{min} = 1\text{e}{-6}, \text{max} = 1.0) \tag{2}$$

The clip function limits $P(u, i)$ to the interval $[1\text{e}{-6}, 1.0]$, ensuring that the propensity score is neither a minimum (close to 0) nor a maximum (close to 1). Avoid the adverse effects of numerical instability on the optimization process. The clipped propensity score $P_{\text{clip}}(u, i)$ is directly used in the

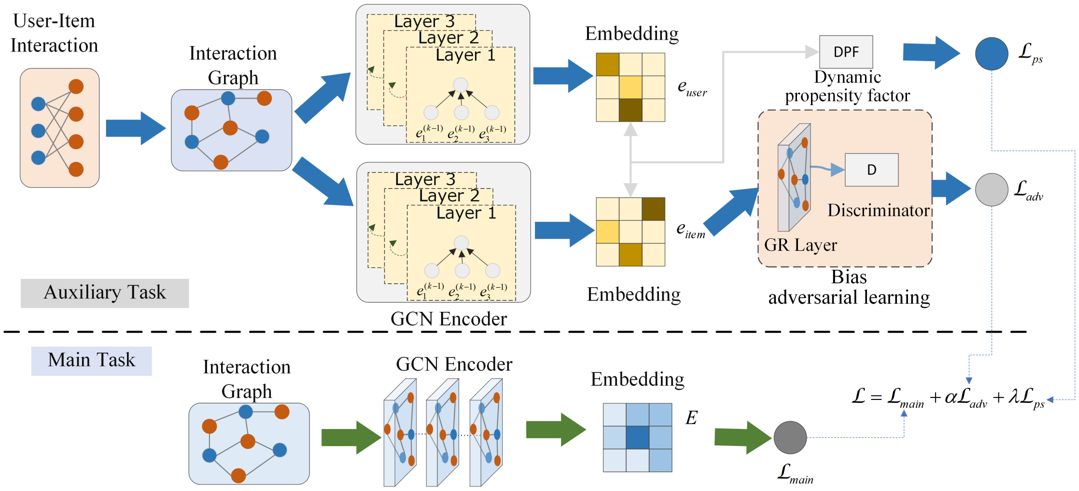


Figure 1: Framework of the proposed APWCF for collaborative filtering.

main loss function of the recommendation task as a weight factor to weight the interaction samples. Interaction samples with low propensity scores (such as long-tail items) are given higher weights, while samples with high propensity scores are given lower weights, thereby effectively alleviating the interference of selection bias on the recommendation results.

3.4 Bias Adversarial Learning

In recommendation tasks, user interest signals and item bias signals (such as popularity bias) are often mixed, which makes the recommendation system prone to excessive bias towards popular items, thereby damaging the recommendation effect of long-tail items. In order to explicitly decouple user interest and bias signals, we designed a bias adversarial learning module.

Discriminator: In order to mark the popularity deviation of items, the popularity label is generated according to the item interaction frequency $f(i)$ as shown in Eq. (3). Where $y_{bias}(i)$ represents the popularity label of item i and median(f) represents the median of the interaction frequencies $f(i)$ of all items. It is the value in the middle after sorting the interaction frequencies of all items from small to large. It is used to distinguish between popular items and non-popular items. In APWCF, we design the bias discriminator as a binary classification model, with the input being the item embedding e_i and the output being the predicted popularity label $\hat{y_{bias}}$.

$$y_{\text{bias}}(i) = \begin{cases} 1, & \text{if } f(i) \geq \text{median}(f) \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

Then in model training, the calculation of adversarial loss is shown in Eq. (4). where n is the number of items, $\hat{y}_{\text{bias}}^{(i)}$ is the predicted bias label for item i, and $y_{\text{bias}}^{(i)}$ is the true bias label.

$$\mathcal{L}_{\text{adv}} = \frac{1}{n}\sum_{i=1}^{n} \text{BCE}(\hat{y}_{\text{bias}}^{(i)}, y_{\text{bias}}^{(i)}) \tag{4}$$

Gradient reversal layer: To prevent the biased discriminator from negatively perturbing the recommendation task, and inspired by [Zhang *et al.*, 2023], we apply a gradient reversal layer, which reverses the gradient direction of the embedding during back-propagation. Where $\mathbf{e}_i^{\text{GRL}}$ is the embedding after gradient reversal, and α is a hyperparameter controlling the strength of the reversal.

$$\mathbf{e}_i^{\text{GRL}} = \mathbf{e}_i, \quad \nabla_{\mathbf{e}_i}\mathcal{L}_{\text{adv}} = -\alpha\nabla_{\mathbf{e}_i}\mathcal{L}_{\text{adv}} \tag{5}$$

The user interest signal is dominated by the dot product of user and item embeddings, and its goal is to maximize the user's preference for the target item. The bias signal is identified by the bias discriminator and suppressed in adversarial learning, ensuring that the item embedding does not carry popularity bias information.

3.5 Multi-task Training

The training of our APWCF combines the main recommendation task with two auxiliary tasks: the propensity score estimation task and the bias adversarial learning task. The main recommendation loss focuses on optimizing the matching of user interests and items to improve recommendation performance; the auxiliary tasks reduce the impact of selection bias through propensity score estimation and explicitly suppress the interference of popularity bias through adversarial learning. The three work together in the joint optimization process to ensure that the model can not only capture the real interests of users, but also effectively remove multiple biases, and ultimately achieve a balance between recommendation accuracy and fairness. Specifically:

Main task: The main recommendation loss of APWCF is based on the BPR loss and is weighted by the propensity score to mitigate the selectivity bias, as shown in Eq. (6). Specifically, in model training, the loss of the recommendation task is weighted using dynamic propensity scores to reduce the weight of high-propensity samples while increasing the influence of low-propensity samples, thereby effectively mitigating the selectivity bias. For example, high-propensity samples are given lower weights to avoid overfitting of frequent interactions; while low-propensity samples are given higher weights to increase the focus on long-tail items.

$$\mathcal{L}_{\text{main}} = -\sum_{(u,i,j)\in D} \ln \sigma(\mathbf{e}_u^\top \mathbf{e}_i - \mathbf{e}_u^\top \mathbf{e}_j) \cdot \frac{1}{P_{\text{clip}}(u,i)} \tag{6}$$

where (u, i, j) represents the positive and negative sample pairs of user u. And $\mathbf{e}_u$, $\mathbf{e}_i$, and $\mathbf{e}_j$ represent the embeddings of user u, positive sample i, and negative sample j, respectively. $\sigma(\cdot)$ is the Sigmoid function, and $P_{\text{clip}}(u, i)$ is the propensity score for the positive sample.

Propensity score estimation task: To assist the main task, APWCF introduces a propensity score estimation task, whose goal is to improve the estimation accuracy of the propensity score. The loss function of this task is the propensity score estimation loss, as shown in Eq. (7):

$$\mathcal{L}_{\text{ps}} = \frac{1}{|D|} \sum_{(u,i)} \text{BCE}(P_{\text{clip}}(u,i), y_{ui}) \tag{7}$$

where $|D|$ is the total number of interaction samples. And y_{ui} is the ground truth label for whether user u interacted with item i, and BCE denotes the binary cross-entropy loss.

Bias adversarial learning task: In addition, the adversarial bias learning task explicitly decouples the user interest signal from the popularity bias signal by optimizing the bias discriminator. The corresponding adversarial loss is shown in Eq. (4).

Finally, a joint optimization strategy is applied to minimize the total loss, as shown in Eq. (8):

$$\mathcal{L} = \mathcal{L}_{\text{main}} + \lambda \mathcal{L}_{\text{ps}} + \alpha \mathcal{L}_{\text{adv}} \tag{8}$$

where λ is the weight of the propensity score estimation loss and α is the strength of the adversarial learning module. We describe the key steps and overall process of the APWCF algorithm in detail in the form of pseudocode, as shown in Algorithm 1.

4 Experiments

In this section, we carry out extensive experiments to evaluate the effectiveness of the proposed model and answer the following research questions:

- **RQ1**: How significant are the improvements of our APWCF compared to state-of-the-art debiasing recommendation methods?
- **RQ2**: What is the contribution of various components in our framework to the overall performance?
- **RQ3**: To what extent does our APWCF exhibit debiasing ability?
- **RQ4**: What effects do hyper-parameters have on our APWCF?

4.1 Experimental Settings

Datasets. We conduct experiments on several public real-world benchmark datasets from various domains: ML-100K, Yahoo!R3, KuaiRec, Douban-book, and Yelp2018. Detailed statistics of these datasets are summarized in Tab. 1.

Dataset	#User	#Item	#Interaction	Field
ML-100K	943	1,682	74,817	Movies
Yahoo!R3	15,400	1,000	365,704	Musics
KuaiRec	7,176	10,612	1,153,787	Videos
Douban-book	12,861	22,296	598,421	Books
Yelp2018	31,668	38,048	8,827,696	Shopping

Table 1: Detailed datasets statistics.

Evaluation Metrics. To verify the effectiveness of the proposed method, we adopt two widely used evaluation metrics: Recall@K and NDCG@K.

Baselines. To comprehensively evaluate the effectiveness of the proposed method, we selected a variety of state-of-the-art models as baselines, including two backbones, MF [Rendle *et al.*, 2009] and LightGCN [He *et al.*, 2020], and several debiasing models, such as IPS [Schnabel *et al.*, 2016], CausE [Bonner and Vasile, 2018], REL [Saito *et al.*, 2020], PDA [Zhang *et al.*, 2021], MACR [Wei *et al.*, 2021], DICE [Zheng *et al.*, 2021], uCTRL [Lee *et al.*, 2023], and PPAC [Ning *et al.*, 2024] to ensure a comprehensive comparison.

Hyper-parameter settings. To ensure a fair comparison among models, for each recommendation model, we initialize the parameters with the Xavier [Glorot and Bengio, 2010] distribution and use Adam [Kingma and Ba, 2015] as the optimizer, with the learning rate set to 0.001. The embedding size of the user and the item is fixed at 64. A batch size of 2048 is used for all datasets. The L_2 regularization cofficient λ_{reg} and the layer of GCN is fixed to 0.0001 and 3 respectively. The number of training epochs is set to 500. To prevent overfitting, we implemented an early stopping strategy for all models. If the NDCG on the validation set does not improve for 20 consecutive epochs, the training is stopped. We also fine-tune other hyperparameters based on the hyperparameter range mentioned in the original paper. For APWCF, we fine-tune the hyperparameters λ, α, and β in {0.1, 0.5, 1, 2, 5, 10, 20, 50}, {0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 10}, and {0.01, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0}, respectively.

Implementation details. Detailed implementation details and training efficiency analysis can be found in Supplemental Materials B and C, which are available at https://github.com/GeneralRec/APWCF.

4.2 Overall Performance Comparison (RQ1)

To verify the effectiveness of our APWCF, we compare it with various existing debias methods. We evaluate it on multiple benchmark datasets across various domains and report the Recall and NDCG results in Tab. 2. The best results are highlighted in bold, and the second-best are underlined. “Base” indicates that no debiasing measures are applied and only the backbone itself is used as the model for the recommendation task. From the results, we observe that:

(1) We conducted experiments on five benchmark datasets based on different backbone models. For each dataset, we evaluated various debiasing methods based on two backbone models and used four metrics: Recall@K and NDCG@K (K=10, 20). We calculated a total of 40 performance metrics on the five datasets. APWCF achieved the best or second-best results in 34 metrics. Specially, on the Yelp2018 dataset,

Dataset	Backbone	Metric	Base	IPS	CausE	MACR	PDA	DICE	REL	uCTRL	PPAC	APWCF
ML-100K	MF	Recall@10	0.0850	0.0866	0.0252	0.0888	0.0938	0.0959	**0.0967**	0.0821	0.0951	0.0863
		Recall@20	0.1538	0.1478	0.0465	0.1619	0.1697	0.1701	**0.1716**	0.1389	0.1610	0.1626
		NDCG@10	0.0970	0.0981	0.0372	0.1037	0.1073	0.1081	**0.1096**	0.0818	0.1085	0.1005
		NDCG@20	0.1194	0.1169	0.0430	0.1267	0.1320	0.1320	**0.1335**	0.1013	0.1286	0.1245
	LightGCN	Recall@10	0.0949	**0.1036**	0.0402	0.0932	0.0978	0.0963	0.1018	0.0895	0.0930	0.1031
		Recall@20	0.1717	0.1803	0.0681	0.1706	0.1773	0.1585	0.1839	0.1510	0.1599	**0.1862**
		NDCG@10	0.1062	0.1152	0.0433	0.1064	0.1105	0.1020	0.1173	0.0931	0.1055	**0.1200**
		NDCG@20	0.1317	0.1401	0.0529	0.1319	0.1363	0.1214	0.1439	0.1116	0.1264	**0.1453**
Yahoo!R3	MF	Recall@10	0.0158	0.0159	0.0143	0.0163	0.0165	0.0164	0.0162	0.0133	**0.0166**	**0.0166**
		Recall@20	0.0283	0.0289	0.0257	0.0286	0.0290	0.0290	0.0282	0.0245	0.0300	**0.0301**
		NDCG@10	0.0132	0.0132	0.0114	0.0139	0.0145	0.0142	0.0141	0.0104	0.0142	0.0144
		NDCG@20	0.0187	0.0188	0.0163	0.0191	0.0197	0.0196	0.0192	0.0152	0.0199	**0.0201**
	LightGCN	Recall@10	0.0159	0.0161	0.0156	0.0162	0.0164	0.0130	0.0163	0.0152	**0.0170**	0.0167
		Recall@20	0.0292	0.0281	0.0280	0.0287	0.0287	0.0249	0.0289	0.0274	**0.0301**	0.0296
		NDCG@10	0.0138	0.0140	0.0125	0.0143	0.0143	0.0106	0.0142	0.0121	0.0144	**0.0148**
		NDCG@20	0.0197	0.0191	0.0177	0.0196	0.0198	0.0156	0.0195	0.0173	0.0200	**0.0202**
KuaiRec	MF	Recall@10	0.1017	0.1112	0.1157	0.1176	0.1304	0.1269	0.1275	0.1008	0.1007	**0.1351**
		Recall@20	0.1590	0.1652	0.1522	0.1546	0.1836	0.1841	0.1810	0.1620	0.1623	**0.1923**
		NDCG@10	0.3365	0.3613	0.4111	0.4178	0.4341	0.4239	0.4304	0.2584	0.3312	**0.4510**
		NDCG@20	0.3092	0.3248	0.3374	0.3470	0.3773	0.3728	0.3731	0.2577	0.3085	**0.3894**
	LightGCN	Recall@10	0.1288	0.1291	0.1287	0.1312	0.1309	0.0702	0.1305	0.1283	0.1101	**0.1332**
		Recall@20	0.1817	**0.1850**	0.1817	0.1842	0.1784	0.1049	0.1817	0.1770	0.1727	0.1816
		NDCG@10	0.4293	0.4274	0.4244	0.4330	0.4337	0.1732	0.4289	0.4170	0.3542	**0.4381**
		NDCG@20	0.3733	0.3741	0.3705	0.3750	0.3674	0.1607	0.3690	0.3580	0.3282	0.3711
Douban-book	MF	Recall@10	0.0431	0.0497	0.0372	0.0680	0.0798	0.0771	0.0710	0.0730	0.0697	**0.0831**
		Recall@20	0.0679	0.0830	0.0583	0.1047	0.1239	0.1080	0.1099	0.1015	0.1064	**0.1274**
		NDCG@10	0.0605	0.0635	0.0551	0.0940	0.1092	0.1081	0.0935	0.0988	0.0812	**0.1094**
		NDCG@20	0.0635	0.0711	0.0568	0.0984	0.1148	0.1086	0.0997	0.1004	0.0908	**0.1158**
	LightGCN	Recall@10	0.0857	0.0876	0.0461	0.0870	0.0900	0.0780	0.0863	0.0892	0.0870	**0.0901**
		Recall@20	0.1318	0.1326	0.0697	0.1303	**0.1358**	0.1111	0.1306	0.1244	0.1323	0.1333
		NDCG@10	0.1200	0.1210	0.0626	0.1199	0.1217	0.0996	0.1178	0.1163	0.1141	**0.1231**
		NDCG@20	0.1250	0.1262	0.0644	0.1241	0.1269	0.1035	0.1230	0.1193	0.1214	**0.1274**
Yelp2018	MF	Recall@10	0.0214	0.0232	0.0214	0.0292	0.0361	0.0260	0.0311	0.0304	0.0283	**0.0405**
		Recall@20	0.0382	0.0400	0.0362	0.0507	0.0619	0.0449	0.0540	0.0528	0.0488	**0.0689**
		NDCG@10	0.0283	0.0311	0.0294	0.0394	0.0489	0.0352	0.0414	0.0251	0.0379	**0.0562**
		NDCG@20	0.0338	0.0363	0.0335	0.0458	0.0564	0.0409	0.0485	0.0327	0.0442	**0.0640**
	LightGCN	Recall@10	0.0456	0.0459	0.0311	0.0464	0.0465	0.0328	0.0254	0.0458	0.0466	**0.0469**
		Recall@20	0.0772	0.0771	0.0519	0.0784	0.0780	0.0540	0.0494	0.0741	0.0621	**0.0786**
		NDCG@10	0.0622	0.0624	0.0415	0.0634	0.0633	0.0435	0.0395	0.0613	**0.0791**	0.0647
		NDCG@20	0.0710	0.0710	0.0474	0.0723	0.0719	0.0493	0.0450	0.0687	0.0716	**0.0732**

Table 2: Performance comparison with state-of-the-art recommendation models. The best results are indicated in bold font and the suboptimal ones are underlined.

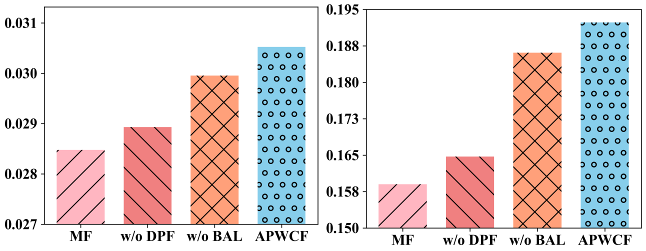
with MF as the backbone model, APWCF's Recall@20 and NDCG@20 are 0.0689 and 0.0640, respectively, which are 11.31% and 13.48% higher than the second-best PDA.

(2) Overall, methods with LightGCN as the backbone network outperform MF-based methods. This may be because LightGCN is better at capturing the complex interactions between users and items, while MF, as a shallow model, has difficulty effectively modeling these high-order relationships.

(3) Although APWCF achieves optimal or suboptimal results in most cases, we also observe that APWCF performs poorly when using MF as the backbone network on the ML-100K dataset. We believe this is related to the small number of nodes in this dataset, which makes it difficult to effectively optimize the dynamic propensity factor and hinders adversarial learning. In addition, the use of shallow models such as MF may also limit the expressive power of the model.

4.3 Ablation Study (RQ2)

APWCF consists of two key components: Dynamic Propensity Factor and Bias Adversarial Learning. To evaluate the contribution of each module, we conducted ablation studies on the Yahoo!R3 and KuaiRec datasets. The Recall@20 results are presented in Fig. 2, where "w/o DPF" and "w/o BAL" denote the model variants obtained by removing the dynamic propensity factor and bias adversarial learning components, respectively. As shown in Fig. 2, the model performance drops significantly after removing either component, indicating that both modules contribute substantially to the overall performance. Furthermore, both "w/o DPF" and "w/o BAL" variants still outperform the baseline method MF. This demonstrates that the dynamic propensity factor effectively models the propensity of the data and corrects the bias in the recommendation results, while adversarial learning further optimizes The deviation distribution of items, both of



(a) Yahoo!R3

(b) KuaiRec

Figure 2: The performance of difference components of our method on Yahoo!R3 and KuaiRec dataset.

which jointly improve the recommendation performance. In addition, similar trends are observed on LightGCN.

4.4 Evaluation of Debiasing Ability (RQ3)

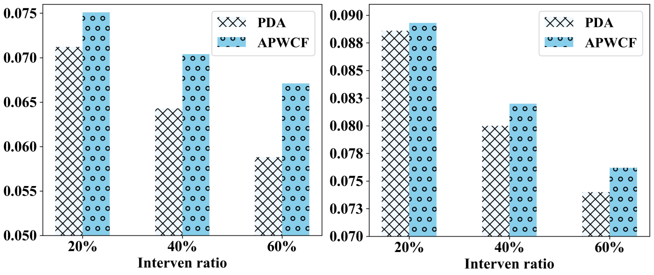
To verify the debiasing ability of APWCF, we compare it with PDA. On the Yelp2018, we evaluate recommendation performance under three intervention levels: low, medium, and high. Specifically, the proportion of the intervention set is set to 20%, 40%, and 60%, corresponding to low, medium, and high intervention levels, respectively. Generally, the size of the intervention set affects the degree to which the debiasing ability is tested. A higher intervention ratio presents a more rigorous test of the model's debiasing ability. In other words, better performance under high intervention indicates stronger debiasing ability. We calculate Recall@20 and NDCG@20 on three sets of different intervention ratio datasets to evaluate the recommendation performance under different intervention ratios.

From the results in Fig. 3, we observe that as the intervention ratio increases, the model faces greater distribution inconsistency, so the requirement for debias ability is higher, and the overall performance of both methods decreases. In addition, at high intervention ratios, APWCF still has an advantage over PDA. Specially, when MF is used as the backbone and the intervention ratios are 20%, 40%, and 60%, the Recall@20 of PDA is 0.0712, 0.0643, and 0.0588, respectively, while APWCF reaches 0.0751, 0.0704, and 0.0671, which is 5.47%, 9.49%, and 14.11% higher than PDA. Furthermore, compared to PDA, APWCF demonstrates more stable debiasing performance across different intervention ratios. Specifically, as the intervention ratio increases from 20% to 40% and from 40% to 60%, the Recall@20 of PDA decreases by 9.69% and 8.55%, respectively, while APWCF only decreases by 6.25% and 4.69%.

4.5 Parameter Sensitivity Study (RQ4)

APWCF has several key hyperparameters, including λ, α, and β. Due to space limitations, we focus discussion on the parameter λ, while the analysis of α and β is provided in the Supplemental Material D.

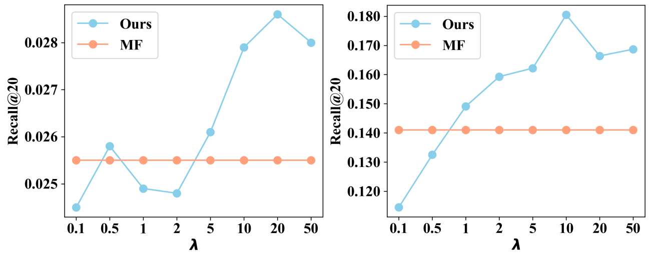
The effect λ. λ controls the weight of the propensity score estimation loss in the overall objective, aiming to reduce bias in the recommendation process. We fine-tune λ on the Yahoo!R3 and KuaiRec datasets, and use MF as the backbone. We report the results of Recall@20 on the validation set are presented in Fig. 4. From the results, We observe that our method outperforms the MF under most λ values. Generally, as λ increases, Recall@20 improves significantly. However, when λ becomes too large, the model focuses excessively on correcting propensity score, and over-emphasis on correcting data bias may cause the model to overfit the propensity estimation task during training. We believe that especially when the training data is unbalanced (such as a high proportion of popular items), the model will over-optimize the loss of propensity estimation and ignore the modeling of the difference between positive and negative samples in the recommendation task, resulting in a decrease in debiasing performance.



(a) MF

(b) LightGCN

Figure 3: Debiasing performance across intervention ratios, with stronger performance at higher ratios indicating better debiasing.



(a) Yahoo!R3

(b) KuaiRec

Figure 4: The performance of difference λ on two datasets.

5 Conclusion and Future Work

In this paper, we propose a debiasing method, APWCF, which integrates dynamic propensity modeling and bias adversarial learning. To overcome the limitations of existing methods, the DPF module dynamically stabilizes propensity score estimation, while the BAL module decouples user interests from bias signals, enabling the joint removal of multiple biases. Extensive experiments on five real-world datasets demonstrate that APWCF significantly outperforms state-of-the-art methods in terms of debiasing. In the future, APWCF can be extended to mitigate biases in feedback loops, which tend to amplify existing biases, further exacerbating the imbalance of user interactions and product exposure over time.

Acknowledgments

The work reported herein was supported by the National Key R&D Program of China (2023YFC3306100), National Natural Science Foundation of China (62172324, 62272379), Key R&D in Shaanxi Province (2023-YBGY-269, 2022-QCY-LL33HZ), Xixian New Area Science and Technology Plan Project (RGZN-2023-002, 2022 ZDJS-001).

References

[Bakhshizadeh, 2024] Mahta Bakhshizadeh. Supporting knowledge workers through personal information assistance with context-aware recommender systems. In Tommaso Di Noia, Pasquale Lops, Thorsten Joachims, Katrien Verbert, Pablo Castells, Zhenhua Dong, and Ben London, editors, *Proceedings of the 18th ACM Conference on Recommender Systems, RecSys 2024, Bari, Italy, October 14-18, 2024*, pages 1296–1301. ACM, 2024.

[Bonner and Vasile, 2018] Stephen Bonner and Flavian Vasile. Causal embeddings for recommendation. In Sole Pera, Michael D. Ekstrand, Xavier Amatriain, and John O'Donovan, editors, *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, pages 104–112. ACM, 2018.

[Chen *et al.*, 2023] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. Bias and debias in recommender system: A survey and future directions. *ACM Trans. Inf. Syst.*, 41(3):67:1–67:39, 2023.

[Gao *et al.*, 2023] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, and Yong Li. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *Trans. Recomm. Syst.*, 1(1):1–51, 2023.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and D. Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org, 2010.

[He *et al.*, 2020] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu, editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 639–648. ACM, 2020.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[Lee *et al.*, 2023] Jae-woong Lee, Seongmin Park, Mincheol Yoon, and Jongwuk Lee. uctrl: Unbiased contrastive representation learning via alignment and uniformity for collaborative filtering. In Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete, editors, *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 2456–2460. ACM, 2023.

[Li *et al.*, 2022] Nian Li, Chen Gao, Jinghua Piao, Xin Huang, Aizhen Yue, Liang Zhou, Qingmin Liao, and Yong Li. An exploratory study of information cocoon on short-form video platform. In Mohammad Al Hasan and Li Xiong, editors, *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 4178–4182. ACM, 2022.

[Ning *et al.*, 2024] Wentao Ning, Reynold Cheng, Xiao Yan, Ben Kao, Nan Huo, Nur Al Hasan Haldar, and Bo Tang. Debiasing recommendation with personal popularity. In Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee, editors, *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pages 3400–3409. ACM, 2024.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In Jeff A. Bilmes and Andrew Y. Ng, editors, *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, pages 452–461. AUAI Press, 2009.

[Saito *et al.*, 2020] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. Unbiased recommender learning from missing-not-at-random implicit feedback. In James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang, editors, *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 501–509. ACM, 2020.

[Schnabel *et al.*, 2016] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1670–1679. JMLR.org, 2016.

[Wang *et al.*, 2019] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer, editors, *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 165–174. ACM, 2019.

[Wei *et al.*, 2021] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In Feida Zhu, Beng Chin Ooi, and Chunyan Miao, editors, *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 1791–1800. ACM, 2021.

[Wu *et al.*, 2024] Cheng Wu, Shaoyun Shi, Chaokun Wang, Ziyang Liu, Wang Peng, Wenjin Wu, Dongying Kong, Han Li, and Kun Gai. Enhancing recommendation accuracy and diversity with box embedding: A universal framework. In Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee, editors, *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pages 3756–3766. ACM, 2024.

[Yalcin and Bilge, 2022] Emre Yalcin and Alper Bilge. Evaluating unfairness of popularity bias in recommender systems: A comprehensive user-centric analysis. *Inf. Process. Manag.*, 59(6):103100, 2022.

[Yang *et al.*, 2024] Jiyuan Yang, Yue Ding, Yidan Wang, Pengjie Ren, Zhumin Chen, Fei Cai, Jun Ma, Rui Zhang, Zhaochun Ren, and Xin Xin. Debiasing sequential recommenders through distributionally robust optimization over system exposure. In Luz Angelica Caudillo-Mata, Silvio Lattanzi, Andrés Muñoz Medina, Leman Akoglu, Aristides Gionis, and Sergei Vassilvitskii, editors, *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM 2024, Merida, Mexico, March 4-8, 2024*, pages 882–890. ACM, 2024.

[Zhang *et al.*, 2021] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. Causal intervention for leveraging popularity bias in recommendation. In Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai, editors, *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 11–20. ACM, 2021.

[Zhang *et al.*, 2023] Xiaoying Zhang, Hongning Wang, and Hang Li. Disentangled representation for diversified recommendations. In Tat-Seng Chua, Hady W. Lauw, Luo Si, Evimaria Terzi, and Panayiotis Tsaparas, editors, *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023 - 3 March 2023*, pages 490–498. ACM, 2023.

[Zhao *et al.*, 2020] Yunwei Zhao, Can Wang, Han Han, Min Shu, and Wenlei Wang. An impact evaluation framework of personalized news aggregation and recommendation systems. In Jing He, Hemant Purohit, Guangyan Huang, Xiaoying Gao, and Ke Deng, editors, *IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, WI/IAT 2020, Melbourne, Australia, December 14-17, 2020*, pages 893–900. IEEE, 2020.

[Zheng *et al.*, 2021] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. Disentangling user interest and conformity for recommendation with causal embedding. In Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia, editors, *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 2980–2991. ACM / IW3C2, 2021.