Unbiased Knowledge Distillation for Recommendation

Gang Chen
University of Science and Technology of China
Hefei, China
gchenx@mail.ustc.edu.cn

Jiawei Chen*
Zhejiang University
Hangzhou, China
sleepyhunt@zju.edu.cn

Fuli Feng
University of Science and Technology of China
Hefei, China
fulifeng93@gmail.com

Sheng Zhou
Zhejiang University
Hangzhou, China
zhousheng_zju@zju.edu.cn

Xiangnan He*
University of Science and Technology of China
Hefei, China
xiangnanhe@gmail.com

ABSTRACT

As a promising solution for model compression, knowledge distillation (KD) has been applied in recommender systems (RS) to reduce inference latency. Traditional solutions first train a full teacher model from the training data, and then transfer its knowledge (*i.e., soft labels*) to supervise the learning of a compact student model. However, we find such a standard distillation paradigm would incur serious bias issue — popular items are more heavily recommended after the distillation. This effect prevents the student model from making accurate and fair recommendations, decreasing the effectiveness of RS.

In this work, we identify the origin of the bias in KD — it roots in the biased soft labels from the teacher, and is further propagated and intensified during the distillation. To rectify this, we propose a new KD method with a stratified distillation strategy. It first partitions items into multiple groups according to their popularity, and then extracts the ranking knowledge within each group to supervise the learning of the student. Our method is simple and teacher-agnostic — it works on distillation stage without affecting the training of the teacher model. We conduct extensive theoretical and empirical studies to validate the effectiveness of our proposal. We release our code at: https://github.com/chengang95/UnKD.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommendation, Knowledge Distillation, Bias and Debias

ACM Reference Format:
Gang Chen, Jiawei Chen, Fuli Feng, Sheng Zhou, and Xiangnan He. 2023. Unbiased Knowledge Distillation for Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27-March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570477

*Corresponding author

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

ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570477

1 INTRODUCTION

Recommender system (RS) has become increasingly important with the universalization of online personalized services. With the increasing scale of items, the trade-off between the accuracy and efficiency in modern RS cannot be ignored. A large model with numerous parameters has a high capacity, and thus is shown to have better accuracy. However, its success requires heavy computational and memory costs, which would incur unacceptable latency during the inference phase, making it hard to be applied in real-time RS.

To deal with such dilemma, *knowledge distillation* (KD) has been applied in recommender system [15, 16, 26], with the purpose of reducing model size while maintaining model performance. KD first trains a large teacher model from the training set, and then learns a small student model with the supervision from the *soft labels* that are generated by the teacher. As the soft labels encode the knowledge learned by the teacher, the student can benefit more from it and achieve better performance than the student directly learning from the training data.

Despite decent performance, we argue that the distillation is severely biased towards popular items. We make an empirical study of existing KDs on three benchmark recommendation datasets. The results are presented in Table 1. The overall improvements of KDs mainly lie on the popular group, while the performance of the unpopular group drops significantly (22.4% on average). This impressive result clearly reveal the severe bias issue in KDs, which is essential to be overcome. This negative effect will hinder the student model from completely understanding user preference. Worse still, it will decrease the level of the diversity and fairness in recommendations, heavily deteriorating user experience.

In view of this phenomenon, we first identify the origin of the bias — *the biased soft labels generated by the teacher* — which is further intensified by the distillation process in training the student. Figure 1 provides the evidence of biased teacher prediction, where we train a standard matrix factorization (MF) [23] and count the ratios of popular/unpopular items in the top-10 recommendation lists. As can be seen, the top-10 items with the largest scores are severely biased towards mainstream. Worse still, such bias would be inherited and amplified during the distillation. Existing KDs [16, 26]

usually simply consider higher-ranked items as positive and give them larger confidence weights. As a result, popular items would exert excessive contribution on student model training, causing the bias of the student.

Being aware of the origin of the distillation bias, now the question lies on how to eliminate this negative effect. A straightforward solution is to directly intervene into the training of the teacher model to generate unbiased soft labels, which however is difficult to achieve. On the one hand, the teacher bias may root in multiple factors, including but not limited to the momentum-based optimizer [27], imperfect loss function [3], and the factorization model architecture [17]. Completely isolating bias from teacher is itself highly challenging. On the other hand, a teacher-agnostic KD strategy is more desirable. In practice, a large teacher model is usually deployed in a complex distributed system, and adjusting its training procedure is difficult objectively, not to mention the teacher can be an ensemble of multiple models. As such, in this work, we propose an **Un**biased **K**nowledge **D**istillation strategy (**UnKD**) that performs debiasing during the training of the student model. Specifically, UnKD resorts to a skillful popularity-aware distillation: it first partitions items into multiple groups according to their popularity, and then extracts the ranking knowledge among each group to supervise the learning of the student. On the basis of causal theory, we prove that such stratification strategy can almost block the causal effect from the teacher bias. Remarkably, UnKD is simple and model-agnostic. We implement it on MF [23] and LightGCN [9] to demonstrate effectiveness.

To summarize, this work makes the following contributions:

- Revealing the bias issue of knowledge distillation in recommender systems.
- Proposing an unbiased teacher-agonistic knowledge distillation (UnKD) that extracts popularity-aware ranking knowledge to guide student learning.
- Conducting extensive experiments on three real datasets to demonstrate the superiority of UnKD over state-of-the-arts.

The rest of the paper is organized as follows. In Section 2, we introduce the background of knowledge distillation. In Section 3, we provide causal view on bias issue in knowledge distillation and then detail our proposed UnKD. The experiments and discussions are presented in Section 4. Finally, we provide related work and conclusions in Section 5 and Section 6.

2 PRELIMINARIES

In this section, we first introduce the basic notations and formulate the recommendation task. We then provide the background of knowledge distillation.

We use uppercase character (*e.g.,* U) to denote a random variable and lowercase character (*e.g.,* u) to denote its specific value. We use characters in calligraphic font (*e.g.,* $\mathcal{U}$) to represent the sample space of the corresponding random variable. We use the notation $|*|$ for the size of the collection, *e.g.,* $|\mathcal{U}|$ denoting the size of $\mathcal{U}$.

Recommendation Task. Suppose we have a recommender system with a user set $\mathcal{U} = \{u_1, ..., u_m\}$ and an item set $\mathcal{I} = \{i_1, ..., i_n\}$. The collected historical user-item feedback can be formulated as a matrix $R \in \{1, 0\}^{m \times n}$, where each entry $r_{ui} \in R$ denotes whether a user u has interacted with an item i. Given user u, $\mathcal{I}_u^+ = \{i \in \mathcal{I} | r_{ui} = 1\}$ is the set of items with known positive feedback, and $\mathcal{I}_u^- = \mathcal{I} \setminus \mathcal{I}_u^+$ is the set of items with missing feedback [11, 20, 38]. For each user, the goal of a recommender system is to find items that are most likely to be interacted.

Table 1: Performance (Recall@10) comparison of various knowledge distillation methods in terms of popular/unpopular items on three real-world datasets. We also report the relative improvements over the baseline ('Student') that is directly learned from the data. The experimental settings and the group partition are detailed in Section 4.

Movielens						
	Student	Teacher	RD[26]	CD[16]	DERRD[12]	HTD[13]
Popular Group	0.2156 —	0.2565 +18.97%	0.2237 +3.75%	0.2258 +4.73%	0.2315 +7.37%	0.2228 +3.33%
Unpopular Group	0.0250 —	0.0517 +106.80%	0.0242 -3.20%	0.0179 -28.40%	0.0113 -54.80%	0.0187 -25.20%
Apps						
	Student	Teacher	RD[26]	CD[16]	DERRD[12]	HTD[13]
Popular Group	0.1031 —	0.1448 +40.44%	0.1144 +10.96%	0.1212 +17.55%	0.1058 +2.61%	0.1195 +15.90%
Unpopular Group	0.0109 —	0.0164 +50.45%	0.0098 -10.09%	0.0090 -17.43%	0.0098 -10.09%	0.0061 -44.03%
CiteULike						
	Student	Teacher	RD[26]	CD[16]	DERRD[12]	HTD[13]
Popular Group	0.0831 —	0.1294 +55.71%	0.0910 +9.50%	0.0887 +6.73%	0.0899 +8.18%	0.0885 +6.49%
Unpopular Group	0.0095 —	0.0537 +465.26%	0.0085 -10.52%	0.0088 -7.36%	0.0075 -21.05%	0.0068 -28.42%

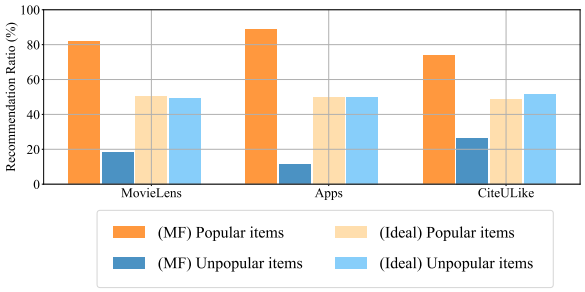
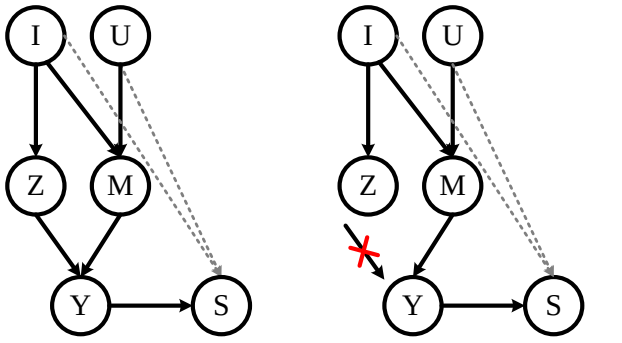


Figure 1: Ratios of popular/unpopular items in the top-10 recommendation lists from a MF model. We also present the ideal ratios from the test data for comparison.

Knowledge Distillation. Knowledge distillation [6, 10, 19] is a promising model compression technique that first trains a large teacher model and then transfers the knowledge from the teacher to the target compact student model. In RS, soft labels — *i.e.,* the teacher predictions on the user-item interactions, are commonly-used for knowledge transfer. These KDs [12, 16, 26] would create or sample the training instances according to soft labels for training a student model. As such, the quality of the soft labels lays a foundation of knowledge distillation. The distillation process is shown in Figure 3(a). It is worth to mention that there is work HTD [13] that utilizes teacher embeddings rather than soft labels for knowledge distillation. However, we also observe serious bias issue in HTD



(a) Generative process of the soft label. (b) We cut off $Z \rightarrow Y$ during distillation.

Figure 2: The causal graph to describe the knowledge distillation. U: user, I: item, M affinity score, Z item popularity, Y: soft label, S: student. The bias origins from the causal effect of Z on Y. Our UnKD is intended to cut off $Z \rightarrow Y$. Admittedly, there may exist other causal paths from U, I to S, but here we only focus on the causal effect through distillation (*i.e.,* via Y).

(Table 1). In fact, our analyses are also suitable for this method, if we simply replace the term 'Soft labels' with 'Teacher embeddings'.

3 METHODOLOGY

In this section, we first resort to a causal graph to trace the origin of bias in knowledge distillation. We then introduce the proposed UnKD and discuss its rationality for eliminating the bias.

3.1 A Causal View on Distillation Bias

Origin of Distillation Bias. To trace the origin of distillation bias and to understand how it affects student model, we resort to the language of causal graph [22] for a qualitive analysis. Figure 2(a) illustrates causal relations behind existing distillation methods, which consists of six random variables including:

- U represents a user node, *e.g.,* user profile or feature (*e.g.,* IDs) that is used for representing a user;
- Similar to U, Node I represents an item node;
- M represents the real affinity score between a user U and an item I, reflecting to what extent that the item matches the preference of the user;
- Z represents the item popularity;
- Y represents the soft label predicted by the teacher model;
- S represents the learned student model.

The edges in the graph describe the causal relations between variables. Specifically, we have:

- Edges $(U, I) \rightarrow M$ depict the causal effect of the features of a user and an item on their affinity;
- Edges $I \rightarrow Z$ depicts that the popularity of an item is affected by its characteristics;
- Edges $(M, Z) \rightarrow Y$ show that the soft label Y is affected by two factors: 1) $M \rightarrow Y$, the desirable effect from the affinity; 2) $Z \rightarrow Y$, the influence from the item popularity, where an item with larger popularity is prone to have higher prediction score. Recent work has validated the effect of $Z \rightarrow Y$ is common in recommendation. It can be original from the biased data (*i.e.,* popular items is usually over-exposed [39]), learning algorithm (momentum-based optimizer is biased towards mainstream [27]) or recommendation architecture[17] (*i.e.,* latent factor models prefer to promote popular groups).
- Edge $Y \rightarrow S$ depicts the student model is learned under the supervision of soft labels.

According to the causal graph, since there exists an additional path $(I \rightarrow Z \rightarrow Y)$ from I to Y, the learned soft label would be deviated from reflecting user's true preference, *e.g.,* an item with higher scores simply because it belongs to a mainstream groups rather than it really meets user preference. Such biases would be further propagated and intensified into the student model, heavily deteriorating its recommendation quality. Typically, the student model would be skewed under the supervision from the biased soft labels; Worse still, note that existing KDs usually employ rank-aware sampling strategy for training a student model. The popular items which usually have abnormally higher scores would obtain more sampling opportunities and thus exert excessive contributions on training. The bias would be amplified during the distillation. As such, it is essential to address bias issue in knowledge distillation. The core lies on blocking the causal effect from I on Y along the path $(I \rightarrow Z \rightarrow Y)$.

Quantifying Bias Effect. Given the importance of cutting off the path $(I \rightarrow Z \rightarrow Y)$, here we refer to the language of causal inference [22] and give a formula of the causal effect that we aim at estimating. We first quantify the causal effect from the bias and then remove it from the total effect to recover the desirable effect from the preference.

Let $Y_{A=a}|U = u$ (short as $Y_a|U = u$) be the random variable with conditional distribution $p(Y|\text{do}(A = a), U = u)$ where a variable A is intervened with a specific value a. The causal effect of a variable I on Y is the magnitude by which the target variable Y is changed by a unit change in an variable I [22]. For example, the conditional total effect of $I = i$ on Y for a specific user u is defined as:

$$\text{TE}_i = Y_i|u - Y_{i^*}|u \tag{1}$$

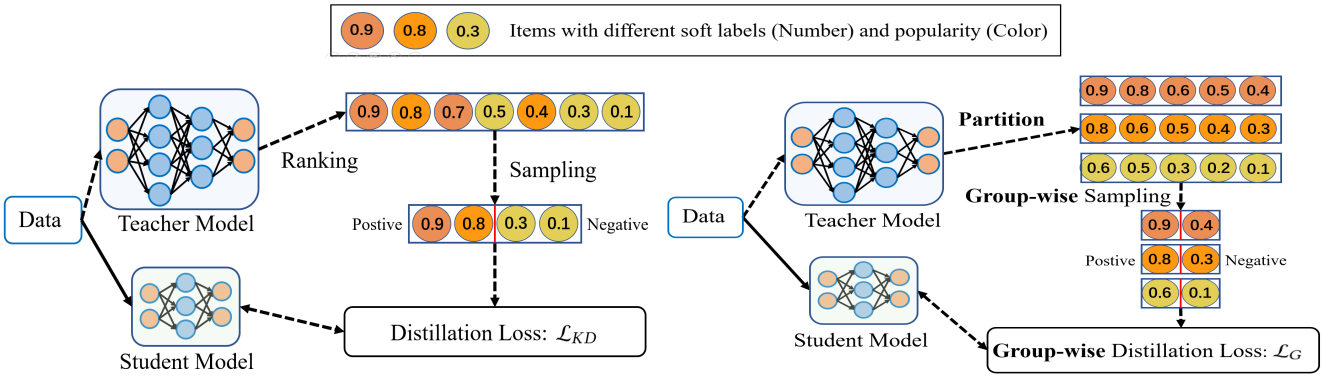
which can be understood as the difference between two hypothetical situations $I = i$ and $I = i^*$. $I = i^*$ can be considered as a benchmark situation for comparison. TE_i can be decomposed into two parts: 1) the desirable causal effect along the path $I \rightarrow M \rightarrow Y$; and 2) the undesirable causal effect along the path $(I \rightarrow Z \rightarrow Y)$. By performing different interventions on I along different causal paths, it is possible to isolate the contribution of the causal effect along different paths.

Specifically, the path-specific causal effect through $(I \rightarrow Z \rightarrow Y)$ expresses the value change of Y with the item popularity Z change from Z_i to Z_i^*:

$$\text{PEZ}_i = Y_{i^*,Z_i}|u - Y_{i^*}|u \tag{2}$$

Accordingly, eliminating the bias can be realized by reducing PEZ from TE, we have:

$$\text{PEM}_i = \text{TE}_i - \text{PEZ}_i = Y_i|u - Y_{i^*,Z_i}|u \tag{3}$$



(a) Traditional Distillation Methods (b) Proposed UnKD

Figure 3: Illustrations of (a) the traditional knowledge distillations and (b) our proposed UnKD. UnKD partitions items into multiple groups according to their popularity, and then extracts the ranking knowledge among each group to learn the student.

which expresses the value change of Y with changing i to i^* while keeping Z unchanged. This formula blocks the effect along $Z \rightarrow Y$ and can be fed into the student model for unbiased distillation.

However, calculating PEM is difficult, as it involves a counterfactual inference since the item popularity for a specific item i^* is intervened from the factual value Z_{i^*} to the Z_i. A naive solution is to directly intervene into the training of the teacher model, *e.g.,* employing some debiasing strategies to mitigate the popularity bias [39]. However, as discussed before, learning an completely unbiased teacher is usually impractical and unsatisfied. Our empirical studies in Section 4 also validate that this strategy does not bring satisfactory results. Therefore, a new unbiased distillation method without requiring to intervene teacher model deserves exploration.

3.2 Unbiased Knowledge Distillation

Towards this end, in this work we propose an unbiased knowledge distillation strategy (UnKD), which conducts debiasing during the learning of the student model. The subtlety of UnKD lies on its popularity-stratified training strategy, where the unfeasible counterfactual terms have been properly offset. To be more specific, UnKD first partitions items into multiple groups according to the item popularity, where the items in one group have similar popularity. After that, for each user, UnKD ranks the items on the same group *w.r.t.* the soft label, and transfers such group-wise knowledge to supervise the learning of the student model. In fact, we have the following lemma:

Lemma 1. *For each user u, for the items with highly similar popularity, the list ranked by Y_i is approximately equal to the list ranked by PEM_i.*

Proof. For arbitrary two items i and j with highly similar popularity, we have $Z_i \approx Z_j$ and thus the equation $Y_{i^*,Z_i}|u = Y_{i^*,Z_j}|u$ almost holds. Then, we have:

$$Y_i|u > Y_j|u \Leftrightarrow Y_i|u - Y_{i^*,Z_i}|u > Y_j|u - Y_{i^*,Z_j}|u \Leftrightarrow \text{PEM}_i > \text{PEM}_j \tag{4}$$

The lemma gets proofed. □

It means that the group-wise ranking lists are approximately unbiased, which provide more accurate evidence on users' true preference. UnKD extracts such accurate popularity-stratified ranking knowledge for training a student model, which avoids disturbance from the terrible popularity effect.

Details of UnKD. The detailed training procedure of UnKD is illustrated in Figure 3(b). UnKD follows the recent advanced strategy CD [12], differing in employing group-wise sampling and training. UnKD consists of the following three steps:

(1) *Group partition.* We partition items into K groups according to the item popularity. The partition procedure refers to the recent work [39]. Specifically, we first sort items according to their popularity in descending order, and then divide the items into K groups. The items with similar popularity are positioned into the same group. Also, we follow [39] and let the sum of popularity over items in each group is the same.

We remark that K is an important hyperparameter balancing the trade-off between the unbiasedness and informativeness. A larger K suggests a more fine-grained partition and the items in each group would have higher similarity on popularity. It means the unbiasedness is more likely to be held. However, larger K would decrease the number of items in each group, and reduced the knowledge about the item ranking relations. On the contrary, a smaller K could bring more information but at the expense of unbiasedness. The empirical results of how K affects distillation performance are shown in Section 4.

(2) *Group-wise Sampling.* For each user, we first rank the items in each group in terms of the soft labels from the teacher. We then sample a set $\mathcal{S}_{ug}$ of positive-negative item pairs (i^+, i^-) for each group g with the rank-aware probability distribution: $p_i \propto e^{-rank(i)/\mu}$ [12]. Here $rank(i)$ represents the ranking position of i in the group, and μ is a hyperparameter.

(3) *Group-wise Learning.* We adopted group-wise distillation loss for training a student model:

$$\mathcal{L}_G = -\sum_u \frac{1}{|\mathcal{U}|} \sum_{g \in \mathcal{G}} \sum_{(i^+,i^-) \in \mathcal{S}_{ug}} \log \sigma(\mathbf{e}_u^T \mathbf{e}_{i^+} - \mathbf{e}_u^T \mathbf{e}_{i^-}) \tag{5}$$

Table 2: Statistics of the datasets.

dataset	Users	Items	Interactions	Sparsity
CiteULike	5219	25181	125580	99.91%
Apps	3898	11797	128105	99.73%
MovieLens	6040	3706	1000209	95.54%

Here the item pairs utilized for loss calculation are sampled from the stage (1) ($(i^+, i^-) \in \mathcal{S}_{ug}$). Note that the item pairs are on the same group and its relations are consistent with user true preference. The distillation loss would be accurate and provide additional useful knowledge for training a better student model. Here, $\mathbf{e}_u$ and $\mathbf{e}_i$ represent the embedding of u and i, respectively. And σ represents sigmoid function.

The final loss function for training a student model is:

$$\mathcal{L} = \mathcal{L}_R + \lambda \mathcal{L}_G \tag{6}$$

where the distillation loss $\mathcal{L}_G$ is usually accompanied with the original supervised loss $\mathcal{L}_R$ from the training data. Hyperparameter λ is utilized to balance their contributions.

4 EXPERIMENTS

In this section, we conduct experiments to evaluate the performance of our proposed UnKD. Our experiments are intended to address the following research questions:

RQ1: How does UnKD perform compared with existing distillation methods? Does UnKD benefit unpopular items?
RQ2: Does UnKD outperform the strong baseline that leverages the debiasing techniques in teacher model training?
RQ3: How does the hyperparameter K (Group numbers) affect distillation performance?

4.1 Experimental Setup

Datasets. Three commonly-used datasets are adopted for testing the model performance including **CiteULike**[1], **Amazon-Apps**[2], and **MovieLens-1M**[3]. For stable evaluation, we filter out users with fewer than 20 interactions. Also we transform the detailed rating value into binary for implicit recommendation as recent work [13]. The statistics of the datasets are shown in Table 2. Besides, for each user we randomly select 90% of historical interactions as the training set, and the remaining 10% data constitutes the testing set. We also randomly partition 10% interactions from training data for model validation.

Compared Methods. We compare our methods with the following baselines:

- RD [26]: A classic KD method for recommendation that treats the Top-N ranked items as positive while reweighs the items according to the position.
- CD [16]: A method that creates positive-negative training instances based on the item ranking position from the teacher.
- DERRD [12]: A KD that trains a student model from both teachers' prediction and teacher latent knowledge.

[1]https://github.com/changun/CollMetric
[2]http://jmcauley.ucsd.edu/data/amazon/links.html
[3]https://grouplens.org/datasets/movielens/

Table 3: Overall performance comparison between our method and baselines. All metrics are based on the top-10 results. where the best performance is bold and the second best underlined.

	Backbone Model	BPRMF		LightGCN	
Dataset	Method	Recall	NDCG	Recall	NDCG
MovieLens	Teacher	0.1810	0.2951	0.1850	0.3012
	Student	0.1435	0.2511	0.1456	0.2581
	RD	<u>0.1473</u>	<u>0.2559</u>	0.1471	0.2583
	CD	0.1445	0.2534	0.1477	0.2602
	DERRD	0.1436	0.2532	<u>0.1487</u>	<u>0.2606</u>
	HTD	0.1441	0.2539	0.1472	0.2592
	UnKD	**0.1547**	**0.2615**	**0.1569**	**0.2672**
	impv-e%	5.02%	2.18%	5.51%	2.53%
Apps	Teacher	0.0991	0.0760	0.1007	0.0782
	Student	0.0719	0.0539	0.0811	0.0643
	RD	0.0768	0.0596	0.0831	0.0647
	CD	<u>0.0790</u>	<u>0.0608</u>	<u>0.0848</u>	<u>0.0658</u>
	DERRD	0.0729	0.0562	0.0832	0.0648
	HTD	0.0732	0.0561	0.0833	0.0652
	UnKD	**0.0853**	**0.0644**	**0.0867**	**0.0678**
	impv-e%	7.97%	5.92%	2.24%	3.04%
CiteULike	Teacher	0.1518	0.1016	0.1657	0.1139
	Student	0.0760	0.0477	0.0783	0.0510
	RD	<u>0.0808</u>	0.0514	0.0833	0.0538
	CD	0.0801	<u>0.0518</u>	0.0936	0.0616
	DERRD	0.0793	0.0511	0.0809	0.0527
	HTD	0.0788	0.0485	<u>0.0958</u>	<u>0.0628</u>
	UnKD	**0.0863**	**0.0550**	**0.1006**	**0.0654**
	impv-e%	6.80%	6.17%	5.01%	4.14%

- HTD [13]: An advanced KD method that distills the topological knowledge from the teacher embedding space.

We test the above distillation methods on two representative backbone models: BPR-MF [23] and LightGCN[9]. We also report the performance of teacher and student models that are directly trained from the training dataset.

Implementation Details. For the backbone model, we closely refer to [12] and set the embedding dimension of the teacher as 100 and the student as 10. Adam is adopted as our optimizer. The search space of the learning rate for all experiments is {0.01,0.001,0.0001}, and the space of the L2 regularization coefficient is {0.01,0.001,0.0001}. We adopt the early stopping strategy that stops training if $NDCG$ on the validation data does not increase for 100 epochs. The total number of training epochs is set to 1000 epochs. For the compared baselines, we closely follow their settings reported in the relevant papers or directly utilize their codes if they are available. We also finely tuned their hyperparameters to ensure optimum.

For our method, during the training phase, the number of groups K is set in the range $\{2, 3, 4, ..., 10\}$. In the testing phase, for better visualization, we only divide items into two groups, popular group and unpopular group. λ is set in the range $\{0.1, 0.2, 0.3, ..., 1.0\}$, and μ is set in the range $\{10, 20\}$. For each user, the number of soft-labels is set in the range $\{30, 40\}$.

Evaluation Metrics. The conventional ranking metrics including normalized discounted cumulative gain (NDCG@N), and Recall (Recall@N) are adopted for evaluating model performance. We also

report Recall for the popular (or unpopular) group, *i.e.,* estimating the fraction of relevant popular (or unpopular) items that are in the top-N ranking list. This metric can reflect how well the model retrievals the popular (or unpopular) items. In this work, we simply choose N as 10.

4.2 Performance Comparison (RQ1)

Overall performance comparison. Table 3 shows the overall performance of our UnKD compared with other KD methods. We observe our UnKD consistently outperforms other KDs on all three datasets. Especially in the dataset CiteULike, the improvements are encouraging. UnKD achieves 5.53% on average improvement over the baselines. Obviously, this result validates that addressing bias issue in knowledge distillation is essential and indeed boosts distillation performance.

Comparison in terms of popular/unpopular groups. To understand how our UnKD addresses bias issue in knowledge distillation, we also report the performance (recall@10) for popular and unpopular item groups. As the results for popular/unpopular groups may have different scale, here we report the relative improvements over the student baseline for better presentation. Figure 4 illustrates the results. We make the following observations: 1) the improvements of existing knowledge distillation methods are mainly from the popular items, while the performance of unpopular items severely suffers. 2) The improvements of UnKD mainly lies on unpopular items. Especially in the dataset CiteULike, UnKD achieves over 100% performance gain for unpopular items. Our UnKD could indeed address bias issue in knowledge distillation, yielding more accurate and fair recommendations.

4.3 Distillation Procedure vs. Teacher Training (RQ2)

Although previously we have discussed that directly intervening the teacher model training for debiasing is not a good choice, we are still curious about its performance. Here we compare UnKD with a strong baseline that leverages an advanced debiasing technology (PD [39]) in teacher model training. PD leverages causal inference to tackle the popularity bias, and usually achieves state-of-the-art performance in a widely range of datasets. We integrate PD into two SOTA KDs (*i.e.,* PD-CD, PD-HTD) for comparisons.

Table 4 presents the results. We make the following observations: Leveraging PD in teacher model training could boost the performance of unpopular items. However, the improvements are not significant as our UnKD. The reason can be attributed to the complexity of the bias in teacher. The bias may roots in many factors. Existing debiasing methods are usually tailored for one or two specific factors and may not eliminate the bias accurately and completely. Also, an improper debiasing may hurt model accuracy. UnKD could circumvent this challenging problem and does not require to intervene the training of the cumbersome teacher model, which is more effective and satisfactory.

4.4 Effect of the Parameter K (RQ3)

It will be interesting to explore how hyper-parameter K affects the performance of UnKD, where K indicates the number of partition groups in the distillation. Figure 5 illustrates the results (Recall@10) on all items and unpopular items, respectively.

As can be seen, with the number of groups (K) increasing, with few exception, the performance on unpopular items will become better first. The reason is that a larger K suggests a more fine-grained partition and the items in each group is prone to have higher popularity similarity. The unbiasedness of the distillation is more likely to be held. However, when K surpasses a threshold, the performance becomes worse with further increase of K. The reason is that a further larger K would make the number of items in each group decrease. The knowledge about some item ranking relations is missing. As such, K balances the trade-off between the informativeness and unbiasedness. Set K to a proper value (*e.g.,* $K = 4$) could achieve best performance for unpopular items. Similar results are observed for the overall performance, except it is relatively stable. The performance of popular items is relatively robust to K. This is because popular items can also benefit from the rich label information from the data.

5 RELATED WORK

In this section, we review the most related work from the following three perspectives.

Knowledge Distillation in Recommendation. Knowledge distillation (KD) is a promising model compression technique that first trains a large teacher model and then transfers the knowledge from the teacher to the target compact student model [5, 26, 37, 40]. KDs have been widely applied in recommender systems to reduce inference latency. They mainly utilize soft labels (*i.e.,* teacher predictions) for knowledge transfer. For example, RD [26] ranked the soft labels from the teacher and treated the top-N ranked items as positive for training a student model; CD [16] utilized soft labels to create positive and negative distillation instances; Soft labels also have been considered by DERRD [12] to create the list-wise distillation loss function. In additional to soft labels, some work considered to transfer the hidden knowledge among the middle layer of teachers (*e.g.,* latent representation). For example, DERRD [12] leveraged expert neural networks to extract useful information from the teacher representations; HTD [13] distilled the topological knowledge built upon the relations in the teacher embedding space. Despite their decent performance, we remark that existing distillation methods are severely biased towards popular items.

Besides model compression, there are also some other applications of KDs in recommender system [7, 18, 29, 31, 34]. For example, in the social recommendation, KD is used to integrate the knowledge from various relational graphs [28]; KD also plays an important role in tackling data selection bias [24, 33]; Some work also considers to leverage KD for model ensemble[41].

Bias in Recommendation. As this work focuses on popularity bias, here we mainly review recent work on this bias. For other types of biases and their debiasing techniques, we simply refer readers to a comprehensive survey [4] for more information.

Popularity bias depicts a common phenomenon [4] that Popular items are recommended even more frequently than their popularity would warrant. Ignoring the popularity bias will result in many severe issues like affecting recommendation accuracy, decreasing recommendation diversity, and even raising "Matthew effect". To

Table 4: Performance comparison (recall@10) between our UnKD and the baselines that leverages debiasing technique in model training. The best performance is shown in bold, and the second best performance is underlined.

	Dataset	MovieLens			Apps			CiteULike		
Backbone Model	Method	Overall	Popular Group	Unpopular Group	Overall	Popular Group	Unpopular Group	Overall	Popular Group	Unpopular Group
BPRMF	Student	0.1435	0.2156	0.0250	0.0719	0.1031	0.0109	0.0760	0.0831	0.0095
	CD	0.1445	**0.2258**	0.0179	0.0790	0.1212	0.0090	0.0801	0.0887	0.0088
	PD-CD	0.1454	0.2205	0.0210	0.0795	0.1176	0.0113	0.0805	0.0890	0.0092
	HTD	0.1441	0.2228	0.0187	0.0732	0.1195	0.0061	0.0788	0.0885	0.0068
	PD-HTD	0.1443	0.2150	0.0263	0.0808	0.1240	0.0076	0.0798	**0.0897**	0.0073
	UnKD	**0.1547**	0.2205	**0.0311**	**0.0853**	**0.1274**	**0.0147**	**0.0863**	0.0854	**0.0208**
LightGCN	Student	0.1456	0.2280	0.0228	0.0811	0.1242	0.0093	0.0783	0.0885	0.0080
	CD	0.1477	0.2316	0.0169	0.0848	0.1310	0.0091	0.0936	0.1067	0.0081
	PD-CD	0.1496	0.2369	0.0172	0.0851	0.1308	0.0095	0.0942	0.1020	0.0110
	HTD	0.1472	0.2328	0.0079	0.0833	0.1291	0.0091	0.0958	0.1093	0.0085
	PD-HTD	0.1485	0.2364	0.0159	0.0835	0.1288	0.0093	0.0979	**0.1102**	0.0128
	UnKD	**0.1569**	**0.2384**	**0.0292**	**0.0867**	**0.1325**	**0.0118**	**0.1006**	0.1076	**0.0162**

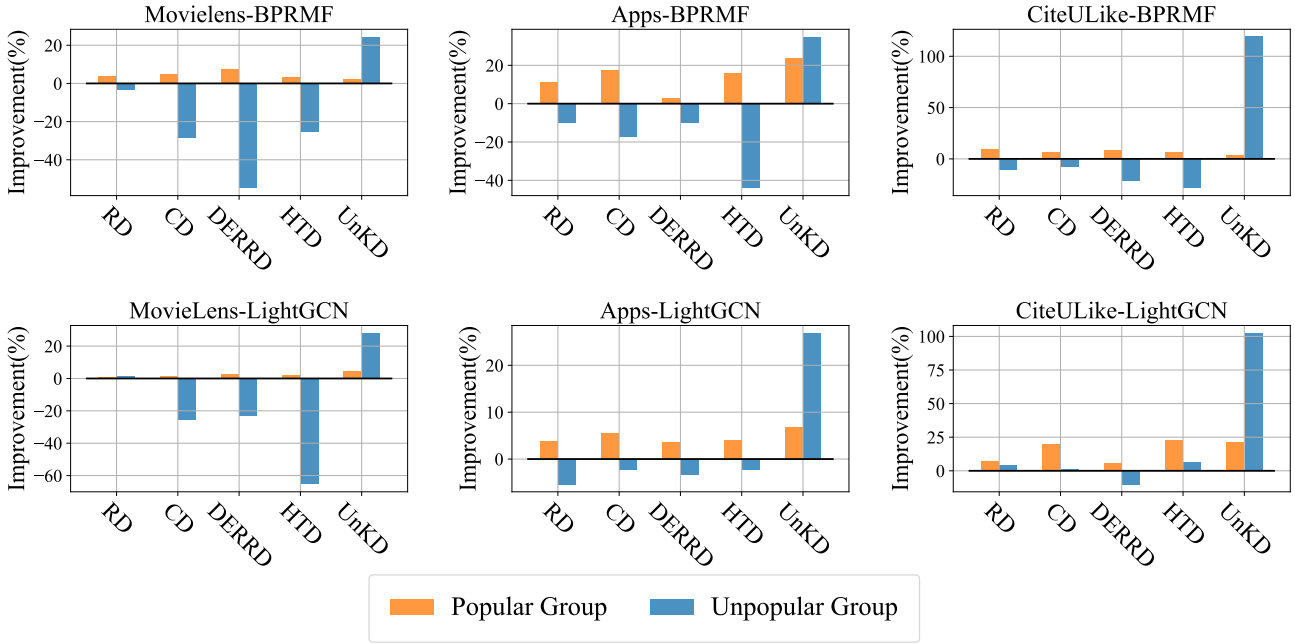


Figure 4: The relative improvements (*w.r.t.* recall@10) of KDs over the baseline that directly trained from the dataset. Here we visualize the results in terms of popular and unpopular group, respectively.

tackle popularity bias, recent work mainly lies on three types: 1) leveraging suitable regularization in model learning or ranking to push the model towards balanced recommendation lists [42]; 2) conducting adversarial training to improve the recommendation opportunity of the niche items [14]; 3) resorting to causal graph to identify the origin of the bias and conduct debiasing accordingly [39]. Although existing methods on popularity bias have achieved great progress, how to completely eliminate popularity bias is still an open problem. Popularity bias is seriously complicated and may root in various components including but not limited to optimizer [27], model architecture [17], or training data [3]. In RS, popularity bias is also occurred during the knowledge distillation, which has not been explored.

Causal Recommendation. Causal inference has received increasing attention in the field of machine learning [2, 8, 21]. In recommender systems, causal inference can be utilized for tackling bias [39], making explainable recommendation [32] or improving model generalization. As this work focuses on bias, here we mainly review recent work on causality-enhanced debiasing. They can be classified into three types: 1) The most well-known causal strategy

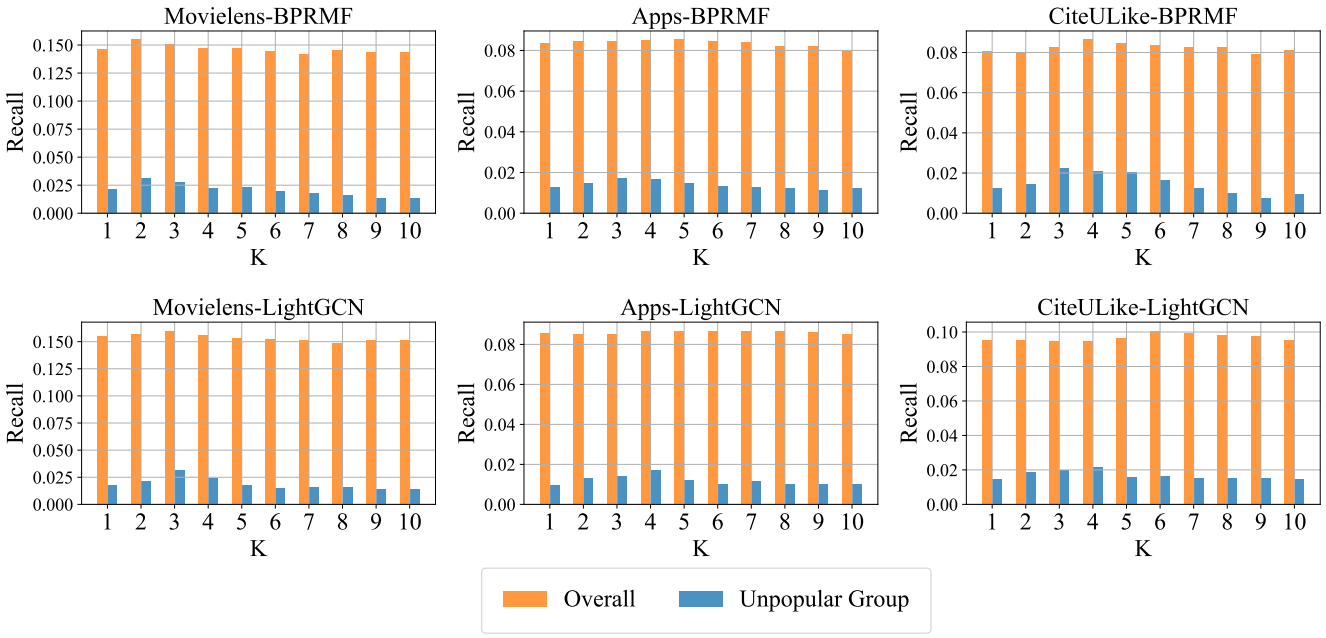


Figure 5: Performance comparison with varying K.

for debiasing is IPS, which reweighs instances with the inverse of the propensity scores. IPS has been widely for tackling various bias, including position bias [1], selection bias [25], and exposure bias [35]. 2) Another type of causality-based debiasing would resort to a causal graph. They leverage the causal graph to trace the origin of bias, and then perform counterfactual inference to cut off the effect from the bias such as PDA [39], MACR [30]. 3) The last relies on constructing counterfactual instances [36]. This method uses counterfactual inference to produce counterfactual instances that are used to offset the bias.

6 CONCLUSION

In this work, we studies on an important but unexplored problem — bias issue in distilling a recommendation model. we first identify the origin of the bias — it roots in the biased soft labels from the teacher, and is further propagated and intensified during the distillation. To rectify this, we proposes an unbiased teacher-agonistic knowledge distillation (UnKD) that extracts popularity-aware ranking knowledge to guide student learning. Our experiments on three real-world datasets validate that our UnKD outperforms state-of-the-arts by a large margin, especially for unpopular item group.

Note that this work only explores distillation bias from the popularity perspective. One interesting direction for future work is to explore more fine-grained bias (*e.g.,* feature-level fairness) in knowledge distillation. Also, considering sequential recommendation is drawing increasingly attention, it will be valuable to explore the model compression technique for the large sequential recommendation models.

7 ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (2021ZD0111802), the National Natural Science Foundation of China (62102382, 62272437, 62106221), the CCCD Key Lab of Ministry of Culture and Tourism and the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study (SN-ZJU-SIAS-001).

REFERENCES

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A General Framework for Counterfactual Learning-to-Rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 5–14. https://doi.org/10.1145/3331184.3331202

[2] Stephen Bonner and Flavian Vasile. 2018. Causal Embeddings for Recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 104–112. https://doi.org/10.1145/3240323.3240360

[3] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. *AutoDebias: Learning to Debias for Recommendation.* Association for Computing Machinery, New York, NY, USA, 21–30. https://doi.org/10.1145/3404835.3462919

[4] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. *CoRR* abs/2010.03240 (2020). arXiv:2010.03240 https://arxiv.org/abs/2010.03240

[5] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. 2017. A Survey of Model Compression and Acceleration for Deep Neural Networks. *CoRR* abs/1710.09282 (2017). arXiv:1710.09282 http://arxiv.org/abs/1710.09282

[6] Xiang Deng and Zhongfei Zhang. 2021. Comprehensive Knowledge Distillation with Causal Intervention. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 22158–22170.

[7] Sihao Ding, Fuli Feng, Xiangnan He, Jinqiu Jin, Wenjie Wang, Yong Liao, and Yongdong Zhang. 2022. Interpolative Distillation for Unifying Biased and Debiased Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 40–49.

[8] Fuli Feng, Weiran Huang, Xiangnan He, Xin Xin, Qifan Wang, and Tat-Seng Chua. 2021. *Should Graph Convolution Trust Neighbors? A Simple Causal Inference*

Method. Association for Computing Machinery, New York, NY, USA, 1208–1218. https://doi.org/10.1145/3404835.3462971

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. *LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation*. Association for Computing Machinery, New York, NY, USA, 639–648. https://doi.org/10.1145/3397271.3401063

[10] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. 2021. Distilling Causal Effect of Data in Class-Incremental Learning. *CVPR*, 3957–3966.

[11] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *2008 Eighth IEEE International Conference on Data Mining*. 263–272. https://doi.org/10.1109/ICDM.2008.22

[12] SeongKu Kang, Junyoung Hwang, Wonbin Kweon, and Hwanjo Yu. 2020. DE-RRD: A Knowledge Distillation Framework for Recommender System *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 605–614. https://doi.org/10.1145/3340531.3412005

[13] SeongKu Kang, Junyoung Hwang, Wonbin Kweon, and Hwanjo Yu. 2021. Topology Distillation for Recommender System. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 829–839. https://doi.org/10.1145/3447548.3467319

[14] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. 2018. An Adversarial Approach to Improve Long-Tail Performance in Neural Collaborative Filtering. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 1491–1494. https://doi.org/10.1145/3269206.3269264

[15] Wonbin Kweon, Seongku Kang, and Hwanjo Yu. 2021. Bidirectional Distillation for Top-K Recommender System. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 3861–3871. https://doi.org/10.1145/3442381.3449878

[16] Jae-woong Lee, Minjin Choi, Jongwuk Lee, and Hyunjung Shim. 2019. Collaborative Distillation for Top-N Recommendation. In *2019 IEEE International Conference on Data Mining (ICDM)*. 369–378. https://doi.org/10.1109/ICDM.2019.00047

[17] Kun Lin, Nasim Sonboli, Bamshad Mobasher, and Robin Burke. 2019. Crank up the volume: preference bias amplification in collaborative recommendation. *CoRR* abs/1909.06362 (2019). arXiv:1909.06362 http://arxiv.org/abs/1909.06362

[18] 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 *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 831–840. https://doi.org/10.1145/3397271.3401083

[19] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. 2019. Structured Knowledge Distillation for Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[20] Benjamin M. Marlin, Richard S. Zemel, Sam Roweis, and Malcolm Slaney. 2007. Collaborative Filtering and the Missing at Random Assumption. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence* (Vancouver, BC, Canada) *(UAI'07)*. AUAI Press, Arlington, Virginia, USA, 267–275.

[21] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. Counterfactual VQA: A Cause-Effect Look at Language Bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12700–12710.

[22] Judea Pearl. 2009. *Causality*. Cambridge University Press.

[23] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. *CoRR* abs/1205.2618 (2012). arXiv:1205.2618 http://arxiv.org/abs/1205.2618

[24] Yuta Saito. 2019. Eliminating Bias in Recommender Systems via Pseudo-Labeling. *CoRR* abs/1910.01444 (2019). arXiv:1910.01444 http://arxiv.org/abs/1910.01444

[25] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. In *Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.). PMLR, New York, New York, USA, 1670–1679. https://proceedings.mlr.press/v48/schnabel16.html

[26] Jiaxi Tang and Ke Wang. 2018. Ranking Distillation: Learning Compact Ranking Models With High Performance for Recommender System. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (London, United Kingdom) *(KDD'18)*. Association for Computing Machinery, New York, NY, USA, 2289–2298. https://doi.org/10.1145/3219819.3220021

[27] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. 2020. Long-Tailed Classification by Keeping the Good and Removing the Bad Momentum Causal Effect. In *NeurIPS*.

[28] Ye Tao, Ying Li, Su Zhang, Zhirong Hou, and Zhonghai Wu. 2022. Revisiting Graph Based Social Recommendation: A Distillation Enhanced Social Graph Network. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 2830–2838. https://doi.org/10.1145/3485447.3512003

[29] Yuening Wang, Yingxue Zhang, and Mark Coates. 2021. Graph Structure Aware Contrastive Knowledge Distillation for Incremental Learning in Recommender Systems. In *Proceedings of the 30th ACM International Conference on Information Knowledge Management* (Virtual Event, Queensland, Australia) *(CIKM '21)*. Association for Computing Machinery, New York, NY, USA, 3518–3522. https://doi.org/10.1145/3459637.3482117

[30] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-Agnostic Counterfactual Reasoning for Eliminating Popularity Bias in Recommender System. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 1791–1800. https://doi.org/10.1145/3447548.3467289

[31] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Guandong Xu, and Quoc Viet Hung Nguyen. 2022. On-Device Next-Item Recommendation with Self-Supervised Knowledge Distillation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 546–555.

[32] Shuyuan Xu, Yunqi Li, Shuchang Liu, Zuohui Fu, Yingqiang Ge, Xu Chen, and Yongfeng Zhang. 2021. Learning causal explanations for recommendation. In *The 1st International Workshop on Causality in Search and Recommendation*.

[33] Zixuan Xu, Penghui Wei, Weimin Zhang, Shaoguo Liu, Liang Wang, and Bo Zheng. 2022. UKD: Debiasing Conversion Rate Estimation via Uncertainty-regularized Knowledge Distillation. *CoRR* abs/2201.08024 (2022). arXiv:2201.08024 https://arxiv.org/abs/2201.08024

[34] C. Yang, J. Pan, X. Gao, T. Jiang, D. Liu, and G. Chen. 2022. Cross-Task Knowledge Distillation in Multi-Task Recommendation. (2022).

[35] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased Offline Recommender Evaluation for Missing-Not-at-Random Implicit Feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 279–287. https://doi.org/10.1145/3240323.3240355

[36] Mengyue Yang, Quanyu Dai, Zhenhua Dong, Xu Chen, Xiuqiang He, and Jun Wang. 2021. Top-N Recommendation with Counterfactual User Preference Simulation. In *Proceedings of the 30th ACM International Conference on Information Knowledge Management* (Virtual Event, Queensland, Australia) *(CIKM '21)*. Association for Computing Machinery, New York, NY, USA, 2342–2351. https://doi.org/10.1145/3459637.3482305

[37] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. 2017. A Gift From Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[38] Hanwang Zhang, Fumin Shen, Wei Liu, Xiangnan He, Huanbo Luan, and Tat-Seng Chua. 2016. Discrete Collaborative Filtering. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Pisa, Italy) *(SIGIR '16)*. Association for Computing Machinery, 325–334.

[39] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. *Causal Intervention for Leveraging Popularity Bias in Recommendation*. Association for Computing Machinery, New York, NY, USA, 11–20. https://doi.org/10.1145/3404835.3462875

[40] Yuan Zhang, Xiaoran Xu, Hanning Zhou, and Yan Zhang. 2020. Distilling structured knowledge into embeddings for explainable and accurate recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 735–743.

[41] Jieming Zhu, Jinyang Liu, Weiqi Li, Jincai Lai, Xiuqiang He, Liang Chen, and Zibin Zheng. 2020. Ensembled CTR prediction via knowledge distillation. In *Proceedings of the 29th ACM International Conference on Information Knowledge Management*. 2941–2958.

[42] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-Opportunity Bias in Collaborative Filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (Virtual Event, Israel) *(WSDM '21)*. Association for Computing Machinery, New York, NY, USA, 85–93. https://doi.org/10.1145/3437963.3441820