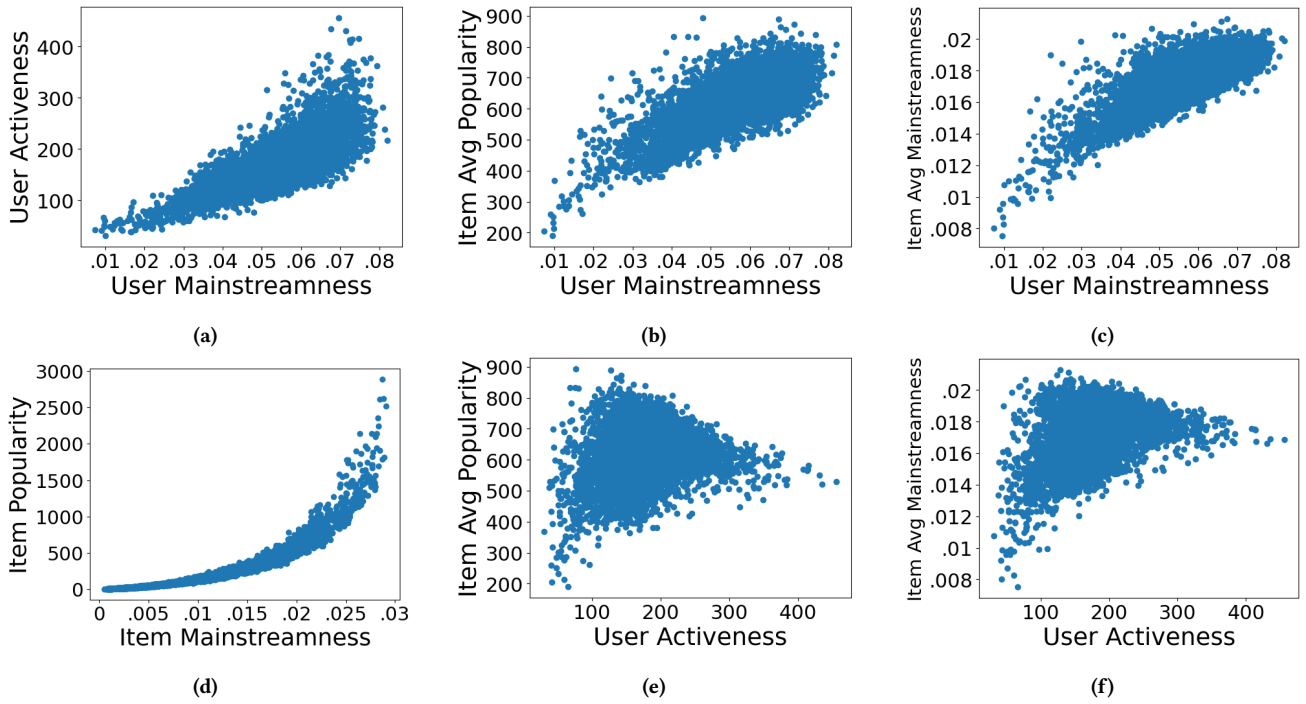


Figure 5: Correlation between dimensions of users and items.

A Correlation between Dimensions of Users and Items

We conduct comprehensive and empirical evidence of relationships of heterogeneous features of users and items in Figure 5 with the KuaiRec dataset. Concretely, other than the correlations we introduced in Section 1, we also present Figure 5c, similar to Figure 5b, to demonstrate that niche users tend to engage with niche items, leading to the connection between user mainstream bias and item mainstream bias. Additionally, Figure 5e exhibits the correlation of user activeness and average item popularity the corresponding users interact with, where each point represents a user, and the y-axis means the average popularity score that the average total appearances of all items this user positively interacted (a higher score indicates the user interact with more popular items). Subsequently, this illustrates the connection between active users and popular items, which shows that some inactive users tend to interact with unpopular items, and vice versa, in which user activeness may correlate with item popularity bias. Similarly, Figure 5f shows evidence that active users tend to engage with mainstream items, illustrating the correlation between user activeness bias and item mainstream bias.

B Evaluating User/Item Levels

Concretely, we take the calculation of user mainstreamness as an example. To differentiate users of different mainstreamness, we calculate a mainstream score for each user to indicate the level of mainstreamness of the user. A higher mainstream score indicates that the user is more likely to be a mainstream user. Toward this, [57] proposes a similarity-based approach, whose core idea is that a user with high overall similarity to other users is likely to be a mainstream user. Specifically, for a user u, the mainstream score is calculated as:

$$MS_u = \sum_{v \in \mathcal{U} \setminus u} Sim(\mathbf{O}_u, \mathbf{O}_v)/(N-1), \tag{9}$$

where $Sim(\mathbf{O}_u, \mathbf{O}_v)$ is the user-user similarity between users u and v. The similarity can be computed by Jaccard similarity between the feedback record $\mathbf{O}_u$ and $\mathbf{O}_v$.

With the mainstream scores of all users, we sort users in a non-descending order of their mainstream scores and evenly divide them into subgroups. Then, we can compute the average recommendation utility of each subgroup and demonstrate the mainstream bias by comparing the average performance of subgroups of different mainstreamness: if subgroups of high mainstreamness receive higher utility than subgroups of lower mainstreamness, then mainstream bias is observed. It is very similar for user activeness calculation, replacing $Sim(\mathbf{O}_u, \mathbf{O}_v)$ with the user's historical interaction count. As for item levels, it is akin to calculate two user side levels, except we replace users with items in the calculation.

C Item-Oriented Recommendation Utility

In general, recommender systems evaluate recommendation utility by user-oriented metrics, such as NDCG@K, in which the quality of a ranking list for a user is evaluated. To gauge the recommendation utility an item gets in a system, we adopt the Mean Discounted Gain (MDG@K) [59]:

$$MDG@K = \sum_{i \in \mathcal{I}} \frac{1}{|\mathcal{U}_i^+|} \sum_{u \in \mathcal{U}_i^+} \frac{\mathbb{I}[\hat{\mathcal{P}}(u,i) < K]}{\log_2(\hat{\mathcal{P}}(u,i)+1)}, \tag{10}$$

Table 5: Cross-Debiasing Effects of Bias-Specific Models on Various Bias Types.

Recommendation Utility	Niche Users	Inactive Users	Niche Items	Unpopular Items
MF	.2878	.2568	.0015	.0013
LOCA	.3043	.2771	.0012	.0008
EnLFT	.3186	.2871	.0017	.0015
LFT	.3257	.2901	.0020	.0019
CFBoost	**.3364**	**.3030**	**.0025**	**.0025**

where $\mathcal{U}_i^+$ is the set of users who have positive interactions for item i in the test set (i.e., intended audiences); $\hat{\mathcal{P}}(u,i)$ returns the position the item i is ranked at for user u; and $\mathbb{I}[\cdot]$ is the indicator function. MDG@K evaluates how well an item is recommended to its intended audiences. A high MDG@K indicates that the item is successfully recommended (ranked at a top position) to users who like it.

D Data Driven Study on Cross-Debiasing Effects

We conduct experiments with three SOTA methods – LOCA [13], EnLFT [57], and LFT [57] – tailored to reduce user mainstream bias, i.e., they aim to promote the utility niche users receive. We want to explore their capability to mitigate other bias types. Experiment details are provided in Section 5. We show the utility of models to different minority user and item groups in Table 5. When compared to traditional Matrix Factorization [26] (MF), all three approaches notably enhance utility for niche users, diminishing user mainstream bias. Additionally, we note improvements in utility for other minority groups as well, underscoring the interconnected nature of these biases. More importantly, it underlines the great potential of addressing heterogeneous biases by a unified framework, which is exemplified by the impressive performance of our proposed CFBoost in Table 5.

E CFAdaBoost

Algorithm 2: CFAdaBoost

Input : $\mathcal{O} = \{(u,i)\}$; number of boosting iterations $\mathcal{T}$.
Output: Final predictions $\widehat{\mathcal{O}}$.

Initialize: $weights(w^1) = 1/(M*N)$;
for $t \leftarrow 1$ *to* $\mathcal{T}$ **do**
  Train sub-model θ_t with weights (w^t) as Equation 1;
  Calculate $\mathcal{L}_{u,i}^t$ for each user-item pair (u,i);
  Calculate the weighted error: $\epsilon_t = \sum_{(u,i)\in\mathcal{O}} w_{u,i}^t \times \mathcal{L}_{u,i}^t$;
  Calculate the ensemble weight: $\alpha_t = \ln(1-\epsilon_t)$;
  Calculate the sample weight of each user-item pair for the next iteration: $w_{u,i}^{t+1} = \frac{w_{u,i}^t \times e^{\alpha_t \mathcal{L}_{u,i}^t}}{Z_t}$;
end for
For the final prediction: $\widehat{\mathcal{O}} = \sum_{t=1}^{\mathcal{T}} \alpha_t \times \widehat{\mathcal{O}}_t$.

We introduce our first algorithm CFAdaBoost inspired by the AdaBoost algorithm [18, 42], which trains a series of sub-models (denoted as $\{\theta_1, \ldots, \theta_{\mathcal{T}}\}$) for total $\mathcal{T}$ boosting iterations. At the boosting iteration t, we have trained sub-models $\{\theta_1, \ldots, \theta_{t-1}\}$ from prior iterations. Our goal at the current iteration is to train a new sub-model θ_t focusing on reducing the training losses for user-item pairs that are overlooked by prior $t-1$ sub-models, while paying less attention to samples that have been effectively modeled. Towards this, we introduce the sample weight $w_{u,i}^t$ for each user-item pair, where $\sum_{(u,i)\in\mathcal{O}} w_{u,i}^t = 1$, and derive a weighted training loss for learning θ_t (Equation 1).

The whole process of CFBoost and CFAdaBoost is the same. The value of $w_{u,i}^t$ is determined by the training losses for (u,i) from $\{\theta_1, \ldots, \theta_{t-1}\}$: if the aggregated training loss for (u,i) is high, $w_{u,i}^t$ should take a high value to push θ_t to reduce the training loss, vice versa.

After $\mathcal{T}$ boosting iterations, we ensemble all $\mathcal{T}$ sub-models, each of which is tailored for specific users and items, aiming to reduce training loss for all users and items. We introduce the ensemble weight $\alpha_t \in \mathbb{R}$ to weighted ensemble $\mathcal{T}$ sub-models: $\widehat{\mathcal{O}} = \sum_{t=1}^{\mathcal{T}} \alpha_t \times \widehat{\mathcal{O}}_t$, where $\widehat{\mathcal{O}}_t$ is the predictions from θ_t. The intuition of α_t is that if the sub-model θ_t is effective (reflected by a low training loss), it should contribute more to the ensembled predictions, reflected by a large value of α_t, and vice versa. Hence, we first calculate a weighted loss to indicate the efficacy of the sub-model θ_t:

$$\epsilon_t = \sum_{(u,i)\in\mathcal{O}} w_{u,i}^t \times \mathcal{L}_{u,i}^t, \tag{11}$$

with which, we define the ensemble weight α_t as:

$$\alpha_t = \ln(1-\epsilon_t). \tag{12}$$

At last, we define the sample weight $w_{u,i}^t$ as:

$$w_{u,i}^t = \frac{w_{u,i}^{t-1} \times e^{\alpha_{t-1}\mathcal{L}_{u,i}^{t-1}}}{Z_{t-1}}, \qquad Z_{t-1} = \sum_{u\in\mathcal{U}, i\in\mathcal{I}} w_{u,i}^{t-1} \times e^{\alpha_{t-1}\mathcal{L}_{u,i}^{t-1}}, \tag{13}$$

where Z_{t-1} is the normalization term; and this recursive product formula aggregates $\{\mathcal{L}_{u,i}^1, \ldots, \mathcal{L}_{u,i}^{t-1}\}$ by $\{\alpha_1, \ldots, \alpha_{t-1}\}$ to indicate how effectively has the user-item pair (u,i) been modeled by prior $t-1$ sub-models. If (u,i) has not been effectively modeled (a high aggregated training loss), we have a high $w_{u,i}^t$ to push the current θ_t to pay more attention to it. The complete algorithm is in Algorithm 2.

By training more and more sub-models that focus on learning samples that were previously overlooked, CFAdaBoost can gradually reduce the training loss for all types of users and items. We further conclude the following theorem, showing that CFAdaBoost can decrease the upper bound of the training loss at an exponential speed for all user-item pairs.

Theorem E.1. *Given the notation of Algorithm 2, the training loss $\mathcal{L}_{u,i}$ for all user-item pairs $\forall(u,i)$ is bounded as: $\mathcal{L}_{u,i} \le e^{-\epsilon^2\mathcal{T}-1}$.*

Proof. The sample weight for (u,i) at boosting iteration $\mathcal{T}+1$ can be written as:

$$\begin{aligned} w_{u,i}^{\mathcal{T}+1} &= w_{u,i}^{1} \times \frac{exp(\alpha_1 \times \mathcal{L}_{u,i}^{1})}{Z_1} \times \cdots \times \frac{exp(\alpha_{\mathcal{T}} \times \mathcal{L}_{u,i}^{\mathcal{T}})}{Z_{\mathcal{T}}} \\ &= \frac{w_{u,i}^{1} \times exp(\sum_{t=1}^{\mathcal{T}} \alpha_t \times \mathcal{L}_{u,i}^{t})}{\prod_{t=1}^{\mathcal{T}} Z_t} = \frac{exp(\sum_{t=1}^{\mathcal{T}} \alpha_t \times \mathcal{L}_{u,i}^{t})}{\prod_{t=1}^{\mathcal{T}} Z_t}. \end{aligned} \tag{14}$$

We can remove $w_{u,i}^{1}$ in the second line because $w_{u,i}^{1}$ is a predefined constant that won't influence our further derivation. And we can derive $exp(\sum_{t=1}^{\mathcal{T}} \alpha_t \times \mathcal{L}_{u,i}^{t})$ from this result as:

$$e^{\sum_{t=1}^{\mathcal{T}} \alpha_t \times \mathcal{L}_{u,i}^{t}} = w_{u,i}^{\mathcal{T}+1} \times \prod_{t=1}^{\mathcal{T}} Z_t. \tag{15}$$

Then, we can formulate the training loss for a user-item pair (u, i) in CFAdaBoost as:

$$\begin{aligned} \mathcal{L}_{u,i} &= \sum_{t=1}^{\mathcal{T}} \alpha_t \times \mathcal{L}_{u,i}^{t} \leq exp\left(\left(\sum_{t=1}^{\mathcal{T}} \alpha_t \times \mathcal{L}_{u,i}^{t}\right) - 1\right) \\ &= \frac{w_{u,i}^{\mathcal{T}+1}}{e} \times \prod_{t=1}^{\mathcal{T}} Z_t \leq \frac{1}{e} \prod_{t=1}^{\mathcal{T}} Z_t. \end{aligned} \tag{16}$$

The second line is derived based on the rule $1+x \leq e^{x}$ for all x [18], and the third line is derived based on Equation 15. We further decompose Z_t as:

$$\begin{aligned} Z_t &= \sum_{u \in \mathcal{U}, i \in \mathcal{I}} w_{u,i}^{t} \times e^{\alpha_t \mathcal{L}_{u,i}^{t}} \\ &\leq \sum_{u \in \mathcal{U}, i \in \mathcal{I}} w_{u,i}^{t}(1 - (1 - e^{\alpha_t})\mathcal{L}_{u,i}^{t}) \\ &= \sum_{u \in \mathcal{U}, i \in \mathcal{I}} w_{u,i}^{t} - \sum_{u \in \mathcal{U}, i \in \mathcal{I}} w_{u,i}^{t}(1 - e^{\alpha_t})\mathcal{L}_{u,i}^{t} \\ &= 1 - \epsilon_t(1 - e^{\alpha_t}) = 1 - \epsilon_t(1 - e^{\ln(1-\epsilon_t)}) \\ &= 1 - \epsilon_t^2 \leq e^{-\epsilon_t^2}, \end{aligned} \tag{17}$$

where we obtain the second line by a convexity argument [18]: $x^y \leq 1-(1-x)y$ for $y \in [0, 1]$ (we can enforce the constraint $0 \leq \mathcal{L}_{u,i}^{t} \leq 1$ in practice). Last, we combine Equation 17 and Equation 16 to have the upper bound of the training loss: $\mathcal{L}_{u,i} \leq e^{-\epsilon^2 \mathcal{T}-1}$, where ϵ is the minimum weighted error for (u, i), which is a positive value. Consequently, the whole equation will exponentially decay when the number of boosting iterations $\mathcal{T}$ increases.

□

In sum, the proposed CFAdaBoost can effectively decrease the training loss for all users and items, no matter if they belong to privileged or minority groups, resulting in a significant reduction of heterogeneous biases. However, a significant limitation of CFAdaBoost lies in its inflexibility in assigning the ensemble weight (α) as a fixed value for all user-item pairs. This contradicts our intuition that a sub-model focusing on a specific group of users and items should contribute more in the final prediction for these target users and items (with higher α values), while contributing less to other users and items that this sub-model cannot predict well (with lower α values). Applying a uniform α can significantly undermine the debiasing efficacy and overall recommendation utility. For instance, assigning a single α^t to a sub-model θ_t tailored for mainstream users fails to account for the unique needs of niche users. This uniform contribution in the final prediction does not differentiate between mainstream and niche users, potentially degrading accuracy for the latter group.

Table 6: Statistics of three datasets.

	#users	#items	density
KuaiRec	5,765	5,800	4.39%
Yelp	20,001	7,643	0.32%
CDs & Vinyl	12,023	8,050	0.32%

F Debiasing Performance

Table 6 summarizes the dataset statistics used in this paper. Then first, we conduct a comparative analysis to show the effectiveness of the proposed CFBoost. In Table 7, Table 8, Table 9, and Table 10, we evaluate the overall NDCG@20/MDG@20 and average NDCG@20/MDG@20 for 5 user subgroups with varying mainstream levels and activeness levels, as well as 5 item subgroups with varying mainstream levels and popularity levels, for all methods and datasets. The best baseline results of each metric and subgroup for all datasets are marked with underlining, the best results of each metric and subgroup for all datasets are marked in bold, and the improvement rate of the proposed CFBoost over the best baselines is exhibited as well. The user subgroups are categorized based on their mainstream scores (Table 7) and activeness scores (Table 8), and the item subgroups are categorized based on their mainstream scores (Table 9) and popularity scores (Table 10), with the 'low' subgroup containing the 20% users or items with the lowest scores, 'med-low' subgroup containing 20% to 40% users or items in the sorted user or item sequence, and so on for 40%-60% ('medium'), 60%-80% ('med-high'), and 80%-100% ('high') users or items.

Our proposed CFBoost and CFAdaBoost effectively address both user-side and item-side biases, with CFBoost exhibiting superior performance. Concretely, both CFBoost and CFAdaboot can effectively address user-side biases, but CFBoost significantly outperforms in item-side recommendation performances than CFAdaBoost. This illustrates that our proposed CFBoost can mitigate various biases simultaneously and effectively.

Concretely, for user mainstream bias and user activeness bias, CFBoost significantly enhances the performance of user groups ('low' and 'med-low') compared to the original MultVAE and MF models. In Table 7 and Table 8, a comparison of CFBoost against the best baselines including SOTA local learning methods (LOCA and LFT) for each subgroup reveals notable improvements, with the 'low' niche user group exhibiting an average improvement rate of approximately 6.39% and with the utility of CFBoost in inactive user groups surpassing other baselines by an average improvement rate of approximately 6.33%. Consequently, our proposed CFBoost significantly alleviates user mainstream bias and user activeness bias, outperforming bias-unaware models (MultVAE and MF) and SOTA local learning methods across most of the user groups.

In addition, for item mainstream bias and item popularity bias, CFBoost significantly improves the recommendation utility of niche

Table 7: Comparison across SOTA debiasing baselines and CFBoost on three datasets (user mainstream bias).

User Mainstream	KuaiRec						Yelp						CDs & Vinyl					
	NDCG	Subgroups of mainstream levels					NDCG	Subgroups of mainstream levels					NDCG	Subgroups of mainstream levels				
	@20	L	ML	M	MH	H	@20	L	ML	M	MH	H	@20	L	ML	M	MH	H
MF	.3048	.2878	.3034	.2988	.3106	.3232	.0748	.0559	.0600	.0608	.0772	.1203	.1292	.1095	.1171	.1330	.1388	.1474
BPR	.1925	.1314	.1661	.1973	.2157	.2519	.0594	.0453	.0463	.0474	.0592	.0991	.1055	.0917	.0986	.1112	.1113	.1149
MultVAE	.3189	.3020	.3174	.3114	.3208	.3428	.0874	.0636	.0680	.0748	.0953	.1357	.1382	.1175	.1244	.1446	.1477	.1570
LOCA	.3342	.3043	.3311	.3302	.3440	**.3613**	.0979	.0763	.0785	.0785	.1035	**.1528**	.1593	.1364	.1479	.1596	.1728	.1799
EnLFT	.3344	.3186	.3367	.3276	.3356	.3536	.0894	.0665	.0705	.0754	.0976	.1371	.1519	.1252	.1369	.1527	.1657	.1788
LFT	.3372	.3257	.3349	.3311	.3375	.3569	.0950	.0734	.0750	.0794	.1022	.1451	.1583	.1322	.1430	.1563	.1747	.1851
PC	.1522	.1056	.1308	.1548	.1704	.1994	.0578	.0415	.0437	.0461	.0587	.0992	.1021	.0884	.0965	.1064	.1078	.1116
BC Loss	.2647	.2605	.2670	.2598	.2593	.2770	.0825	.0622	.0640	.0682	.0878	.1301	.1360	.1135	.1275	.1429	.1437	.1521
Zero Sum	.2012	.1517	.1825	.2045	.2173	.2499	.0557	.0417	.0426	.0433	.0563	.0947	.1006	.0843	.0933	.1054	.1075	.1123
CFAdaBoost	.3427	.3309	**.3422**	.3351	**.3457**	.3595	**.1017**	.0821	.0835	.0851	**.1049**	.1526	**.1660**	**.1481**	.1532	**.1681**	**.1755**	**.1852**
CFBoost	**.3449**	**.3364**	.3416	**.3399**	.3453	.3611	.1004	**.0824**	**.0843**	**.0852**	.1010	.1491	.1644	.1472	**.1541**	.1658	.1698	.1851
Δ_{best}(%)	2.28	3.29	1.47	2.65	0.37	-0.06	2.55	7.99	7.39	7.30	-2.42	-2.42	3.20	7.88	4.19	3.87	-1.74	0
Avg Δ_{best}(%)		1.54						4.06						2.84				

L: low, ML: med-low, M: medium, MH: med-high, H: high

Table 8: Comparison across SOTA debiasing baselines and CFBoost on three datasets (user activeness bias).

User Activeness	KuaiRec						Yelp						CDs & Vinyl					
	NDCG	Subgroups of activeness levels					NDCG	Subgroups of activeness levels					NDCG	Subgroups of activeness levels				
	@20	L	ML	M	MH	H	@20	L	ML	M	MH	H	@20	L	ML	M	MH	H
MF	.3048	.2568	.2959	.3088	.3202	.3424	.0748	.0674	.0675	.0700	.0757	.0936	.1292	.1101	.1157	.1255	.1358	.1587
BPR	.1925	.1251	.1737	.1983	.2173	.2479	.0594	.0556	.0541	.0553	.0596	.0725	.1055	.0962	.1005	.1074	.1053	.1183
MultVAE	.3189	.2738	.3074	.3214	.3336	.3583	.0874	.0735	.0780	.0812	.0903	.1142	.1382	.1171	.1245	.1338	.1438	.1721
LOCA	.3342	.2771	.3223	.3351	.3517	**.3847**	.0979	.0847	.0890	.0920	.1000	.1237	.1593	.1331	.1424	.1535	.1675	.2003
EnLFT	.3344	.2871	.3254	.3362	.3495	.3738	.0894	.0749	.0818	.0830	.0920	.1155	.1519	.1259	.1304	.1452	.1582	.1995
LFT	.3372	.2901	.3251	.3380	.3516	.3797	.0950	.0809	.0868	.0881	.0971	.1221	.1583	.1318	.1396	.1529	.1644	**.2027**
PC	.1522	.0981	.1376	.1553	.1721	.1980	.0578	.0538	.0533	.0537	.0582	.0703	.1021	.0958	.1002	.1049	.1019	.1078
BC Loss	.2647	.2226	.2648	.2694	.2717	.2618	.0825	.0720	.0767	.0760	.0842	.1032	.1360	.1184	.1338	.1348	.1422	.1505
Zero Sum	.2012	.1364	.1876	.2088	.2229	.2501	.0557	.0521	.0504	.0515	.0559	.0689	.1006	.0880	.0929	.1028	.1021	.1169
CFAdaBoost	.3427	.2982	.3337	.3443	.3552	.3817	**.1017**	.0909	.0932	.0948	.1028	**.1264**	**.1660**	.1413	**.1506**	.1618	.1741	.2024
CFBoost	**.3449**	**.3030**	**.3377**	**.3463**	**.3586**	.3788	.1004	**.0916**	**.0936**	**.0964**	**.1043**	.1162	.1644	**.1416**	.1505	**.1637**	**1785**	.1876
Δ_{best}(%)	2.28	4.45	3.77	2.46	1.96	-1.53	2.55	8.15	5.17	4.78	4.30	-6.06	3.20	6.40	5.71	6.63	6.55	-7.45
Avg Δ_{best}(%)		2.22						3.27						3.57				

L: low, ML: med-low, M: medium, MH: med-high, H: high

Table 9: Comparison across SOTA debiasing baselines and CFBoost on three datasets (item mainstream bias).

Item Mainstream	KuaiRec						Yelp						CDs & Vinyl					
	MDG	Subgroups of mainstream levels					MDG	Subgroups of mainstream levels					MDG	Subgroups of mainstream levels				
	@20	L	ML	M	MH	H	@20	L	ML	M	MH	H	@20	L	ML	M	MH	H
MF	.0102	.0015	.0023	.0038	.0069	.0362	.0143	.0050	.0114	.0124	.0163	.0265	.0266	.0206	.0172	.0205	.0276	.0473
BPR	.0078	0	0	0	0	.0391	.0063	.0011	.0057	.0067	.0073	.0107	.0088	.0032	.0018	.0047	.0093	.0252
MultVAE	.0126	.0014	.0031	.0051	.0115	.0416	.0142	.0015	.0096	.0114	.0162	.0322	.0234	.0096	.0091	.0158	.0272	.0554
LOCA	.0102	.0012	.0022	.0028	.0065	.0381	.0193	.0048	.0159	.0185	.0207	.0367	.0427	.0305	.0284	.0345	.0472	.0731
EnLFT	.0119	.0017	.0025	.0031	.0092	.0430	.0155	.0019	.0110	.0126	.0178	.0345	.0433	.0265	.0285	.0350	.0490	.0773
LFT	.0105	.0020	.0029	.0028	.0075	.0374	.0170	.0031	.0131	.0142	.0191	.0355	.0457	.0259	.0301	.0386	.0527	.0810
PC	.0022	.0015	0	0	0	.0094	.0068	.0050	.0057	.0064	.0068	.0099	.0143	.0293	.0070	.0063	.0081	.0209
BC Loss	.0065	.0007	.0014	.0012	.0020	.0274	.0170	.0078	.0149	.0157	.0189	.0278	.0453	.0478	.0381	.0399	.0446	.0561
Zero Sum	.0026	0	0	0	0	.0127	.0059	.0011	.0053	.0062	.0067	.0099	.0083	.0014	.0010	.0027	.0095	.0267
CFAdaBoost	.0125	.0017	.0029	.0039	.0095	**.0445**	.0191	.0061	.0156	.0172	.0205	.0362	.0371	.0306	.0249	.0290	.0381	.0629
CFBoost	**.0133**	**.0025**	**.0041**	**.0073**	**.0137**	.0390	**.0249**	**.0097**	**.0224**	**.0258**	**.0280**	**.0389**	**.0613**	**.0482**	**.0494**	**.0597**	**.0674**	**.0821**
Δ_{best}(%)	5.56	25.47	32.40	42.34	18.61	-9.42	29.01	24.35	40.88	39.46	35.27	5.99	34.14	0.73	29.66	49.35	27.85	1.39
Avg Δ_{best}(%)		21.88						29.19						21.80				

L: low, ML: med-low, M: medium, MH: med-high, H: high

Table 10: Comparison across SOTA debiasing baselines and CFBoost on three datasets (item popularity bias).

Item Popularity	KuaiRec						Yelp						CDs & Vinyl					
	MDG	Subgroups of popularity levels					MDG	Subgroups of popularity levels					MDG	Subgroups of popularity levels				
	@20	L	ML	M	MH	H	@20	L	ML	M	MH	H	@20	L	ML	M	MH	H
MF	.0102	.0013	.0021	.0032	.0072	.0370	.0143	.0010	.0024	.0055	.0147	.0483	.0266	.0079	.0115	.0165	.0237	.0736
BPR	.0078	0	0	0	0	.0391	.0063	0	7.7e-6	.0001	.0029	.0285	.0088	.0005	.0006	.0008	.0016	.0406
MultVAE	.0126	.0012	.0027	.0046	.0117	.0426	.0142	0	.0001	.0003	.0076	.0629	.0234	.0014	.0029	.0065	.0175	.0908
LOCA	.0102	.0008	.0023	.0023	.0065	.0389	.0193	.0009	.0026	.0071	.0208	**.0674**	.0427	.0143	.0233	.0281	.0418	.1063
EnLFT	.0119	.0015	.0023	.0027	.0090	.0439	.0155	0	.0002	.0010	.0112	.0653	.0433	.0116	.0202	.0273	.0438	.1135
LFT	.0105	.0019	.0026	.0024	.0074	.0384	.0170	.0001	.0009	.0034	.0148	.0658	.0457	.0109	.0206	.0319	.0501	**.1147**
PC	.0022	.0015	0	0	0	.0094	.0068	.0030	.0009	.0006	.0027	.0266	.0143	.0158	.0088	.0061	.0053	.0355
BC Loss	.0065	.0006	.0013	.0010	.0020	.0278	.0170	.0019	.0037	.0077	.0173	.0545	.0453	.0242	.0338	.0365	.0458	.0862
Zero Sum	.0026	0	0	0	0	.0127	.0059	0	0	.0002	.0027	.0264	.0083	.0001	.0002	.0003	.0006	.0409
CFAdaBoost	.0125	.0016	.0025	.0033	.0096	**.0454**	.0191	.0012	.0033	.0069	.0187	.0655	.0371	.0138	.0195	.0243	.0324	.0954
CFBoost	**.0133**	**.0025**	**.0036**	**.0061**	**.0144**	.0399	**.0249**	**.0066**	**.0118**	**.0174**	**.0291**	.0598	**.0613**	**.0458**	**.0554**	**.0590**	**.0588**	.0876
$\Delta_{best}(\%)$	5.56	30.98	33.47	34.16	23.43	-9.18	29.01	120.00	218.92	125.97	39.90	-11.28	34.14	89.57	63.98	61.62	17.36	-23.65
Avg $\Delta_{best}(\%)$		22.57						98.7						41.78				

L: low, ML: med-low, M: medium, MH: med-high, H: high

item groups ('low' and 'med-low') compared to the basic recommendation models, MultVAE and MF. In Table 9 and Table 10, the utility of CFBoost for niche item groups outperforms the best baselines, including SOTA item-side debiasing methods (PC and BC Loss), with an average improvement rate of approximately 16.85% for the 'low' mainstreamness group and outperforms other baselines with an average improvement rate of approximately 80.18% for 'low' popularity group. This demonstrates our proposed CFBoost can mitigate item mainstream bias and item popularity bias, and CFBoost showcases superior recommendation utility compared to bias-unaware models and SOTA item-side debiasing methods across diverse item groups.

In sum, from Table 7, Table 8, Table 9, and Table 10, we see that for all datasets, the proposed CFBoost produces the greatest NDCG@20 and MDG@20 improvement for each subgroup across user mainstream levels, user activeness levels, item mainstream levels, and item popularity levels, especially for minority groups, and leads to the SOTA overall model performance. This indicates that our proposed CFBoost can effectively address multiple biases all at once. On the other hand, although our two boosting designs, CFAdaBoost and CFBoost both contain the ability to solve multiple biases simultaneously, we can observe that CFBoost with adaptive α designed for reducing each user-item pair training loss stronger performance not only in user-side biases, but also in item-side biases compared to CFAdaBoost. Furthermore, CFBoost can outperform baselines with lower and the same model complexity, and it can even outperform the baseline model that is way more complex than it, such as LFT. On the other hand, our proposed CFBoost does not consistently yield the highest utility for the most privileged groups because it focuses on each user-item pair, thereby potentially neglecting the needs of privileged groups and resulting in a trade-off between minority and privileged groups. In many scenarios, CFBoost allocates more attention to minority groups in each boosting iteration, which were previously overlooked by other sub-models, while privileged groups typically learn effectively without requiring additional iterations of attention. In contrast, other bias-unaware models may continuously train all or some privileged groups to maximize their utility. However, CFBoost still delivers significantly higher recommendation utility than MF, BPR, and MultVAE.

G Hyper-parameter Study

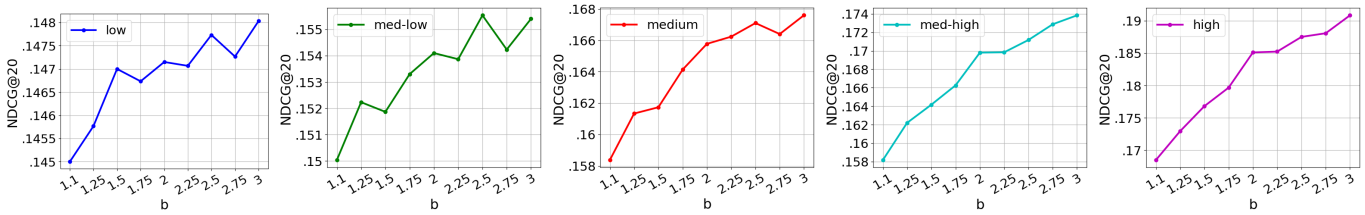
Next, we study the impacts of three hyper-parameters in CFBoost: (1) b in α calculation; (2) the number of boosting iterations ($\mathcal{T}$); and (3) the number of negative sampling rounds ($\mathcal{J}$).

G.1 b

The goal of b is to control the skewness of α for each user-item pair. Different users and items have different needs of α as well. If b is high, then the distribution of α is more skewed, the α value difference between minority users/items and privileged users/items is bigger, and vice versa. The intuition is a larger b can lead to a more skewed distribution and stronger debiasing performance. Here, we run CFBoost on the CDs&Vinyl dataset with the values(b) varying in {1.1, 1.25, 1.5, 2, 2.25, 2.5, 2.75, 3} and other settings the same as in Section 5.2. The overall NDCG@20 and MDG@20 results for user groups and item groups with all user mainstream levels, user activeness levels, item mainstream levels, and item popularity levels are presented in Figure 6, Figure 7, Figure 8, and Figure 9 respectively. We observe that when the b value becomes bigger, the utilities of niche user groups and inactive user groups increase, however, the utilities of all niche item subgroups first increase and then decrease along with increasing b. Moreover, the 'high' subgroup for all user-side and item-side settings is increasing with increasing b. These indicate items generally do not need to have larger α, whereas users often benefit from larger α values to effectively enhance recommendation performance. And we take $b = 2$ to have the best-balanced debiasing ability for both user-side and item-side.

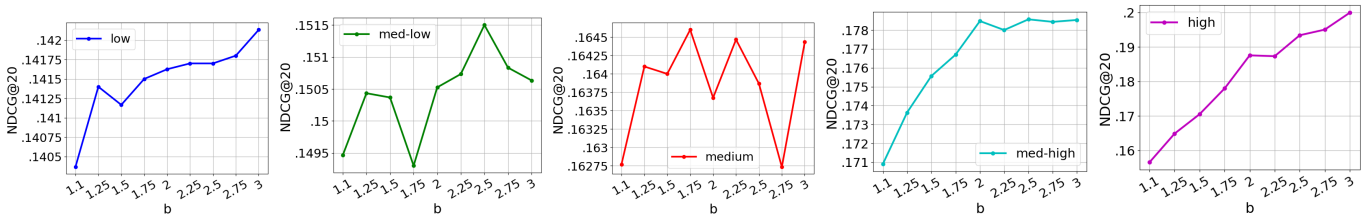
G.2 Number of Booting Iterations $\mathcal{T}$

This hyper-parameter controls how many sub-models we implement in the training and ensemble. It is expected the framework can be more effective in addressing multiple biases when the number of training iterations increases, and then the framework approaches



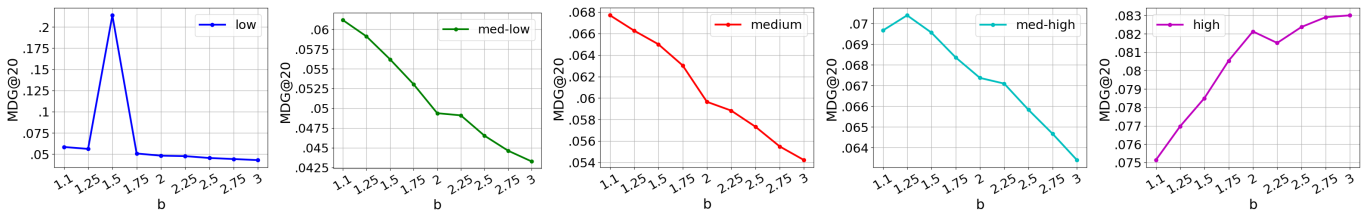
(a) low (b) med-low (c) medium (d) med-high (e) high

Figure 6: Performance of CFBoost on five subgroups of user mainstreamness on CD dataset with varying b.



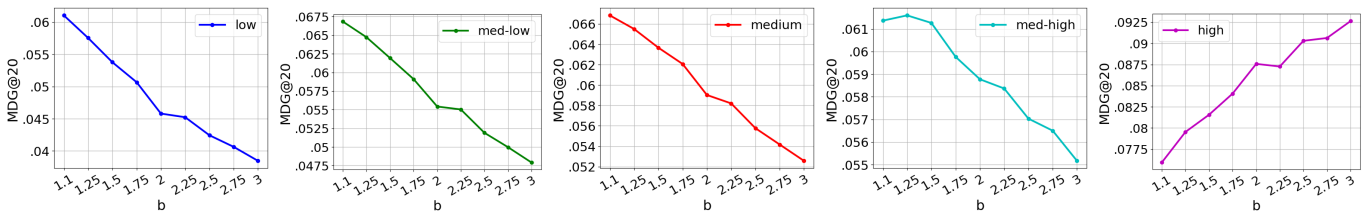
(a) low (b) med-low (c) medium (d) med-high (e) high

Figure 7: Performance of CFBoost on five subgroups of user activeness on CD dataset with varying b.



(a) low (b) med-low (c) medium (d) med-high (e) high

Figure 8: Performance of CFBoost on five subgroups of item mainstreamness on CD dataset with varying b.



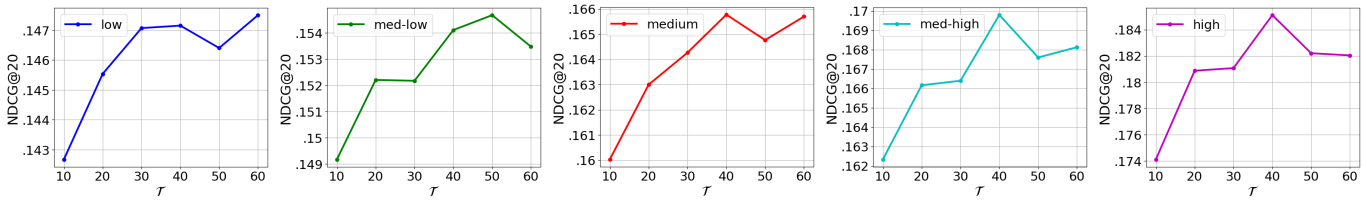
(a) low (b) med-low (c) medium (d) med-high (e) high

Figure 9: Performance of CFBoost on five subgroups of item popularity on CD dataset with varying b.

peak performance and converges. In this experiment, we vary $\mathcal{T}$ in $\{10, 20, 30, 40, 50, 60\}$, and evaluate the average NDCG@20 and MDG@20 changes for users with different mainstream levels, users with different active levels, items with different mainstream levels, and items with different popularity levels in Figure 10, Figure 11, Figure 12, and Figure 13 respectively. Our results demonstrate that the utility first increases and then decreases and converges along with increasing $\mathcal{T}$ for all five subgroups including both users and items. Subsequently, we choose 40 as the optimal hyper-parameter to effectively mitigate user-side and item-side biases with the balanced trade-off between minority groups and privileged groups.

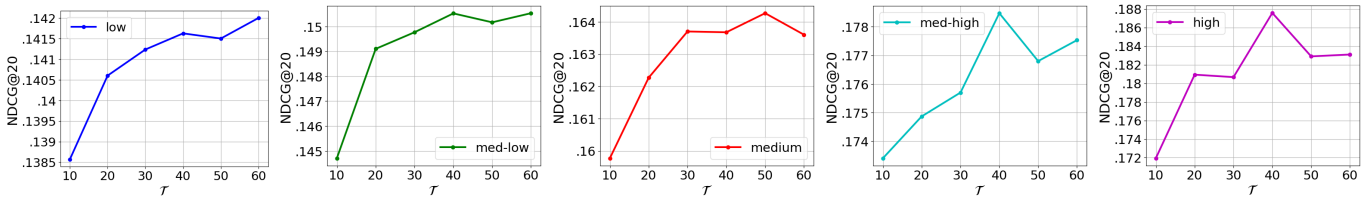
G.3 Number of Negative Sampling Rounds $\mathcal{J}$

Last, we investigate how the number of negative sampling rounds in CFBoost influences the performance of the user-side and item-side. Generally, more rounds of negative sampling result in a more stable and robust estimation of user-item pair losses, which can improve the efficacy of the model. In the experiment, we run CFBoost



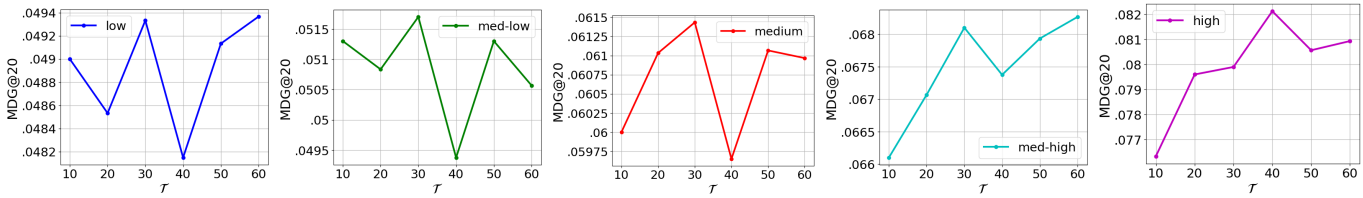
(a) low (b) med-low (c) medium (d) med-high (e) high

Figure 10: Performance of CFBoost on five subgroups of user mainstreamness on CD dataset with varying $\mathcal{T}$.



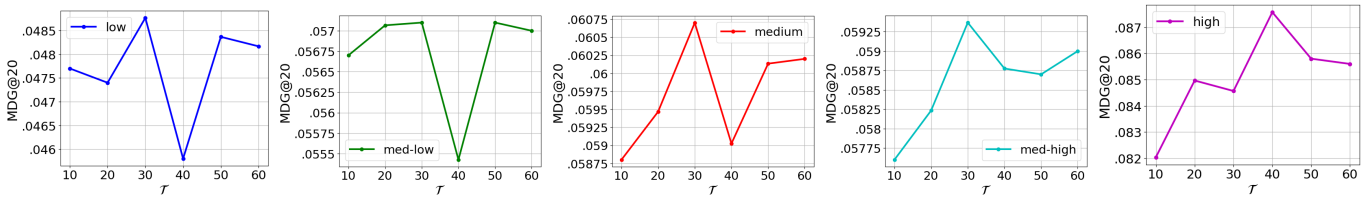
(a) low (b) med-low (c) medium (d) med-high (e) high

Figure 11: Performance of CFBoost on five subgroups of user activeness on CD dataset with varying $\mathcal{T}$.



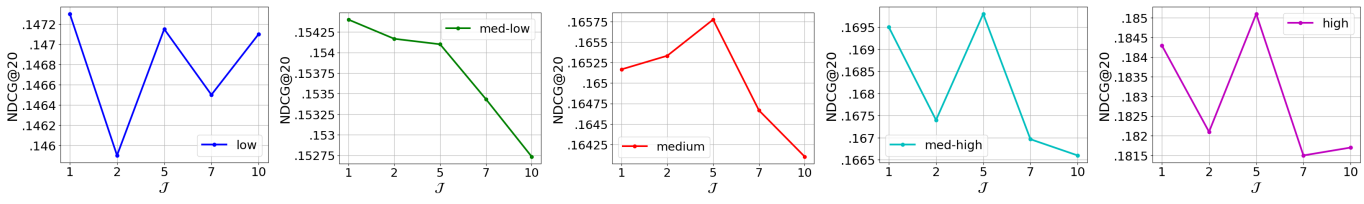
(a) low (b) med-low (c) medium (d) med-high (e) high

Figure 12: Performance of CFBoost on five subgroups of item mainstreamness on CD dataset with varying $\mathcal{T}$.



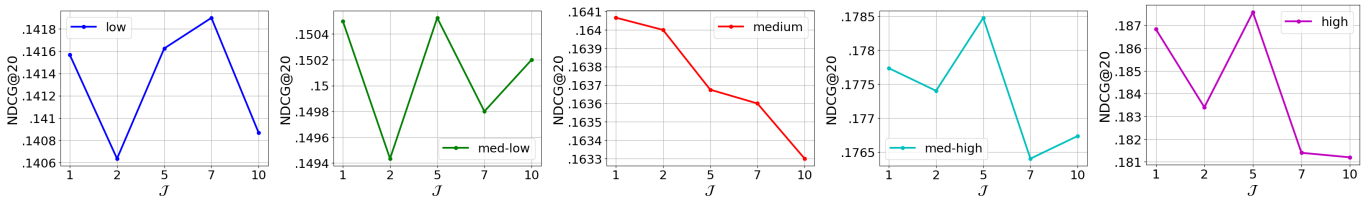
(a) low (b) med-low (c) medium (d) med-high (e) high

Figure 13: Performance of CFBoost on five subgroups of item popularity on CD dataset with varying $\mathcal{T}$.



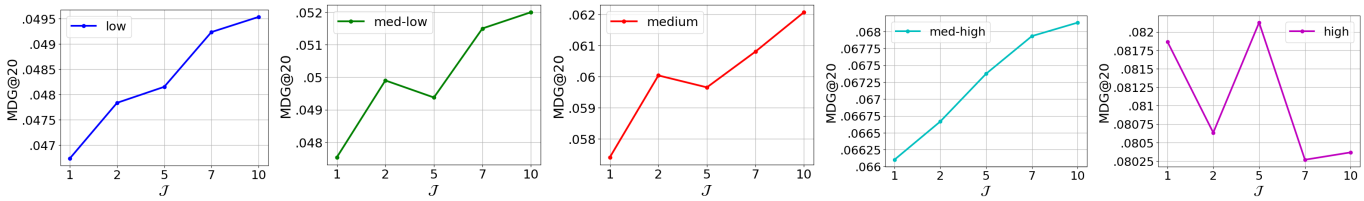
(a) low (b) med-low (c) medium (d) med-high (e) high

Figure 14: Performance of CFBoost on five subgroups of user mainstreamness on CD dataset with varying $\mathcal{J}$.



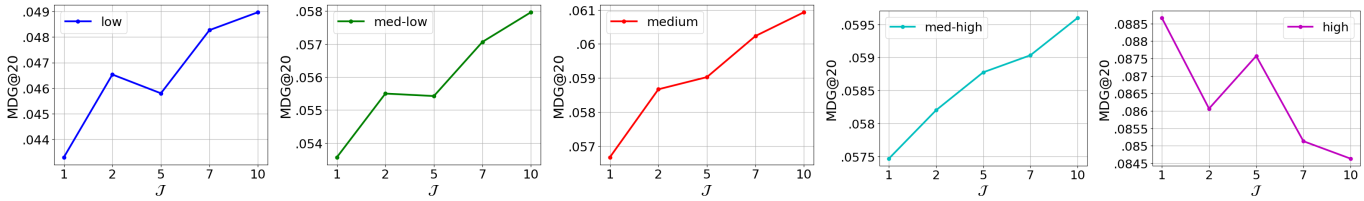
(a) low (b) med-low (c) medium (d) med-high (e) high

Figure 15: Performance of CFBoost on five subgroups of user activeness on CD dataset with varying $\mathcal{J}$.



(a) low (b) med-low (c) medium (d) med-high (e) high

Figure 16: Performance of CFBoost on five subgroups of item mainstreamness on CD dataset with varying $\mathcal{J}$.



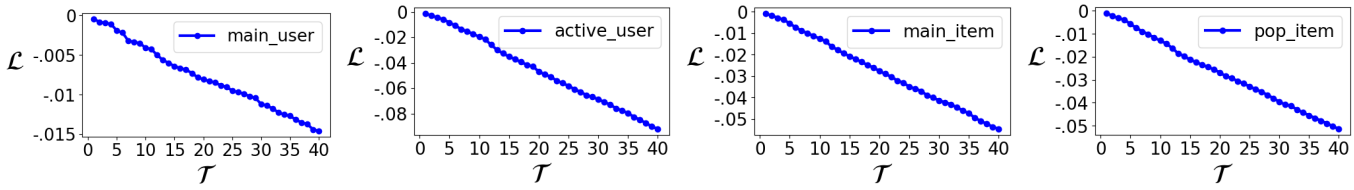
(a) low (b) med-low (c) medium (d) med-high (e) high

Figure 17: Performance of CFBoost on five subgroups of item popularity on CD dataset with varying $\mathcal{J}$.

on the CDs&Vinyl dataset with the number of negative sampling rounds (denoted as $\mathcal{J}$) varying in {1, 2, 5, 7, 10} and other settings the same as in Section 5.2. Then, we evaluate average NDCG@20 and MDG@20 changes for users with different mainstream levels, users with different active levels, items with different mainstream levels, and items with different popularity levels in Figure 14, Figure 15, Figure 16, and Figure 17 respectively. Our findings reveal that as $\mathcal{J}$ increases, the overall performance of most subgroups, particularly on most of the user-side, initially rises and then declines as anticipated. Notably, subgroups of items, excluding the most mainstream and popular items, exhibit greater sensitivity, showcasing improved recommendation utility with an increase in $\mathcal{J}$. This is possible that increased $\mathcal{J}$ may be particularly crucial for enhancing item-side performance, given the abundance of implicit feedback for overall items compared to all users. Consequently, all item groups excluding the 'high' subgroup derive benefits from the increased $\mathcal{J}$, leading to an improvement in their recommendation utility. Accordingly, we identify 5 as the optimal hyper-parameter, showcasing optimal performances for both user-side and item-side with minimal trade-offs.

H Decreasing of Training Loss

In this experiment, we want to show that as the boosting iteration increases, our proposed CFBoost can effectively reduce the training loss for all user-item pairs no matter if they belong to privileged groups or minority groups. We have already demonstrated the decrease of training loss for each of the minority groups in Section 5.5. Then we run experiments and record the $\mathcal{L}_{u,i}$ by choosing privileged groups. Likewise, we also observe a decreasing trend of the $\mathcal{L}_{u,i}$ for privileged groups with increasing boosting iterations ($\mathcal{T}$). Figure 18a and Figure 3a depict the decline in $\mathcal{L}_{u,i}$ for both selected mainstream and niche users, demonstrating the model's effectiveness in mitigating user mainstream bias by progressively reducing training loss of niche user groups and mainstream user groups. This pattern extends to other biases, such as user activeness bias (refers to Figure 18b and Figure 3b), item mainstream bias (refers to Figure 18c and Figure 3c), and item popularity bias (refers to Figure 3d and Figure 18d), where training loss decrease for both minority users/items and privileged users/items. Once again, this aligns with our theorem, reinforcing our primary motivation for debiasing: to mitigate both user-side and item-side biases by reducing training loss in each user-item pair (no matter if they are minority groups or privileged groups).



(a) Mainstream user. (b) Active user. (c) Mainstream item. (d) Popular item.

Figure 18: Training loss $\mathcal{L}$ of privileged groups over $\mathcal{T}$.

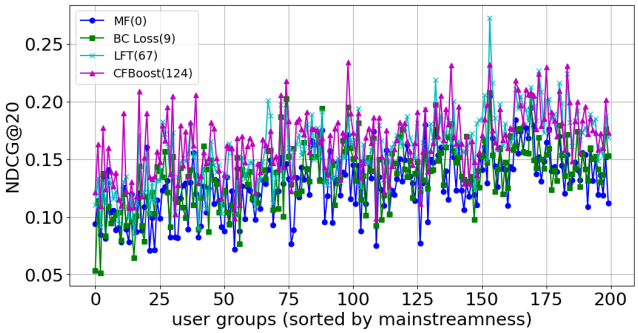


Figure 19: NDCG@20 for four models of user mainstream subgroups.

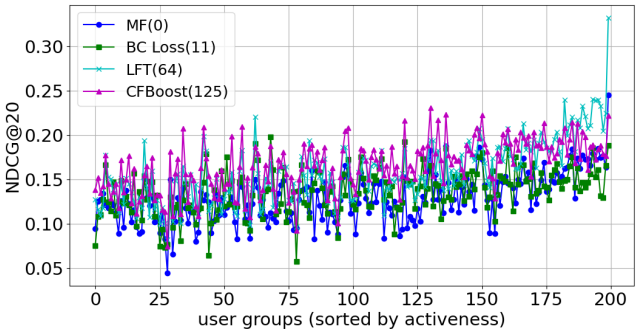


Figure 20: NDCG@20 for four models of user activeness subgroups.

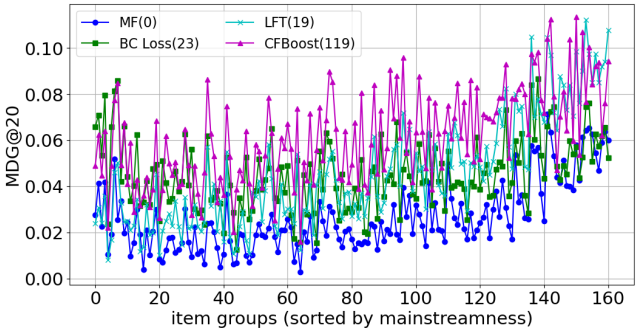


Figure 21: MDG@20 for four models of item mainstream subgroups.

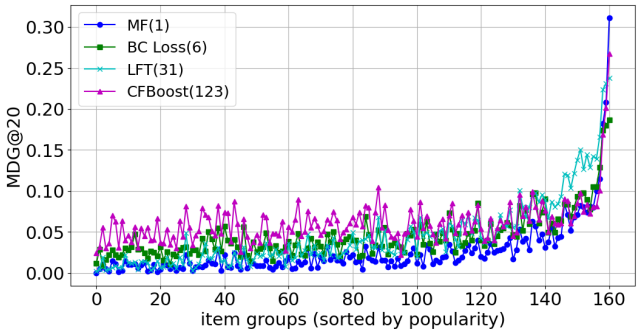


Figure 22: MDG@20 for four models of item popularity subgroups.

I Fine-Grained Bias Visualization

Finally, we conduct empirical evidence confirming and supporting the observations from Section 5.6. In Figure 19, Figure 20, Figure 21, and Figure 22, we divide sorted users in the CD dataset (based on mainstream levels and activeness levels) evenly into 200 users groups, containing about 60 users in each subgroup, and split sorted items (based on mainstream scores and popularity scores) into 161 item groups, containing 50 items in each subgroup, with the same plot settings as Section 5.6. Similarly, when we compare different methods with this fine-grained bias result, the same trend exhibits as Section 5.6: when users or items are more mainstream, active, or popular, recommendation utility significantly increases; and CFBoost is always above other curves with superior debiasing performance. We also mark how many wins (best NDCG@20 for users or MDG@20 for items) among each subgroup for each model. Specifically, CFBoost obtains 124 wins for user mainstream subgroups, 125 wins for user activeness subgroups, 119 wins for item mainstream subgroups, and 123 wins for item popularity subgroups. In sum, across heterogeneous bias levels, CFBoost consistently outperforms other baselines and demonstrates the efficacy of addressing multiple biases simultaneously.