Fairness Through Domain Awareness: Mitigating Popularity Bias for Music Discovery

Rebecca Salganik[1(✉)], Fernando Diaz[3], and Golnoosh Farnadi[1,2]

[1] Université de Montreal/MILA, Montreal, QC, Canada
rebecca.salganik@umontreal.ca, farnadig@mila.quebec
[2] McGill University, Montreal, QC, Canada
[3] Carnegie Mellon University, Pittsburgh, PA, USA
diazf@acm.org

Abstract. As online music platforms continue to grow, music recommender systems play a vital role in helping users navigate and discover content within their vast musical databases. At odds with this larger goal, is the presence of popularity bias, which causes algorithmic systems to favor mainstream content over, potentially more relevant, but niche items. In this work we explore the intrinsic relationship between music discovery and popularity bias through the lens of individual fairness. We propose a domain-aware, individual fairness-based approach which addresses popularity bias in graph neural network based recommender systems. Our approach uses individual fairness to reflect a ground truth listening experience, i.e., if two songs sound similar, this similarity should be reflected in their representations. In doing so, we facilitate meaningful music discovery that is resistant to popularity bias and grounded in the music domain. We apply our BOOST methodology to two discovery based tasks, performing recommendations at both the playlist level and user level. Then, we ground our evaluation in the cold start setting, showing that our approach outperforms existing fairness benchmarks in both performance and recommendation of lesser-known content. Finally, our analysis makes the case for the importance of domain-awareness when mitigating popularity bias in music recommendation.

Keywords: Recommendation · Algorithmic Fairness · Graph Neural Networks

1 Introduction

The proliferation of market activity on digital platforms has acted as a catalyst for research in recommendation, search, and information retrieval [33]. At its core, the goal of this research is to design systems which can facilitate users' exploration of an extensive item catalogue: be it in the domain of journalism [60], films [29], fashion [20], music [38,52,53], or otherwise. Within this larger goal of recommendation, each domain comes with its own specifics that differentiate it from other settings [9,25,46]. Particular to the music streaming

N. Goharian et al. (Eds.): ECIR 2024, LNCS 14611, pp. 351–368, 2024.
https://doi.org/10.1007/978-3-031-56066-8_27

domain, an extensive body of work has explored the importance of discovery, exploration, and novelty in the larger goal of performing music recommendation [17,23,27,39,44,49]. Broadly, discovery can be considered the ability of a curatorial system to expose users to relevant content that *they would not have manually discovered themselves* [27,32,49]. And, most significantly, the importance of this subtask is substantiated by numerous works indicating that music discovery is a crucial factor for maintaining and improving customer loyalty [39,44,49].

Recent work in this domain has begun to uncover an inverse relationship between novelty, one of the keys to discovery, and the notion of popularity bias [36,60]. Within the broader recommendation community, popularity bias has long been an important topic of research. This phenomenon manifests itself when algorithmic reliance on pre-existing data causes new, or less well known items, to be disregarded in favor of previously popular items [3,12,16,35,47,57]. And, particularly in the context of discovery, where purpose of a user's engagement with algorithmic curation hinges on exposure to musical items which they would not have already been familiar with, the presence of popularity bias can clearly hinder a system's ability to serve this need. In this work, we apply our methodology to graph neural network (GNN) based recommender systems [26,61]. In the graph space, popularity is deeply interlaced with the degree of a node, or the number of edges that connect a node and to others in the graph. This is because the innate process of representation learning is affected by the number of neighbors a node has [37]. And, thus, a node's centrality can dictate the quality of its learned representation. This suggests that duplicating the feature information of an extremely popular song, creating a new song using these duplicate features, and randomly placing it once at the edge of a graph, will significantly impact its learned representation, even if everything about the song remains *exactly the same*. Currently, the state of the art approaches to mitigating popularity bias, do so from a domain agnostic approach [2,42,51,59,64]. This methodology has two important drawbacks. First, it often requires a method to rely on the presence of sensitive attributes in order to define popularity, which are often unavailable. Second, such an approach is unable to recognize musical similarities among items, thus increasing the complexity of disentangling popularity bias in learned representation.

In this work, we propose a **domain aware**, individual fairness based approach for facilitating engaging music discovery. Unlike domain-agnostic approaches, our method does not rely on sensitive attributes to define popularity. Instead, we design an intuitive, simple framework that uses music features to fine-tune item representations such that they are reflective of information that is, in essence, a ground truth to the listening experience: two songs that sound similar should, at least somewhat, reflect this similarity in their learned representations. In order to facilitate the domain awareness of our approach we generate nuanced multi-modal track features, extensively augmenting two publicly available datasets. Using these novel feature sets, we show the importance of integrating musical similarity into a debiasing technique and show the effects of our method at learning expressive representations of items that are robust to the effects of popularity bias in the graph setting. Grounding our approach in the musical domain empowers us to leverage a ranking-based individual fairness

definition and extend it to the bipartite graph setting. We compare our individual fairness-based method with three other methods which are grounded in other canonical fairness notions and are not domain-aware. Through a series of empirical results, we show that our fairness framework enables us to outperform a series of accepted fairness benchmarks in both performance and recommendation of lesser known content on two important music recommendation tasks. In summary, the contributions of this paper are the following:

1. **Problem Setting**: we define the task of music discovery through the lens of domain-aware individual fairness, showing the intrinsic connections between individual fairness, musical similarity, popularity bias, and music discovery.
2. **Dataset Design**: we extensively augment two classic music recommendation datasets to generate a set of nuanced multi-modal track features.
3. **Method**: (1) we provide a novel technical formulation of popularity bias (2) design a domain-aware ranking based individual fairness approach to mitigating popularity bias in graph-based recommendation.[1]

2 Related Work

2.1 Popularity Bias in Recommendation

Most broadly, popularity bias refers to a disparity between the treatment of popular and unpopular items at the hands of a recommender system. As such, this term is loosely tied to a collection of complementary terms including exposure bias [19], superstar economics [5], long tail recommendation [42], the Matthew effect [45], and aggregate diversity [4,13]. There have been several different approaches to formulating popularity through some quantitative definition. One body of work defines popularity with respect to individual items' visibility [19,42,65]. Another group of approaches attempts to simplify this process by applying some form of binning to the raw appearance values. Most notably, the long tail model [12,22,28,47,62] has risen to prominence as a popularity definition. Due to the exponential decay in item interactions, the first 20% of items, called *short head*, take up a vast majority of the user interactions and the remaining 80%, or *long tail* and *distant tail*, have, even in aggregate, significantly fewer interactions.

In addition to providing formal definitions, a large body of work has formed around analyzing and mitigating popularity bias in recommender systems [10,16,34,35]. These mitigation strategies are often based on the instrumentation of various canonical fairness notions such as **group fairness** [2,7,51,55,64], **counterfactual fairness** [59,65,67], or **individual fairness** [14,58]. We contrast our work with previous individual fairness approaches in our use of the music feature space as a form of domain expertise in definition of item-item similarity. We argue that without this "anchoring" an individual fairness method that uses the output of a recommender model, whether it be in learned representation [58] or the relevance score [14], is already influenced by an item's

[1] Github repository https://github.com/Rsalganik1123/Domain_Aware_ECIR2024.

popularity. Finally, in addition to the classical formulation of popularity bias, a group of works have explored the connection between popularity bias and novelty [41,66,68] where various metrics are designed to evaluate the novelty of a recommended list. We see our work as complementary to the exploration in this area however, we differentiate our problem formulation because while novelty is an important aspect of discovery, without domain awareness novelty alone does not account for musical similarity - a critical aspect of the discovery setting.

2.2 GNNs in Recommendation

In recent years various graph neural network (GNN) architectures have been proposed for the recommendation domain [61]. For brevity, we will focus only on the two methods that are used as the backbone recommenders to the fairness mitigation techniques discussed later in this paper and refer readers to the following surveys [26,61] for recent innovations in this domain. In particular, *PinSage* [63] is an industry solution to graph-based recommendation. PinSage trains on a randomly sampled subset of the graph. In order to construct neighborhoods, PinSage uses k random walks to select the top m most relevant neighbors to use as the neighbor set. However, it is important to note that PinSage learns representations of items but not users. Meanwhile, *LightGCN* (LGCN) [31], is a method that learns both user and item embeddings simultaneously. Since its proposal in 2020, it is still considered state of the art.

3 Methodology

In this section we detail the dataset augmentation procedure and architecture of our domain-aware, individually fair music recommendation system. First, we introduce our datasets in Sect. 3.1. Then, following the problem setting in Sect. 3.2, we reformulate popularity bias in Sect. 3.3 and introduce our domain-aware, individually fair music recommender system in Sect. 3.4.

3.1 Dataset Augmentation Procedure

We augment both of our datasets to include a rich set of features scraped from Spotify API [1]. The details of the augmented features are as follows.

1. **Sonic features.** Spotify has a series of 9 proprietary features which are used to define the audio elements associated with a track. They are *danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, and tempo*. Each of these features is a continuous scalar value. We apply 10 leveled binning to the values.
2. **Genre features.** We identify the primary artist associated with each collect all the genre tags associated with them.
3. **Track Name features.** For each song in the dataset, we extract the song title and pass it through a pre-trained language transformer model, *BERT* [18], into an embedding of dimension 512.

4. **Image features.** For each song in the dataset we extract the associated album artwork. We use this image to generate *ResNet50* [30] embeddings of dimension 1024.

3.2 Problem Setting

The task of performing recommendation can be seen as link prediction an undirected bipartite graph. We denote such undirected bipartite graph as $G = (V, E)$. The set $V = T \cup P$ consists of a set containing song (or track) nodes, T, and playlist (or user) nodes, P (or U). The edge set E are defined between a playlist p_k (or user u_k) and a song t_i if t_i is contained in p_k (or listened to by u_k). Following this setting, our goal (link prediction) is to predict whether any two song nodes $t_i, t_j \in T$ share a common parent playlist p.

3.3 Reformulating Popularity Bias

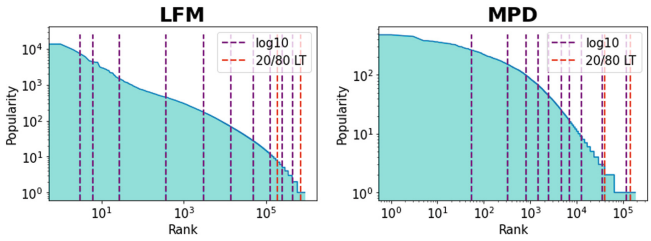

Fig. 1. Binning procedure for popularity definition. We contrast our popularity definition with the classic long tail model [42], showing that our proposed method empowers for a granular visualization of popularity between various item groups.

Defining Popularity. As mentioned in Sect. 2.1, there is no true consensus within the community on how to define popularity. Here, we present a methodology which we believe allows for both the granularity and expressiveness necessary to highlight differences among various mitigation methods. Broadly, our method consists of important steps (1) logarithmic smoothing and (2) binning. In doing so, we combine the best of each methodology. Applying a logarithmic transformation to the raw values, solves the scaling issues that are caused by the extremes of the long-tail distribution. Meanwhile, binning concisely highlights large scale patterns. And, in contrast to previous methods using binning [2,19,51], logarithmic smoothing guarantees that none of the bins are left empty. Please note that we select 10 bins based on the distribution of the datasets and the formulation of our BOOST methodology (see Sect. 3.4) but this number can be tuned to the granularity needs.

Our popularity measure is achieved by first, counting the number of times each song track, t_i appears within playlist (or user) training interactions such that for each t_i, $a_{t_i} = |\{p_i : t_j \in p_i\}|$. Then, we apply the base 10 logarithmic smoothing to these values such that for each t_i, $\text{pop}_{t_i} = \log_{10}(a_{t_i})$. Finally, we apply binning

onto these values to split them into 10 groups such that for each t_i, pop_bin $(t_i) \in \{0, \ldots, 9\}$ where bin 9 has a higher popularity value than bin 0. The visualization of this binning procedure and its comparison with the long tail method can be seen in Fig. 1. As demonstrated by our visualizations, transforming the raw values into the logarithmic space shows that the bins are filled in relatively even intervals, where, as the popularity increases, so does the number of songs included in a bin. We showcase the gains that our method has over the canonical long tail model in Fig. 1 where we compare the positioning of our binning methodology with the classic long tail model. Furthermore, as we later show Figs. 3 and 4 our formulation of popularity is able to elucidate crucial differences among both the datasets and baseline model performances on these datasets.

Popularity Bias and Music Discovery. In addition we formalize the inverse relationship between music discovery and popularity bias. For each song track, $t_i \in T$, we generate a counterfactual example song, $t_i^* \in T_{CF}$, where everything about the features is *exactly the same* and the only difference is that t_i has a high degree while t_i^* has a degree of one. We calculate the distance between an original song node, t_i and its counterfactual duplicate, t_i^*. A system with high potential for musical discovery will have a low distance between the songs, showing a low popularity bias and an understanding of musical similarity. We will return to this formulation in Sect. 5.1, showing that a node's placement and degree in the graph can exacerbate the presence of popularity bias, reflecting itself in the node's learned representation.

3.4 Mitigating Popularity Bias Through Individual Fairness

Ranking-Based Individual Fairness. *REDRESS* is an individual fairness framework proposed by Dong et al. [21] for learning fair representations in single node graphs. Our work extends this framework to the bipartite recommendation setting and integrates it into our popularity bias mitigation approach. Here, the crucial formulation of individual fairness requires that nodes which were similar in their initial feature space should remain similar in their learned representation embeddings [24]. More concretely, for each song node, t_i, and node pair t_u, t_v in a graph G, similarity is defined on the basis of the pairwise cosine similarity metric, $s(\cdot, \cdot)$, as applied to either a feature $X[v] \in \mathfrak{R}^d$, or learned embedding set, $Z[v] \in \mathfrak{R}^m$. Applying this procedure in a pairwise fashion produces two similarity matrices. The first, or *apriori similarity*, S_G, in which similarity is calculated on input features and the second, or *learned similarity*, S_Z, in which similarity is calculated between learned embeddings generated by some GNN model, M. Drawing on principles from learn to rank [8], each entry in these similarity matrices is re-cast as the probability that node t_i is more similar to node t_u than t_v and transformed into an *apriori probability tensor*, $P_G \in \mathfrak{R}^{|T|\times|T|\times|T|}$, and a *learned probability tensor*, $P_Z \in \mathfrak{R}^{|T|\times|T|\times|T|}$. For more details on the calculations of these probabilities, please see the original formulation in [21]. Having defined these two probability

tensors, each individual node the fairness loss, $L_{t_u,t_v}(t_i)$, is the canonical cross entropy loss aggregated over all nodes $t_i \in V$ as:

$$L_{\text{fairness}} = \sum_{i}^{|T|} \sum_{u}^{|T|} \sum_{v}^{|T|} L_{t_u,t_v}(t_i) \tag{1}$$

Individually Fair Music Discovery. The original formulation of individual fairness requires some form of domain expertise [24] to determine how similar (or dis-similar) two items are. For the music discovery domain, we use music features as the basis for calculating cosine similarity. Thus, our *apriori similarity*, S_G, is defined as the cosine similarity between the musical features, $X[v] \in \mathfrak{R}^{|T|\times 9}$, associated with song nodes. Meanwhile, our *learned similarity* contains the song-level embeddings, $S_Z \in \mathfrak{R}^{|T|\times m}$, learned by PinSage. In this way, REDRESS acts as a regularizer that ensures that rank-based similarity between songs is preserved between the input and embedding space. Thus, our similarity notion is domain-aware and grounded in the essence of musical experiences: acoustics.

Bringing Popularity Into Individual Fairness. The *REDRESS* framework does not explicitly encode any attributes of popularity in its training regimen. To extend this technique for explicitly counteracting popularity bias, we define the BOOST technique which is used to further increase the penalty on misrepresentation of items that come from diverse popularity categories. We define 10 popularity bins by applying a logarithmic transformation and binning the degrees of a node i (i.e., deg_i) such that $\text{pop\_bin}(i) = \text{bin}\left(\log_{10}\left(deg_i\right)\right)$. Given the learned representation matrix, $S_Z \in \mathfrak{R}^{|T|\times|T|}$, we define another matrix B in which

$$B_{ij} = |\text{pop\_bin}(i) - \text{pop\_bin}(j)| \tag{2}$$

Then, in the *BOOST* loss formulation, in place of S_Z we use $S_Z' = S_Z + B$.

Objective Function. The training objective is:

$$L_{\text{total}} = L_{\text{utility}} + \gamma L_{\text{fairness}} \tag{3}$$

where γ is a scalable hyperparameter which controls the focus given to fairness used to balance between utility (L_{utility}) and fairness (L_{fairness}). For L_{utility}, we apply the aforementioned focal loss [40]. And L_{fairness} is Eq. 1 defined above.

Generating Recommendations. Notably, the PinSage architecture is only designed to learn embeddings for songs, not for playlists (or users). Thus we design our own procedure for generating playlist (or user) embeddings using the learned song embeddings. For each playlist (or user) node, p_i, we have a set of songs, $T(p_i) = \{t_i \in T, e_{p_i,t_i} \in E\}$, which are contained in a playlist. For a test playlist, p_t, we split the associated track set into two groups: $T(t_p) = \{t_i : t_i \in u_i\} =$

$t_{seed} \cup t_{holdout}$ such that t_{seed} is a set of k songs that are used to generate the playlist representation and $t_{holdout}$ are masked for evaluation. Thus, in order to generate a playlist (or user) embedding we define:

$$z_{p_t} = MEAN(\{z_{t_j} : t_j \in t_{seed}\})$$

where the $z_k \in \Re^{1\times d}$ are the learned representations of dimension d. Having learned these playlist representations, we find the k-nearest neighbouring tracks and consider these the recommendations (Table 1).

4 Experimental Settings

Table 1. Dataset statistics

Dataset	Recommendation Setting	Feedback Type	#Users/Playlists	#Songs	#Artists
MPD	*Automatic Playlist Continuation*	Explicit	11,100	183,408	37,509
LFM	*Weekly Discovery*	Implicit	10,267	890,568	100,638

Recommendation Scenario. We evaluate our method on two important music discovery tasks: *Automatic Playlist Continuation* [54] and *Weekly Discovery* [56]. *Automatic Playlist Continuation* requires the recommender system to perform *next k recommendation* on a user generated playlist. Meanwhile, *Weekly Discovery*, involves the creation of a new playlist based on a user's aggregated listening habits. Following the paradigm of the cold start setting [54], we extract splits on the playlist level by randomly sampling without replacement such that each split trains on a distinct subset of the playlist pool.

Datasets. As introduced in Sect. 3.1, we extensively augment two publicly available datasets, LastFM (LFM) [43] and the Million Playlist Dataset (MPD) [15], with rich multi-modal track-level feature sets. For more details please refer back to Sect. 3.1.

Music Performance Metrics. We design a series of musical evaluation metrics to complement classic utility measures (see Table 2 for further detail). For example, we use *Artist Recall* to evaluate the correct identification of artists in a recommendation pool, an auxiliary task in music recommendation [38]. In addition, we design *Flow* to capture the musical cohesiveness of the recommended songs in a playlist [6].

Fairness Metrics. In order to assess the debiasing techniques used to promote of long tail songs, we follow common evaluation practices in the fairness literature [2,12,48]. Percentage metrics capture the ratio of niche to popular content that is being recommended on a playlist (or user) level. Meanwhile, *coverage* looks at the aggregate sets of niche songs and artists over all recommendations (see 2 for further detail) (Table 2).

Table 2. Music and fairness performance metrics. We define a ground truth set, G, and a recommended set, R, we define the set of unique artists in a playlist as $A(.)$ and the d-dimensional musical feature matrix associated with the tracks of a playlist as $F(.) \in \mathfrak{R}^{|.| \times d}$.

Metric	Category	Formulation
Artist Recall@100	Music	$\frac{1}{\lvert P_{test}\rvert} \sum_{p \in P_{test}} \frac{1}{\lvert A(G_p)\rvert} \lvert A(G_p) \cap A(R_p)\rvert$
Flow@100	Music	$\frac{1}{\lvert P_{test}\rvert} \sum_{p \in P_{test}} \cos(F(t_i), F(t_j)) \; \forall (t_i, t_j) \in R_p$
Artist Diversity (per playlist)	Fairness	$\frac{1}{\lvert P_{test}\rvert} \sum_{p \in P_{test}} \frac{1}{\lvert A(P)\rvert} \lvert \{A(R_p)\}\rvert$
Percentage of Long Tail Items	Fairness	$\frac{1}{\lvert P_{test}\rvert} \sum_{p \in P_{test}} \frac{1}{\lvert p\rvert} \lvert \{t_i : t_i \in R_p \cap t_i \in LT\}\rvert$
Coverage over Long Tail Items	Fairness	$\frac{1}{\lvert LT\rvert} \lvert \{t_i : t_i \in R \bigcap t_i \in \lvert LT\rvert\}$
Coverage over Artists	Fairness	$\frac{1}{\lvert A\rvert} \lvert \{arid(t_i) : t_i \in R\rvert\}$

Baselines. We use two naive baselines, first using bare features in place of learned representations (**Features**) and, second, recommending the top 100 most popular tracks (**MostPop**). Then, we evaluate against three state of the art bias mitigation techniques: a group fairness-based, in processing method (**ZeroSum** [51]), a causality-based in-processing method (**MACR** [59]), and a re-ranking, post-processing method (**xQuAD** [2]).

Parameter Settings and Reproducibility. Each of the baseline methods was tested with learning rates ~ $(0.01, 0.0001)$, embedding sizes of $[10, 24, 64, 128]$ and batch sizes of $[256, 512, 1024]$. For the values in the tables below, each stochastic method was run 5 times and averaged. All details and further hyperparameter settings can be found on our Github repository.

5 Results

5.1 RQ1: How Does Incorporating Individual Fairness Improve the Mitigation of Popularity Bias and Facilitate Music Discovery?

To showcase the performance of our algorithm in the discovery setting and motivate the need for individual fairness in the mitigation of popularity bias, we draw on the definition of music discovery presented in Sect. 3.3 and evaluate the effects of popularity bias on learned representations of songs. Simulating a situation of maximal popularity bias, we consider the hypothetical example in which extremely popular songs are reversed to become unpopular and measure the effect on their learned representations. More formally, for each song track, $t_i \in T$, we generate a counterfactual example song, $t_i^* \in T_{CF}$, where everything about the features is *exactly the same* and the only difference is that t_i appears in many playlists while t_i^* appears only once. We augment the original dataset to include these counterfactual songs, $T_{AUG} = T \cup T_{CF}$. Then, we use five methods to learn the item level representations: one baseline recommender, PinSage, and four bias mitigation methods,

ZeroSum [51], MACR [59], REDRESS, and BOOST. We apply 2-dimensional PCA to each embedding set and analyze the Euclidean distance between the centroids of original track embeddings, $\bar{T}$, and counterfactual track embeddings, $\bar{T}_{CF}$. Due to the size of our dataset, we run this metric using the 100 most popular tracks in the MPD dataset and leave further exploration of this phenomenon for future work.

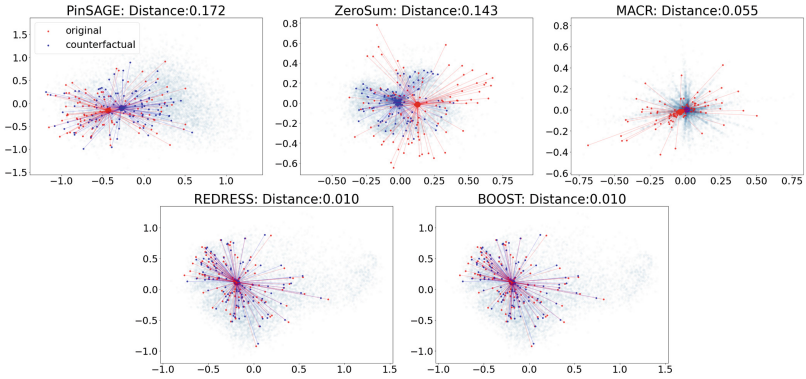

Fig. 2. Simulating Popularity Bias: We select 100 of the most popular songs in MPD and [54], duplicate features, and give them a degree of 1. We find that REDRESS and BOOST have the lowest distance between the originals and their unpopular duplicates, showing the least amount of popularity bias.

As shown in Fig. 2, we find that all fairness interventions decrease the distance between the two centroids. Furthermore, as the granularity of fairness increases, the distance between the centroids of learned representations decreases. For example, PinSage, which has no mitigation of popularity bias, has the largest distance of 0.172. ZeroSum [51], which considers group fairness, decreases the distance to 0.143, MACR [59], which uses counterfactual estimation, shrinks to 0.055. Finally, our methods, REDRESS and BOOST are able to achieve both the lowest distance and the correct orientation between the two embedding spaces. In these results, we see that the domain-awareness of our methodology, which enables it to understand musical similarity between items, allows it to be resistant to the effects of popularity bias on a learned song embedding. Thus, in the setting of musical discovery, it is able to uncover proximity between items which are musically coherent even if they are not necessarily of similar popularity status. And, in doing so, we build representations that are complex, expressive, and effective for music recommendation.

5.2 RQ2: How Does Our Individual Fairness Approach Compare to Existing Methods for Mitigating Popularity Bias?

Table 3. Comparison between all methods. Note: We use **bold** text to represent the best performance within a column. In addition, we calculate statistical significance using the Wilcoxon signed-rank test [50] to results between PinSage and BOOST. We show that the *BOOST* method achieves the best performance along all fairness metrics when compared with debiasing benchmarks.

		Classic		Music		Fairness			
Data	Model	Recall@100	NDCG@100	Artist Recall@100	Flow	Diversity	%LT	LT Cvg	Artist Cvg
MPD	Features	0.041	0.073	0.073	0.900	0.841	**0.588**	0.022	0.073
	MostPop	0.044	0.048	0.141	0.908	0.680	0.0	0.0	0.001
	LightGCN	**0.106 ± 0.004**	0.119 ± 0.004	**0.272 ± 0.011**	0.905 ± 0.000	0.672 ± 0.025	0.002 ± 0.000	0.000 ± 0.000	0.025 ± 0.001
	PinSage	0.068 ± 0.002	**0.144 ± 0.003**	0.139 ± 0.003	0.931 ± 0.001	0.707 ± 0.003	0.476 ± 0.002	0.032 ± 0.000	0.105 ± 0.000
	ZeroSum	0.044 ± 0.002	0.043 ± 0.002	0.220 ± 0.008	0.904 ± 0.001	0.765 ± 0.013	0.000 ± 0.003	0.000 ± 0.000	0.048 ± 0.002
	xQuAD	0.064 ± 0.005	0.104 ± 0.006	0.135 ± 0.013	0.927 ± 0.004	0.703 ± 0.059	0.226 ± 0.001	0.017 ± 0.000	0.098 ± 0.004
	MACR	0.028 ± 0.014	0.030 ± 0.015	0.149 ± 0.022	0.902 ± 0.002	0.831 ± 0.034	0.019 ± 0.006	0.000 ± 0.001	0.011 ± 0.003
	REDRESS	0.045 ± 0.002	0.100 ± 0.003	0.162 ± 0.004	0.969 ± 0.032	0.829 ± 0.001	0.504 ± 0.003	0.036 ± 0.004	0.117 ± 0.000
	BOOST	0.020 ± 0.004	0.047 ± 0.003	0.137 ± 0.002	**0.979 ± 0.000**	**0.899 ± 0.002**	**0.522 ± 0.001**	**0.037 ±0.003**	**0.125 ±0.000**
	p values	4.408083e-16	1.768725e-19	0.727897	3.751961e-61	1.168816e-29	0.000596	-	-
LFM	Features	0.033	0.037	0.041	0.996	0.919	0.486	0.005	0.034
	MostPop	0.015	0.011	0.046	0.926	0.600	0.000	0.000	0.001
	LightGCN	0.026 ± 0.001	0.023 ± 0.001	0.068 ± 0.001	0.998± 0.000	0.505 ± 0.012	0.000 ± 0.000	0.000 ± 0.000	0.003 ± 0.001
	PinSage	**0.064 ± 0.001**	**0.095 ± 0.002**	**0.077 ± 0.002**	0.969 ± 0.000	0.775 ± 0.003	0.437 ± 0.001	0.008 ± 0.000	0.053 ± 0.001
	ZeroSum	0.001 ± 0.003	0.001 ± 0.001	0.045 ± 0.004	0.996 ± 0.008	0.866 ± 0.000	0.007 ± 0.000	0.000 ± 0.000	0.032 ± 0.001
	xQuAD	0.055 ± 0.001	0.064 ± 0.001	0.068 ± 0.002	0.998 ± 0.000	0.801 ± 0.008	0.212 ± 0.000	0.004 ± 0.000	0.053 ± 0.001
	MACR	0.014 ± 0.001	0.014 ± 0.001	0.049 ± 0.007	0.996 ± 0.003	0.777 ± 0.050	0.002 ± 0.004	0.000 ± 0.000	0.001 ± 0.000
	REDRESS	0.038 ± 0.002	0.053 ± 0.004	0.057 ± 0.001	0.998 ± 0.002	0.862 ± 0.004	0.451 ± 0.000	0.008 ± 0.002	0.056 ± 0.000
	BOOST	0.005 ± 0.001	0.007 ± 0.001	0.029 ± 0.002	**0.999 ± 0.000**	**0.941 ± 0.003**	**0.498 ± 0.006**	**0.010 ± 0.000**	**0.068 ± 0.001**
	p values	5.696989e-08	1.179627e-15	1.914129e-07	0.001408	1.112495e-34	2.477700e-11	-	-

Analyzing Utility Performance: First, we look at the comparison between the backbone recommender systems and their debiasing counterparts. Within the greater fairness community it is typical to see a trade-off between recommendation utility and the effectiveness of a debiasing technique [32]. Indeed, in our experiments this trade-off is present. For example, evaluating the columns of *Recall* and *NDCG* on Table 3 we can see that both recommender systems outperform their debiasing counterparts. However, we argue that the presence of this trade-off in the discovery setting is not only expected but also, potentially desirable. Since the premise of the canonical recommendation utility metrics is to reward a system that can accurately recover the exact tracks a user liked, any attempts to promote long tail content that wasn't originally listened to is penalized, even if it is well aligned with a user's taste. In recent years, several recommendation works have suggested that this trade-off, though present in offline testing, doesn't necessarily carry over into online testing [11,32]. Even more so, in the discovery setting, where the premise of algorithmic curation is facilitating user interactions with music that isn't already top-of-mind, the drop in performance can be attributed to the systems' purposeful avoidance of previously popular items, in favor of other musically coherent and relevant content.

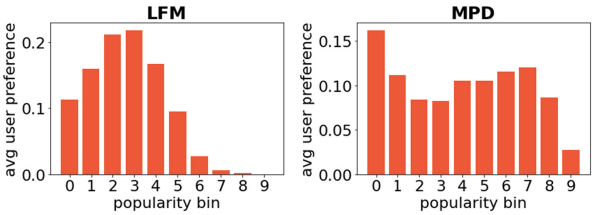

Fig. 3. Dataset Breakdown by Long Tail Definition: Using our formulation of popularity we can see that the two datasets have different distributions of popularity in their training data which, in turn, helps explain fairness/performance tradeoffs.

Analyzing Musical Performance: In order to further analyze the performance of our debiasing method, we look at the in performance on the music metrics, *Artist Recall* and *Flow*. In particular, *Flow* plays an important role in the music discovery task because studies have indicated that users are drawn to homogeneous listening suggestions when engaging with algorithmic curation [6,32]. As we can see in both datasets, REDRESS and BOOST consistently achieve the highest *Flow*. By harnessing musical features and in our debiasing technique, our method generates representations that are indicative of musical similarity. Crucially, if we consider the implications of such a debiasing technique on a mainstream user, these results indicate that our debiasing method's awareness of musical similarity will enable it to maintain the stylistic elements that such a user is drawn to, even if it is promoting niche content.

Analyzing Fairness Performance: Next, we compare the performance among the various fairness promotion methods. Looking at the columns of *Recall* and *NDCG* on Table 3, we can see that, as expected, xQuAD [2] which is a re-ranking method is able to preserve the highest utility. However, among the in-processing

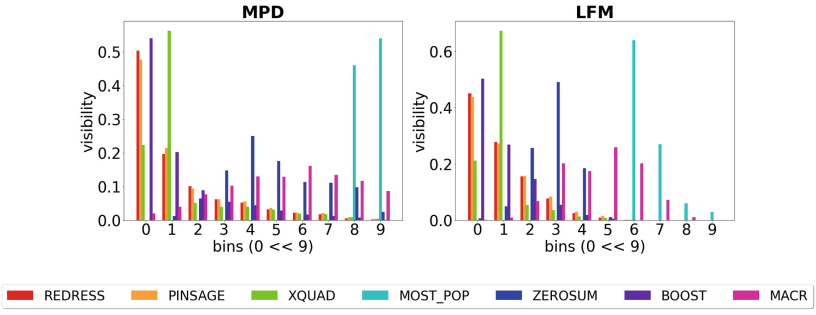

Fig. 4. Group By Group Analysis of Recommendations: We show that REDRESS and BOOST select the largest amount of items from the lowest bins. Note: visibility indicated the number of item from group k appearing in the final recommendations

methods, REDRESS is able to achieve the second highest utility. Meanwhile, looking at the fairness metrics, it is clear that REDRESS and BOOST are the highest performing methods. In particular, looking at the columns for *%LT* and *LT Cvg*, we can see that REDRESS is noticeably better than the other methods and BOOST is able to improve on its performance. Crucially, our method is able to have high values in both coverage and percentage of long tail items indicating that REDRESS/BOOST is not just prioritizing niche items but also choosing a diverse selection from among them.

Effects of Popularity Definition: As we can see in Fig. 4 the definition of popularity plays a significant role in the model selection method *especially* in the case where user preferences encoded in the training data skew towards popular items. In particular, using a less granular definition for popularity bins can synthetically inflate the performance of $\%LT$ and $LTCvg$. For example, we can see that methods like xQuAD and ZeroSum are selecting a majority of their items from bins mid-popularity bins. Using a classical long tail methodology, these differences would not be as visible, masking distinctions among the baselines' fairness.

6 Limitations of Our Work

First, it is important to remember that recommender systems are responsible for serving the tastes of listeners, not policing them, and we do not deny the validity of mainstream listening practices. Thus, the intention of this work is to serve the needs of all users, mainstream and niche equally. However, due to our lack of access to online evaluation settings we cannot confirm that the effects of debiasing will not affect mainstream users' listening experiences. We leave this analysis for future work. Second, due to the nature of publicly available data, both of these datasets skew heavily towards Western, anglophone content and are not representative of the wide array of music that is available for consumption. Finally, we acknowledge that our definition of discovery is grounded in qualitative metrics and cannot encompass the entire complexity of a discovery experience.

7 Conclusion

In this work, we address the problem of mitigating popularity bias in music recommendation. We focus our objective on the task of facilitating meaningful music discovery. In particular, we emphasize the critical aspects of discovery which differentiate it from generalized music recommendation and underscore the negative effects that popularity bias can have on users' ability to uncover novel music. On the basis of this motivation, we unravel the intrinsic ties between popularity bias and individual fairness, proposing a domain-aware debiasing method that uses musical similarity to counteract the effects of popularity on learned representations. Finally, we perform extensive evaluation on two music datasets showing the improvements of our domain aware method in comparison with three state of the

art popularity bias mitigation techniques. While we have designed this method with the explicit focus of music recommendation, we hope that these promising findings can inspire future approaches which are grounded in concrete, domain specific attributes in a wide variety of applications.

Acknowledgements. Funding support for project activities has been partially provided by Canada CIFAR AI Chair, Facebook award, IVADO scholarship, and NSERC Discovery Grants program. We also express our gratitude to Compute Canada for their support in providing facilities for our evaluations.

References

1. Spotipy: Spotify API in Python (2014). https://spotipy.readthedocs.io/en/2.19.0/
2. Abdollahpouri, H., Burke, R., Mobasher, B.: Managing popularity bias in recommender systems with personalized re-ranking (2019)
3. Abdollahpouri, H., Mansoury, M., Burke, R., Mobasher, B.: The connection between popularity bias, calibration, and fairness in recommendation. In: Proceedings of the 14th ACM Conference on Recommender Systems, p. 726–731. RecSys 2020, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3383313.3418487
4. Adomavicius, G., Kwon, Y.: Improving aggregate recommendation diversity using ranking-based techniques. IEEE Trans. Knowl. Data Eng. **24**(5), 896–911 (2012). https://doi.org/10.1109/TKDE.2011.15
5. Bauer, C., Kholodylo, M., Strauss, C.: Music recommender systems challenges and opportunities for non-superstar artists. In: Bled eConference (2017)
6. Bontempelli, T., Chapus, B., Rigaud, F., Morlon, M., Lorant, M., Salha-Galvan, G.: Flow moods: recommending music by moods on deezer. In: RecSys 2022 (2022)
7. Boratto, L., Fenu, G., Marras, M.: Interplay between upsampling and regularization for provider fairness in recommender systems. User Model. User-Adap. Inter. **31**(3), 421–455 (2021). https://doi.org/10.1007/s11257-021-09294-8
8. Burges, C.: From ranknet to lambdarank to lambdamart: an overview. Learning **11**, 23–581 (2010)
9. Burke, R., Ramezani, M.: Matching recommendation technologies and domains (2011). https://doi.org/10.1007/978-0-387-85820-311
10. Cañamares, R., Castells, P.: Should i follow the crowd? a probabilistic analysis of the effectiveness of popularity in recommender systems. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, p. 415–424. SIGIR 2018, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3209978.3210014
11. Castells, P., Moffat, A.: Offline recommender system evaluation: challenges and new directions. AI Magazine **43**(2), 225–238 (2022). https://doi.org/10.1002/aaai.12051, https://onlinelibrary.wiley.com/doi/abs/10.1002/aaai.12051
12. Celma, O., Cano, P.: From hits to niches? or how popular artists can bias music recommendation and discovery. In: Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition. NETFLIX 2008, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1722149.1722154
13. Celma, Ò., Herrera, P.: A new approach to evaluating novel recommendations. In: Proceedings of the ACM Conference on Recommender Systems, RecSys 2008 (2008). https://api.semanticscholar.org/CorpusID:7572506

14. Chakraborty, A., Hannák, A., Biega, A.J., Gummadi, K.P.: Fair sharing for sharing economy platforms (2017)
15. Chen, C.W., Lamere, P., Schedl, M., Zamani, H.: RecSys challenge 2018: automatic music playlist continuation. In: RecSys 2018 (2018)
16. Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., He, X.: Bias and debias in recommender system: a survey and future directions (2020). https://doi.org/10.48550/ARXIV.2010.03240
17. Cunningham, S.J., Bainbridge, D., Mckay, D.: Finding new music: a diary study of everyday encounters with novel songs. In: International Society for Music Information Retrieval Conference (2007)
18. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding (2019)
19. Diaz, F., Mitra, B., Ekstrand, M.D., Biega, A.J., Carterette, B.: Evaluating stochastic rankings with expected exposure. In: CIKM 2020 (2020)
20. Ding, Y., Mok, P., Ma, Y., Bin, Y.: Personalized fashion outfit generation with user coordination preference learning. Inform. Process. Manag. **60**(5), 103434 (2023). https://doi.org/10.1016/j.ipm.2023.103434, https://www.sciencedirect.com/science/article/pii/S0306457323001711
21. Dong, Y., Kang, J., Tong, H., Li, J.: Individual fairness for graph neural networks: a ranking based approach. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 300–310. KDD 2021, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3447548.3467266 event-place: Virtual Event, Singapore
22. Downey, A.B.: Evidence for long-tailed distributions in the internet. In: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, p. 229–241. IMW 2001, Association for Computing Machinery, New York, NY, USA (2001). https://doi.org/10.1145/505202.505230
23. Drott, E.: Why the next song matters: streaming, recommendation, scarcity. Twentieth-Century Music **15**, 325–357 (2018). https://doi.org/10.1017/S1478572218000245
24. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness (2011)
25. Ekstrand, M.D., Harper, F.M., Willemsen, M.C., Konstan, J.A.: User perception of differences in recommender algorithms. In: Proceedings of the 8th ACM Conference on Recommender Systems, p. 161–168. RecSys 2014, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2645710.2645737
26. Gao, C., et al.: A survey of graph neural networks for recommender systems: challenges, methods, and directions. ACM Trans. Recomm. Syst. **1**(1), 1–51 (2023). https://doi.org/10.1145/3568022
27. Garcia-Gathright, J., St. Thomas, B., Hosey, C., Nazari, Z., Diaz, F.: Understanding and evaluating user satisfaction with music discovery. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, p. 55–64. SIGIR 2018, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3209978.3210049
28. Goel, S., Broder, A., Gabrilovich, E., Pang, B.: Anatomy of the long tail: ordinary people with extraordinary tastes. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, p. 201–210. WSDM 2010, Association for Computing Machinery, New York, NY, USA (2010). https://doi.org/10.1145/1718487.1718513

29. Harper, F.M., Konstan, J.A.: The MovieLens datasets: History and context. ACM Trans. Interact. Intell. Syst. **5**(4), 1–19 (2015). https://doi.org/10.1145/2827872
30. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR 2016 (2016)
31. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: simplifying and powering graph convolution network for recommendation. In: SIGIR 2020 (2020)
32. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**(1), 5–53 (2004). https://doi.org/10.1145/963770.963772
33. Hossain, I., et al.: A survey of recommender system techniques and the ecommerce domain (2023)
34. Jadidinejad, A.H., Macdonald, C., Ounis, I.: How sensitive is recommendation systems' offline evaluation to popularity? (2019)
35. Jannach, D., Lerche, L., Kamehkhosh, I., Jugovac, M.: What recommenders recommend: an analysis of recommendation biases and possible countermeasures. User Model. User-Adap. Inter. **25**(5), 427–491 (2015). https://doi.org/10.1007/s11257-015-9165-3
36. Kamehkhosh, I., Jannach, D.: User perception of next-track music recommendations. In: Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, p. 113–121. UMAP 2017, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3079628.3079668
37. Kang, J., Zhu, Y., Xia, Y., Luo, J., Tong, H.: RawlsGCN: towards Rawlsian difference principle on graph convolutional network. In: WWW 2022 (2022)
38. Korzeniowsky, F., Oramas, S., Gouyon, F.: Artist similarity with graph neural networks. In: Proceedings of the 18th International Society for Music Information Retrieval Conference. ISMIR (2021)
39. Lavranos, C., Kostagiolas, P., Martzoukou, K.: Theoretical and applied issues on the impact of information on musical creativity: an information seeking behavior perspective, pp. 1–16 (2016). https://doi.org/10.4018/978-1-5225-0270-8.ch001
40. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell. (2020)
41. Lo, K., Ishigaki, T.: Matching novelty while training: Novel recommendation based on personalized pairwise loss weighting. In: 2019 IEEE International Conference on Data Mining (ICDM), pp. 468–477 (2019)
42. Mansoury, M., Abdollahpouri, H., Pechenizkiy, M., Mobasher, B., Burke, R.: A graph-based approach for mitigating multi-sided exposure bias in recommender systems. ACM Trans. Inform. Syst. **40**(2), 1–31 (2021). https://doi.org/10.1145/3470948
43. Melchiorre, A., Rekabsaz, N., Parada-Cabaleiro, E., Brandl, S., Lesota, O., Schedl, M.: Investigating gender fairness of recommendation algorithms in the music domain. Inform. Process. Manag. **58**, 102666 (2021)
44. Mäntymäki, M., Islam, N.: Gratifications from using freemium music streaming services: differences between basic and premium users. In: International Confererence on Information Systems (2015)
45. Möller, J., Trilling, D., Helberger, N., van Es, B.: Do not blame it on the algorithm: an empirical assessment of multiple recommender systems and their impact on content diversity. Information, Communication & Society (2018)
46. Noh, T., Yeo, H., Kim, M., Han, K.: A study on user perception and experience differences in recommendation results by domain expertise: the case of fashion domains.

In: Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems. CHI EA 2023, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3544549.3585641
47. Park, Y.J., Tuzhilin, A.: The long tail of recommender systems and how to leverage it. In: Proceedings of the 2008 ACM Conference on Recommender Systems, p. 11–18. RecSys 2008, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1454008.1454012
48. Patro, G.K., Biswas, A., Ganguly, N., Gummadi, K.P., Chakraborty, A.: FairRec: two-sided fairness for personalized recommendations in two-sided platforms. In: Proceedings of The Web Conference 2020. ACM (2020). https://doi.org/10.1145/3366423.3380196
49. Raff, A., Mladenow, A., Strauss, C.: Music discovery as differentiation strategy for streaming providers. In: Proceedings of the 22nd International Conference on Information Integration and Web-Based Applications & Services, p. 476–480. iiWAS 2020, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3428757.3429151
50. Rey, D., Neuhäuser, M.: Wilcoxon-signed-rank test (2011)
51. Rhee, W., Cho, S.M., Suh, B.: Countering popularity bias by regularizing score differences. In: Proceedings of the 16th ACM Conference on Recommender Systems, p. 145–155. RecSys 2022, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3523227.3546757
52. Salha-Galvan, G., Hennequin, R., Chapus, B., Tran, V.A., Vazirgiannis, M.: Cold start similar artists ranking with gravity-inspired graph autoencoders (2021). https://doi.org/10.48550/ARXIV.2108.01053
53. Saravanou, A., Tomasi, F., Mehrotra, R., Lalmas, M.: Multi-task learning of graph-based inductive representations of music content. In: Proceedings of the 22nd International Society for Music Information Retrieval Conference, pp. 602–609. ISMIR, Online (2021). https://doi.org/10.5281/zenodo.5624379
54. Schedl, M., Zamani, H., Chen, C.W., Deldjoo, Y., Elahi, M.: Current challenges and visions in music recommender systems research. Int. J. Multimedia Inform. Retrieval **7**(2), 95–116 (2018). https://doi.org/10.1007/s13735-018-0154-2
55. Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., Joachims, T.: Recommendations as treatments: debiasing learning and evaluation (2016)
56. Stanisljevic, D.: The impact of Spotify features on music discovery in the streaming platform age. Master's thesis (2020). http://hdl.handle.net/2105/55511
57. Steck, H.: Item popularity and recommendation accuracy. In: RecSys 2011. ACM (2011)
58. Wang, X., Wang, W.H.: Providing item-side individual fairness for deep recommender systems. In: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, p. 117–127. FAccT 2022, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3531146.3533079
59. Wei, T., Feng, F., Chen, J., Wu, Z., Yi, J., He, X.: Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, p. 1791–1800. KDD 2021, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3447548.3467289
60. Wu, C., Wu, F., Huang, Y., Xie, X.: Personalized news recommendation: methods and challenges (2022)
61. Wu, S., Sun, F., Zhang, W., Xie, X., Cui, B.: Graph neural networks in recommender systems: a survey (2020)

62. Yang, C.C., Chen, H., Hong, K.: Visualization of large category map for internet browsing. Decis. Support Syst. **35**(1), 89–102 (2003). https://doi.org/10.1016/S0167-9236(02)00101-X
63. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM (2018). https://doi.org/10.1145/3219819.3219890
64. Zhang, A., Ma, W., Wang, X., Chua, T.S.: Incorporating bias-aware margins into contrastive loss for collaborative filtering. In: Advances in Neural Information Processing Systems, vol. 35: Annual Conference on Neural Information Processing Systems, NeurIPS (2022)
65. Zhang, Y., et al.: Causal intervention for leveraging popularity bias in recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, p. 11–20. SIGIR 2021, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3404835.3462875
66. Zhao, M., et al.: Investigating accuracy-novelty performance for graph-based collaborative filtering. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM (2022). https://doi.org/10.1145/3477495.3532005
67. Zheng, Y., Gao, C., Li, X., He, X., Li, Y., Jin, D.: Disentangling user interest and conformity for recommendation with causal embedding. In: Proceedings of the Web Conference 2021, p. 2980–2991. WWW 2021, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3442381.3449788
68. Zhu, Z., He, Y., Zhao, X., Zhang, Y., Wang, J., Caverlee, J.: Popularity-opportunity bias in collaborative filtering. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, p. 85–93. WSDM 2021, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3437963.3441820