Auditing Consumer- and Producer-Fairness in Graph Collaborative Filtering

Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Daniele Malitesta*,
Vincenzo Paparella, and Claudio Pomo*

Politecnico di Bari, Bari, Italy, name.surname@poliba.it

Abstract. To date, *graph* collaborative filtering (CF) strategies have been shown to outperform pure CF models in generating accurate recommendations. Nevertheless, recent works have raised concerns about fairness and potential biases in the recommendation landscape since unfair recommendations may harm the interests of Consumers and Producers (CP). Acknowledging that the literature lacks a careful evaluation of graph CF on CP-aware fairness measures, we initially evaluated the effects on CP-aware fairness measures of eight state-of-the-art graph models with four pure CF recommenders. Unexpectedly, the observed trends show that graph CF solutions do not ensure a large item exposure and user fairness. To disentangle this performance puzzle, we formalize a taxonomy for graph CF based on the mathematical foundations of the different approaches. The proposed taxonomy shows differences in node representation and neighbourhood exploration as dimensions characterizing graph CF. Under this lens, the experimental outcomes become clear and open the doors to a multi-objective CP-fairness analysis[1].

Keywords: Graph Collaborative Filtering · Fairness · Multi-Objective Analysis

1 Introduction and Motivations

Recommender systems (RSs) are ubiquitous and utilized in a wide range of domains from e-commerce and retail to media streaming and online advertising. Personalization, or the system's ability to suggest relevant and engaging products to users, has long served as a key indicator for gauging the success of RSs. In recent decades, collaborative filtering (CF) [10], the predominant modeling paradigm in RSs, has shifted from neighborhood techniques [10, 30, 31] to frameworks based on the learning of users' and items' latent factors [16, 29, 49]. More recently, deep learning (DL) models have been proposed to overcome the linearity of traditional latent factors approaches.

Among these DL algorithms, graph-based methods view the data in RSs from the perspective of graphs. By modeling users and items as nodes with latent representations and their interactions as edges, the data can be naturally

[*] Authors are listed in alphabetical order. Corresponding authors: Daniele Malitesta (daniele.malitesta@poliba.it) and Claudio Pomo (claudio.pomo@poliba.it).

[1] Codes are available at: https://github.com/sisinflab/ECIR2023-Graph-CF.

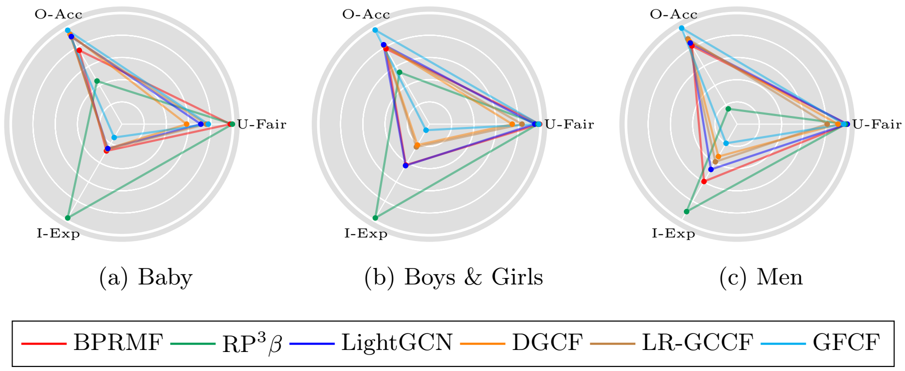
represented as a user-item bipartite graph. By iteratively aggregating contributions from near- and long-distance neighborhoods, the so-called message-passing schema updates nodes' initial representations and effectively distills the collaborative signal [43]. Early works [5, 50] adopted the vanilla graph convolutional network (GCN) [15] architecture and paved the way to advanced algorithms lightening the message-passing schema [8, 14] and exploring different graph sampling strategies [47]. Recent approaches propose simplified formulations [21, 26] that optionally transfer the graph CF paradigm to different spaces [33, 34]. As some graph edges may provide noisy contributions to the message-passing schema [39], a research line focuses on meaningful user-item interactions [36, 42, 45]. In this context, explainability is the natural next step [18] towards the disentanglement of user-item connections into a set of user intents [44, 46].

On the other side, the adoption of DL (and, often, black-box) approaches to the recommendation task has raised issues regarding the fairness of RSs. The concept of fairness in recommendation is multifaceted. Specifically, the two core aspects to categorize recommendation fairness may be summarized as (1) the primary parties engaged (consumers vs. producers) and (2) the type of benefit provided (exposure vs. relevance). Item suppliers are more concerned about exposure fairness than customers because they want to make their products better known and visible (**P**roducer fairness). However, from the customer's perspective, relevance fairness is of utmost importance, and hence system designers must ensure that exposure of items is equally effective across user groups (**C**onsumer fairness). A recent study highlights that nine out of ten publications on recommendation fairness concentrated on either C-fairness or P-fairness [22], disregarding the joint evaluation between C-fairness, P-fairness, and the accuracy.

The various graph CF *strategies* described above have historically centered on the enhancement of system accuracy, but, actually, never focused on the recommendation fairness dimensions. Despite some recent graph-based approaches have specifically been designed to address C-fairness [11, 17, 27, 40, 41, 48] and P-fairness [6, 19, 20, 35, 51, 52], there is a notable *knowledge gap* in the literature about the effects of the state-of-the-art graph *strategies* on the three objectives of C-fairness, P-fairness, and system accuracy. This work intends to complement the previous research and provide answers to pending research problems such as how different graph models perform for the three evaluation objectives. By measuring these dimensions in terms of **overall accuracy**, **user fairness**, and **item exposure**, we observe these aspects in detail[2].

Motivating example. A preliminary comparison of the leading graph and classical CF models is carried out to provide context for our study. The graph-based models include LightGCN [14], DGCF [44], LR-GCCF [8], and GFCF [33], which are tested against two classical CF baselines, namely BPRMF [28] and RP$^3\beta$ [25], on the Baby, Boys & Girls, and Men datasets from the Amazon catalog [23]. We train each baseline using a total of 48 unique hyper-parameter settings and select the optimal configuration for each baseline as the one achieving the highest

[2] In the rest of the paper, when no confusion arises, we will refer to C-fairness with user fairness, to P-fairness with item exposure, and to their combination as CP-fairness.



(a) Baby

(b) Boys & Girls

(c) Men

Fig. 1: Kiviat diagrams indicating the performance of selected pure and graph CF recommenders on overall accuracy (i.e., O-Acc, calculated with the *nDCG@20*), item exposure (i.e., I-Exp, calculated with the *APLT@20* [1]), and user fairness (U-Fair, calculated with the *UMADrat@20* [9]). Higher means better.

accuracy on the validation set (as in the original papers). Overall accuracy, user fairness, and item exposure (as introduced above) are evaluated. Figure 1 displays the performance of the selected baselines on the three considered recommendation objectives. For better visualization, all values are scaled between 0 and 1 using min-max normalization, and, when needed, they are replaced by their 1's complement to adhere to the "higher numbers are better" semantics. As a result, in each of the three dimensions, the values lay in $[0, 1]$ with higher values indicating the better. Please, note that such an experimental evaluation is not the main focus of this work but it is the motivating example for the more extensive analysis we present later. The interested reader may refer to **Appendix A** for a presentation of the full experimental settings to reproduce these results and the ones reported in the following sections of the paper.

First, according to Figure 1, graph CF models are significantly more accurate than the classical CF ones, even if the latter perform far better in terms of item exposure. Moreover, the displayed trends suggest there is no clear winner on the user fairness dimension: classical CF models show promising performance, while some graph CF models do not achieve remarkable results. As a final observation, an underlying trade-off between the three evaluation goals seems to exist, and it might be worth investigating it in-depth. Such outcomes open to a more complete study on how **different strategy patterns** recognized in graph CF may affect the three recommendation objectives, which is the scope of this work.

Research questions and contributions. In the remainder of this paper, we therefore attempt to answer the following two research questions (RQs):

RQ1. Given the different graph CF strategies, the raising question is *"Can we explain the variations observed when testing several graph models on overall accuracy, item exposure, and user fairness separately?"* According to a recent benchmark that identifies some state-of-the-art graph techniques [54], the sug-

Table 1: Categorization of the chosen graph baselines according to the proposed taxonomy. For each model, we refer to the technical description reported in the original paper and try to match it with our taxonomy.

Models	**Nodes Representation**				**Neighborhood Exploration**			
	Latent representation		**Weighting**		**Explored nodes**		**Message passing**	
	low	*high*	*weighted*	*unweighted*	*same*	*different*	*implicit*	*explicit*
GCN-CF* [15]		✓		✓	✓			✓
GAT-CF* [39]		✓	✓		✓			✓
NGCF [43]	✓			✓		✓		✓
LightGCN [14]	✓			✓		✓		✓
DGCF [44]	✓		✓			✓		✓
LR-GCCF [8]	✓			✓	✓	✓		✓
UltraGCN [21]	✓				✓	✓	✓	
GFCF [33]						✓	✓	

*The postfix -CF indicates that we re-adapted the original implementations (tailored for the task of node classification) to the task of personalized recommendation.

gested graph CF taxonomy (Table 1) extends the set of graph-based models introduced in the motivating example by examining eight state-of-the-art graph CF baselines through their strategies for *nodes representation* and *neighborhood exploration*. We present a more nuanced view of prior findings by analyzing the impact of each taxonomy dimension on overall accuracy and CP-fairness.
RQ2. The demonstrated performance prompts the questions: *"How and why nodes representation and neighborhood exploration algorithms can strike a trade-off between overall accuracy, item exposure, and user fairness?"* We employ the Pareto optimality to determine the influence of such dimensions in two-objective scenarios, where the objectives include overall accuracy, item exposure, and user fairness. The Pareto frontier is computed for three 2-dimensional spaces: accuracy/item exposure, accuracy/user fairness, and item exposure/user fairness.

2 Nodes Representation and Neighborhood Exploration in Graph Collaborative Filtering: A Formal Taxonomy

2.1 Preliminaries

Let $\mathcal{U}$ be the set of N users, and $\mathcal{I}$ the set of M items in the system, respectively. We represent the observed interactions between users and items in a binary format (i.e., implicit feedback). Specifically, let $\mathbf{R} \in \mathbb{R}^{N \times M}$ be the user-item feedback matrix, where $r_{u,i} = 1$ if user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$ have a recorded interaction, $r_{u,i} = 0$ otherwise. Following the above preliminaries, we introduce $\mathcal{G} = (\mathcal{U}, \mathcal{I}, \mathbf{R})$ as the bipartite and undirected graph connecting users and items (the graph nodes) when there exists a recorded bi-directional interaction among them (the graph edges). Nodes features for user $u \in \mathcal{U}$ and $i \in \mathcal{I}$ are suitably encoded as the embeddings $\mathbf{e}_u \in \mathbb{R}^d$ and $\mathbf{e}_i \in \mathbb{R}^d$, with $d << N, M$. Given the dual nature of user and item derivations, we only report user-side formulas.

2.2 Updating node representation through message-passing

The representation of users' and items' nodes are updated by leveraging the graph topology from $\mathcal{G}$. In this respect, the message-passing schema has recently gained attention in the literature. The algorithm works by aggregating the information (i.e., the *messages*) from the *neighbor* nodes into the *ego* node, and the process is recursively performed for multiple hops thus exploring wider neighborhood portions. In general, the message-passing for l hops is:

$$\mathbf{e}_u^{(l)} = \omega\left(\left\{\mathbf{e}_{i'}^{(l-1)}, \forall i' \in \mathcal{N}(u)\right\}\right), \tag{1}$$

where $\omega(\cdot)$ and $\mathcal{N}(\cdot)$ are the aggregation function and neighborhood node set, respectively, while l is in $1 \leq l \leq L$, where L is a hyper-parameter. Note that the following statements hold: $\mathbf{e}_u^{(0)} = \mathbf{e}_u$ and $\mathbf{e}_i^{(0)} = \mathbf{e}_i$. A reworking of Equation (1) for $l \in \{2, 3\}$ allows *same-* and *different*-type node representation emerge [3]:

$$\begin{array}{ll}
\textit{\textbf{Same-type node representation}} & \left\{ \underbrace{\mathbf{e}_u^{(2)}}_{\text{(user)}} = \omega\Big(\Big\{\omega\Big(\Big\{ \underbrace{\mathbf{e}_{u''}^{(0)}}_{\text{(user)}}, \forall u'' \in \mathcal{N}(i') \setminus \{u\}\Big\}\Big), \forall i' \in \mathcal{N}(u)\Big\}\Big) \right. \\
\textit{\textbf{Different-type node representation}} & \left\{ \begin{array}{l} \underbrace{\mathbf{e}_u^{(3)}}_{\text{(user)}} = \omega\Big(\Big\{\omega\Big(\Big\{\omega\Big(\Big\{ \underbrace{\mathbf{e}_{i'''}^{(0)}}_{\text{(item)}}, \forall i''' \in \mathcal{N}(u'') \setminus \{i''\}\Big\}\Big), \\ \qquad \forall u'' \in \mathcal{N}(i') \setminus \{u''\}\Big\}\Big), \forall i' \in \mathcal{N}(u)\Big\}\Big). \end{array} \right.
\end{array} \tag{2}$$

To better clarify the extent of Equation (2), after an **even** and an **odd** number of explored hops, *ego* node updates leverage by design *same-* and *different*-type node connections, i.e., user-user/item-item and user-item/item-user as evident from Equation (2). While the existing literature does not always consider the two scenarios as distinct, we underline the importance of investigating the influence of different node-node connections explored during the message-passing. In light of the above, we will count the number of explored hops as follows: $\mathbf{e}_*^{(2l)}, \forall l \in \{1, 2, \ldots, \frac{L}{2}\}$ as obtained through l **same-**type node connections (denoted as *same-l*), and $\mathbf{e}_*^{(2l-1)}, \forall l \in \{1, 2, \ldots, \frac{L}{2}\}$ as obtained through l **different-**type node connections (denoted as *different-l*). In the following, we introduce the graph convolutional network (GCN) and its recent CF applications.

The baseline: graph convolutional network (GCN). The standard graph convolutional network from Kipf and Welling [15] performs feature transformation, message aggregation, application of a one-layer neural network, element-wise addition, and ReLU activation, respectively. Let us consider $\mathbf{W}^{(l)} \in \mathbb{R}^{d_{l-1} \times d_l}$ and $\mathbf{b}^{(l)} \in \mathbb{R}^{d_l}$ as the weight matrix and the bias for the l-th explored hop. The message-passing for user u is:

$$\mathbf{e}_u^{(l)} = \mathrm{ReLU}\left(\sum_{i' \in \mathcal{N}(u)} \left(\mathbf{W}^{(l)}\mathbf{e}_{i'}^{(l-1)} + \mathbf{b}^{(l)}\right)\right). \tag{3}$$

GCN for collaborative filtering. Inspired by the GCN message-passing approach, the authors from Wang et al. [43] propose neural graph collaborative filtering (NGCF). At each hop exploration, the model aggregates the neighborhood information and the inter-dependencies among the *ego* and the neighborhood nodes. Formally, the aggregation could be formulated as follows:

$$\mathbf{e}_u^{(l)} = \text{LeakyReLU}\left(\sum_{i' \in \mathcal{N}(u)} \left(\mathbf{W}_{\text{neigh}}^{(l)} \mathbf{e}_{i'}^{(l-1)} + \mathbf{W}_{\text{inter}}^{(l)} \left(\mathbf{e}_{i'}^{(l-1)} \odot \mathbf{e}_u^{(l-1)}\right) + \mathbf{b}^{(l)}\right)\right), \tag{4}$$

where LeakyReLU is the activation function, $\mathbf{W}_{\text{neigh}}^{(l)} \in \mathbb{R}^{d_{l-1} \times d_l}$ and $\mathbf{W}_{\text{inter}}^{(l)} \in \mathbb{R}^{d_{l-1} \times d_l}$ are the neighborhood and inter-dependencies weight matrices, respectively, while $\odot$ is the Hadamard product.

He et al. [14] propose a light convolutional network, namely LightGCN, with the rationale to simplify the message-passing schema from GCN and NGCF by dropping feature transformations (i.e., the weight matrices and biases) and the non-linearity applied after the message aggregation. Specifically, they implement:

$$\mathbf{e}_u^{(l)} = \sum_{i' \in \mathcal{N}(u)} \mathbf{e}_{i'}^{(l-1)}. \tag{5}$$

The variation shows superior accuracy to the state-of-the-art. A slightly different solution [8] can outperform LightGCN regarding the accuracy level.

2.3 Weighting the importance of graph edges

The message-passing schema is inherently designed to aggregate into the *ego* node all messages coming from its neighborhood. Nevertheless, the *binary* nature of the user-item feedback (i.e., 0/1) would suggest that not all recorded user-item interactions necessarily hide the same importance to the nodes they involve.

In general, let $a_{y \rightarrow x}^{(l)}$ be the importance of the neighbor node y on its ego node x after l explored hops. We re-write the formulation of the message-passing after l explored hops (presented in Equation (1)) as:

$$\mathbf{e}_u^{(l)} = \omega\left(\left\{a_{i' \rightarrow u}^{(l)} \mathbf{e}_{i'}^{(l-1)}, \forall i' \in \mathcal{N}(u)\right\}\right). \tag{6}$$

The baseline: graph attention network (GAT). Attention mechanisms have reached considerable success in the GCN-related literature to weight the contribution of neighbor messages before aggregation. The original study [39] proposes the following message-passing formulation:

$$\begin{aligned}\mathbf{e}_u^{(l)} &= \sum_{i' \in \mathcal{N}(u)} \left(a_{i' \rightarrow u}^{(l)} \mathbf{W}_{\text{neigh}}^{(l)} \mathbf{e}_{i'}^{(l-1)} + \mathbf{b}^{(l)}\right) \\ &= \sum_{i' \in \mathcal{N}(u)} \left(\alpha\left(\mathbf{e}_{i'}^{(l-1)}, \mathbf{e}_u^{(l-1)}\right) \mathbf{W}_{\text{neigh}}^{(l)} \mathbf{e}_{i'}^{(l-1)} + \mathbf{b}^{(l)}\right),\end{aligned} \tag{7}$$

where $\alpha(\cdot)$ is the importance function depending on the lastly-calculated embeddings of the neighbor and the ego nodes, e.g., $a_{i' \rightarrow u}^{(l)} = \alpha\left(\mathbf{e}_{i'}^{(l-1)}, \mathbf{e}_u^{(l-1)}\right)$.

GAT for collaborative filtering. The authors from Wang et al. [44] design a message-passing schema that calculates the importance of neighborhood nodes for *ego* nodes by disentangling the intents underlying each user-item interaction. Similarly to He et al. [14] and Chen et al. [8], they therefore propose the following embedding update formulation:

$$
\begin{aligned}
\mathbf{e}_u^{(l)} &= \sum_{i' \in \mathcal{N}(u)} a_{i' \to u}^{(l)} \mathbf{e}_{i'}^{(l-1)} \\
&= \sum_{i' \in \mathcal{N}(u)} \alpha \left(\mathbf{e}_{i'}^{(l-1)}, \mathbf{e}_u^{(l-1)}, K, T \right) \mathbf{e}_{i'}^{(l-1)},
\end{aligned}
\tag{8}
$$

where $\alpha\left(\cdot, K, T\right)$ is the importance function of the lastly-calculated embeddings from the neighbor and the *ego* nodes, e.g., $a_{i' \to u}^{(l)} = \alpha\left(\mathbf{e}_{i'}^{(l-1)}, \mathbf{e}_u^{(l-1)}, K, T\right)$, K is the total number of intents, and T is the total number of routing iterations to repeat the disentangling procedure.

2.4 Going beyond message-passing

The recent graph learning literature [7, 53] has outlined the phenomenon of *over-smoothing*, that leads node representations to become more similar as more hops are explored. The issue is generally tackled by limiting the neighborhood exploration to (maximum) three hops, and to two hops when attention mechanisms are introduced. However, the idea of improving accuracy by restricting the number of explored neighborhoods is counter-intuitive and "conflicts" with the rationale behind collaborative filtering [4]. This awareness led works such as Mao et al. [21] and Shen et al. [33] to surpass and simplify the traditional concept of message-passing. UltraGCN [21] adopts negative sampling to contrast over-smoothing and additional objective terms to (i) approximate the infinite neighborhood exploration and (ii) mine relevant "unexpected" node-node interactions such as the item-item ones. Conversely, GFCF [33] translates the graph-based recommendation task into the graph signal processing domain to obtain a closed-form formulation for approximating the infinite neighborhood exploration. Given that such recent strategies do not *explicitly* perform the message-passing schema as presented above, in the remaining sections of this paper, we will adopt the terms *explicit* and *implicit* message-passing as shorthands to denote the two model families, respectively.

2.5 A taxonomy of graph CF approaches

We propose (see Table 1) a taxonomy to classify the state-of-the-art graph models. The taxonomy considers the recurrent **strategy patterns** as emerged by conducting an in-depth review and analyzing the different graph CF approaches.

- **Node representation** indicates the representation strategy to model users' and items' nodes. It involves the *dimensionality* of node embeddings, and the possibility of *weighting* the neighbor node contributions.

- **Neighborhood exploration** refers to the procedure for exploring the multi-hop neighborhoods of each node to update the node latent representation. It involves the type of *node-node connections* which are explored, and the *message-passing* schema (i.e., *explicit* or *implicit* as previously defined).

In the next two sections, we will assess the performance of the graph CF models from the taxonomy in Table 1. Thus, we consider GCN-CF [15], GAT-CF [39], NGCF [43], LightGCN [14], DGCF [44], LR-GCCF [8], UltraGCN [21], and GFCF [33] for a total of eight graph CF solutions.

3 Taxonomy-aware evaluation

This section aims to answer RQ1 (*"Can we explain the variations observed when testing several graph models on overall accuracy, item exposure, and user fairness separately?"*) by showing how the proposed taxonomy of graph strategies can explain the recommendation evaluation on CP-Fairness and overall accuracy. We experiment with 48 hyper-parameter configurations to investigate various combinations of graph CF techniques for *message-passing*, *explored nodes*, *edge weighting*, and *latent representations.* Results refer to the Amazon Men dataset and top-20 lists (Table 2). Please note that we report the **best** metric result for each <dimension, value> pair (the corresponding best graph recommendation model is displayed below each metric result) to ease the interpretation of results and provide meaningful insights.

- **Message-passing.** We investigate the two widely-recognized message-passing strategies: *implicit* and *explicit.* The most obvious pattern indicates that both sets have almost the same number of top-performing models in each of the evaluation criteria. *Explicit* graph approaches perform better on item exposure, where they outperform *implicit* techniques (i.e., on *Gini* and *APLT*) two out of three times by a significant margin. On the one hand, this tendency may be due to the absence of a direct message (information) propagating along the user-item graph in *implicit* techniques, which prevents the user node from exploring vast item segments. On the other hand, it appears that models from both families perform similarly on accuracy and user fairness, indicating that there is no obvious reason to favor *implicit* over *explicit* or vice versa.
- **Explored nodes.** Here, we examine four methods to explore nodes (adopting the message-passing re-formulation from Equation (2)): *same* and *different*, with 1 and 2 hops. Similarly to the trend found for the message-passing dimension, the results demonstrate that the two primary categories (*same* and *different*) are nearly equally performing across all measurements, with *same-2* and *different-1* being the prominent ones. In detail, the *different-1* exploration outperforms the *same-2* on the overall accuracy level (GFCF is the leading model here). Conversely, *same-2* is the best strategy for item exposure (with LR-GCCF and GAT-CF leading). As observed for the message-passing, user fairness does not give a reason to choose between *same* and *different.* The exploration of 1 hop in *same* and *different* settings is the preferable technique, even if 2 hops connections lead to a better item exposure.

Table 2: Best metric results (and corresponding graph CF model) for each <dimension, value> pair, on the Amazon Men dataset for top-20 lists. **Bold** is used to indicate the best result in the pairs having a two-valued dimension, while † is used only for the "explored nodes" dimension to indicate also the best results on *same* and *different*. The symbols ↑ and ↓ indicate whether better stands for high or low values. We use "*rank*" and "*rat*" as the *UMADrank@k* and *UMADrat@k*.

Dimensions	**Values**	**Overall Accuracy**		**Item Exposure**			**User Fairness**	
		Recall↑	*nDCG*↑	*EFD*↑	*Gini*↑	*APLT*↑	*rank*↓	*rat*↓
Message passing	*implicit*	0.1222 (GFCF)	**0.0911** **(GFCF)**	**0.2615** **(GFCF)**	0.2871 (UltraGCN)	0.1808 (UltraGCN)	0.0123 (UltraGCN)	**0.0022** **(UltraGCN)**
	explicit	**0.1223** **(LR-GCCF)**	0.0884 (LR-GCCF)	0.2536 (LR-GCCF)	**0.5090** **(LR-GCCF)**	**0.3823** **(GAT-CF)**	**0.0002** **(DGCF)**	0.0169 (LightGCN)
Explored nodes	*same-1*	0.1221† (LR-GCCF)	0.0884† (LR-GCCF)	0.2500† (LR-GCCF)	0.4377 (LR-GCCF)	0.3433 (GAT-CF)	**0.0002†** **(DGCF)**	**0.0022†** **(UltraGCN)**
	same-2	0.1184 (LightGCN)	0.0841 (LightGCN)	0.2380 (LightGCN)	**0.5090†** **(LR-GCCF)**	**0.3823†** **(GAT-CF)**	**0.0002†** **(DGCF)**	0.0209 (NGCF)
	different-1	**0.1222†** **(GFCF)**	**0.0911†** **(GFCF)**	**0.2615†** **(GFCF)**	0.4093 (NGCF)	0.3424 (GAT-CF)	**0.0002†** **(DGCF)**	**0.0022†** **(UltraGCN)**
	different-2	0.1210 (DGCF)	0.0850 (DGCF)	0.2407 (LightGCN)	0.4934† (LR-GCCF)	0.3438† (LR-GCCF)	**0.0002†** **(DGCF)**	0.0388 (LightGCN)
Weighting	*weighted*	0.1210 (DGCF)	0.0857 (DGCF)	0.2428 (DGCF)	0.3240 (DGCF)	**0.3823** **(GAT-CF)**	**0.0002** **(DGCF)**	0.0301 (DGCF)
	unweighted	**0.1223** **(LR-GCCF)**	**0.0884** **(LR-GCCF)**	**0.2536** **(LR-GCCF)**	**0.5090** **(LR-GCCF)**	0.3438 (LR-GCCF)	0.0101 (GCN-CF)	**0.0169** **(LightGCN)**
Latent representations	*emb-64*	0.1193 (LR-GCCF)	0.0871 (LR-GCCF)	0.2479 (LR-GCCF)	**0.5090** **(LR-GCCF)**	0.3627 (GAT-CF)	**0.0002** **(DGCF)**	0.0054 (UltraGCN)
	emb-128	0.1221 (LR-GCCF)	0.0883 (LR-GCCF)	**0.2536** **(LR-GCCF)**	**0.5090** **(LR-GCCF)**	0.3644 (GAT-CF)	**0.0002** **(DGCF)**	0.0111 (UltraGCN)
	emb-256	**0.1223** **(LR-GCCF)**	**0.0884** **(LR-GCCF)**	0.2532 (LR-GCCF)	0.5038 (LR-GCCF)	**0.3823** **(GAT-CF)**	**0.0002** **(DGCF)**	**0.0022** **(UltraGCN)**

- **Weighted.** This study examines *weighted* and *unweighted* graph CF techniques. Differently from above, we observe that *unweighted* solutions provide the best performance on almost all CP-fairness metrics, with LR-GCCF steadily being the superior approach. The only trend deviation refers to GAT-CF (i.e., a *weighted* method) surpassing *unweighted* solutions on the *APLT* level, that is, recommending items from the long-tail. The behavior is likely attributable to the design of *weighted* techniques, which can investigate farther neighbors of the *ego* node (observe the performance of GAT-CF on the *same-2* dimension), leading user profiles to match distant (and possibly niche) products in the catalog. On the contrary, it is interesting to notice how the other two metrics accounting for item exposure (i.e., *EFD* as item novelty measure and *Gini* as item diversity measure) seem to privilege *unweighted* graph techniques (i.e., LR-GCCF). The observed behaviors differ as the three metrics provide completely different perspectives of the *item exposure*, and thus they are uncorrelated.

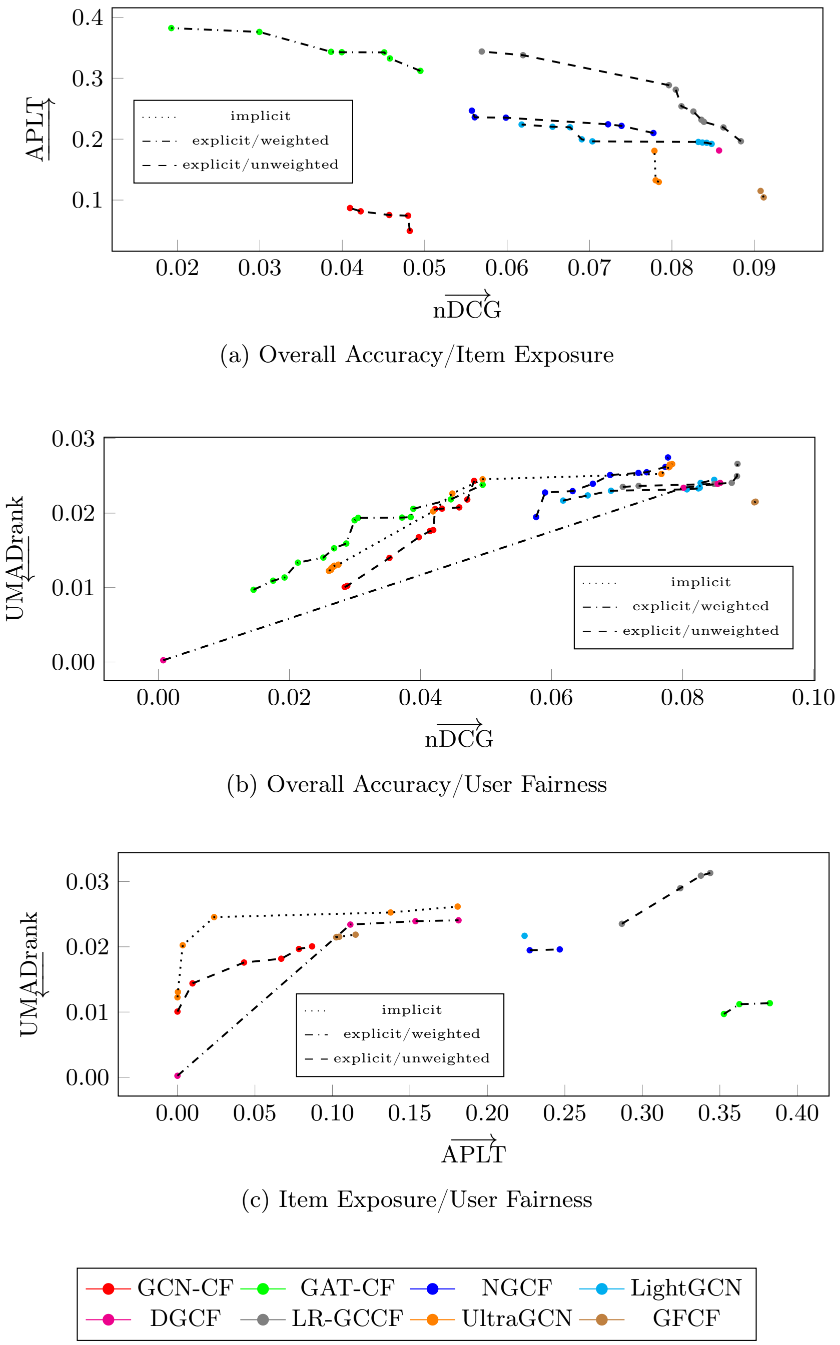
- **Latent representations.** We compare the performance of graph CF techniques adopting latent representations with *64*, *128*, and *256* features, respectively. It is worth noticing that higher latent representations (i.e., *128* and *256*) result in better performance on all measurements. Specifically, it appears that the *128* dimension is the turning point after which the trend becomes stable (i.e., the metric values for *128* and *256* are frequently comparable). This may be an important insight since the majority of research works in recent literature tend to employ *64*-embedded representations of nodes without exploring further dimensionalities (see Table 1 as a reference).

4 Trade-off Analysis

This section analyses how the graph CF baselines balance the trade-off among accuracy, item exposure, and user fairness, and aims to answer RQ2 (*"How and why nodes representation and neighborhood exploration algorithms can strike a trade-off between overall accuracy, item exposure, and user fairness?"*). Due to space constraints, we report the results only for the Amazon Men dataset. The negative Pearson correlation values for accuracy/item exposure (*nDCG*/*APLT*) and accuracy/user fairness (*nDCG*/*UMADrank*) suggest that a trade-off may be necessary, and desirable. In addition, the same correlation metric indicates the necessity of a trade-off for item exposure/user fairness (*APLT*/*UMADrank*). Among the strategy patterns identified in the proposed taxonomy (see Table 1), we select the most important architectural dimensions, **message-passing** and **weighting** of graph edges, to conduct this study. In detail, the analysis studies three combined categories: (1) models with implicit message-passing (denoted as *implicit*); (2) models with explicit message-passing and neighborhood weighting (denoted as *explicit/weighted*); (3) models with explicit message-passing without neighborhood weighting (denoted as *explicit/unweighted*). For each analyzed trade-off, we select the Pareto optimal solutions[3] of the baselines laying on the model-specific Pareto frontier [24]. Figure 2 plots graph models Pareto frontiers in the common *objective function spaces* related to the considered trade-offs. The careful reader may notice the different axis' scales across the graphics due to the metric values. The colors of Pareto optimal solutions are model-specific, while the line style is used to distinguish the categories: dotted lines for *implicit*, dash-dot lines for *explicit/weighted*, and dashed lines for *explicit/unweighted*.

- **Accuracy/Item Exposure.** Figure 2a shows that the *explicit/weighted* models exhibit a trade-off, as they maximize either *nDCG* (i.e., DGCF) or *APLT* (i.e., GAT-CF), but not both. This is expected since DGCF is designed as a version of GAT-CF with improved accuracy. It is worth mentioning that DGCF's trade-off is reached at the expense of item exposure. In contrast to these models, *explicit/unweighted* baselines show a balanced trade-off because

[3] A solution is Pareto optimal if no other solution can improve an objective without hurting the other one.



(a) Overall Accuracy/Item Exposure

(b) Overall Accuracy/User Fairness

(c) Item Exposure/User Fairness

Fig. 2: Overall Accuracy/Item Exposure, Overall Accuracy/User Fairness, and Item Exposure/User Fairness trade-offs on Amazon Men, assessed through *nDCG/APLT*, *nDCG/UMADrank*, and *APLT/UMADrank*, respectively. Each point depicts a model hyper-parameter configuration set belonging to the corresponding Pareto frontier. Colors refer to a particular baseline, while lines styles discern their technical strategies based on the proposed taxonomy. Arrows indicates the optimization direction for each metric on x and y axes.

they do not prioritize accuracy or item exposure exclusively. In detail, LR-GCCF provides the best performance in terms of *nDCG* and *APLT* simultaneously. From a visual inspection, LR-GCCF's Pareto frontier dominates those of the other *explicit/unweighted* models. Conversely, GCN-CF exhibits the worst trade-off because it is neither ideal for *nDCG* nor *APLT*. As for the *implicit* models, they appear to prioritize precision over the provision of long-tail items. *Under this lens, the latest (i.e., implicit) approaches seem to increase accuracy, even if this is to the detriment of the niche items exposure.*

- **Accuracy/User Fairness.** To ease the interpretation of Figure 2b, we recall that *UMADrank* (used to measure User Fairness) measures to what extent the model ranking performance differs among the user groups (partitioned based on their activity on the platform). Figure 2b shows that, for GAT-CF and GCN-CF, the poor performance in terms of *nDCG* is associated with high variability in terms of user fairness. In fact, for these two models, the *UMADrank* value indicates high variability across user groups. Something different emerges for models such as NGCF, LightGCN, LR-GCF, and GFCF. These models, GFCF in particular, exhibit valuable recommendation accuracy with better stability in terms of ranking performance across the different user groups. As a consequence, the Pareto frontiers associated with these models dominate the others. In detail, GFCF is the best-performing one regarding this trade-off. Conversely, UltraGCN and DCGF do not show consistent behavior demonstrating a strong sensitivity to the chosen hyper-parameters set. *In this setting, no graph CF strategy emerges as the absolute winner. Specifically, every graph CF strategy is not enough to guarantee adequate fairness among different user groups. Then, the positive results are associated with particular configurations of some models and are lost when the hyper-parameter set changes.*
- **Item Exposure/User Fairness.** The trade-off indicates to what extent graph CF models can treat final users fairly and recommend items from the long tail. In Figure 2c, it is possible to identify two groups of baselines: the models that show poor performance in terms of item exposure (UltraGCN, DGCF, GCN-CF, and GFCF) and the models that exhibit an acceptable exposure for long-tail items (LightGCN, NGCF, LR-GCCF, and GAT-CF). In detail, a cluster of models that belong to the *explicit/unweighted* category stands out in this second group. Not only are these models able to recommend niche items, but also they are stable (among the user groups) in terms of accuracy. On the contrary, although GAT-CF lies close to the *utopia point*[4], it exhibits greater variability regarding the accuracy metric. Indeed, comparing Figure 2c with Figure 2a, GAT-CF demonstrates to achieve adequate user fairness, but its performance is still very poor in terms of accuracy. *To summarize, even if a system designer could be more interested in promoting models solely guaranteeing the best value for APLT (Producer Fairness), the explicit/unweighted strategies can generally ensure a satisfactory (for Consumers and Producers) trade-off between user fairness and item exposure.*

[4] The point that simultaneously minimizes (maximizes) all the metrics.

5 Conclusion and Future work

We assess the performance of graph CF models on Consumer and Producer (CP)-fairness metrics showing that their superior accuracy capabilities is reached at the expense of user fairness, item exposure, and their combination. By recognizing nodes representation and neighborhood exploration as the two main dimensions of a novel graph CF taxonomy, we study their influence on CP-fairness and overall accuracy separately and simultaneously. The outcomes raise concerns about the effective application of recent approaches in graph CF (e.g., implicit message-passing techniques). On such basis, we are performing further investigations on other datasets and algorithms, and we are working on new graph models balancing accuracy and CP-Fairness.

A Experimental Settings and Protocols

Datasets. As a pre-processing stage, for each dataset, we randomly sample 60k interactions and drop users and items with less than five interactions to avoid the cold-start effect [12, 13]. The final dataset statistics are: (1) Baby has 5,842 users, 7,925 items, 35,475 interactions; (2) Boys & Girls has 3,042 users, 12,912 items, 35,762 interactions; (3) Men has 3,909 users, 27,656 items, 51,519 interactions.
Reproducibility. Datasets are split using the 70/10/20 train/validation/test hold-out strategy. Baselines are trained through grid search (48 explored configurations), with a batch size of 256 and 400 epochs. Datasets and codes (implemented with Elliot [2]) are available at this **link**.
Evaluation. As for the *overall accuracy*, we use the recall (*Recall*@*k*) and the normalized discounted cumulative gain (*nDCG*@*k*). Concerning the *item exposure*, we focus on: (1) item novelty [37, 38] through the expected free discovery (*EFD*@*k*) measuring the expected portion of relevantly-recommended items that have already been seen by the users; (2) item diversity [32] with the 1's complement of the Gini index (*Gini*@*k*), a statistical dispersion measure which estimates how a model suggests heterogeneous items to users; (3) the average percentage of items from the long-tail (*APLT@k*) which are recommended in users' lists [1] to calculate recommendation's bias towards popular items. *User fairness* indicates how equally each user group receives accurate recommendations. Users are split into quartiles based on the number of items they interacted with. We then measure *UMADrat@k* and the *UMADrank@k* [9], where the former stands for the average deviation in the predicted ratings among users groups, while the latter represents the average deviation in the recommendation accuracy (calculated in terms of *nDCG@k*) among users groups. The best hyper-parameter configurations are found by considering *Recall@20* on the validation.

Acknowledgment. This work was partially supported by the following projects: IPZS-PRJ4_IA_NORMATIVO, Codice Pratica VHRWPD7 – CUP B97I19000980007 – COR 1462424 ERP 4.0, Grant Agreement Number 101016956 H2020 PASSEPARTOUT, Secure Safe Apulia, Codice Pratica 3PDW2R7 SERVIZI LOCALI 2.0, MISE CUP: I14E20000020001 CTEMT - Casa delle Tecnologie Emergenti Comune di Matera, PON ARS01_00876 BIO-D, CT_FINCONS_II.

References

[1] Abdollahpouri, H., Burke, R., Mobasher, B.: Controlling popularity bias in learning-to-rank recommendation. In: RecSys, pp. 42–46, ACM (2017)

[2] Anelli, V.W., Bellogín, A., Ferrara, A., Malitesta, D., Merra, F.A., Pomo, C., Donini, F.M., Noia, T.D.: Elliot: A comprehensive and rigorous framework for reproducible recommender systems evaluation. In: SIGIR, pp. 2405–2414, ACM (2021)

[3] Anelli, V.W., Deldjoo, Y., Noia, T.D., Sciascio, E.D., Ferrara, A., Malitesta, D., Pomo, C.: How neighborhood exploration influences novelty and diversity in graph collaborative filtering. In: MORS@RecSys, CEUR Workshop Proceedings, vol. 3268, CEUR-WS.org (2022)

[4] Anelli, V.W., Deldjoo, Y., Noia, T.D., Sciascio, E.D., Ferrara, A., Malitesta, D., Pomo, C.: Reshaping graph recommendation with edge graph collaborative filtering and customer reviews. In: DL4SR@CIKM, CEUR Workshop Proceedings, vol. 3317, CEUR-WS.org (2022)

[5] van den Berg, R., Kipf, T.N., Welling, M.: Graph convolutional matrix completion. CoRR **abs/1706.02263** (2017)

[6] Boltsis, G., Pitoura, E.: Bias disparity in graph-based collaborative filtering recommenders. In: SAC, pp. 1403–1409, ACM (2022)

[7] Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., Sun, X.: Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In: AAAI, pp. 3438–3445, AAAI Press (2020)

[8] Chen, L., Wu, L., Hong, R., Zhang, K., Wang, M.: Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In: AAAI, pp. 27–34, AAAI Press (2020)

[9] Deldjoo, Y., Anelli, V.W., Zamani, H., Bellogín, A., Noia, T.D.: A flexible framework for evaluating user and item fairness in recommender systems. User Model. User Adapt. Interact. **31**(3), 457–511 (2021)

[10] Ekstrand, M.D., Riedl, J., Konstan, J.A.: Collaborative filtering recommender systems. Found. Trends Hum. Comput. Interact. **4**(2), 175–243 (2011)

[11] Fu, Z., Xian, Y., Gao, R., Zhao, J., Huang, Q., Ge, Y., Xu, S., Geng, S., Shah, C., Zhang, Y., de Melo, G.: Fairness-aware explainable recommendation over knowledge graphs. In: SIGIR, pp. 69–78, ACM (2020)

[12] He, R., McAuley, J.J.: Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In: WWW, pp. 507–517, ACM (2016)

[13] He, R., McAuley, J.J.: VBPR: visual bayesian personalized ranking from implicit feedback. In: AAAI, pp. 144–150, AAAI Press (2016)

[14] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: SIGIR, pp. 639–648, ACM (2020)

[15] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (Poster), OpenReview.net (2017)

[16] Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)

[17] Li, C., Hsu, C., Zhang, Y.: Fairsr: Fairness-aware sequential recommendation through multi-task learning with preference graph embeddings. ACM Trans. Intell. Syst. Technol. **13**(1), 16:1–16:21 (2022)

[18] Ma, J., Cui, P., Kuang, K., Wang, X., Zhu, W.: Disentangled graph convolutional networks. In: ICML, Proceedings of Machine Learning Research, vol. 97, pp. 4212–4221, PMLR (2019)

[19] Mansoury, M., Abdollahpouri, H., Pechenizkiy, M., Mobasher, B., Burke, R.: Fairmatch: A graph-based approach for improving aggregate diversity in recommender systems. In: UMAP, pp. 154–162, ACM (2020)

[20] 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. Inf. Syst. **40**(2), 32:1–32:31 (2022)

[21] Mao, K., Zhu, J., Xiao, X., Lu, B., Wang, Z., He, X.: Ultragcn: Ultra simplification of graph convolutional networks for recommendation. In: CIKM, pp. 1253–1262, ACM (2021)

[22] Naghiaei, M., Rahmani, H.A., Deldjoo, Y.: Cpfair: Personalized consumer and producer fairness re-ranking for recommender systems. In: SIGIR, pp. 770–779, ACM (2022)

[23] Ni, J., Li, J., McAuley, J.J.: Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In: EMNLP/IJCNLP (1), pp. 188–197, Association for Computational Linguistics (2019)

[24] Paparella, V.: Pursuing optimal trade-off solutions in multi-objective recommender systems. In: RecSys, pp. 727–729, ACM (2022)

[25] Paudel, B., Christoffel, F., Newell, C., Bernstein, A.: Updatable, accurate, diverse, and scalable recommendations for interactive applications. ACM Trans. Interact. Intell. Syst. **7**(1), 1:1–1:34 (2017)

[26] Peng, S., Sugiyama, K., Mine, T.: SVD-GCN: A simplified graph convolution paradigm for recommendation. In: CIKM, pp. 1625–1634, ACM (2022)

[27] Rahman, T.A., Surma, B., Backes, M., Zhang, Y.: Fairwalk: Towards fair graph embedding. In: IJCAI, pp. 3289–3295, ijcai.org (2019)

[28] Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: UAI, pp. 452–461, AUAI Press (2009)

[29] Rendle, S., Krichene, W., Zhang, L., Anderson, J.R.: Neural collaborative filtering vs. matrix factorization revisited. In: RecSys, pp. 240–248, ACM (2020)

[30] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. In: CSCW, pp. 175–186, ACM (1994)

[31] Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW, pp. 285–295, ACM (2001)

[32] Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Recommender Systems Handbook, pp. 257–297, Springer (2011)

[33] Shen, Y., Wu, Y., Zhang, Y., Shan, C., Zhang, J., Letaief, K.B., Li, D.: How powerful is graph convolution for recommendation? In: CIKM, pp. 1619–1629, ACM (2021)

[34] Sun, J., Cheng, Z., Zuberi, S., Pérez, F., Volkovs, M.: HGCF: hyperbolic graph convolution networks for collaborative filtering. In: WWW, pp. 593–601, ACM / IW3C2 (2021)

[35] Sun, J., Guo, W., Zhang, D., Zhang, Y., Regol, F., Hu, Y., Guo, H., Tang, R., Yuan, H., He, X., Coates, M.: A framework for recommending accurate and diverse items using bayesian graph convolutional neural networks. In: KDD, pp. 2030–2039, ACM (2020)

[36] Tao, Z., Wei, Y., Wang, X., He, X., Huang, X., Chua, T.: MGAT: multimodal graph attention network for recommendation. Inf. Process. Manag. **57**(5), 102277 (2020)

[37] Vargas, S.: Novelty and diversity enhancement and evaluation in recommender systems and information retrieval. In: SIGIR, p. 1281, ACM (2014)

[38] Vargas, S., Castells, P.: Rank and relevance in novelty and diversity metrics for recommender systems. In: RecSys, pp. 109–116, ACM (2011)

[39] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: ICLR (Poster), OpenReview.net (2018)

[40] Voit, M.M., Paulheim, H.: Bias in knowledge graphs - an empirical study with movie recommendation and different language editions of dbpedia. In: LDK, OASIcs, vol. 93, pp. 14:1–14:13, Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021)

[41] Wang, N., Lin, L., Li, J., Wang, H.: Unbiased graph embedding with biased graph observations. In: WWW, pp. 1423–1433, ACM (2022)

[42] Wang, X., He, X., Cao, Y., Liu, M., Chua, T.: KGAT: knowledge graph attention network for recommendation. In: KDD, pp. 950–958, ACM (2019)

[43] Wang, X., He, X., Wang, M., Feng, F., Chua, T.: Neural graph collaborative filtering. In: SIGIR, pp. 165–174, ACM (2019)

[44] Wang, X., Jin, H., Zhang, A., He, X., Xu, T., Chua, T.: Disentangled graph collaborative filtering. In: SIGIR, pp. 1001–1010, ACM (2020)

[45] Wang, Y., Tang, S., Lei, Y., Song, W., Wang, S., Zhang, M.: Disenhan: Disentangled heterogeneous graph attention network for recommendation. In: CIKM, pp. 1605–1614, ACM (2020)

[46] Wu, J., Shi, W., Cao, X., Chen, J., Lei, W., Zhang, F., Wu, W., He, X.: Disenkgat: Knowledge graph embedding with disentangled graph attention network. In: CIKM, pp. 2140–2149, ACM (2021)

[47] Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J., Xie, X.: Self-supervised graph learning for recommendation. In: SIGIR, pp. 726–735, ACM (2021)

[48] Wu, L., Chen, L., Shao, P., Hong, R., Wang, X., Wang, M.: Learning fair representations for recommendation: A graph-based perspective. In: WWW, pp. 2198–2208, ACM / IW3C2 (2021)

[49] Wu, Y., DuBois, C., Zheng, A.X., Ester, M.: Collaborative denoising auto-encoders for top-n recommender systems. In: WSDM, pp. 153–162, ACM (2016)

[50] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: KDD, pp. 974–983, ACM (2018)

[51] Zhao, M., Wu, L., Liang, Y., Chen, L., Zhang, J., Deng, Q., Wang, K., Shen, X., Lv, T., Wu, R.: Investigating accuracy-novelty performance for graph-based collaborative filtering. In: SIGIR, pp. 50–59, ACM (2022)

[52] Zheng, Y., Gao, C., Chen, L., Jin, D., Li, Y.: DGCN: diversified recommendation with graph convolutional networks. In: WWW, pp. 401–412, ACM / IW3C2 (2021)

[53] Zhou, K., Huang, X., Li, Y., Zha, D., Chen, R., Hu, X.: Towards deeper graph neural networks with differentiable group normalization. In: NeurIPS (2020)

[54] Zhu, J., Dai, Q., Su, L., Ma, R., Liu, J., Cai, G., Xiao, X., Zhang, R.: BARS: towards open benchmarking for recommender systems. In: SIGIR, pp. 2912–2923, ACM (2022)