Challenging the Myth of Graph Collaborative Filtering: a Reasoned and Reproducibility-driven Analysis

Vito Walter Anelli*
vitowalter.anelli@poliba.it
Politecnico di Bari
Italy

Daniele Malitesta*
daniele.malitesta@poliba.it
Politecnico di Bari
Italy

Claudio Pomo*
claudio.pomo@poliba.it
Politecnico di Bari, Italy
Italy

Alejandro Bellogín
alejandro.bellogin@uam.es
Universidad Autónoma de Madrid
Spain

Eugenio Di Sciascio
eugenio.disciascio@poliba.it
Politecnico di Bari
Italy

Tommaso Di Noia
tommaso.dinoia@poliba.it
Politecnico di Bari
Italy

ABSTRACT

The success of graph neural network-based models (GNNs) has significantly advanced recommender systems by effectively modeling users and items as a bipartite, undirected graph. However, many original graph-based works often adopt results from baseline papers without verifying their validity for the specific configuration under analysis. Our work addresses this issue by focusing on the replicability of results. We present a code that successfully replicates results from six popular and recent graph recommendation models (NGCF, DGCF, LightGCN, SGL, UltraGCN, and GFCF) on three common benchmark datasets (Gowalla, Yelp 2018, and Amazon Book). Additionally, we compare these graph models with traditional collaborative filtering models that historically performed well in offline evaluations. Furthermore, we extend our study to two new datasets (Allrecipes and BookCrossing) that lack established setups in existing literature. As the performance on these datasets differs from the previous benchmarks, we analyze the impact of specific dataset characteristics on recommendation accuracy. By investigating the information flow from users' neighborhoods, we aim to identify which models are influenced by intrinsic features in the dataset structure. The code to reproduce our experiments is available at: https://github.com/sisinflab/Graph-RSs-Reproducibility.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommendation, Graph Collaborative Filtering, Reproducibility

ACM Reference Format:
Vito Walter Anelli, Daniele Malitesta, Claudio Pomo, Alejandro Bellogín, Eugenio Di Sciascio, and Tommaso Di Noia. 2023. Challenging the Myth of Graph Collaborative Filtering: a Reasoned and Reproducibility-driven Analysis. In *Seventeenth ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3604915.3609489

1 INTRODUCTION AND RELATED WORK

The world of recommender systems (RSs) is experiencing a revolutionary shift, thanks to the emergence of graph neural network-based models [9, 26, 58] (GNNs). These groundbreaking models are designed to represent users and items as a bipartite, undirected graph, unlocking a whole new level of high-order relationships that were previously almost unattainable. Not only they do achieve better accuracy than their predecessors, but they are also setting a new standard for modern recommender systems [20, 28, 47, 79]. In recent years, great effort has been devoted in creating GNN-based models that address the critical issues of existing models, such as the over-smoothing phenomenon [12] and scalability issues [87]. These cutting-edge models are taking the world of recommender systems by storm and ushering in a new era of accuracy [41, 47, 51, 59, 81]. Over the past ten years, the application of neural techniques rooted in graph representation learning, such as graph convolutional networks [35] (GCNs), has introduced a fresh perspective on traditional collaborative filtering (CF) approaches. Rather than relying solely on user-item interactions for optimization [29, 36, 55], GCN-based methods enable the extraction of both short- and long-distance user preferences toward items [71]. By incorporating multi-hop relationships into the embeddings of users and items, these learned profiles yield more precise recommendations, as evidenced in the literature [28, 47]. Nevertheless, more researchers obtained different accuracy outcomes in independent experiments and began questioning the graph collaborative filtering (graph CF) prominence [96].

The original GCN layer employs message-passing techniques to refine the node representations of users and items through the iterative aggregation of their respective multi-hop neighbor nodes. While early attempts focused on simple aggregation methods [68, 87], recent solutions have advanced the field by exploring the inter-dependencies between nodes and their neighbors [71], designing simplified versions of the graph convolutional layer [14, 28] and learning multiple nodes' views [78, 89] augmented via self-supervised and contrastive learning to improve model accuracy.

[*] Corresponding authors: Vito Walter Anelli (vitowalter.anelli@poliba.it), Daniele Malitesta (daniele.malitesta@poliba.it), and Claudio Pomo (claudio.pomo@poliba.it).

RecSys '23, September 18–22, 2023, Singapore, Singapore

ACM ISBN 979-8-4007-0241-9/23/09.
https://doi.org/10.1145/3604915.3609489

Moreover, current trends aim to simplify message-passing formulations [47, 51, 59], explore other spaces for graph-based recommendation tasks [59, 62, 92], and use hypergraphs to capture complex user-item dependencies [76, 82]. To filter out noisy neighbors and uncover hidden preference patterns, a complementary research field emerged that focuses on learning importance weights through attention mechanisms, such as those employed in the graph attention network [69] (GAT). While some models aim to recognize meaningful user-item interactions at a higher level [67, 70], others disentangle relations on a finer-grained scale [73, 92]. The recent advancements in GCN-based techniques have opened up new avenues for more accurate and effective recommendation systems.

Reproducibility is the cutting-edge research task in which researchers replicate experimental results using the same data and methods [7, 15, 16, 65]. In the case of graph CF, several factors contribute to the lack of reproducibility. Firstly, many graph CF studies copy previous results found in the literature for the same datasets, which makes it challenging to compare and reproduce results across different studies. Secondly, such studies do not provide the implementation of the adopted baselines, which makes it difficult to assess the effectiveness of different models. Furthermore, graph CF studies frequently do not provide complete information since they do not always share the experimental setups, such as hyper-parameter settings and training procedures. This lack of transparency makes it challenging to reproduce results and verify the validity of the findings. The lack of reproducibility in graph CF is a significant issue because it undermines the research's credibility and hinders the field's progress. To address this problem, researchers should strive to provide more detailed descriptions of their experimental setups and make their code and datasets publicly available. Additionally, the research community should work together to establish standard evaluation metrics and experimental protocols to promote reproducibility and facilitate comparison across different studies.

To this aim, this work reports on a notable reproducibility effort to re-implement and replicate the results of six state-of-the-art (both well-established and recent) papers on graph collaborative filtering, namely, NGCF [71], DGCF [73], LightGCN [28], SGL [78], UltraGCN [47], and GFCF [59]. In particular, we provide an in-depth experimental analysis of the papers, conducting the experiments from scratch on the three datasets adopted in the original papers: Gowalla [37], Yelp 2018 [28], and Amazon Book [27]. Notably, the investigation extends the previous works by incorporating state-of-the-art classical collaborative filtering baselines such as UserkNN [56], ItemkNN [57], RP$^3\beta$ [49], and EASE^R [61] to correctly position the graph CF methods in the recommender systems state-of-the-art.

The study's findings reveal that RP$^3\beta$ ranks as the second-best method with the Yelp 2018 dataset, indicating that the original papers would have needed a more comprehensive evaluation. To this end, the evaluation benchmark incorporates two additional datasets, Allrecipes [21] and BookCrossing [97], which are common in the recommendation literature but uncommon in the graph CF-specific literature. However, surprisingly, the rankings significantly differ on the Allrecipes dataset, and the mathematical formulation of the graph CF methods is not sufficient to account for these outcomes. This observation leads to further investigation to comprehend the experimental results. Examining the dataset *topological* characteristics shows that the overall number of users and items and the average user and item degree vary from dataset to dataset. This observation may indicate the amount of information transmitted from node to node in the computational graph. According to the mathematical background, the analysis of the results is then threefold, focusing on the impact of (i) the *coldness/warmness* of a user, (ii) the *popularity* of the enjoyed items, and (iii) the *size* of the user *neighborhood* and the coldness/warmness of the neighbors. The users are partitioned in quartiles accordingly, and the experiments are re-evaluated to obtain more fine-grained results that motivate the outcomes for all the considered datasets.

Overall, the study aims to comprehensively answer several research questions, including:

RQ1. Is the state-of-the-art (i.e., the six most important papers) of graph collaborative filtering (graph CF) replicable?
RQ2. How does the state-of-art of graph CF position with respect to classic CF state-of-the-art?
RQ3. How does the state-of-art of graph CF perform on datasets from different domains and with different topological aspects, not commonly adopted for graph CF recommendation?
RQ4. What information (or lack of it) impacts the performance of the graph CF methods across the various datasets?

The following introduces the background and the experiments to answer the outlined research questions. First, in Section 2, we present the background technologies and the reproducibility details to conduct our study. Then, in Section 3, we report the reproducibility results, whose insights are complemented by adding novel classic CF baselines (i.e., Section 4). Furthermore, an investigation upon graph topology sheds light on the discrepancies of the graph CF approaches on two introduced datasets (i.e., Section 5). By reinterpreting the concept of users' node degree as *information flow* from the multi-hop neighborhoods to the user, we unveil the behavior of the graph and classic CF. Finally, Section 6 wraps up the main take-home messages and paves the way to novel directions for future work. Codes and datasets to reproduce our paper are available here: https://github.com/sisinflab/Graph-RSs-Reproducibility.

2 BACKGROUND AND REPRODUCIBILITY ANALYSIS

The current section is aimed to provide the background about selected state-of-the-art methodologies in graph CF and their reproducibility details as presented in the original papers. First, the main aspects about graph-based models are introduced to conduct a chronological analysis of the strategies behind each algorithm (Section 2.1). Then, we assess the experimental settings as reported in the original works by focusing on the chosen baselines (Section 2.2), the datasets involved (Section 2.3), and the training-testing protocol adopted in each case (Section 2.4).

2.1 Graph collaborative filtering

In graph CF, users, items, and their interconnections are viewed as a bipartite and undirected graph. Let $\mathcal{U}$ and $\mathcal{I}$ be the sets of users and items in the recommendation system, respectively. Then, let $\mathbf{R} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$ be the user-item interaction matrix where, in an implicit feedback scenario, $R_{ui} = 1$ if user $u \in \mathcal{U}$ interacted with item $i \in \mathcal{I}$, 0 otherwise. We build the adjacency matrix $\mathbf{A} \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{I}|) \times (|\mathcal{U}|+|\mathcal{I}|)}$ indicating the bi-directional connections linking users and items in $\mathbf{R}$:

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{R} \\ \mathbf{R}^\top & 0 \end{bmatrix}. \tag{1}$$

We use the set of users and items, along with the adjacency matrix, to formally define the user-item bipartite and undirected graph $\mathcal{G} = \{\mathcal{U} \cup \mathcal{I}, \mathbf{A}\}$. By associating users' and items' nodes to embeddings, the vast majority of approaches iteratively update their representations at different hop distances through the message-passing schema [9, 23].

For this work, we select and reproduce the results for six widely-recognized state-of-the-art approaches in graph CF, namely, NGCF [71], DGCF [73], LightGCN [28], SGL [78], UltraGCN [47], and GFCF [59] (refer to Section 3). This selection is motivated by two aspects: (i) such models are adopted as baselines in recent works from top-tier venues (see the second column in Table 1); (ii) their strategies cover a wide spectrum of techniques in graph CF. To provide a chronological overview of such techniques, in the following, we report their main aspects:

- **NGCF.** Neural graph collaborative filtering [71] (NGCF) is among the pioneer approaches in graph CF. Its message-passing schema works by aggregating the neighborhood information and the inter-dependencies among the *ego* and the *neighborhood* nodes (note that a normalized Laplacian adjacency matrix is used during the message-passing).
- **DGCF.** Disentangled graph collaborative filtering [73] (DGCF) assumes that user-item interactions can be disentangled into independent intents, where each stands for a specific aspect describing the user's preference towards the item. The model learns a set of weighted adjacency matrices refining the user-item importance related to a specific intent.
- **LightGCN.** Light graph convolutional network [28] (LightGCN) suggests that a more light-weight formulation of the graph convolutional layer proposed by Kipf and Welling [35] can lead to superior accuracy performance in the recommendation scenario. Specifically, the architecture removes feature transformations and non-linearities.
- **SGL.** Self-supervised graph learning [78] (SGL) is among the first attempts to bring the lesson-learned from self-supervised [31] and contrastive [34] learning to graph CF. Built upon a LightGCN-based convolutional layer, the model learns different views of nodes by performing node/edge dropout and random walk operations on the graph topology. A self-supervised contrastive loss component is added to encourage the consistency among different views of the same node and the divergence among different nodes.
- **UltraGCN.** Ultra simplification of graph convolutional network [47] (UltraGCN) addresses some crucial issues in graph CF. Specifically, the authors propose a novel message-passing schema that mathematically approximates the infinite-layer propagation through a single (simplified) node update iteration. The adjacency matrix is normalized through a modified Laplacian formulation that accounts for the asymmetric weighting of connected nodes in user-user and item-item connections. Moreover, two loss components are introduced to tackle the over-smoothing effect and learn from the usually-unexplored type of node relationships such as item-item.
- **GFCF.** Graph filter-based collaborative filtering [59] (GFCF) questions the role of graph convolutional network into recommendation by leveraging graph signal processing theory. By showing that several existing approaches in CF may fall into one unified framework based upon graph convolution, the authors eventually propose a closed-form algorithm that proves to be a strong baseline against other trainable and computationally-expensive (graph-based) approaches in CF. Thus, the method represents the only exception to the message-passing models presented above.

2.2 Analysis on reported baselines

Table 1 reports on the baselines each graph-based approach was tested against in the original paper. By categorizing them into *classic* and *graph* CF we first observe that, with the only exception of UltraGCN, all graph-based recommendation systems are generally compared only against 1-2 classical CF solutions (MF [29, 55]- and/or VAE [38, 44]-based approaches in most cases). However, the recent literature [2, 3, 15, 16, 96] has raised several concerns about usually-untested strong CF baselines, such as nearest-neighborhood approaches (e.g., UserkNN [56] and ItemkNN [57]), random-walk techniques (e.g., RP$^3\beta$ [49]), and other autoencoder-based solutions (e.g., EASE^R [61]). Differently from the classical CF baselines, we notice that most of the works compare their proposed approaches against a wide (and shared) range of graph CF solutions. This is easily explainable given the conceptual and logical similarities among the graph CF baselines and the proposed approaches. Moreover, besides a limited subset of graph CF baselines (i.e., HOP-Rec [83] and GRMF [53]), the vast majority of tested graph algorithms [14, 43, 64, 68, 87] are based upon the graph convolutional network architecture. Interestingly, we observe that only a subgroup of our selected six graph CF approaches (up to a maximum of three approaches if we consider UltraGCN) is generally compared against the proposed approach. While we could justify this point with chronological motivations (e.g., DGCF could have not been tested on SGL, UltraGCN, and GFCF), we deem this to be an important lack in the existing literature.

Under the above considerations, and differently from the previous works, we compare the accuracy performance of the selected six graph CF approaches against strong CF techniques (UserkNN, ItemkNN, RP$^3\beta$ and EASE^R), while providing a complete evaluation setting which involves all the selected graph methods, where they are put against one another (refer to Section 4). To our knowledge, this work is one of the first attempts [96] to fill this gap.

2.3 Analysis on reported datasets

Table 2 displays the datasets adopted to train and test the reviewed graph-based recommender systems, as reported in the original papers. Notably, we recognize a total of seven recommendation datasets spanning different domains such as social networks (i.e., Gowalla), points-of-interest (i.e., Yelp 2018), e-commerce (i.e., the Amazon product categories and Alibaba-iFashion), and movies (i.e., Movielens 1M). It is worth pointing out that when we set the '✓'

Table 1: **Analysis of baselines used in each of the selected graph-based models, categorized into *classic* and *graph* CF. A colored tick '✓' denotes when one of the baselines is also among the selected set of graph-based approaches for our study.**

Families	Baselines	Models					
		NGCF [71]	DGCF [73]	LightGCN [28]	SGL [78]	UltraGCN [47]	GFCF [59]
		Used as graph CF baseline in (2021 — present)					
		[10, 13, 32, 62, 77, 84]	[19, 39, 46, 74, 75, 92]	[40, 54, 78, 82, 88, 89]	[22, 46, 77, 82, 85, 93]	[17, 24, 42, 48, 95, 96]	[4, 5, 41, 50, 80, 96]
Classic CF	MF-BPR [55]	✓	✓			✓	
	NeuMF [29]	✓					
	CMN [18]	✓					
	MacridVAE [44]		✓				
	Mult-VAE [38]			✓	✓		✓
	DNN+SSL [86]				✓		
	ENMF [11]					✓	
	CML [30]					✓	
	DeepWalk [52]					✓	
	LINE [66]					✓	
	Node2Vec [25]					✓	
	NBPO [91]					✓	
Graph CF	HOP-Rec [83]	✓					
	GC-MC [68]	✓	✓				
	PinSage [87]	✓					
	NGCF [71]		✓	✓	✓	✓	✓
	DisenGCN [43]		✓				
	GRMF [53]			✓			✓
	GRMF-Norm [28]			✓			✓
	NIA-GCN [64]					✓	
	LightGCN [28]				✓	✓	✓
	DGCF					✓	
	LR-GCCF [14]					✓	
	SCF [94]					✓	
	BGCF [63]					✓	
	LCFN [90]					✓	

for the same dataset on different models, we are stating that the authors from the original works used the exact same dataset setting, that is, the original user-item interaction data and splitting/filtering strategies. A deeper analysis shows that there exists a subset of three datasets (i.e., Gowalla [37], Yelp 2018 [28], and Amazon Book [27]) which is utilized in the majority of graph CF works. For the sake of reproducibility, we replicate the original results calculated on such datasets for the six graph CF approaches (although the SGL paper does not provide results on Gowalla). Given the limited set of shared datasets among all the approaches, we include novel, *never-investigated* datasets to assess if their recommendation accuracy remains consistent on other domains and/or topologies.

2.4 Analysis on experimental comparison

As a final analyzed dimension, we discuss the protocol for the experimental comparison between the baselines and the proposed approach in each selected work. Being the pioneer model in the domain, the authors from NGCF train all proposed baselines from scratch. In the DGCF paper, the authors directly report the results of some baselines which are shared with NGCF and train the other baselines from scratch. In a similar manner, the authors by LightGCN, SGL, and UltraGCN copy the result values from the original papers, while the remaining models are trained from scratch. Finally, the authors from GFCF reproduce the results from LightGCN as the baselines are exactly the same.

Regarding the copy-paste of baseline results, authors often justify this approach by stating that they used the same experimental settings (such as dataset splitting/filtering) as their (graph) CF baselines. Additionally, it is worth noting that some authors are shared among the studies being investigated.

To remove all doubts, and differently from the mentioned works, we re-implement all algorithms by carefully following their original codes, and train/evaluate them through Elliot [1, 45]. Our goal is to provide a fair and repeatable experimental environment for the selected graph CF approaches, by using the hyper-parameter settings as indicated in each paper and/or shared online code to assess to

Table 2: Analysis of the datasets adopted in each graph-based approach.

Models	Gowalla	Yelp 2018	Amazon Book	Alibaba-iFashion	Movielens 1M	Amazon Electronics	Amazon CDs
NGCF	✓	✓	✓				
DGCF	✓	✓	✓				
LightGCN	✓	✓	✓				
SGL		✓	✓	✓			
UltraGCN	✓	✓	✓		✓	✓	✓
GFCF	✓	✓	✓				

what extent we can reproduce the original results. The reader may refer to Section 3.1 for a whole description of our settings.

3 REPLICATION OF PRIOR RESULTS (RQ1)

This section focuses on how the replication of the experiments from the six state-of-the-art papers on graph CF stated before has been set up. It starts by defining the evaluation protocol applied to compare these methods in their respective works (Section 3.1). After that, we present our replication results (Section 3.2).

3.1 Settings

The experimental setup adopted in the first part of this study is designed primarily to replicate the results of the models included in this analysis [28, 31, 43, 47, 59, 71]. As mentioned earlier, we use the three most common datasets in this scenario to show the results of our replicability study. Specifically, we use Gowalla, Yelp 2018, and Amazon Book as provided in the public repositories of NGCF[1] and LightGCN[2]. All the proposed models (except SGL) use the same datasets with the same filtering/splitting. The authors state that they adopt a random split based on the 80/20 hold-out (i.e., for each user, 80% of the interactions is used to create the training set, while the remaining 20% constitutes the test set). Thus, each user-item interaction is treated as positive; all others are considered unfavorable. In addition, the authors leave 10% of the training as a validation set for tuning the hyper-parameters. However, this portion of the dataset is not indicated in the papers' extra material.

The adopted evaluation protocol, all-unrated-item [60], is shared across all the analyzed papers: for each user, we retain all candidate items with whom she does not interact with in the training set. The quality of recommendations is assessed by the Recall and the nDCG on the top-20 recommendation lists for each user. Each work performs its own tuning of the hyper-parameters (the Recall@20 is used as validation metric), by reporting on the search hyper-parameter spaces. Moreover, the best configurations on each dataset are usually provided in the respective papers and/or repositories. Thus, we set the hyper-parameters on each model-dataset as the best ones declared by the authors. The careful reader would notice that the results reported in Table 3 for NGCF (see the '**Original**' column) differ from those shown in the in-proceedings version [71]. The reason is that the authors modified and recalculated the results obtained for the model and baselines due to errors in the pre-processing of the Yelp 2018 dataset and in the calculation of the nDCG. Thus, for the sake of fair reproducibility, and only in this case, we consider the results reported in the arXiv (most updated) version of the paper [72].

3.2 Results

Table 3 compares the results reported by the six papers focused on our study with those obtained in our implementation (using the tuned parameters specified in each work, as explained before). The new experiments closely approximate the original ones, with the most significant performance shift being in the 10^{-3} order. There are no noticeable distinctions in metrics, dataset, or algorithm used.

More specifically, in an algorithm basis, we observe the performance of GFCF is the best replicated one. This might be due to this method being the only one with a closed-form algorithm, hence, no perturbations from random initializations are expected. The rest of the approaches evidence a similar (high) level of replication, although the shift for NGCF and DGCF rarely achieves the 10^{-4} order for the two metrics in all the datasets. In any case, considering the random initializations and stochastic learning processes [33], our replication of these approaches could be considered a success.

No significant differences were found among the three datasets. SGL was not originally reported for Gowalla, so it was omitted from the table as we compared reported results with our implementations using the same hyper-parameters.

In summary, these results confirm that, as discussed before, even though authors of these papers re-used the performance values from other papers just by copy-pasting them, this did not hurt the reproducibility of these approaches. As previously stated, our assumption for this behavior (which is not a safe practice in general [7]) is that the experiments of the original papers were all comparable because some authors are shared across contributions, which should guarantee that the settings and implementations of the algorithms are the same.

4 BENCHMARKING GRAPH CF APPROACHES USING ALTERNATIVE BASELINES (RQ2)

In line with recent reproducibility works (such as [15]) that evidenced certain problems regarding the choice and optimization of the baselines used for comparison, in this section we assess how graph CF approaches perform relatively to classical CF baselines. Here, we specify first how the experiments are prepared (Section 4.1), and the corresponding results are shown (Section 4.2).

[1] https://github.com/xiangwang1223/neural_graph_collaborative_filtering.
[2] https://github.com/kuandeng/LightGCN.

Table 3: Results of our replicability study on Gowalla, Yelp 2018, and Amazon Book for the selected state-of-the-art graph-based recommender systems. We calculate the performance shift between our conducted experiments and the original ones (as reported in their papers). Note that models have been sorted out according to the chronological order.

Datasets	Models	Ours		Original		Performance Shift	
		Recall	nDCG	Recall	nDCG	Recall	nDCG
Gowalla	NGCF	0.1556	0.1320	0.1569	0.1327	$-1.3 \cdot 10^{-03}$	$-7 \cdot 10^{-04}$
	DGCF	0.1736	0.1477	0.1794	0.1521	$-5.8 \cdot 10^{-03}$	$-4.4 \cdot 10^{-03}$
	LightGCN	0.1826	0.1545	0.1830	0.1554	$-4 \cdot 10^{-04}$	$-9 \cdot 10^{-04}$
	SGL*	—	—	—	—	—	—
	UltraGCN	0.1863	0.1580	0.1862	0.1580	$+1 \cdot 10^{-04}$	0
	GFCF	0.1849	0.1518	0.1849	0.1518	0	0
Yelp 2018	NGCF	0.0556	0.0452	0.0579	0.0477	$-2.3 \cdot 10^{-03}$	$-2.5 \cdot 10^{-03}$
	DGCF	0.0621	0.0505	0.0640	0.0522	$-1.9 \cdot 10^{-03}$	$-1.7 \cdot 10^{-03}$
	LightGCN	0.0629	0.0516	0.0649	0.0530	$-2 \cdot 10^{-03}$	$-1.4 \cdot 10^{-03}$
	SGL	0.0669	0.0552	0.0675	0.0555	$-6 \cdot 10^{-04}$	$-3 \cdot 10^{-04}$
	UltraGCN	0.0672	0.0553	0.0683	0.0561	$-1.1 \cdot 10^{-03}$	$-8 \cdot 10^{-04}$
	GFCF	0.0697	0.0571	0.0697	0.0571	0	0
Amazon Book	NGCF	0.0319	0.0246	0.0337	0.0261	$-1.8 \cdot 10^{-03}$	$-1.5 \cdot 10^{-03}$
	DGCF	0.0384	0.0295	0.0399	0.0308	$-1.5 \cdot 10^{-03}$	$-1.3 \cdot 10^{-03}$
	LightGCN	0.0419	0.0323	0.0411	0.0315	$+8 \cdot 10^{-04}$	$+8 \cdot 10^{-04}$
	SGL	0.0474	0.0372	0.0478	0.0379	$-4 \cdot 10^{-04}$	$-7 \cdot 10^{-04}$
	UltraGCN	0.0688	0.0561	0.0681	0.0556	$+7 \cdot 10^{-04}$	$+5 \cdot 10^{-04}$
	GFCF	0.0710	0.0584	0.0710	0.0584	0	0

**Results are not provided since SGL was not originally trained and tested on Gowalla [78].*

4.1 Settings

We expand our investigation by examining four *classic* CF models to enhance the replicability analysis. Specifically, we select four models whose accuracy performance has rarely been compared with the *graph-based* CF approaches replicated in this study. The decision to include UserkNN, ItemkNN, RP$^3\beta$, and EASE^R is purposeful. We refer to [16] and (more recently) [2], which demonstrated the competitiveness of these baseline models compared to more recent approaches when a shared benchmark for comparison is employed among all involved methodologies. Furthermore, we also consider two unpersonalized approaches (i.e., MostPop and Random). The two models act as benchmarks to assess the effectiveness of customized methods compared to a user-agnostic solution.

For a fair comparison, the configuration delineated herein elucidates how the four *classic* CF models are tuned following the exact same training/test splitting reported in Section 3 and the same experimental protocol. The only difference is that (for obvious reasons) we need to explore the hyper-parameters of each classic CF model introduced in the comparison. Similarly to what the authors do in the original graph CF works, we retain the 10% of the training to generate a validation set, but decide to explore 20 distinct configurations for each model through the state-of-the-art Tree-structured Parzen Estimator (TPE) hyper-parameter search [8]. For every model, the final results correspond to the accuracy measure on the test set by setting the hyper-parameter configuration providing the best Recall@20 results on the validation set.

4.2 Results

Table 4 shows the results of the graph CF models (as previously replicated in Table 3) with the additional baselines. First, it is worth noting that, even though none of these baselines gets the best results in any of the three datasets considered, they achieve the second-best performance in Yelp 2018 (refer to RP$^3\beta$ with nDCG).

Second, none of the models in the reference family achieve competitive performance. While this is expected for the Random algorithm, it is an indication that either none of these datasets evidence a strong popularity bias or (considering the way they were processed) such bias was removed.

Third, some of the classic CF approaches (such as RP$^3\beta$ and UserkNN in Gowalla, and RP$^3\beta$ and EASE^R in Yelp 2018) demonstrate better performance than some of the state-of-the-art graph CF methods, in particular, they perform better than NGCF, DGCF, and LightGCN. This result is in line with recent experimental comparisons [3, 6, 15] where these baselines outperform other methods based on matrix factorization or neural networks. Moreover, to some extent, the fact that some graph CF methods are outperformed should not be surprising, since, as shown in Table 1, none of these baselines were included in the original papers where the graph CF approaches were proposed.

5 EXTENDING THE EXPERIMENTAL COMPARISON TO NEW DATASETS (RQ3 — RQ4)

This section aims to provide a full picture from an experimental point of view on two new datasets: Allrecipes and BookCrossing. First, Section 5.1 introduces the experimental settings followed to obtain the results presented in Section 5.2. Then, Section 5.3 discusses these results in more detail, aiming to explain the insights derived from them.

5.1 Settings

Motivated by the previous results, we further enrich our analysis by investigating the behavior of all tested models on two datasets that have never been considered in any previous study involving graph-based approaches for recommendation, namely, Allrecipes [21]

Table 4: Graph-based CF solutions tested against unpersonalized (i.e., reference) and classical CF approaches on Gowalla, Yelp 2018, and Amazon Book. While results for the graph-based approaches have been directly reported from our reproducibility study (see above), classical CF recommender systems have been fine-tuned on the two datasets to find their best configurations. Boldface and underline refer to best and second-to-best values, respectively.

Families	Models	Gowalla		Yelp 2018		Amazon Book	
		Recall	nDCG	Recall	nDCG	Recall	nDCG
Reference	MostPop	0.0416	0.0316	0.0125	0.0101	0.0051	0.0044
	Random	0.0005	0.0003	0.0005	0.0004	0.0002	0.0002
Classic CF	UserkNN	0.1685	0.1370	0.0630	0.0528	0.0582	0.0477
	ItemkNN	0.1409	0.1165	0.0610	0.0507	0.0634	0.0524
	RP$^3\beta$	0.1829	0.1520	0.0671	0.0559	0.0683	0.0565
	EASE^R *	0.1661	0.1384	0.0655	0.0552	**0.0710**	0.0567
Graph CF	NGCF	0.1556	0.1320	0.0556	0.0452	0.0319	0.0246
	DGCF	0.1736	0.1477	0.0621	0.0505	0.0384	0.0295
	LightGCN	0.1826	0.1545	0.0629	0.0516	0.0419	0.0323
	SGL	—	—	0.0669	0.0552	0.0474	0.0372
	UltraGCN	**0.1863**	**0.1580**	0.0672	0.0553	0.0688	0.0561
	GFCF	0.1849	0.1518	**0.0697**	**0.0571**	**0.0710**	**0.0584**

*Results for EASE^R on Amazon Book are taken from BARS Benchmark [96].

and BookCrossing [97]. Table 5 shows some statistics of these datasets, where we purposely decide to report both the benchmarking datasets for graph CF (i.e., Gowalla, Yelp 2018, and Amazon Book) and the newly introduced ones. On the one hand, Allrecipes exhibits quite discordant characteristics compared to the other datasets. Although it has a comparable density, users are more numerous than items, with a much lower average user and item node degrees compared to the other standard graph CF datasets. On the other hand, BookCrossing displays the lowest ratio between the number of users to items across all datasets, and a much higher density than all the others. In summary, the newly introduced datasets serve as a foundation to assess the performance in different (and never-explored) *topological* settings for graph CF baselines.

To adhere to the experimental setup presented so far, we adopt the all-unrated-item evaluation protocol, and split the two datasets with a random hold-out solution, ensuring an 80:20 proportion. Differently from the replicability study, we now perform a TPE-based hyper-parameter tuning for *all* models, as the best hyper-parameters for each graph-based approach is not known in advance; for this, we (again) use the 10% portion of the training set as validation set. We run 20 different settings within the search space provided in the original papers. The models' best configurations are selected through the Recall@20 on the validation.

5.2 Results

Table 6 provides a full comparison between unpersonalized methods, classical CF approaches, and the graph CF methods under analysis. In line with our previous section experiments, classic CF methods (in particular, RP$^3\beta$ and EASE^R) are very competitive compared to graph CF approaches, even in novel datasets like the ones included in this analysis. Specifically, the results in BookCrossing are dominated by these baselines, while in Allrecipes, the MostPop stands out, evidencing a strong popularity bias.

Table 5: Statistics calculated on the training sets of Gowalla, Yelp 2018, Amazon Book, Allrecipes, and BookCrossing. We indicate the number of user-item interactions through' Edges' while 'Avg. Deg. (U)' and 'Avg. Deg. (I)' refer to users' and items' average node degree (i.e., average interaction number).

Statistics	Gowalla	Yelp 2018	Amazon Book	Allrecipes	BookCrossing
Users	29,858	31,668	52,643	10,084	6,754
Items	40,981	38,048	91,599	8,407	13,670
Edges	810,128	1,237,259	2,380,730	80,540	234,762
Density	0.0007	0.0010	0.0005	0.0010	0.0025
Avg. Deg. (U)	27.1327	39.0697	45.2241	7.9869	34.7590
Avg. Deg. (I)	19.7684	32.5184	25.9908	9.5801	17.1735

Table 6: Graph-based CF solutions tested against unpersonalized (i.e., reference) and classical CF approaches on Allrecipes and BookCrossing. Boldface and underline refer to best and second-to-best values, respectively.

Families	Models	Allrecipes		BookCrossing	
		Recall	nDCG	Recall	nDCG
Reference	MostPop	0.0472	0.0242	0.0352	0.0319
	Random	0.0024	0.0010	0.0013	0.0011
Classic CF	UserkNN	0.0339	0.0188	0.0871	0.0769
	ItemkNN	0.0326	0.0180	0.0779	0.0739
	RP$^3\beta$	0.0170	0.0089	**0.0941**	0.0821
	EASE^R	0.0351	0.0192	0.0925	**0.0847**
Graph CF	NGCF	0.0291	0.0144	0.0670	0.0546
	DGCF	0.0448	0.0234	0.0643	0.0543
	LightGCN	0.0459	0.0236	0.0803	0.0660
	SGL	0.0365	0.0192	0.0716	0.0600
	UltraGCN	**0.0475**	**0.0248**	0.0800	0.0651
	GFCF	0.0101	0.0051	0.0819	0.0712

These results highlight that, among the graph CF techniques, those that maintain their performance in novel domains are UltraGCN (best one in Allrecipes and third among its type) and LightGCN (second best in both domains). While the nature of these two datasets is clearly different (as shown in Table 2, Allrecipes is smaller and it contains more users than items, instead of the other way around as in BookCrossing), the relative performance of the best graph CF methods is competitive. However, for some of them, the performance drop is significant, reaching an accuracy lower than that of any other classic CF baseline.

To bring light into some of these behaviors, the next section discusses in more detail how the ranking of the graph CF methods changes depending on the dataset, and hypothesize which dataset characteristics may be tied to these effects.

5.3 Discussion

To further validate and explain the reasons behind the results reported in Table 6, in the following we perform a twofold analysis. First, we rank all the selected graph-based recommendation models on all the tested datasets to assess their relative improvement across all settings and provide another perspective on the results from Table 6. Then, we propose a more nuanced study on the measured accuracy performance by investigating its (possible) dependence

Table 7: Graph-based recommender systems, ranked according to their Recall@20 and nDCG@20 on all the tested datasets. For each model, we also report its relative improvement with respect to the worst-performing approach on the same dataset (in green).

Metric		Gowalla	Yelp 2018	Amazon Book	Allrecipes	BookCrossing
Recall	**1.**	UltraGCN *(+19.73%)*	GFCF *(+25.36%)*	GFCF *(+122.57%)*	UltraGCN *(+370.30%)*	GFCF *(+27.37%)*
	2.	GFCF *(+18.83%)*	UltraGCN *(+20.86%)*	UltraGCN *(+115.67%)*	LightGCN *(+354.46%)*	LightGCN *(+24.88%)*
	3.	LightGCN *(+17.35%)*	SGL *(+20.32%)*	SGL *(+48.59%)*	DGCF *(+343.56%)*	UltraGCN *(+24.42%)*
	4.	DGCF *(+11.57%)*	LightGCN *(+13.13%)*	LightGCN *(+31.35%)*	SGL *(+261.39%)*	SGL *(+11.35%)*
	5.	NGCF (−)	DGCF *(+11.69%)*	DGCF *(+20.38%)*	NGCF *(+188.12%)*	NGCF *(+4.20%)*
	6.	*SGL** (−)	NGCF (−)	NGCF (−)	GFCF (−)	DGCF (−)
nDCG	**1.**	UltraGCN *(+19.70%)*	GFCF *(+26.33%)*	GFCF *(+137.40%)*	UltraGCN *(+386.27%)*	GFCF *(+31.12%)*
	2.	LightGCN *(+17.05%)*	UltraGCN *(+22.35%)*	UltraGCN *(+128.05%)*	LightGCN *(+362.75%)*	LightGCN *(+21.55%)*
	3.	GFCF *(+15.00%)*	SGL *(+22.12%)*	SGL *(+51.22%)*	DGCF *(+358.82%)*	UltraGCN *(+19.89%)*
	4.	DGCF *(+11.89%)*	LightGCN *(+14.16%)*	LightGCN *(+31.30%)*	SGL *(+276.47%)*	SGL *(+10.50%)*
	5.	NGCF (−)	DGCF *(+11.73%)*	DGCF *(+19.92%)*	NGCF *(+182.35%)*	NGCF *(+0.55%)*
	6.	*SGL** (−)	NGCF (−)	NGCF (−)	GFCF (−)	DGCF (−)

**SGL is not classifiable on the Gowalla dataset as results were not calculated in the original paper [78].*

on the specific dataset characteristics, namely, the node degree as viewed at multiple hops.

5.3.1 Graph-based models' ranking. In Table 7, we rank the six graph CF recommender systems under analysis according to the calculated Recall@20 and nDCG@20, for both the original datasets (i.e., Gowalla, Yelp 2018, and Amazon Book) and the novel datasets we introduced (i.e., Allrecipes and BookCrossing). Moreover, we also indicate the relative improvement of each model with respect to the worst-performing algorithm on that dataset.

The trend on the three original datasets is quite steady, with UltraGCN and GFCF being the two best-performing approaches in almost all cases, and the remaining graph techniques ranked as in descending chronological order (confirming the findings from the recent literature). In terms of relative improvements, we observe large performance differences mainly on the Amazon Book setting.

By focusing on the two additional datasets (i.e., Allrecipes and BookCrossing), the rankings corroborate some of the previous outcomes, but also introduce novel and unexpected considerations. While UltraGCN seems to preserve its role of leading approach in the two scenarios (in BookCrossing it is ranked as third but with minimum margin to the second one), we notice how GFCF's performance is very fluctuating, as it even stands in the last position on Allrecipes with large performance difference to the other models (the same goes for DGCF). Noticeably, LightGCN gets up to the top of the ranking in both settings, indicating that a careful hyper-parameter tuning could be beneficial to outperform most of the other approaches, even the ones that should surpass it according to the literature (such as SGL). As final remarks, NGCF poor performance is again confirmed in such different dataset settings.

5.3.2 Analysis on the node degree. As already observed in Table 5, the average node degree of users and items represents one of the main aspects discerning each dataset from the other ones. For this reason, we decide to reason about its possible influence on the models' performance. In this respect, instead of limiting our analysis to the sole definition of node degree (i.e., number of recorded interactions for each user and item), and given the ability of graph-based approaches to distill the collaborative signal by stacking multiple layers [71], we propose a novel investigation which reinterprets the node degree as *information flow* from neighbor nodes to the user nodes after multiple hops. Note that we only consider users as the ending nodes of such a flow because we are interested in assessing how the accuracy recommendation measures (which are generally calculated user-wise) may be influenced by this aspect.

Before diving into the results and discussion, we provide some useful intuitions and formulations which may help understand our analysis. With reference to Figure 1, we introduce the definition of information flow at one, two, and three hops. We decide to limit our focus on the first three explored hops because (i) graph-based recommender systems built upon the message-passing schema usually tend not to iterate over the third aggregation layer, and (ii) the investigation of more than three hops would not be meaningful from a recommendation perspective. As a matter of fact, we interpret each of the three hops as follows:

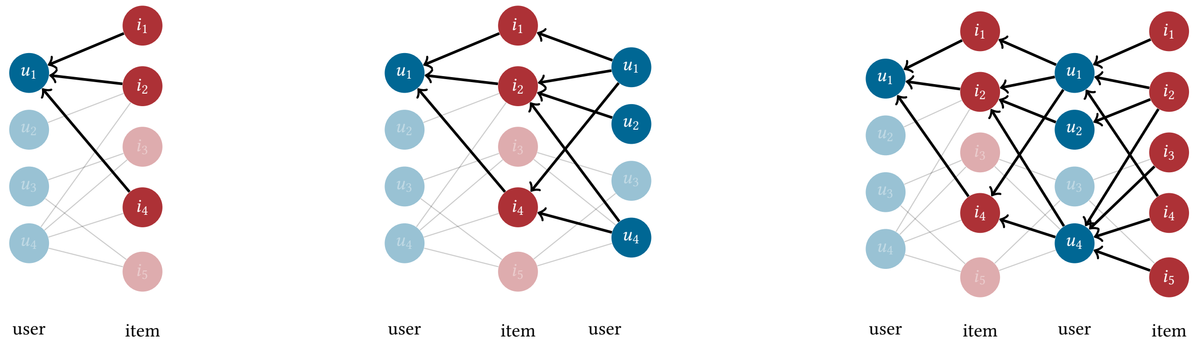
- at **one** hop (Figure 1a), users receive the information coming from the items they interacted with; in other words, this is an indication of the *activeness* of users on the platform;
- at **two** hops (Figure 1b), users receive the information of the other users co-interacting with the same items; in other words, this is an indication of the influence of *items' popularity* on users;
- at **three** hops (Figure 1c), users receive the information coming from the items interacted by the other users involved in co-interactions; i.e., this is an indication of the influence of *co-interacting* users' *activeness* on users.

Let us formalize such definitions. The information received by users at one, two, and three hops is calculated as:

$$\Upsilon_{\mathcal{U}}^{(1)} = \mathbf{R}\mathbf{1}_{\mathcal{I}}, \qquad \Upsilon_{\mathcal{U}}^{(2)} = (\mathbf{R} \odot (\mathbf{1}_{\mathcal{U}}\mathbf{R}))\mathbf{1}_{\mathcal{I}}, \qquad \Upsilon_{\mathcal{U}}^{(3)} = (\mathbf{R}\mathbf{R}^{\top} \odot \mathbf{R}\mathbf{1}_{\mathcal{I}})\mathbf{1}_{\mathcal{I}}, \quad (2)$$

where $\Upsilon_{\mathcal{U}}^{(h)} \in \mathbb{R}^{|\mathcal{U}| \times 1}$ is the vector of the information that all users receive from the nodes in their h-hop, $\mathbf{1}_{\mathcal{U}} \in \mathbb{R}^{1 \times |\mathcal{U}|}$ and $\mathbf{1}_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}| \times 1}$ are row and column vectors with 1 repeated $|\mathcal{U}|$ and $|\mathcal{I}|$ times, respectively, while $\odot$ is the Hadamard product performed in broadcast.

In light of the above, the study assesses the accuracy performance of graph-based recommender systems on user groups considering the information received from the one, two, and three hops neighborhood. Following other analyses in the literature, we decide to split users into quartiles according to the information values (i.e., $\Upsilon_{\mathcal{U}}^{(h)}$). Thus, we consider four groups: (i) users whose values are



(a) 1-hop **(b) 2-hop** **(c) 3-hop**

Figure 1: A toy user-item graph where the ego user node (highlighted) receives the *information flow* from the (a) 1-, (b) 2-, and (c) 3-hop neighbor nodes (highlighted). Arrows' direction is a visual representation of the information flow.

below the 25% of the distribution, (ii) users whose values are above the 25% and below the 50% of the distribution, (iii) users whose values are above the 50% and below the 75% of the distribution, and (iv) users whose values are above the 75% of the distribution.

Figure 2 displays the percentage variation in accuracy performance (measured by nDCG) across quartiles relative to the average value reported in Table 6. The figure illustrates how the quality of recommendation performance fluctuates amongst different clusters of users. For example, a method indicating a 50% improvement in the fourth quartile would suggest that users in this cluster, typically more active (1-hop) or also interested in popular items (2-hop), receive more accurate recommendations with respect to the average user. This observation implies that a non-discriminatory recommendation system should produce no variation across quartiles, with values overlapping the 0% dashed line. The second necessary preliminary to understand the outcome of the experiments is the interpretation of the quartiles for the different hops. In the 1-hop, the fourth quartile pertains to warm users interacting most with the platform, while the first quartile represents cold users interacting less frequently. In the 2-hop, high values in the fourth quartile indicate active users who enjoy popular items, resulting in dense subgraphs. The first quartile, in contrast, consists of less active users interacting with niche items in less dense subgraphs. The 3-hop, which includes user neighbors, generates the highest values when active users interact with popular items enjoyed by warm users (i.e., their neighbors). However, it is essential to note that the plots offer no insight into overall accuracy (which is in Table 6).

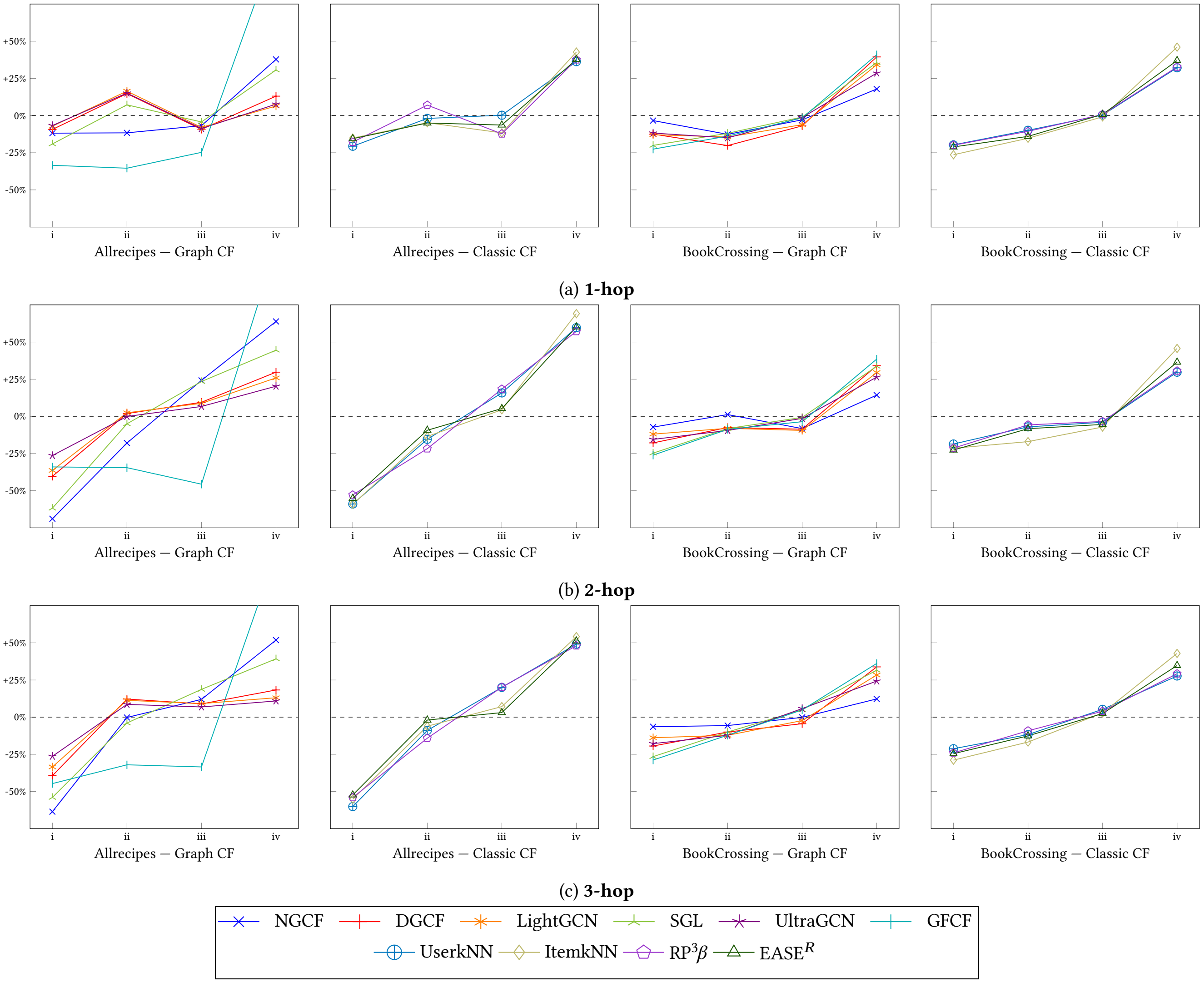
When considering the recommendation performance according to the corresponding cluster (depicted in Figure 2), it is crucial to note that none of them demonstrate ideal recommendation behavior. Instead, these systems tend to favor warm users or densely interconnected subgraphs located in the fourth quartile. Despite this trend, the 1-hop plots for graph Collaborative Filtering (CF) and classic CF methods in Allrecipes and BookCrossing graphs demonstrate minimal disparities between different recommendation approaches. Even though they all favor the fourth quartile over the first one, the coldness/warmness of a user marginally impacts how much the method is biased toward these types of users. The lone exception to this trend is GFCF, which exhibits even greater penalization towards the first three quartiles (varying on the three hops from, approximately, -45% to +115%, and thus exceeding the plots' upper bound). As such, this system only provides satisfactory recommendation performance for users in the fourth quartile.

Regarding the 2-hop, there are several interesting insights to be gained. Firstly, the recommendation methods exhibit a higher overall slope, favoring the users who enjoyed popular items over the cold users who enjoyed niche items. While this may seem like an obvious observation, the plot confirms that user coldness/warmness alone is not a sufficient indicator of high-quality recommendations. Instead, the 2-hop reveals that combining user coldness/warmness and item popularity is useful for identifying such users. A second noteworthy aspect is that the Allrecipes dataset highlights three distinct behaviors among the graph CF methods. UltraGCN, DGCF, and LightGCN exhibit similar performance and display less discriminatory behavior across quartiles. It is interesting to note that these models also perform best overall (see Table 6). On the other hand, SGL and NGCF show a higher slope that is comparable to classic CF methods. Also, their corresponding performance is similar in Table 6. A third observation concerns GFCF, which performs poorly across all quartiles except for the fourth. Its behavior is even more accentuated than in the 1-hop analysis. Additionally, NGCF, SGL, and GFCF are graph CF algorithms performing differently according to user warmness and item popularity. Meanwhile, all algorithms in BookCrossing, and the classic CF in Allrecipes, exhibit the distribution over the quartiles across methods.

Finally, in the 3-hop, for the BookCrossing dataset, the information pertaining to neighbors does not contribute significantly to the results, as indicated by the similarity between the 2- and 3-hop plots. Meanwhile, in Allrecipes, the best models (UltraGCN, DGCF, and LightGCN) exhibit more consistency in performance across all quartiles, as demonstrated by a more even distribution of results (less variations across the quartiles). However, this pattern is not evident in NGCF, SGL, and GFCF, which exhibit a more disparate range of results across the quartiles.

6 CONCLUSION AND FUTURE WORK

This study replicates the results of six graph CF methods, namely NGCF, DGCF, LightGCN, SGL, UltraGCN, and GFCF, and expands



(a) **1-hop**

(b) **2-hop**

(c) **3-hop**

Figure 2: Percentage variation between the nDCG on user quartiles and the average nDCG value across all users (indicated as the dashed line), for each model-dataset setting. Rows refer to user quartiles when considering (a) 1-, (b) 2-, and (c) 3-hop.

the research to include state-of-the-art recommendation strategies like UserkNN, ItemkNN, RP$^3\beta$, and EASE^R. The observed high rankings of the latter ones highlight the need for more comprehensive evaluations. After the initial study on the standard Gowalla, Yelp 2018, and Amazon Book datasets, experiments are extended to two additional datasets, Allrecipes and BookCrossing, which reveal substantial ranking variations compared to the initial datasets. Thus, the study introduces and analyzes the information flow in the graph and discovers that 2-hop information (combining user activeness and item popularity) is a valid indicator of CF behavior and could motivate the recommendation performance. The experimental results call for further investigations into the diversity and fairness of the considered methods and whether graph methods effectively mitigate popularity bias.

ACKNOWLEDGMENTS

This work was partially supported by the following projects: Secure Safe Apulia, MISE CUP: I14E20000020001 CTEMT - Casa delle Tecnologie Emergenti Comune di Matera, CT_FINCONS_II, CT_FINCONS_III, OVS Fashion Retail Reloaded, LUTECH DIGITALE 4.0, PID2019-108965GB-I00.

REFERENCES

[1] Vito Walter Anelli, Alejandro Bellogín, Antonio Ferrara, Daniele Malitesta, Felice Antonio Merra, Claudio Pomo, Francesco Maria Donini, and Tommaso Di Noia. 2021. Elliot: A Comprehensive and Rigorous Framework for Reproducible Recommender Systems Evaluation. In *SIGIR*. ACM, 2405–2414.

[2] Vito Walter Anelli, Alejandro Bellogín, Tommaso Di Noia, Dietmar Jannach, and Claudio Pomo. 2022. Top-N Recommendation Algorithms: A Quest for the State-of-the-Art. In *UMAP*. ACM, 121–131.

[3] Vito Walter Anelli, Alejandro Bellogín, Tommaso Di Noia, and Claudio Pomo. 2021. Reenvisioning the comparison between Neural Collaborative Filtering and Matrix Factorization. In *RecSys*. ACM, 521–529.

[4] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Daniele Malitesta, Vincenzo Paparella, and Claudio Pomo. 2023. Auditing Consumer- and Producer-Fairness in Graph Collaborative Filtering. In *ECIR (1) (Lecture Notes in Computer Science, Vol. 13980)*. Springer, 33–48.

[5] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Eugenio Di Sciascio, Antonio Ferrara, Daniele Malitesta, and Claudio Pomo. 2022. How Neighborhood Exploration influences Novelty and Diversity in Graph Collaborative Filtering. In *MORS@RecSys (CEUR Workshop Proceedings, Vol. 3268)*. CEUR-WS.org.

[6] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Eugenio Di Sciascio, Antonio Ferrara, Daniele Malitesta, and Claudio Pomo. 2022. Reshaping Graph Recommendation with Edge Graph Collaborative Filtering and Customer Reviews. In *DL4SR@CIKM (CEUR Workshop Proceedings, Vol. 3317)*. CEUR-WS.org.

[7] Alejandro Bellogín and Alan Said. 2021. Improving accountability in recommender systems research through reproducibility. *User Model. User Adapt. Interact.* 31, 5 (2021), 941–977.

[8] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for Hyper-Parameter Optimization. In *NIPS*. 2546–2554.

[9] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velickovic. 2021. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *CoRR* abs/2104.13478 (2021).

[10] Jiangxia Cao, Xixun Lin, Shu Guo, Luchen Liu, Tingwen Liu, and Bin Wang. 2021. Bipartite Graph Embedding via Mutual Information Maximization. In *WSDM*. ACM, 635–643.

[11] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient Neural Matrix Factorization without Sampling for Recommendation. *ACM Trans. Inf. Syst.* 38, 2 (2020), 14:1–14:28.

[12] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and Relieving the Over-Smoothing Problem for Graph Neural Networks from the Topological View. In *AAAI*. AAAI Press, 3438–3445.

[13] Hanxiong Chen, Yunqi Li, Shaoyun Shi, Shuchang Liu, He Zhu, and Yongfeng Zhang. 2022. Graph Collaborative Reasoning. In *WSDM*. ACM, 75–84.

[14] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting Graph Based Collaborative Filtering: A Linear Residual Graph Convolutional Network Approach. In *AAAI*. AAAI Press, 27–34.

[15] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2021. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *ACM Trans. Inf. Syst.* 39, 2 (2021), 20:1–20:49.

[16] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *RecSys*. ACM, 101–109.

[17] Xiaoyu Du, Zike Wu, Fuli Feng, Xiangnan He, and Jinhui Tang. 2022. Invariant Representation Learning for Multimedia Recommendation. In *ACM Multimedia*. ACM, 619–628.

[18] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative Memory Network for Recommendation Systems. In *SIGIR*. ACM, 515–524.

[19] Wenqi Fan, Xiaorui Liu, Wei Jin, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2022. Graph Trend Filtering Networks for Recommendation. In *SIGIR*. ACM, 112–121.

[20] Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. 2022. Graph Neural Networks for Recommender System. In *WSDM*. ACM, 1623–1625.

[21] Xiaoyan Gao, Fuli Feng, Xiangnan He, Heyan Huang, Xinyu Guan, Chong Feng, Zhaoyan Ming, and Tat-Seng Chua. 2020. Hierarchical Attention Network for Visually-Aware Food Recommendation. *IEEE Trans. Multim.* 22, 6 (2020), 1647–1659.

[22] Yunjun Gao, Yuntao Du, Yujia Hu, Lu Chen, Xinjun Zhu, Ziquan Fang, and Baihua Zheng. 2022. Self-Guided Learning to Denoise for Robust Recommendation. In *SIGIR*. ACM, 1412–1422.

[23] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *ICML (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 1263–1272.

[24] Kaiqi Gong, Xiao Song, Senzhang Wang, Songsong Liu, and Yong Li. 2022. ITSM-GCN: Informative Training Sample Mining for Graph Convolutional Network-based Collaborative Filtering. In *CIKM*. ACM, 614–623.

[25] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *KDD*. ACM, 855–864.

[26] William L. Hamilton. 2020. *Graph Representation Learning*. Morgan & Claypool Publishers.

[27] Ruining He and Julian J. McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *WWW*. ACM, 507–517.

[28] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. ACM, 639–648.

[29] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. ACM, 173–182.

[30] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge J. Belongie, and Deborah Estrin. 2017. Collaborative Metric Learning. In *WWW*. ACM, 193–201.

[31] Chao Huang, Lianghao Xia, Xiang Wang, Xiangnan He, and Dawei Yin. 2022. Self-Supervised Learning for Recommendation. In *CIKM*. ACM, 5136–5139.

[32] Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. 2021. MixGCF: An Improved Training Method for Graph Neural Network-based Recommender Systems. In *KDD*. ACM, 665–674.

[33] Keller Jordan. 2023. Calibrated Chaos: Variance Between Runs of Neural Network Training is Harmless and Inevitable. arXiv:2304.01910 [cs.LG]

[34] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. In *NeurIPS*.

[35] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (Poster)*. OpenReview.net.

[36] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[37] Dawen Liang, Laurent Charlin, James McInerney, and David M. Blei. 2016. Modeling User Exposure in Recommendation. In *WWW*. ACM, 951–961.

[38] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *WWW*. ACM, 689–698.

[39] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. In *WWW*. ACM, 2320–2329.

[40] Fan Liu, Zhiyong Cheng, Lei Zhu, Zan Gao, and Liqiang Nie. 2021. Interest-aware Message-Passing GCN for Recommendation. In *WWW*. ACM / IW3C2, 1296–1305.

[41] Jiahao Liu, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, Li Shang, and Ning Gu. 2023. Personalized Graph Signal Processing for Collaborative Filtering. *CoRR* abs/2302.02113 (2023).

[42] Siwei Liu, Iadh Ounis, and Craig Macdonald. 2022. An MLP-based Algorithm for Efficient Contrastive Graph Recommendations. In *SIGIR*. ACM, 2431–2436.

[43] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled Graph Convolutional Networks. In *ICML (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 4212–4221.

[44] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *NeurIPS*. 5712–5723.

[45] Daniele Malitesta, Claudio Pomo, Vito Walter Anelli, Tommaso Di Noia, and Antonio Ferrara. 2023. An Out-of-the-Box Application for Reproducible Graph Collaborative Filtering extending the Elliot Framework. In *UMAP (Adjunct Publication)*. ACM, 12–15.

[46] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. SimpleX: A Simple and Strong Baseline for Collaborative Filtering. In *CIKM*. ACM, 1243–1252.

[47] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation. In *CIKM*. ACM, 1253–1262.

[48] Yi Ouyang, Peng Wu, and Li Pan. 2022. Asymmetrical Context-aware Modulation for Collaborative Filtering Recommendation. In *CIKM*. ACM, 1595–1604.

[49] Bibek Paudel, Fabian Christoffel, Chris Newell, and Abraham Bernstein. 2017. Updatable, Accurate, Diverse, and Scalable Recommendations for Interactive Applications. *ACM Trans. Interact. Intell. Syst.* 7, 1 (2017), 1:1–1:34.

[50] Shaowen Peng, Kazunari Sugiyama, and Tsunenori Mine. 2022. Less is More: Reweighting Important Spectral Graph Features for Recommendation. In *SIGIR*. ACM, 1273–1282.

[51] Shaowen Peng, Kazunari Sugiyama, and Tsunenori Mine. 2022. SVD-GCN: A Simplified Graph Convolution Paradigm for Recommendation. In *CIKM*. ACM, 1625–1634.

[52] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. In *KDD*. ACM, 701–710.

[53] Nikhil Rao, Hsiang-Fu Yu, Pradeep Ravikumar, and Inderjit S. Dhillon. 2015. Collaborative Filtering with Graph Information: Consistency and Scalable Methods. In *NIPS*. 2107–2115.

[54] Xuan Rao, Lisi Chen, Yong Liu, Shuo Shang, Bin Yao, and Peng Han. 2022. Graph-Flashback Network for Next Location Recommendation. In *KDD*. ACM, 1463–1471.

[55] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*. AUAI Press, 452–461.

[56] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *CSCW*. ACM, 175–186.

[57] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*. ACM, 285–295.

[58] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *IEEE Trans. Neural Networks* 20, 1 (2009), 61–80.

[59] Yifei Shen, Yongji Wu, Yao Zhang, Caihua Shan, Jun Zhang, Khaled B. Letaief, and Dongsheng Li. 2021. How Powerful is Graph Convolution for Recommendation?. In *CIKM*. ACM, 1619–1629.

[60] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *RecSys*. ACM, 213–220.

[61] Harald Steck. 2019. Embarrassingly Shallow Autoencoders for Sparse Data. In *WWW*. ACM, 3251–3257.

[62] Jianing Sun, Zhaoyue Cheng, Saba Zuberi, Felipe Pérez, and Maksims Volkovs. 2021. HGCF: Hyperbolic Graph Convolution Networks for Collaborative Filtering. In *WWW*. ACM / IW3C2, 593–601.

[63] Jianing Sun, Wei Guo, Dengcheng Zhang, Yingxue Zhang, Florence Regol, Yaochen Hu, Huifeng Guo, Ruiming Tang, Han Yuan, Xiuqiang He, and Mark Coates. 2020. A Framework for Recommending Accurate and Diverse Items Using Bayesian Graph Convolutional Neural Networks. In *KDD*. ACM, 2030–2039.

[64] Jianing Sun, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, Xiuqiang He, Chen Ma, and Mark Coates. 2020. Neighbor Interaction Aware Graph Convolution Networks for Recommendation. In *SIGIR*. ACM, 1289–1298.

[65] Zhu Sun, Di Yu, Hui Fang, Jie Yang, Xinghua Qu, Jie Zhang, and Cong Geng. 2020. Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison. In *RecSys*. ACM, 23–32.

[66] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *WWW*. ACM, 1067–1077.

[67] Zhulin Tao, Yinwei Wei, Xiang Wang, Xiangnan He, Xianglin Huang, and Tat-Seng Chua. 2020. MGAT: Multimodal Graph Attention Network for Recommendation. *Inf. Process. Manag.* 57, 5 (2020), 102277.

[68] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph Convolutional Matrix Completion. *CoRR* abs/1706.02263 (2017).

[69] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR (Poster)*. OpenReview.net.

[70] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *KDD*. ACM, 950–958.

[71] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*. ACM, 165–174.

[72] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. *CoRR* abs/1905.08108 (2019).

[73] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *SIGIR*. ACM, 1001–1010.

[74] Xiangmeng Wang, Qian Li, Dianer Yu, Peng Cui, Zhichao Wang, and Guandong Xu. 2022. Causal Disentanglement for Semantics-Aware Intent Learning in Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2022), 1–1. https://doi.org/10.1109/TKDE.2022.3159802

[75] Zhenyi Wang, Huan Zhao, and Chuan Shi. 2022. Profiling the Design Space for Graph Neural Networks based Collaborative Filtering. In *WSDM*. ACM, 1109–1119.

[76] Chunyu Wei, Jian Liang, Bing Bai, and Di Liu. 2022. Dynamic Hypergraph Learning for Collaborative Filtering. In *CIKM*. ACM, 2108–2117.

[77] Wei Wei, Chao Huang, Lianghao Xia, Yong Xu, Jiashu Zhao, and Dawei Yin. 2022. Contrastive Meta Learning with Behavior Multiplicity for Recommendation. In *WSDM*. ACM, 1120–1128.

[78] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised Graph Learning for Recommendation. In *SIGIR*. ACM, 726–735.

[79] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2023. Graph Neural Networks in Recommender Systems: A Survey. *ACM Comput. Surv.* 55, 5 (2023), 97:1–97:37.

[80] Jiafeng Xia, Dongsheng Li, Hansu Gu, Jiahao Liu, Tun Lu, and Ning Gu. 2022. FIRE: Fast Incremental Recommendation with Graph Signal Processing. In *WWW*. ACM, 2360–2369.

[81] Lianghao Xia, Chao Huang, Jiao Shi, and Yong Xu. 2023. Graph-less Collaborative Filtering. *CoRR* abs/2303.08537 (2023).

[82] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy X. Huang. 2022. Hypergraph Contrastive Collaborative Filtering. In *SIGIR*. ACM, 70–79.

[83] Jheng-Hong Yang, Chih-Ming Chen, Chuan-Ju Wang, and Ming-Feng Tsai. 2018. HOP-rec: high-order proximity for implicit recommendation. In *RecSys*. ACM, 140–144.

[84] Menglin Yang, Min Zhou, Jiahong Liu, Defu Lian, and Irwin King. 2022. HRCF: Enhancing Collaborative Filtering via Hyperbolic Geometric Regularization. In *WWW*. ACM, 2462–2471.

[85] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge Graph Contrastive Learning for Recommendation. In *SIGIR*. ACM, 1434–1443.

[86] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix X. Yu, Aditya Krishna Menon, Lichan Hong, Ed H. Chi, Steve Tjoa, Jieqi Kang, and Evan Ettinger. 2020. Self-supervised Learning for Deep Models in Recommendations. *CoRR* abs/2007.12865 (2020).

[87] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *KDD*. ACM, 974–983.

[88] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation. In *WWW*. ACM / IW3C2, 413–424.

[89] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are Graph Augmentations Necessary?: Simple Graph Contrastive Learning for Recommendation. In *SIGIR*. ACM, 1294–1303.

[90] Wenhui Yu and Zheng Qin. 2020. Graph Convolutional Network for Recommendation with Low-pass Collaborative Filters. In *ICML (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 10936–10945.

[91] Wenhui Yu and Zheng Qin. 2020. Sampler Design for Implicit Feedback Data by Noisy-label Robust Learning. In *SIGIR*. ACM, 861–870.

[92] Yiding Zhang, Chaozhuo Li, Xing Xie, Xiao Wang, Chuan Shi, Yuming Liu, Hao Sun, Liangjie Zhang, Weiwei Deng, and Qi Zhang. 2022. Geometric Disentangled Collaborative Filtering. In *SIGIR*. ACM, 80–90.

[93] Minghao Zhao, Le Wu, Yile Liang, Lei Chen, Jian Zhang, Qilin Deng, Kai Wang, Xudong Shen, Tangjie Lv, and Runze Wu. 2022. Investigating Accuracy-Novelty Performance for Graph-based Collaborative Filtering. In *SIGIR*. ACM, 50–59.

[94] Lei Zheng, Chun-Ta Lu, Fei Jiang, Jiawei Zhang, and Philip S. Yu. 2018. Spectral collaborative filtering. In *RecSys*. ACM, 311–319.

[95] Yuchen Zhou, Yanan Cao, Yanmin Shang, Chuan Zhou, Shirui Pan, Zheng Lin, and Qian Li. 2022. Explainable Hyperbolic Temporal Point Process for User-Item Interaction Sequence Generation. *ACM Trans. Inf. Syst.* (nov 2022). https://doi.org/10.1145/3570501

[96] Jieming Zhu, Quanyu Dai, Liangcai Su, Rong Ma, Jinyang Liu, Guohao Cai, Xi Xiao, and Rui Zhang. 2022. BARS: Towards Open Benchmarking for Recommender Systems. In *SIGIR*. ACM, 2912–2923.

[97] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *WWW*. ACM, 22–32.