How Neighborhood Exploration influences Novelty and Diversity in Graph Collaborative Filtering

VITO WALTER ANELLI, Politecnico di Bari, Italy
YASHAR DELDJOO, Politecnico di Bari, Italy
TOMMASO DI NOIA, Politecnico di Bari, Italy
EUGENIO DI SCIASCIO, Politecnico di Bari, Italy
ANTONIO FERRARA, Politecnico di Bari, Italy
DANIELE MALITESTA*, Politecnico di Bari, Italy
CLAUDIO POMO*, Politecnico di Bari, Italy

Graph convolutional networks (GCNs) have recently been shown to improve the recommendation accuracy of collaborative filtering algorithms. Their message-passing schema refines user and item node representation by aggregating the informative content from the neighborhood. However, after multiple hops, noisy contributions can flatten the differences among nodes, as not all user-item interactions are equally important. This impact is mitigated by (i) restricting the exploration depth in the graph and optionally weighting the neighbor contribution and (ii) going beyond the traditional message propagation at multiple hops. Nevertheless, it remains unclear how these exploration strategies affect the recommendation of novel and diverse products. This study investigates the influence of such GCN techniques on novelty and diversity of recommendations. It also assesses and motivates the impact of the number of exploration hops on the same metrics by analyzing interactions between same-type and different-type nodes, such as user-user and user-item, respectively. Code and datasets are available at: https://github.com/sisinflab/Novelty-Diversity-Graph.

CCS Concepts: • **Information systems → Recommender systems**; • **Computing methodologies → Neural networks**.

Additional Key Words and Phrases: Collaborative Filtering, Recommendation, Graph Convolutional Networks

1 INTRODUCTION

In the challenge of bridging the gap between supply and demand, popular companies (e.g., Amazon, Booking) have opted to integrate recommendation systems into their online platforms. These algorithms attempt to present customers with personalized lists of preferred products by identifying preference patterns among users and items. Among the existing recommendation paradigms, collaborative filtering (CF) [5] has long settled as the dominant approach, suggesting that like-minded users could interact with similar items. CF models optimize an objective score function between users and items, where both of them are mapped into embeddings and combined linearly (e.g., inner product [17]) or non-linearly (e.g., neural networks [12] and probabilistic models [20]).

The natural representation of users and items in a recommendation system is a bipartite, undirected graph, where users and items are the nodes and recorded interactions are the edges linking them. For this reason, graph convolutional networks (GCNs) [16] have gained traction in recommendation, from pioneering works [35, 39] to more recent solutions [11, 21, 25].

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

Authors' addresses: Vito Walter Anelli, vitowalter.anelli@poliba.it, Politecnico di Bari, Italy; Yashar Deldjoo, yashar.deldjoo@poliba.it, Politecnico di Bari, Italy; Tommaso Di Noia, tommaso.dinoia@poliba.it, Politecnico di Bari, Italy; Eugenio Di Sciascio, eugenio.disciascio@poliba.it, Politecnico di Bari, Italy; Antonio Ferrara, antonio.ferrara@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.

Graph convolution relies upon the concept of message-passing networks [7] to refine nodes' representation, where each *ego* node embedding is refined by aggregating its' *neighbors* node embeddings (i.e., whose contribution is called *message*). The procedure is performed iteratively over multiple hops, therefore exploring ever-expanding neighborhoods surrounding the ego node. Differently from previous CF approaches, the adoption of a message-passing schema helps explicitly incorporate user and item high-order relationships into their embedding representations, therefore effectively distilling the **collaborative signal** [35]. Nevertheless, GCN performance has been shown to decrease as the number of explored hops increases since graph convolution indiscriminately aggregates all contributions from the neighbor nodes (even unimportant ones), eventually smoothing the differences in the neighborhood [3, 42].

To mitigate this over-smoothing effect, graph-based techniques for collaborative filtering limit the exploration of neighborhood to three hops [4, 11, 35]. Similar approaches are designed to weight the importance of each neighbor node on its ego node through attention mechanisms [33], which allows the exploration of even smaller portions of the neighborhood to reach remarkable results [36].

Conversely, recent works [21, 25] highlight critical limitations in the adoption of graph convolution to explore users' and items' neighborhoods. Starting from the idea described in [11], they propose alternative reformulations of GCN for the recommendation task, providing simplified and lighter versions which go beyond the traditional concept of multi-hop message-passing.

Although the literature has widely shown the recommendation accuracy boost of such models to traditional CF baselines, their ability to produce novel and diverse recommendation lists [30, 31] remains poorly investigated. While the topic has been addressed only recently by few works in graph CF [28, 41], modern recommender systems are more and more required to reach a sufficient trade-off between accurate and novel/diverse recommendations [19, 27, 38], as a renewed need from both user's and business's perspectives [1, 18].

This paper seeks to understand how and why the neighborhood exploration strategy and (optionally) depth may influence novelty and diversity recommendation metrics in graph collaborative filtering. To this aim, we run extensive experiments by training and evaluating six state-of-the-art graph models for CF on three popular recommendation datasets. Our contributions are threefold: *(i)* to the best of our knowledge, no previous work has evaluated approaches from the two recognized graph recommendation families (i.e., with and without explicit message-passing) on a grid of accuracy/novelty/diversity recommendation metrics, *(ii)* to provide a fair comparison, we train all message-passing models exploring the whole hop range 1-4, which also allows examining the accuracy/novelty/diversity trade-off on the neighborhood size, and *(iii)* we propose a simple reformulation of the message-passing schema where *same*-type node connections (e.g., user-user) and *different*-type node connections (e.g., user-item) are explicitly highlighted, in an effort to unveil new (and non-evident) performance patterns.

2 RELATED WORK

Graph Collaborative Filtering. After pioneer works [29, 39] adopting vanilla GCN [16] for recommendation, other approaches propose finer neighborhood explorations built upon it. Wang et al. [35] aggregate the messages from the neighborhood considering the similarity between each neighbor node and its ego node, while the works in [4, 11, 14] improve accuracy when removing non-linearities and feature transformations. As neighbor nodes are not equally important to their ego node, noisy messages tend to over-smooth the existing node differences after multiple hops [3, 42]. To tackle the issue, messages are propagated to a maximum of three hops [11, 35], optionally leveraging attention mechanisms [33] to learn the importance of users' intents on the interacted items [36, 37]. Conversely, more recent approaches take a step further and try to rethink the message-passing schema by allowing theoretically unlimited

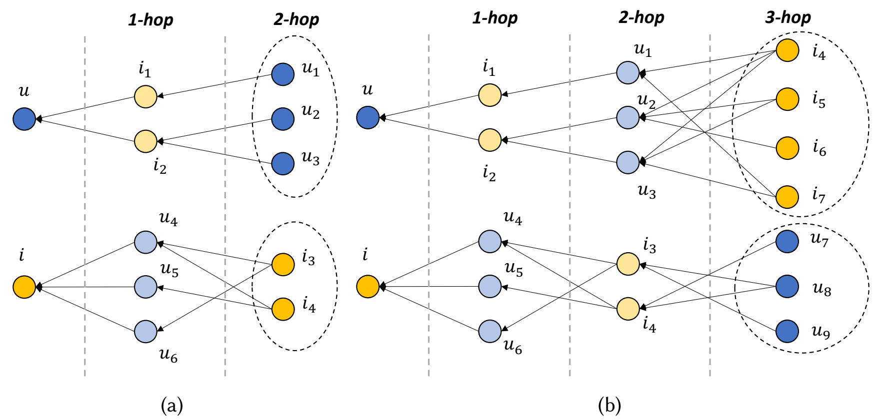

Fig. 1. User and item neighborhood exploration after (a) 2 and (b) 3 hops. Contributions to the ego node update are highlighted through dashed ovals. Edge direction indicates the message propagation from neighbor to ego nodes.

propagation hops [21] and revisiting the concept of graph convolution and node embedding smoothness through the lens of graph signal processing [25].

Novelty and Diversity in Recommendation. User experience is becoming crucial on recommendation platforms [13, 15, 26] as the suggestion of interesting lists of items satisfies users and entices them to remain loyal to the platform, thus increasing profits [34]. A good user experience requires the recommended items to be nontrivial, as diverse as possible, and possibly unexpected [6, 26]. However, designing dedicated models is particularly challenging due to the inherent difficulty of evaluating them without a user study. For this reason, researchers have dedicated a considerable effort to the beyond-accuracy dimensions over the past two decades [24, 32, 40]. While the search for the accuracy/novelty/diversity trade-off has gained momentum in recommendation [19, 27, 38], to the best of our knowledge, only two studies investigate novelty and diversity dimensions in the field of graph collaborative filtering [28, 41]. Their focus is on identifying the accuracy/diversity trade-off by proposing specific models that could achieve competitive performance. *However, they do not analyze the link between neighborhood exploration and these dimensions. Instead, we assess the state-of-the-art, most accurate models for graph recommendation and inspect how they behave on novelty and diversity, exploring the potential motivations.*

3 NEIGHBORHOOD EXPLORATION

We propose a simple reformulation of the message-passing schema for graph recommendation, where interactions among *same*-type nodes (e.g., user-user) and *different*-type nodes (e.g., item-user) are explicitly highlighted.

Preliminaries. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$ and $\mathcal{I} = \{i_1, i_2, \ldots, i_M\}$ be the sets of users and items. Let $\mathbf{Y} \in \mathbb{R}^{N \times M}$ be the user-item interaction matrix, where $y_{ui} = 1$ when user u interacted with item i, and $y_{ui} = 0$ otherwise. The adjacency matrix for all user-item and item-user interactions is:

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

Let $\mathcal{G} = (\{\mathcal{U}, \mathcal{I}\}, \mathbf{A})$ be the bipartite and undirected graph connecting pairs of nodes (i.e., users and items) for an existing interaction among them. User and item node features are the embeddings $\mathbf{e}_u \in \mathbb{R}^d, \forall u \in \mathcal{U}$ and $\mathbf{e}_i \in \mathbb{R}^d, \forall i \in \mathcal{I}$, respectively.

Message-Passing. Let u and i be the nodes for the user and the item to update (*ego* nodes), and let $\mathcal{N}(u)$ and $\mathcal{N}(i)$ be the sets of nodes at one hop from u and i, respectively (*neighbor* nodes). The schema aggregates the embeddings from the neighborhood (*messages*) to refine the ego nodes:

$$\mathbf{e}_u^{(1)} = \omega\left(\left\{\mathbf{e}_{i'}^{(0)}, \forall i' \in \mathcal{N}(u)\right\}\right) \quad \mathbf{e}_i^{(1)} = \omega\left(\left\{\mathbf{e}_{u'}^{(0)}, \forall u' \in \mathcal{N}(i)\right\}\right) \tag{2}$$

where $\mathbf{e}_u^{(1)}$ and $\mathbf{e}_i^{(1)}$ are the refined embedding versions of user u and item i after one hop, $\omega(\cdot)$ is the aggregation function (e.g., the summation), while $\mathbf{e}_{u'}^{(0)} = \mathbf{e}_{u'}$ and $\mathbf{e}_{i'}^{(0)} = \mathbf{e}_{i'}$. To explore wider and wider neighborhoods of the ego nodes, aggregation is usually iterated. After two hops, the embeddings of user u and item i are:

$$\mathbf{e}_u^{(2)} = \omega\left(\left\{\mathbf{e}_{i'}^{(1)}, \forall i' \in \mathcal{N}(u)\right\}\right) \quad \mathbf{e}_i^{(2)} = \omega\left(\left\{\mathbf{e}_{u'}^{(1)}, \forall u' \in \mathcal{N}(i)\right\}\right) \tag{3}$$

The general message-passing formulation after l hops is:

$$\mathbf{e}_u^{(l)} = \omega\left(\left\{\mathbf{e}_{i'}^{(l-1)}, \forall i' \in \mathcal{N}(u)\right\}\right) \quad \mathbf{e}_i^{(l)} = \omega\left(\left\{\mathbf{e}_{u'}^{(l-1)}, \forall u' \in \mathcal{N}(i)\right\}\right) \tag{4}$$

Reformulation. The two-hop node update in Equation (3) is further expanded through the one-hop node update in Equation (2):

$$\begin{aligned} \mathbf{e}_u^{(2)} &= \omega\Big(\Big\{\omega\Big(\Big\{\mathbf{e}_{u''}^{(0)}, \underbrace{\forall u'' \in \mathcal{N}(i') \setminus \{u\}}_{\text{2-hop}}\Big\}\Big), \underbrace{\forall i' \in \mathcal{N}(u)}_{\text{1-hop}}\Big\}\Big) \\ \mathbf{e}_i^{(2)} &= \omega\Big(\Big\{\omega\Big(\Big\{\mathbf{e}_{i''}^{(0)}, \underbrace{\forall i'' \in \mathcal{N}(u') \setminus \{i\}}_{\text{2-hop}}\Big\}\Big), \underbrace{\forall u' \in \mathcal{N}(i)}_{\text{1-hop}}\Big\}\Big) \end{aligned} \tag{5}$$

where set differences are used to avoid node duplicates. After two hops, the node embeddings of user u and item i get the contributions of those users u'' and items i'' for whom there exists a ***user***-*item*-***user*** path connecting u with u'', and an ***item***-*user*-***item*** path connecting i with i'', respectively (Figure 1a). Such paths link *same*-type nodes (in bold). In a similar manner, let us apply the general formula from Equation (4) to the three-hop node update:

$$\mathbf{e}_u^{(3)} = \omega\left(\left\{\mathbf{e}_{i'}^{(2)}, \forall i' \in \mathcal{N}(u)\right\}\right) \quad \mathbf{e}_i^{(3)} = \omega\left(\left\{\mathbf{e}_{u'}^{(2)}, \forall u' \in \mathcal{N}(i)\right\}\right) \tag{6}$$

which we expand through Equation (5):

$$\begin{aligned} \mathbf{e}_u^{(3)} &= \omega\Big(\Big\{\omega\Big(\Big\{\omega\Big(\Big\{\mathbf{e}_{i'''}^{(0)}, \underbrace{\forall i''' \in \mathcal{N}(u'') \setminus \{i''\}}_{\text{3-hop}}\Big\}\Big), \\ &\qquad \underbrace{\forall u'' \in \mathcal{N}(i') \setminus \{u''\}}_{\text{2-hop}}\Big\}\Big), \underbrace{\forall i' \in \mathcal{N}(u)}_{\text{1-hop}}\Big\}\Big) \\ \mathbf{e}_i^{(3)} &= \omega\Big(\Big\{\omega\Big(\Big\{\omega\Big(\Big\{\mathbf{e}_{u'''}^{(0)}, \underbrace{\forall u''' \in \mathcal{N}(i'') \setminus \{u''\}}_{\text{3-hop}}\Big\}\Big), \\ &\qquad \underbrace{\forall i'' \in \mathcal{N}(u') \setminus \{i''\}}_{\text{2-hop}}\Big\}\Big), \underbrace{\forall u' \in \mathcal{N}(i)}_{\text{1-hop}}\Big\}\Big) \end{aligned} \tag{7}$$

After three hops, the node embeddings of user u and item i get the contributions of those items i''' and users u''' for whom there exists a ***user**-item-user-**item*** path connecting u with i''', and an ***item**-user-item-**user*** path connecting i with u''', respectively (Figure 1b). In this case, such paths link *different*-type nodes (in bold).

This reformulation outlines two neighborhood exploration types, propagating messages through *same*- and *different*-type nodes after an **even** and an **odd** number of hops, respectively. While previous works assess recommendation performance when **indistinctly** increasing the hop numbers, we provide a finer evaluation based on the explored node types. In the next sections, we will count hops following the introduced categorization. For example, *same*-type node explorations after **1** and **2** hops refer to the paths ***user**-item-**user*** and ***user**-item-user-item-**user***, respectively, while *different*-type node explorations after **1** and **2** hops refer to the paths ***user-item*** and ***user**-item-user-**item***, respectively.

4 EXPERIMENTS AND DISCUSSION

In the following, we describe datasets, baselines, reproducibility details, evaluation protocol, and results of our work.

4.1 Experimental Setup

Datasets. We adopt Movielens-1M [8], Amazon Digital Music [22], and Epinions [23]. They are binarized by retaining interactions with score > 3 (Epinions already has an implicit version), and filtered through the p-core to avoid the cold-start effect [9, 10] which is out of the scope of this paper. Movielens-1M counts 5,915 users, 2,753 items, and 570,622 interactions, Amazon Digital Music counts 8,328 users, 6,275 items, and 99,400 interactions, and Epinions counts 14,341 users, 13,145 items, and 269,170 interactions.

Baselines. We evaluate graph recommendation models explicitly adopting message propagation (i.e., NGCF [35], LightGCN [11], DGCF [36], and LR-GCCF [4]) and going beyond the message-passing schema (i.e., UltraGCN [21] and GFCF [25]).

Reproducibility. Datasets are split into train/validation/test with 80/10/10 hold-out. Models are trained by searching the best hyperparameters as in [2] and setting search spaces according to the original works while fixing the number of epochs to 400 and batch size to 1024. Datasets and codes are made accessible[1].

Evaluation. Following [30–32], we select the expected popularity complement ($EPC@k$) and the expected free discovery ($EFD@k$) as *Novelty* metrics [32], along with the 1's complement of the Gini index ($Gini@k$) and the Shannon entropy ($SE@k$) as *Diversity* metrics [24]. Both the $EPC@k$ and the $EFD@k$ account for long-tail items, and measure the expected number of recommended unknown and known items which are also relevant, respectively. The $Gini@k$ and the $SE@k$ calculate how unequally a recommender system shows different items to users. We set the $Recall@20$ as validation metric to follow the original papers, and also report results for it and $nDCG@k$ to assess the accuracy/novelty/diversity trade-off.

4.2 Results and Discussion

This section shows the recommendation performance and a finer evaluation of the accuracy/novelty/diversity trade-offs. Results refer to top-20 recommendation lists.

Overall Recommendation Performance. Table 1 depicts recommendation performance on accuracy, novelty, and diversity.

[1]https://github.com/sisinflab/Novelty-Diversity-Graph.

Table 1. Overall recommendation performance on accuracy, novelty, and diversity metrics for top-20 recommendation lists. Bold and underline stand for best and second-to-best values, respectively.

Models	**Movielens-1M**						**Amazon Digital Music**						**Epinions**					
	Accuracy		**Novelty**		**Diversity**		**Accuracy**		**Novelty**		**Diversity**		**Accuracy**		**Novelty**		**Diversity**	
	Recall	*nDCG*	*EPC*	*EFD*	*Gini*	*SE*	*Recall*	*nDCG*	*EPC*	*EFD*	*Gini*	*SE*	*Recall*	*nDCG*	*EPC*	*EFD*	*Gini*	*SE*
MostPop	0.1380	0.1099	0.0473	0.5365	0.0105	5.2156	0.0319	0.0154	0.0029	0.0263	0.0031	4.3832	0.0467	0.0224	0.0054	0.0489	0.0015	4.4358
Random	0.0077	0.0060	0.0036	0.0414	0.9105	11.4085	0.0017	0.0007	0.0002	0.0021	0.8929	12.5890	0.0015	0.0006	0.0002	0.0024	0.8789	13.6486
NGCF	0.2535	0.1985	0.0929	1.0214	**0.1479**	8.9930	0.1127	0.0606	0.0109	0.1270	**0.4130**	**11.6953**	0.0792	0.0394	0.0096	0.1079	**0.2107**	**11.6255**
LightGCN	0.2712	0.2167	0.1013	1.1129	<u>0.1465</u>	<u>9.0079</u>	0.1189	0.0628	0.0113	0.1310	0.3148	11.2940	0.0914	0.0466	0.0115	0.1217	0.0759	9.7898
DGCF	<u>0.2791</u>	<u>0.2231</u>	<u>0.1047</u>	<u>1.1490</u>	0.1462	**9.0111**	<u>0.1264</u>	0.0674	<u>0.0123</u>	<u>0.1400</u>	0.2483	10.8904	**0.1046**	<u>0.0536</u>	**0.0132**	**0.1407**	0.0599	9.6502
LR-GCCF	**0.2876**	**0.2274**	**0.1056**	**1.1589**	0.1245	8.7438	0.1246	0.0664	0.0119	0.1388	<u>0.4037</u>	<u>11.6542</u>	0.0990	0.0504	0.0124	0.1377	<u>0.1367</u>	<u>10.8977</u>
UltraGCN	0.2540	0.2045	0.0901	0.9921	0.0766	8.0334	0.1256	<u>0.0675</u>	<u>0.0123</u>	0.1382	0.1737	10.0458	<u>0.1041</u>	**0.0541**	<u>0.0131</u>	<u>0.1397</u>	0.0586	9.0948
GFCF	0.1685	0.1398	0.0583	0.6577	0.0117	5.4064	**0.1287**	**0.0744**	**0.0137**	**0.1544**	0.2392	10.4923	0.0946	0.0496	0.0115	0.1158	0.0277	7.5926

Coherently with the literature, DGCF and LR-GCCF are steadily the best or the second-to-best models on accuracy (e.g., DGCF reaches the second-to-best *Recall* on Amazon Digital Music, while LR-GCCF obtains the best *nDCG* on Movielens-1M). Approaches without explicit message aggregation (i.e., UltraGCN and GFCF) still compete with the other baselines on accuracy (e.g., GFCF is the best model on Amazon Digital Music for the *Recall* and the *nDCG*, and UltraGCN is the best technique on Epinions for the *nDCG*).

As for the accuracy/novelty/diversity trade-off, we see that, independently of the adoption of message-passing, accurate approaches can also produce novel recommendations (e.g., LR-GCCF and DGCF are the best and second-to-best approaches for accuracy and novelty on Movielens-1M, and GFCF and UltraGCN provide superior accuracy performance on Amazon Digital Music and Epinions, respectively, with GFCF outperforming all other baselines on novelty, and UltraGCN getting slightly lower *EPC* and *EFD* values than DGCF). Unexpectedly, NGCF settles as the approach producing the most diverse lists of recommended items on all datasets (i.e., see *Gini* and *SE*) but cannot cope with the other baselines in terms of *Recall* and *nDCG* (similarly to Random). Other graph models with explicit message-passing (especially DGCF and LR-GCCF) are placed in the best accuracy/diversity trade-off spot, as they are often the second-to-best approaches on diversity, with limited observable drops in the accuracy. Contrarily, techniques without message-passing always show the lowest diversity.

<u>***Observation 1.*** *While the accuracy/novelty trade-off does not depend on the explicit/non-explicit message-passing, the accuracy/diversity trade-off is preserved only when propagating messages, at the expense of (limited) recommendation accuracy drops.*</u>

Effect of Neighborhood Exploration. Figure 2 shows the accuracy/novelty/diversity trade-off on Amazon Digital Music for different exploration depths (i.e., explicit message-passing) and strategies (i.e., explicit and implicit messages). Using the reformulation from Section 3, we separate explicit message propagation results into *different*- and *same*-type node explorations at 1/2 hops. We confirm that, while UltraGCN and GFCF can compete well on the accuracy/novelty trade-off with the other baselines (whatever the explored number of hops and node type), the opposite occurs on the accuracy/diversity trade-off. Indeed, higher accuracy values for UltraGCN and GFCF are obtained at the expense of significant drops in their diversity, even compared to message propagation at 1 hop (e.g., DGCF surpasses them on diversity at the expense of a slightly lower accuracy in the *same*-node setting).

As for the influence of *different*- and *same*-type node explorations, wider explorations of the neighborhood almost always lead to improved accuracy/novelty and accuracy/diversity performance, independently of the explored node types (apart from the *same*-type settings for NGCF on the *Recall* and LR-GCCF on the *Recall* and the *EPC*). Noticeably, the exploration of 1 hop in the *same*-type node setting leads to a better trade-off in accuracy/novelty/diversity than

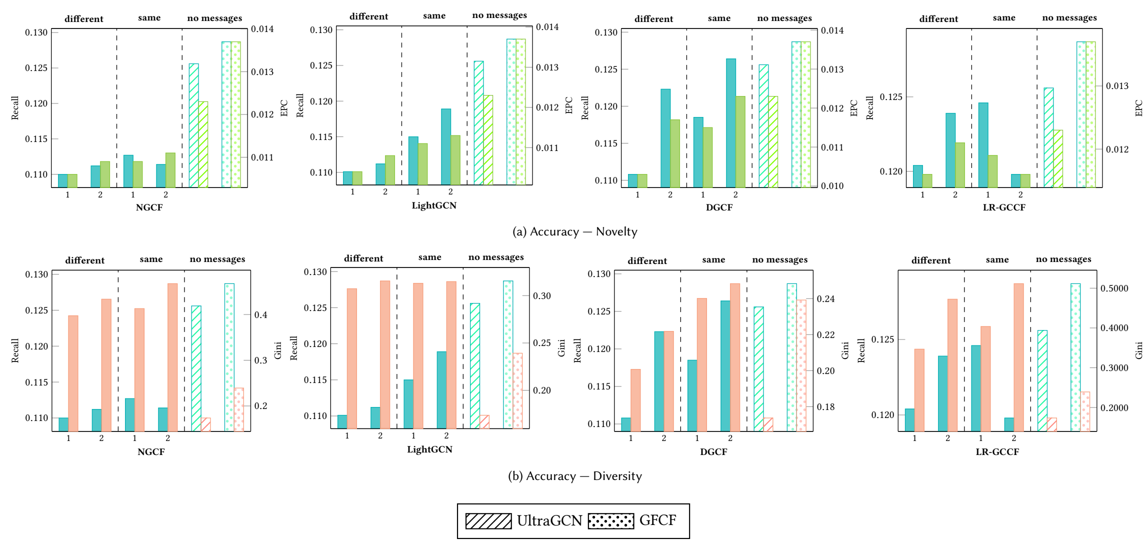

Fig. 2. Accuracy/Novelty (a) and Accuracy/Diversity (b) trade-offs of graph models explicitly propagating messages (i.e., filled bar plots) and going beyond message-passing (i.e., patterned bar plots) on Amazon Digital Music for top-20 recommendation lists. As for explicit message-passing, results are further categorized into *different*- and *same*-node type explorations (i.e., the leftmost and central tabs in each plot, respectively), when varying the number of hops from 1 to 2. Accuracy, novelty, and diversity are assessed through *Recall* (in teal blue), *EPC* (in lime green), and *Gini* (in melon), respectively. Best viewed in color.

the exploration of 2 hops in the *different*-node setting (e.g., LightGCN increases the *Recall* and the *EPC* without a significant variation of *Gini*, and DGCF slightly decreases the *Recall* and the *EPC*, but improves *Gini*).

Observation 2. *To confirm observation 1, message propagation (even at 1 hop) can reach a better accuracy/diversity trade-off than no message propagation; then, same-type node explorations may lead to improved accuracy/novelty and accuracy/diversity trade-offs.*

5 CONCLUSION AND FUTURE WORK

This work studies the accuracy/novelty/diversity trade-off in graph collaborative filtering for different neighborhood exploration strategies (i.e., with and without explicit message-passing) and depths (i.e., number of explored hops). Results for six state-of-the-art graph models on three e-commerce datasets reveal that the accuracy/diversity trade-off is reachable only when explicitly propagating messages. Thanks to a message-passing reformulation, we show that user-user and item-item explorations may improve accuracy/diversity/novelty trade-off. We plan to expand the evaluation to recent graph models which optimize diversity and better investigate the same-type node setting.

REFERENCES

[1] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Augusto Pizzato. 2020. Multistakeholder recommendation: Survey and research directions. *User Model. User Adapt. Interact.* 30, 1 (2020), 127–158.

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

[3] 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.

[4] 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.

[5] Michael D. Ekstrand, John Riedl, and Joseph A. Konstan. 2011. Collaborative Filtering Recommender Systems. *Found. Trends Hum. Comput. Interact.* 4, 2 (2011), 175–243.

[6] Alireza Gharahighehi and Celine Vens. 2021. Diversification in session-based news recommender systems. *Personal and Ubiquitous Computing* (29 Jul 2021). https://doi.org/10.1007/s00779-021-01606-4

[7] 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.

[8] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2016), 19:1–19:19.

[9] 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.

[10] Ruining He and Julian J. McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *AAAI*. AAAI Press, 144–150.

[11] 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.

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

[13] Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. 2015. What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Model. User Adapt. Interact.* 25, 5 (2015), 427–491.

[14] Weiguang Jiang, Su Wang, Jun Zheng, and Wenxin Hu. 2021. LGACN: A Light Graph Adaptive Convolution Network for Collaborative Filtering. In *ICANN (3) (Lecture Notes in Computer Science, Vol. 12893)*. Springer, 113–126.

[15] Marius Kaminskas and Derek Bridge. 2017. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Trans. Interact. Intell. Syst.* 7, 1 (2017), 2:1–2:42.

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

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

[18] Mounia Lalmas. 2020. Personalising and Diversifying the Listening Experience. In *ICTIR*. ACM, 3.

[19] Alfonso Landin, Javier Parapar, and Álvaro Barreiro. 2020. Novel and Diverse Recommendations by Leveraging Linear Models with User and Item Embeddings. In *ECIR (2) (Lecture Notes in Computer Science, Vol. 12036)*. Springer, 215–222.

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

[21] 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.

[22] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *SIGIR*. ACM, 43–52.

[23] Matthew Richardson, Rakesh Agrawal, and Pedro M. Domingos. 2003. Trust Management for the Semantic Web. In *ISWC (Lecture Notes in Computer Science, Vol. 2870)*. Springer, 351–368.

[24] Guy Shani and Asela Gunawardana. 2011. Evaluating Recommendation Systems. In *Recommender Systems Handbook*. Springer, 257–297.

[25] 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.

[26] Thiago Silveira, Min Zhang, Xiao Lin, Yiqun Liu, and Shaoping Ma. 2019. How good your recommender system is? A survey on evaluations in recommendation. *Int. J. Mach. Learn. Cybern.* 10, 5 (2019), 813–831.

[27] Dusan Stamenkovic, Alexandros Karatzoglou, Ioannis Arapakis, Xin Xin, and Kleomenis Katevas. 2022. Choosing the Best of Both Worlds: Diverse and Novel Recommendations through Multi-Objective Reinforcement Learning. In *WSDM*. ACM, 957–965.

[28] 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.

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

[30] Saúl Vargas. 2014. Novelty and diversity enhancement and evaluation in recommender systems and information retrieval. In *SIGIR*. ACM, 1281.

[31] Saul Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *RecSys*. ACM, 109–116.

[32] Saul Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *RecSys*. ACM, 109–116.

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

[34] Sanne Vrijenhoek, Mesut Kaya, Nadia Metoui, Judith Möller, Daan Odijk, and Natali Helberger. 2021. Recommenders with a Mission: Assessing Diversity in News Recommendations. In *CHIIR*. ACM, 173–183.

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

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

[37] Junkang Wu, Wentao Shi, Xuezhi Cao, Jiawei Chen, Wenqiang Lei, Fuzheng Zhang, Wei Wu, and Xiangnan He. 2021. DisenKGAT: Knowledge Graph Embedding with Disentangled Graph Attention Network. In *CIKM*. ACM, 2140–2149.

[38] Wenzhuo Yang, Jia Li, Chenxi Li, Latrice Barnett, Markus Anderle, Simo Arajärvi, Harshavardhan Utharavalli, Caiming Xiong, and Steven C. H. Hoi. 2021. On the Diversity and Explainability of Recommender Systems: A Practical Framework for Enterprise App Recommendation. In *CIKM*. ACM,

4302–4311.

[39] 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.

[40] ChengXiang Zhai, William W. Cohen, and John D. Lafferty. 2003. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR*. ACM, 10–17.

[41] Yu Zheng, Chen Gao, Liang Chen, Depeng Jin, and Yong Li. 2021. DGCN: Diversified Recommendation with Graph Convolutional Networks. In *WWW*. ACM / IW3C2, 401–412.

[42] Kaixiong Zhou, Xiao Huang, Yuening Li, Daochen Zha, Rui Chen, and Xia Hu. 2020. Towards Deeper Graph Neural Networks with Differentiable Group Normalization. In *NeurIPS*.