Enhancing Graph Collaborative Filtering with FourierKAN Feature Transformation

Jinfeng Xu

The University of Hong Kong Hong Kong, China jinfeng@connect.hku.hk

Zheyu Chen

Hong Kong Polytechnic University Hong Kong, China zheyu.chen@connect.polyu.hk

Jinze Li

The University of Hong Kong Hong Kong, China lijinze-hku@connect.hku.hk

Shuo Yang

The University of Hong Kong Hong Kong, China shuoyang.ee@gmail.com

Wei Wang

Shenzhen MSU-BIT University Shenzhen, China ehomewang@ieee.org

Xiping Hu

Beijing Institute of Technology Beijing, China huxp@bit.edu.cn

Edith Ngai*

The University of Hong Kong Hong Kong, China chngai@eee.hku.hk

ABSTRACT

Graph Collaborative Filtering (GCF) has emerged as a dominant paradigm in modern recommendation systems, excelling at modeling complex user-item interactions and capturing high-order collaborative signals through graph-structured learning. Most existing GCF models predominantly rely on simplified graph architectures like LightGCN, which strategically remove feature transformation and activation functions from vanilla graph convolution networks. Through systematic analysis, we reveal that feature transformation in message propagation can enhance model representation, though at the cost of increased training difficulty. To this end, we propose FourierKAN-GCF, a novel GCN framework that adopts Fourier Kolmogorov-Arnold Networks as efficient transformation modules within graph propagation layers. This design enhances model representation while decreasing training difficulty. Our FourierKAN-GCF can achieve higher recommendation performance than most widely used GCF backbone models. In addition, it can be integrated into existing advanced self-supervised models as a backbone, replacing their original backbone to achieve enhanced performance. Extensive experiments on three public datasets demonstrate the superiority of FourierKAN-GCF.

KEYWORDS

Recommendation, Collaborative Filtering, Graph Neural Network, Kolmogorov-Arnold Network, Fourier Coefficients

ACM Reference Format:

Jinfeng Xu, Zheyu Chen, Jinze Li, Shuo Yang, Wei Wang, Xiping Hu, and Edith Ngai*. 2025. Enhancing Graph Collaborative Filtering with FourierKAN Feature Transformation. In Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3746252.3760909

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '25, November 10-14, 2025, Seoul, Republic of Korea

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 979-8-4007-2040-6/2025/11...\$15.00 https://doi.org/10.1145/3746252.3760909

1 INTRODUCTION

Recommender systems are widely used to alleviate information overload on the Web [2–4, 8, 20, 26], aiming to recommend suitable items for users based on their historical behavior. Collaborative filtering (CF) addresses this by learning user preferences from similar users. Recently, graph-based CF (GCF) models [7, 10, 18, 22–25] have achieved notable success, as user-item interactions are naturally graph-structured. For example, NGCF [18] adopts whole standard GCN in recommender systems, which retains the feature transformation and nonlinear operation. However, LightGCN [7] states that both feature transformation and nonlinear operation are unnecessary in the recommendation field and further proposes a lightweight GCN. Consequently, most subsequent GCF models [1, 2, 21] adopt LightGCN as the backbone for further exploration.

However, there are many unfair analyses of NGCF in LightGCN: a) There are two different feature transformations in NGCF, but LightGCN removed them without performing separate fine-grained ablation experiments. b) LightGCN also removed the interaction information representation part of NGCF without giving reasons. Therefore, we question: 'Is feature transformation during message passing in GCN really unnecessary in recommendation?'

We provide an empirical analysis for NGCF and LightGCN in Section 2. Then, we state that feature transformation in NGCF can enhance the interaction representation and boost the performance of GCN, but increases the training difficulty. In this work, we introduce a simple yet powerful graph-based recommendation model called FourierKAN-GCF. Specifically, FourierKAN-GCF incorporates a unique Fourier Kolmogorov-Arnold Network (KAN) in place of the traditional multilayer perceptron (MLP) within the feature transformation during message passing in GCNs. This substitution enhances the representational capabilities and reduces the difficulty of training for GCFs. FourierKAN-GCF can achieve higher recommendation performance than most widely-used GCF backbone models. In addition, FourierKAN-GCF can be integrated into existing advanced self-supervised models as a backbone, replacing their original backbone to achieve enhanced performance. Extensive experiments on public datasets demonstrate the superiority of FourierKAN-GCF over state-of-the-art methods. Our work is intended to reawaken researchers' thinking about feature transformation in GCF, rather than arbitrarily removing it.

2 PRELIMINARY

2.1 NGCF Brief

NGCF [18] retains feature transformation and nonlinear operation during the message passing in GCN, formally:

$$\mathbf{e}_{u}^{(l+1)} = \sigma(\mathbf{W}_{1}\mathbf{e}_{u}^{(l)} + \sum_{i \in \mathcal{N}_{u}} \frac{\mathbf{W}_{1}\mathbf{e}_{i}^{(l)} + \mathbf{W}_{2}(\mathbf{e}_{i}^{(l)} \odot \mathbf{e}_{u}^{(l)})}{\sqrt{|\mathcal{N}_{u}||\mathcal{N}_{i}|}}),$$

$$\mathbf{e}_{i}^{(l+1)} = \sigma(\mathbf{W}_{1}\mathbf{e}_{i}^{(l)} + \sum_{u \in \mathcal{N}_{i}} \frac{\mathbf{W}_{1}\mathbf{e}_{u}^{(l)} + \mathbf{W}_{2}(\mathbf{e}_{u}^{(l)} \odot \mathbf{e}_{i}^{(l)})}{\sqrt{|\mathcal{N}_{u}||\mathcal{N}_{i}|}}),$$
(1)

where $\mathbf{e}_u^{(l)}$ and $\mathbf{e}_i^{(l)}$ represent the embedding of user u and item i after l layers message propagation, respectively. \mathcal{N}_u and \mathcal{N}_i denote the interacted item set with u and interacted user set with i, respectively. \mathbf{W}_1 and \mathbf{W}_2 are two trainable weight matrices to perform feature transformation in each layer. σ is the nonlinear activation function. It is worth noting that $\mathbf{W}_1\mathbf{e}_i^{(l)}$ and $\mathbf{W}_1\mathbf{e}_u^{(l)}$ can be regarded as the aggregated representation from neighbors, while $\mathbf{W}_2(\mathbf{e}_i^{(l)}\odot\mathbf{e}_u^{(l)})$ and $\mathbf{W}_2(\mathbf{e}_u^{(l)}\odot\mathbf{e}_i^{(l)})$ can be regarded as the aggregated representation information. The final embeddings are calculated by $\hat{\mathbf{e}}_u = \{\mathbf{e}_u^{(1)} || \mathbf{e}_u^{(2)} || \dots || \mathbf{e}_u^{(L)} \}$ and $\hat{\mathbf{e}}_i = \{\mathbf{e}_i^{(1)} || \mathbf{e}_i^{(2)} || \dots || \mathbf{e}_u^{(L)} \}$, where || is the concatenation operation. NGCF only concatenates layer-1 to layer-L and ignores ego layer-0, since ego layer-0 has already been considered in the first term $\mathbf{W}_1\mathbf{e}_u^{(l)}$ and $\mathbf{W}_1\mathbf{e}_i^{(l)}$ of message passing and propagation.

2.2 LightGCN Brief

LightGCN [7] analyzes the feature transformation and nonlinear operation in NGCF. It offers four observations: a) Removing the entire feature transformation, including \mathbf{W}_1 and \mathbf{W}_2 , leads to consistent improvements over NGCF. b) Removing only nonlinear operation σ will lead to a small deterioration of performance. c) Removing both entire feature transformation and nonlinear operation can improve performance significantly. d) The deterioration of NGCF stems from the training difficulty rather than over-fitting.

To this end, LightGCN removes feature transformation and non-linear operation during the message passing in GCN. Formally, the user-item graph to propagate embeddings as:

$$\mathbf{e}_{u}^{(l+1)} = \sum_{i \in \mathcal{N}_{u}} \frac{\mathbf{e}_{i}^{(l)}}{\sqrt{|\mathcal{N}_{u}||\mathcal{N}_{i}|}}, \quad \mathbf{e}_{i}^{(l+1)} = \sum_{u \in \mathcal{N}_{i}} \frac{\mathbf{e}_{u}^{(l)}}{\sqrt{|\mathcal{N}_{u}||\mathcal{N}_{i}|}}, \quad (2)$$
where $\mathbf{e}_{u}^{(l)}$ and $\mathbf{e}_{i}^{(l)}$ represent the embedding of user u and item

where $\mathbf{e}_u^{(l)}$ and $\mathbf{e}_i^{(l)}$ represent the embedding of user u and item i after l layers message propagation, respectively. \mathcal{N}_u and \mathcal{N}_i denote the interacted item set with u and interacted user set with i, respectively. LightGCN only retains the aggregation of neighbors' representation but removes the aggregation of interaction information representation. The final embeddings are calculated as $\hat{\mathbf{e}}_u = \sum_{l=0}^L \frac{1}{L+1} \mathbf{e}_u^{(l)}$ and $\hat{\mathbf{e}}_i = \sum_{l=0}^L \frac{1}{L+1} \mathbf{e}_i^{(l)}$, where L is the total layer number. LightGCN considers the ego layer-0 cause it removes the first term $\mathbf{W}_1 \mathbf{e}_u^{(l)}$ and $\mathbf{W}_1 \mathbf{e}_i^{(l)}$.

We argue that observation (d) is the main reason why LightGCN performs better than NGCF in most cases. Observations (a-c) were not sufficient to verify that the entire feature transformation and nonlinear operation do not contribute to feature extraction. Besides, we argue that LightGCN verified the feature transformation not contributing to learning better features by comparing NGCF

with removed both \mathbf{W}_1 and the whole interaction representation aggregation part $\mathbf{W}_2(\mathbf{e}_i^{(l)} \odot \mathbf{e}_u^{(l)})$, which is not a fair comparison. We point out that $\mathbf{e}_u^{(l)}$ and $\mathbf{e}_i^{(l)}$ naturally contain information about user preferences and item properties that can be adequately described through multiple feature dimensions. Therefore, \mathbf{W}_1 is an unnecessary feature transformation part. However, we point out that the interaction representation aggregation part $\mathbf{W}_2(\mathbf{e}_i^{(l)} \odot \mathbf{e}_u^{(l)})$ contains valuable interaction information, which can not be easily extracted by the heuristic rule. In this case, \mathbf{W}_2 is a necessary feature transformation part that contributes to feature extraction. In the next subsection, we provide an empirical analysis.

Table 1: Performance comparison of LightGCN, NGCF, and six variants of NGCF in terms of Recall@20 (R@20) and NDCG@20 (N@20).

Datasets	MO	ос	Am	azon	Gowalla			
Metrics	R@20 N@20		R@20	N@20	R@20	N@20		
LightGCN	0.3307	0.1811	0.0447	0.0227	0.1830	0.1152		
NGCF	0.3361	0.1894	0.0379	0.0196	0.1755	0.1013		
NGCF-f1	0.3377↑	0.1926↑	$0.0414\uparrow$	0.0209↑	0.1791	0.1081		
NGCF-f2	0.3357↓	0.1897↑	0.0388↑	$0.0192 \downarrow$	0.1764↑	0.1020↑		
NGCF-i	0.3301↓	0.1824↓	0.0362↓	$0.0181 \downarrow$	0.1739↓	$0.1010 \downarrow$		
NGCF-f1-f2	0.3374↑	0.1913↑	0.0407	0.0205↑	0.1784↑	0.1077		
NGCF-f1-i	0.3332↓	0.1868↓	0.0372↓	0.0190↓	0.1752↓	0.1013 -		
NGCF-n	0.3343↓	0.1878↓	0.0373↓	0.0196 -	0.1750↓	$0.1008 \downarrow$		

2.3 Re-Analysis for NGCF

We implement six variants of NGCF: 1) NGCF-f1 removes feature transformation matrix \mathbf{W}_1 . 2) NGCF-f2 removes feature transformation matrix \mathbf{W}_2 . 3) NGCF-i removes the whole interaction representation aggregation part $\mathbf{W}_2(\mathbf{e}_i^{(l)}\odot\mathbf{e}_u^{(l)})$. 4) NGCF-f1-f2 removes both \mathbf{W}_1 and \mathbf{W}_2 . 5) NGCF-f1-i removes both \mathbf{W}_1 and $\mathbf{W}_2(\mathbf{e}_i^{(l)}\odot\mathbf{e}_u^{(l)})$. 6) NGCF-n removes the nonlinear operation σ .

We keep all optimal hyper-parameter settings as the NGCF reported. As Table 1 shows, we conclude the findings: a) For all three datasets, removing the feature transformation matrix W_1 will lead to observed improvements. Therefore, the feature transformation matrix W_1 is unnecessary. b) For the MOOC dataset, removing the feature transformation matrix W2 leads to a slight deterioration. However, for the other two datasets, it will lead to a small improvement. Moreover, removing both W_1 and W_2 will lead to a small deterioration than only removing W_1 for all datasets. We own this phenomenon to that removing only W1 can make the model optimization focus on training W_2 , and remove both W_1 and W2 will lose the representation power from feature transformation. c) Removing the whole interaction representation aggregation part $\mathbf{W}_2(\mathbf{e}_i^{(l)} \odot \mathbf{e}_u^{(l)})$ will lead to a performance degradation on all datasets. Additionally, removing both \mathbf{W}_1 and $\mathbf{W}_2(\mathbf{e}_i^{(l)}\odot\mathbf{e}_u^{(l)})$ will make a obviously degration than only remove W_1 and even NGCF. This shows that this part can bring valuable information about interaction. d) Nonlinear operation only brings a small positive effect. e) NGCF outperforms LightGCN on the MOOC dataset verifying that NGCF has better representation power than LightGCN, but the training is more difficult [7]. Then, we can draw some conclusions. First, the feature transformation matrix W_1 is unnecessary

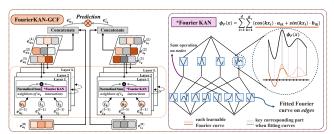


Figure 1: Overview of FourierKAN-GCF.

for NGCF. Besides, the feature transformation matrix \mathbf{W}_2 and the whole interaction representation aggregation part $\mathbf{W}_2(\mathbf{e}_i^{(l)} \odot \mathbf{e}_u^{(l)})$ is beneficial for feature extraction. Last, nonlinear operation only has a minor influence.

To this end, a feature transformation approach, simpler to train than MLPs but with strong representational capabilities, could be effective in graph learning for recommendations. KAN [14] is regarded as a promising alternative to MLP. While KAN shares the same theoretically unlimited representational capacity as MLP [19], the practical representational capacity of MLP is constrained by the hidden dimensionality. In contrast, the practical representational capacity of KAN depends on its ability to fit trainable activation functions. Our work can be seen as exploring adopting KAN as a feature transformation approach within GCF.

3 METHODOLOGY

We detail the overview of FourierKAN-GCF¹ in Figure 1. FourierKAN-GCF introduces a promising feature transformation for GCF, which significantly boosts performance and simplifies training. Note that FourierKAN-GCF can be adopted as the backbone in existing advanced self-supervised models.

3.1 Kolmogorov-Arnold Network (KAN)

Kolmogorov-Arnold Network [14] is a promising alternative to Multi-Layer Perceptron (MLP). MLP is inspired by the universal approximation theorem [5]. KAN focuses on the Kolmogorov-Arnold representation theorem [12]. Specifically, unlike MLPs, which have fixed activation functions on nodes, KANs contain learnable activation functions on edges (weights). This unique architecture enables KANs to learn nonlinear functions more effectively. Formally:

KAN =
$$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q(\sum_{p=1}^n \phi_{q,p}(\mathbf{x}_p)),$$
 (3)

where $\phi_{q,p}$ are univariate functions that map each input variable \mathbf{x}_p such $\phi_{q,p}:[0,1]\to\mathbb{R}$ and $\phi_q:\mathbb{R}\to\mathbb{R}$. $\phi_{q,p}(\mathbf{x}_p)$ is trainable activation function. In the KolmogovArnold theorem, the inner functions form a KAN layer with $n_{in}=n$ and $n_{out}=2n+1$, and the outer functions form a KAN layer with $n_{in}=2n+1$ and $n_{out}=n$. So the Kolmogorov-Arnold representations in this formula are simply compositions of two KAN layers. A useful trick is that it includes a basis function $b(\mathbf{x})$ such that the activation function $\mathrm{silu}(\mathbf{x})=\frac{\mathbf{x}}{1+e^{-\mathbf{x}}}$. $\phi(\mathbf{x})=\mathbf{w}(b(\mathbf{x})+\mathrm{spline}(\mathbf{x}))$ is the sum of the basis function $b(\mathbf{x})$ and function $\mathrm{spline}(\mathbf{x})=\sum_i c_i B_i(\mathbf{x})$ is a linear combination of B-splines, where c_i is a trainable parameter.

3.2 Fourier KAN

To further reduce the training difficulty and be able to adopt to different scenarios. Our goal can be converted into finding the split from a complex function into multiple relatively simple nonlinear functions. Naturally, the Fourier Coefficients [16] is a potential choice. Therefore, we propose the following equation:

$$\phi_F(\mathbf{x}) = \sum_{i=1}^d \sum_{k=1}^g (\cos(k\mathbf{x}_i) \cdot a_{ik} + \sin(k\mathbf{x}_i) \cdot b_{ik}), \qquad (4)$$

where d is the dimension number of features. Fourier coefficients a_{ik} and b_{ik} are trainable. Hyper-parameter g is the gridsize, which plays a critical role in controlling the number of terms (frequencies) used in the Fourier series expansion. Specifically, g determines how many different sine and cosine terms are included in the Fourier Coefficients corresponding to each input dimension. The Fourier Coefficients has a significant advantage in computational efficiency and reduces the training difficulty caused by the spline function.

3.3 FourierKAN-GCF

The message passing in FourierKAN-GCF is defined as:

$$\mathbf{e}_{u}^{(l+1)} = \sigma(\mathbf{e}_{u}^{(l)} + \sum_{i \in \mathcal{N}_{u}} \frac{\mathbf{e}_{i}^{(l)} + \phi_{F}(\mathbf{e}_{i}^{(l)} \odot \mathbf{e}_{u}^{(l)})}{\sqrt{|\mathcal{N}_{u}||\mathcal{N}_{i}|}}),$$

$$\mathbf{e}_{i}^{(l+1)} = \sigma(\mathbf{e}_{i}^{(l)} + \sum_{u \in \mathcal{N}_{i}} \frac{\mathbf{e}_{u}^{(l)} + \phi_{F}(\mathbf{e}_{u}^{(l)} \odot \mathbf{e}_{i}^{(l)})}{\sqrt{|\mathcal{N}_{u}||\mathcal{N}_{i}|}}),$$
(5)

where $\phi_F(\cdot)$ is simplified single layer Fourier KAN function. We remove the unnecessary transform matrix \mathbf{W}_1 in NGCF and utilize our Fourier KAN function to replace the transform matrix \mathbf{W}_2 . The final user embedding and item embedding are calculated by $\hat{\mathbf{e}}_u = \{\mathbf{e}_u^{(1)}||\mathbf{e}_u^{(2)}||...||\mathbf{e}_u^{(L)}\}$ and $\hat{\mathbf{e}}_i = \{\mathbf{e}_i^{(1)}||\mathbf{e}_i^{(2)}||...||\mathbf{e}_0^{(L)}\}$, where || is the concatenation operation.

3.4 Dropout Strategies

To mitigate overfitting in FourierKAN-GCF, we employ message dropout and node dropout strategies, similar to NGCF [18]. In message dropout, a fraction $1 - p_m$ of message passing in Eq. 5 is randomly set to zero, where p_m is the dropout ratio. In node dropout, $1 - p_n$ of the nodes in the matrix are randomly dropped, with p_n as the dropout ratio.

3.5 Model Training

For model optimization, we adopt the Bayesian Personalized Ranking (BPR) [17] loss function as our optimization criterion. The core objective of BPR is to enhance the divergence in the predictive preference between positive and negative items within each user-item triplet $(u, i_p, i_n) \in \mathcal{D}$, where \mathcal{D} signifies the collection of training data, the term positive item p pertains to an item with which the user u has interacted, whereas the negative item n is selected randomly from the pool of items without interaction with user u.

$$\mathcal{L} = \sum_{(u, i_p, i_n) \in \mathcal{D}} -\ln \sigma(\hat{\mathbf{e}}_u^T \hat{\mathbf{e}}_{i_p} - \hat{\mathbf{e}}_u^T \hat{\mathbf{e}}_{i_n}) + \lambda \|\Theta\|^2, \tag{6}$$

where λ controls the L_2 regularization strength, σ is the Sigmoid function, and Θ denotes model parameters.

 $^{^{1}} Code \ is \ available \ at: https://anonymous.4 open.science/r/FourierKAN-GCF-r.$

Table 2: Statistics of experimental datasets.

Dataset	# Users	# Items	# Interaction	Sparsity
MOOC	82,535	1,302	458,453	99.57%
Amazon	50,677	16,897	454,529	99.95%
Gowalla	29,859	40,989	1,027,464	99.92%

4 EXPERIMENT

In this section, we compare the performance of FourierKAN-GCF with popular graph-based backbone models and demonstrate its compatibility with advanced self-supervised graph-based models.

4.1 Datasets and Evaluation Metrics

We conduct experiments on three real-world datasets: MOOC, Amazon Video Games (Amazon), and Gowalla. Details can be found in Table 2. For a fair comparison, we sort all observed user-item interactions chronologically based on the interaction timestamps. Then, we split each dataset with a ratio of 7:1:2 for training, validation, and testing. Regarding evaluation metrics, we adopt two well-established metrics [9]: Recall@K (R@K) and NDCG@K (N@K).

4.2 Baselines and Experimental Settings

To verify the effectiveness of FourierKAN-GCF, we select five GCN-based backbone models (NGCF [18], LR-GCCF [2], LightGCN [7], UltraGCN [15], and IMP-GCN [13]) as baselines. Moreover, FourierKAN-GCF can also be a more powerful backbone to enhance existing advanced self-supervised enhanced graph-based models (SimGCL [27], LightGCL [1], and RecDCL [28]). For a fair comparison, we fix the embedding size of both users and items to 64 for all models, initialize embedding parameters with the Xavier initialization [6], and use Adam [11] as optimizer. Besides, we tune the hyper-parameters of each baseline following their published papers. For FourierKAN-GCF, we fix the $\lambda = 1$ for L_2 regularization, and tune layer number L from 1 to 4. The gridsize g is searched from $\{1, 2, 4, 8\}$. Message and node dropout ratios p_m and p_n is searched from $\{0.0, 0.1, 0.2, 0.3\}$.

4.3 Performance Comparison

The results of our experiments are listed in Table 3. Our FourierKAN-GCF outperforms all baselines in three datasets across various metrics. Moreover, FourierKAN-GCF is second only to LightGCN in efficiency, which demonstrates it effectively reduces the training difficulty associated with feature transformation. We owe our superiority to FourierKAN, which is easier to train and has greater representation power than MLP. Note that FourierKAN-GCF can adjust the training difficulty by adjusting the grid size g.

4.4 Compatibility Analysis

We further evaluate the compatibility of FourierKAN-GCF with advanced self-supervised graph-based models by replacing their backbones. As shown in Table 4, adopting FourierKAN-GCF significantly improves performance, highlighting the importance of feature transformation in GCF and establishing FourierKAN-GCF as an effective solution.

4.5 Ablation Study

Table 3 also demonstrates the significance of dropout strategies. We use w/o MD and w/o ND to denote without message dropout and without node dropout, respectively. This ablation study shows that both message dropout and node dropout play a distinct role in improving model representation power and model robustness. We use KAN-GCF to denote a variant that utilizes our standard KAN function to replace transform matrix \mathbf{W}_2 in NGCF. It shows that standard KAN is better than MLP but worse than our Fourier KAN.

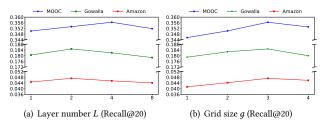


Figure 2: Effective of layer number L and grid size q.

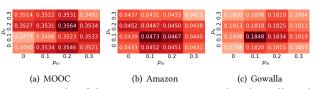


Figure 3: Study of dropout ratio pair p_m and p_n (Recall@20).

4.6 Sensitivity Analysis

To analyze the hyper-parameter sensitivity of FourierKAN-GCF, we test the performance of FourierKAN-GCF on three datasets with different hyper-parameters. The optimal layer number L is 3 on all datasets, shown in Figure 2(a). Besides, Figure 2(b) demonstrates that for MOOC dataset, the optimal (p_m, p_n) pair is (0.2, 0.2), and for Amazon and Gowalla datasets, the optimal pairs are all (0.1, 0.1). As Figure 3 illustrated, for the relatively dense dataset MOOC, g=4 is the best grid size. For the relatively sparse datasets Amazon and Gowalla, g=2 is the best grid size. This further demonstrates that FourierKAN can be easily adapted to various datasets.

5 CONCLUSION

In this paper, we revisit feature transformation and nonlinear operations in the message-passing mechanism of GCNs. While feature transformation enhances interaction representation and boosts performance, it increases training complexity. To this end, we introduce a new feature transformation for GCF called FourierKAN. Inspired by KAN, FourierKAN employs the Fourier Coefficients instead of the Spline function in standard KAN. We further propose a simple yet effective GCF model (FourierKAN-GCF), which reduces the difficulty of training. In addition, FourierKAN-GCF can be integrated into existing advanced self-supervised models as a backbone, replacing their original backbone to achieve enhanced performance. Extensive experiments on public datasets verify the superiority of our model over the advanced methods.

GENAI USAGE DISCLOSURE

No GenAI tools were used in any stage of the research, nor in the writing.

Table 3: Performance comparison of baselines and FourierKAN-GCF in terms of R@K and N@K. The superscript * indicates the improvement is statistically significant where the p-value is less than 0.05. #T denotes seconds per epoch.

Datasets			MOOC		Amazon					Gowalla					
Metrics	R@20	R@50	N@20	N@50	#T	R@20	R@50	N@20	N@50	#T	R@20	R@50	N@20	N@50	#T
NGCF	0.3361	0.4799	0.1894	0.2349	5.2s	0.0379	0.0782	0.0196	0.0274	4.4s	0.1755	0.2811	0.1013	0.1270	37.5s
LR-GCCF	0.3336	0.4809	0.1938	0.2294	5.1s	0.0440	0.0815	0.0224	0.0317	4.7s	0.1803	0.2971	0.1101	0.1369	40.1s
LightGCN	0.3307	0.4773	0.1811	0.2217	4.3s	0.0447	0.0844	0.0227	0.0326	3.5s	0.1830	0.3044	0.1152	0.1414	26.4s
UltraGCN	0.3194	0.4701	0.1962	0.2307	4.9s	0.0459	0.0844	0.0230	0.0331	4.0s	0.1798	0.2909	0.1059	0.1328	33.8s
IMP-GCN	0.2788	0.4183	0.1717	0.2057	54.2s	0.0461	0.0839	0.0232	0.0323	44.9s	0.1808	0.2932	0.1060	0.1345	143.6s
KAN-GCF	0.3417	0.4984	0.2024	0.2396	4.9s	0.0451	0.0837	0.0229	0.0325	4.0s	0.1922	0.3023	0.1142	0.1403	32.9s
FourierKAN-GCF	0.3564	0.5065	0.2147	0.2462	4.6s	0.0473	0.0856	0.0252	0.0342	3.7s	0.1962	0.3077	0.1179	0.1436	29.8s
w/o MD	0.3523	0.4912	0.2116	0.2449	4.5s	0.0452	0.0825	0.0221	0.0314	3.7s	0.1920	0.3031	0.1150	0.1408	29.5s
w/o ND	0.3527	0.4839	0.2071	0.2439	4.5s	0.0452	0.0809	0.0219	0.0309	3.6s	0.1908	0.3017	0.1133	0.1389	29.6s

Table 4: Performance comparison of different backbones on advanced recommendation models in terms of R@K and N@K.

Datasets			MC	OC			Ama	azon		Gowalla			
Metrics	Backbones	R@20	R@50	N@20	N@50	R@20	R@50	N@20	N@50	R@20	R@50	N@20	N@50
SimGCL	LightGCN	0.3503	0.5032	0.2109	0.2428	0.0462	0.0837	0.0230	0.0327	0.2028	0.3126	0.1184	0.1459
	FourierKAN-GCF	0.3639	0.5188	0.2216	0.2543	0.0482	0.0868	0.0258	0.0352	0.2084	0.3163	0.1234	0.1521
LightGCL	LightGCN	0.2742	0.4139	0.1697	0.2018	0.0453	0.0818	0.0225	0.0318	0.1888	0.2976	0.1081	0.1348
	FourierKAN-GCF	0.2823	0.4291	0.1733	0.2072	0.0479	0.0864	0.0250	0.0347	0.2007	0.3100	0.1198	0.1452
RecDCL	LightGCN	0.3531	0.5009	0.2113	0.2423	0.0469	0.0848	0.0235	0.0334	0.1993	0.3052	0.1160	0.1433
	FourierKAN-GCF	0.3608	0.5120	0.2175	0.2514	0.0484	0.0863	0.0253	0.0349	0.2051	0.3113	0.1203	0.1472

ACKNOWLEDGMENTS

This work was supported by the Hong Kong UGC General Research Fund no. 17203320 and 17209822, and the project grants from the HKU-SCF FinTech Academy.

REFERENCES

- [1] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *The Eleventh International Conference on Learning Representations*.
- [2] 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 Proceedings of the AAAI conference on artificial intelligence, Vol. 34. 27–34.
- [3] Zheyu Chen, Jinfeng Xu, and Haibo Hu. 2025. Don't Lose Yourself: Boosting Multimodal Recommendation via Reducing Node-neighbor Discrepancy in Graph Convolutional Network. In ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 1–5.
- [4] Zheyu Chen, Jinfeng Xu, Yutong Wei, and Ziyue Peng. 2025. Squeeze and Excitation: A Weighted Graph Contrastive Learning for Collaborative Filtering. In Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2769–2773.
- [5] George Cybenko. 1989. Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems 2, 4 (1989), 303–314.
- [6] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 249–256.
- [7] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. 639–648.
- [8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web. 173–182.
- [9] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems (TOIS) 20, 4 (2002), 422–446.
- [10] Yangqin Jiang, Chao Huang, and Lianghao Huang. 2023. Adaptive graph contrastive learning for recommendation. In Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining. 4252–4261.
- [11] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations.
- [12] Andrey Nikolaevich Kolmogorov. 1961. On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables. American Mathematical Society.
- [13] Fan Liu, Zhiyong Cheng, Lei Zhu, Zan Gao, and Liqiang Nie. 2021. Interest-aware message-passing GCN for recommendation. In Proceedings of the web conference

- 2021. 1296-1305.
- [14] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. 2024. Kan: Kolmogorov-arnold networks. arXiv preprint arXiv:2404.19756 (2024).
- [15] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: ultra simplification of graph convolutional networks for recommendation. In Proceedings of the 30th ACM international conference on information & knowledge management. 1253–1262.
- [16] Henri J Nussbaumer and Henri J Nussbaumer. 1982. The fast Fourier transform. Springer.
- [17] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. 452–461.
- [18] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval. 165–174.
- [19] Yixuan Wang, Jonathan W Siegel, Ziming Liu, and Thomas Y Hou. 2024. On the expressiveness and spectral bias of KANs. arXiv preprint arXiv:2410.01803 (2024).
- [20] Jinfeng Xu, Zheyu Chen, Jinze Li, Shuo Yang, Hewei Wang, Yijie Li, Mengran Li, Puzhen Wu, and Edith CH Ngai. 2025. MDVT: Enhancing Multimodal Recommendation with Model-Agnostic Multimodal-Driven Virtual Triplets. arXiv preprint arXiv:2505.16665 (2025).
- [21] Jinfeng Xu, Zheyu Chen, Jinze Li, Shuo Yang, Hewei Wang, and Edith CH Ngai. 2024. AlignGroup: Learning and Aligning Group Consensus with Member Preferences for Group Recommendation. In Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. 2682–2691.
- [22] Jinfeng Xu, Zheyu Chen, Zixiao Ma, Jiyi Liu, and Edith CH Ngai. 2024. Improving Consumer Experience With Pre-Purify Temporal-Decay Memory-Based Collaborative Filtering Recommendation for Graduate School Application. IEEE Transactions on Consumer Electronics (2024).
- [23] Jinfeng Xu, Zheyu Chen, Wei Wang, Xiping Hu, Sang-Wook Kim, and Edith CH Ngai. 2025. COHESION: Composite Graph Convolutional Network with Dual-Stage Fusion for Multimodal Recommendation. In Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval. 1830–1839.
- [24] Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, and Edith CH Ngai. 2025. The Best is Yet to Come: Graph Convolution in the Testing Phase for Multimodal Recommendation. arXiv preprint arXiv:2507.18489 (2025).
- [25] Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, Hewei Wang, and Edith CH Ngai. 2025. Mentor: multi-level self-supervised learning for multimodal recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 39. 12908–12917.
- [26] Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, Wei Wang, Xiping Hu, Steven Hoi, and Edith Ngai. 2025. A Survey on Multimodal Recommender Systems: Recent Advances and Future Directions. arXiv preprint arXiv:2502.15711 (2025).
- [27] 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 Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval. 1294–1303.

[28] Dan Zhang, Yangliao Geng, Wenwen Gong, Zhongang Qi, Zhiyu Chen, Xing Tang, Ying Shan, Yuxiao Dong, and Jie Tang. 2024. RecDCL: Dual Contrastive

Learning for Recommendation. In Proceedings of the ACM on Web Conference 2024. 3655-3666.