AFDGCF: Adaptive Feature De-correlation Graph Collaborative Filtering for Recommendations

Wei Wu
School of Data Science, University of Science and Technology of China
Hefei, China
urara@mail.ustc.edu.cn

Chao Wang*
Guangzhou HKUST Fok Ying Tung Research Institute
Guangzhou, China
School of Computer Science and Technology, University of Science and Technology of China
Hefei, China
chadwang2012@gmail.com

Dazhong Shen
Shanghai Artificial Intelligence Laboratory
Shanghai, China
dazh.shen@gmail.com

Chuan Qin
Career Science Lab, BOSS Zhipin
Beijing, China
PBC School of Finance, Tsinghua University
Beijing, China
chuanqin0426@gmail.com

Liyi Chen
School of Data Science, University of Science and Technology of China
Hefei, China
liyichencly@gmail.com

Hui Xiong*
Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
Department of Computer Science and Engineering, The Hong Kong University of Science and Technology
Hong Kong SAR, China
xionghui@ust.hk

ABSTRACT

Collaborative filtering methods based on graph neural networks (GNNs) have witnessed significant success in recommender systems (RS), capitalizing on their ability to capture collaborative signals within intricate user-item relationships via message-passing mechanisms. However, these GNN-based RS inadvertently introduce excess linear correlation between user and item embeddings, contradicting the goal of providing personalized recommendations. While existing research predominantly ascribes this flaw to the over-smoothing problem, this paper underscores the critical, often overlooked role of the over-correlation issue in diminishing the effectiveness of GNN representations and subsequent recommendation performance. Up to now, the over-correlation issue remains unexplored in RS. Meanwhile, how to mitigate the impact of over-correlation while preserving collaborative filtering signals is a significant challenge. To this end, this paper aims to address the aforementioned gap by undertaking a comprehensive study of the over-correlation issue in graph collaborative filtering models. Firstly, we present empirical evidence to demonstrate the widespread prevalence of over-correlation in these models. Subsequently, we dive into a theoretical analysis which establishes a pivotal connection between the over-correlation and over-smoothing issues. Leveraging these insights, we introduce the **A**daptive **F**eature **D**e-correlation **G**raph **C**ollaborative **F**iltering (AFDGCF) framework, which dynamically applies correlation penalties to the feature dimensions of the representation matrix, effectively alleviating both over-correlation and over-smoothing issues. The efficacy of the proposed framework is corroborated through extensive experiments conducted with four representative graph collaborative filtering models across four publicly available datasets. Our results show the superiority of AFDGCF in enhancing the performance landscape of graph collaborative filtering models. The code is available at https://github.com/U-rara/AFDGCF.

CCS CONCEPTS

- **Information systems → Collaborative filtering**.

KEYWORDS

Recommender Systems, Collaborative Filtering, Graph Neural Networks, Over-correlation, Over-smoothing

ACM Reference Format:
Wei Wu, Chao Wang, Dazhong Shen, Chuan Qin, Liyi Chen, and Hui Xiong. 2024. AFDGCF: Adaptive Feature De-correlation Graph Collaborative Filtering for Recommendations. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3626772.3657724

[*] Corresponding authors

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657724

1 INTRODUCTION

Recommender system (RS) has become widely adopted in various online applications as a solution to the challenge of information overload, by providing personalized item recommendations [45]. Among the popular RS techniques, collaborative filtering (CF) stands out for its ability to extract informative user and item representations from historical interactions [35]. With the rapid advancement of graph neural networks (GNNs) and their effectiveness in processing graph-structured data, GNN-based CF has emerged as a prominent research focus in RS, capturing collaborative signals in the high-order connectivity between users and items [41]. Despite the remarkable achievements facilitated by GNNs in RS, significant challenges persist in attaining high robustness and accuracy, such as feature over-correlation and over-smoothing problems.

In the context of GNN-based CF models, a single GNN layer predominantly considers the immediate neighboring nodes of users and items, potentially limiting the ability to capture deep collaborative signals. Addressing this limitation, conventional GNN-based CF models stack multiple GNN layers to expand their receptive fields. However, this practice can lead to performance degradation as the number of stacked layers increases [41, 52]. Prior studies on GNN-based CF [4, 24, 32, 52] often attribute this degradation to the widely discussed over-smoothing problem, where node representations tend to converge towards similarity with escalating layer count. However, it is important to note that another critical factor contributing to performance decline is the feature over-correlation issue. As the number of GNN layers grows, the dimensions of node representations' features become increasingly correlated [17], negatively impacting the quality of the learned embeddings. While over-correlation and over-smoothing exhibit similar tendencies and effects, their primary distinction lies in their focus on relationships: over-smoothing pertains to relationships between node representations (in the row direction of the representation matrix), while over-correlation pertains to relationships between feature dimensions of the representations (in the column direction of the representation matrix).

In the literature, while various GNN-based CF studies, including GCMC [1], SpectralCF [54], PinSage [50], and NGCF [42], have primarily concentrated on GNN structures, they have often overlooked the aforementioned issues. Subsequent investigations [2, 9, 16, 23, 25, 28, 47, 48] have started addressing these problems, with an emphasis on mitigating the over-smoothing issue but largely neglecting feature over-correlation. Consequently, several unresolved challenges remain in GNN-based CF. Firstly, the influence of feature over-correlation on RS and its relationship with over-smoothing remains unclear. If both issues concurrently impact the recommendation performance, it may be possible to leverage their interconnection to address both problems simultaneously. Secondly, devising dedicated strategies to alleviate feature over-correlation in GNN-based CF models remains an unexplored but essential avenue for enhancing embedding capability. Lastly, the varying severity of over-correlation and over-smoothing across different GNN layers underscores the importance of crafting layer-wise adaptive techniques to appropriately control feature learning.

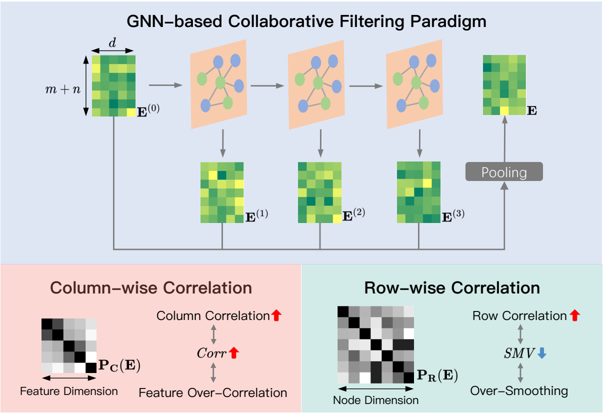

Figure 1: Illustration of the GNN-based CF, highlighting the challenges of over-correlation and over-smoothing.

To address these challenges, in this paper, we propose a comprehensive de-correlation paradigm for GNN-based CF models, named the <u>A</u>daptive <u>F</u>eature <u>D</u>e-correlation <u>G</u>raph <u>C</u>ollaborative <u>F</u>iltering (AFDGCF) framework. To the best of our knowledge, this is the first investigation into the impact of feature over-correlation issue in RS. We commence by providing empirical validation, demonstrating the prevalence of the feature over-correlation issue in GNN-based methods. Subsequently, through theoretical analysis, we establish a fundamental connection between feature over-correlation and over-smoothing. Building upon this insight, we propose the model-agnostic AFDGCF framework, which adaptively penalizes correlations among feature dimensions within the output representations of each GNN layer to mitigate the impact of over-correlation while preserving collaborative filtering signal. The efficacy of our approach is extensively validated through the experiments on different implementations employing diverse GNN-based CF methods and various publicly available datasets. These results demonstrate the general effectiveness of our AFDGCF framework in alleviating both the over-correlation and over-smoothing issues, and thus enhancing model performance. In summary, the main contributions of this paper are as follows: 1) We introduce a new perspective on the feature over-correlation problem in GNN-based CF. We theoretically discuss the relevance and differences between over-correlation and the commonly addressed over-smoothing problem. 2) We propose the Adaptive Feature De-correlation Graph Collaborative Filtering (AFDGCF) framework, a novel model-agnostic approach to mitigate feature over-correlation issue adaptively in existing GNN-based CF. 3) We present extensive experimental results using various public datasets to validate the effectiveness of the AFDGCF framework compared to many state-of-the-art methods.

2 RELATED WORK

This subsection presents pertinent literature relevant to the paper, encompassing GNN-based CF models and studies on over-correlation and over-smoothing issues.

2.1 GNN-based Collaborative Filtering

Within the area of RS, GNN-based CF emerged as a forefront avenue of research. It leverages the inherent higher-order connectivity present in historical user-item interactions to attain enhanced

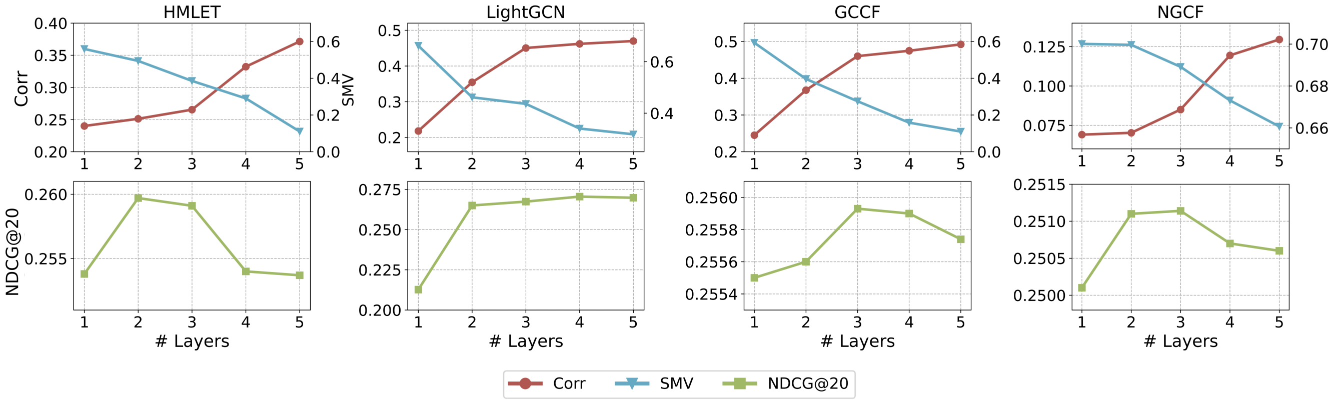

Figure 2: The figures at the top illustrate the feature correlation (measured by *Corr* on the left axis) and smoothness (measured by *SMV* on the right axis) of representations learned by state-of-the-art GNN-based CF models on the Movielens dataset. The bottom figures present the corresponding recommendation performance (measured by NDCG@20).

performance compared to conventional matrix factorization techniques [20, 27]. During the nascent phase, *Li and Chen* pioneered the exploration of the recommendation problem through a graph-oriented approach, treating CF as a link prediction task within bipartite graphs [21]. Subsequently, propelled by the progression of GNNs, researchers progressively delved into GNNs' integration within the realm of RS. As an illustration, GCMC [1] established a graph-based auto-encoder framework employing GNNs to facilitate the augmentation of the rating matrix. SpectralCF [54] directly performed spectral domain learning on the user-item bipartite graph, alleviating the cold-start problem in CF. By leveraging efficient random walks, PinSage [50] was the first to apply GNNs to industrial-scale RS. NGCF [42] injected the collaborative signal latent in the high-order connectivity between users and items into embeddings via propagation on the user-item graph. Building on this, LightGCN [16] and GCCF [9] subsequently streamlined the message-passing process, devising GNN paradigms tailored for CF tasks. In addition, there were also some efforts that attempted to make GNNs more suitable for CF tasks through unique designs. DGCF [43], in order to capture diverse intentions when users selected items, designed a graph disentangling module to decouple these factors and obtain disentangled representations. HMLET [19] attempted to amalgamate linear and non-linear methodologies by applying a gating module to choose linear or non-linear propagation for each user and item. Additionally, there were other efforts aimed at expanding GNN-based recommendations by delving into aspects such as training paradigms and application. [29, 30, 38–40, 51]

2.2 Over-smoothing and Over-correlation

The proliferation of layers in GNNs gives rise to two significant challenges: over-smoothing and over-correlation. The former leads to highly similar node representations, making it difficult to distinguish between them, while the latter results in redundant information among feature representations. Both issues elevate the risk of over-fitting in GNN-based models and curtail their expressive capacity. Currently, in the field of GNNs, numerous studies have been dedicated to exploring how these two issues affect the performance of deeper GNNs in graph-related tasks [4, 17, 24, 32, 52]. In the area of RS, various studies have also endeavored to mitigate the over-smoothing issue. One category of methodologies aims to streamline the process of message passing. For instance, LightGCN [16] eliminated feature transformation and non-linear activation from GNNs, and GCCF [9] incorporated a residual network structure while discarding non-linear activation. UltraGCN [25] skipped explicit message-passing through infinite layers. GCNII [10] preserved the non-linear activation in the residual network structure, and LMC [34] scaled deep GCNII to large-scale graphs with provable convergence. The second set of approaches performs feature propagation within subdivided sub-graphs. For example, IMP-GCN [23] employed graph convolutions within sub-graphs composed of similar-interest users and the items they interacted with. Similarly, Xia et al. proposed a graph resizing technique that recursively divides a graph into sub-graphs. Furthermore, additional investigations [2, 47, 48] endeavored to tackle the over-smoothing challenge in GNN-based CF using strategies including contrastive learning [7] and knowledge distillation [5]. Besides the aforementioned efforts to alleviate over-smoothing, it is noticeable that over-smoothing may not invariably be detrimental to RS. This is because the smoothness of embeddings plays a pivotal role in the effectiveness of GNN-based CF models [14, 16, 44].

Although significant progress has been made in the research of over-smoothing, there is currently a lack of studies on the over-correlation issue in GNN-based CF. In this paper, we analyze the impact of the over-correlation issue on GNN-based CF methods and propose a suitable solution.

3 PRELIMINARIES

In this section, we will introduce the paradigm of GNN-based CF. In general, the input of CF methods consists of a user set $\mathcal{U} = \{u_1, u_2, \cdots, u_p\}$, an item set $\mathcal{I} = \{i_1, i_2, \cdots, i_q\}$, and the interaction matrix between users and items $\mathbf{R} \in \{0, 1\}^{p \times q}$, where each element $r_{u,i} \in \mathbf{R}$ denotes the interaction behavior between user u and item i, such as a purchase or click, with 1 indicating an interaction has occurred and 0 indicating no interaction. In GNN-based CF, the interaction matrix above is reformulated into a user-item graph

$\mathcal{G} =< \mathcal{U} \cup \mathcal{I}, \mathbf{A} >$, where $\mathbf{A}$ represents the adjacency matrix:

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

Aggregation operations and update operations can represent the message-passing process on the user-item graph [46]:

$$\begin{aligned} \mathbf{e}_i^{(l+1)} &= \text{Updater}\left(\mathbf{e}_i^{(l)}, \text{Aggregator}\left(\left\{\mathbf{e}_u^l, \forall u \in \mathcal{N}_i\right\}\right)\right) \\ \mathbf{e}_u^{(l+1)} &= \text{Updater}\left(\mathbf{e}_u^{(l)}, \text{Aggregator}\left(\left\{\mathbf{e}_i^l, \forall i \in \mathcal{N}_u\right\}\right)\right) \end{aligned}, \tag{2}$$

where $\mathbf{e}_u^{(l)}, \mathbf{e}_i^{(l)}$ represents the representation of users or items at layer l. $\mathcal{N}_u, \mathcal{N}_i$ represent the neighboring nodes of user u or item i, respectively. Aggregator denotes the function responsible for aggregation operations, such as mean-pooling and attention mechanisms. Updater refers to the operation function that updates the current node representation, for instance, sum operation and nonlinear transformation.

After the message-passing process, node representation matrix:

$$\mathbf{E}^{(l)} = \left[\mathbf{e}_{u_1}^{(l)}, \cdots, \mathbf{e}_{u_p}^{(l)}, \mathbf{e}_{i_1}^{(l)}, \cdots, \mathbf{e}_{i_q}^{(l)}\right] \tag{3}$$

can be obtained at each layer (as shown in Figure 1). The final node representation matrix, denoted by $\mathbf{E}$, can be obtained through various pooling operations, such as mean/sum/weighted-pooling and concatenation:

$$\mathbf{E} = \text{Pooling}(\mathbf{E}^{(0)}, \mathbf{E}^{(1)}, \cdots, \mathbf{E}^{(L)}), \tag{4}$$

where L denotes the number of layers, $\mathbf{E}^{(0)}$ represents learnable embedding matrix. Without losing generalization, this paper's analysis will focus on LightGCN [16], one of the most representative methods in GNN-based CF. Its message-passing process can be formulated as follows:

$$\mathbf{E}^{(l+1)} = (\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{E}^{(l)}, \tag{5}$$

where $\mathbf{D}$ is the degree matrix of $\mathbf{A}$.

4 THE GENERAL AFDGCF FRAMEWORK

In this section, we commence with an empirical analysis of the impact of over-correlation and over-smoothing issues on GNN-based CF models. Subsequently, we delve into their theoretical connection. Building upon this, we propose an Adaptive Feature De-correlation Graph Collaborative Filtering (AFDGCF) framework, as illustrated in Figure 3, which addresses the issues of over-correlation and over-smoothing in GNN-based CF simultaneously while finding a trade-off between over-smoothing and representation smoothness through layer-wise adaptive feature correlation constraint.

4.1 Empirical Analysis on Over-correlation and Over-smoothing

Over-correlation issue refers to the increasing correlation between the feature dimensions of the GNN model's output representations as the number of GNN layers increases [6, 8, 36]. Over-smoothing issue refers to the growing similarity between the node representations of the GNN model's output as the number of GNN layers increases. The former will lead to feature redundancy among the learned representations, while the latter will lead to convergence of node representations, making them difficult to be distinguished. In recommender systems, both of them will constrain the quality of learned user and item representations, increase the risk of overfitting, and result in sub-optimal model performance. To investigate their impact on GNN-based CF models, we adopt *Corr* [17] and *SMV* [24] to assess them respectively.

Corr is utilized to measure the correlation between feature dimensions, which calculates the average Pearson correlation coefficient [12] between any two feature dimensions:

$$Corr(\mathbf{E}) = \frac{1}{d(d-1)} \sum_{i \neq j} |\rho(\mathbf{E}_{*i}, \mathbf{E}_{*j})|, \quad i, j \in \{1, 2, \ldots, d\}, \tag{6}$$

where d represents the dimension of the features, $\mathbf{E}_{*i}$ denotes the i-th feature of users and items, and $\rho(\mathbf{x}, \mathbf{y})$ represents the Pearson correlation coefficient between vectors $\mathbf{x}$ and $\mathbf{y}$:

$$\rho(\mathbf{x}, \mathbf{y}) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}, \tag{7}$$

which ranging from −1 to 1. The larger the absolute value, the stronger the correlation between the corresponding vectors.

SMV is utilized to gauge the similarity between user and item representations, which computes the average normalized Euclidean distance between any two nodes:

$$SMV(\mathbf{E}) = \frac{1}{m(m-1)} \sum_{i,j \in \mathcal{U} \cup \mathcal{I}, i \neq j} D\left(\mathbf{E}_{i*}, \mathbf{E}_{j*}\right), \tag{8}$$

where $D(\mathbf{x}, \mathbf{y})$ denotes the normalized euclidean distance between any two vectors $\mathbf{x}$ and $\mathbf{y}$:

$$D(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \left\| \frac{\mathbf{x}}{\|\mathbf{x}\|} - \frac{\mathbf{y}}{\|\mathbf{y}\|} \right\|. \tag{9}$$

Based on the two metrics above, we explore over-correlation and over-smoothing issues in GNN-based CF models. From Figure 2, we observe the following trends for these four models:

- *Corr* increases with layer count, indicating an increase in the correlation among the learned representations.
- *SMV* decreases with layer count, indicating an increase in the smoothness among the learned representations.
- The recommendation performance does not consistently improve with the increase in the number of layers; instead, performance degradation occurs after reaching 2 to 3 layers.

The above three points reveal the widespread existence of over-correlation and over-smoothing issues in GNN-based CF models, leading to the suboptimal performance of the models. Moreover, the severity of both issues increases with the number of layers, indicating a certain association between them. Next, we will analyze their association theoretically and propose solutions.

4.2 Theoretical Analysis on Over-correlation and Over-smoothing

In this subsection, we endeavor to analyze the relationship between over-correlation and over-smoothing theoretically. Firstly, we define the concepts of column correlation and row correlation of the representation matrix, which correspond to the feature correlation and smoothness, respectively. Then, through mathematical proof, we demonstrate a proportional relationship between the column correlation and row correlation of the matrix, thus inferring the connection between over-correlation and over-smoothing issues.

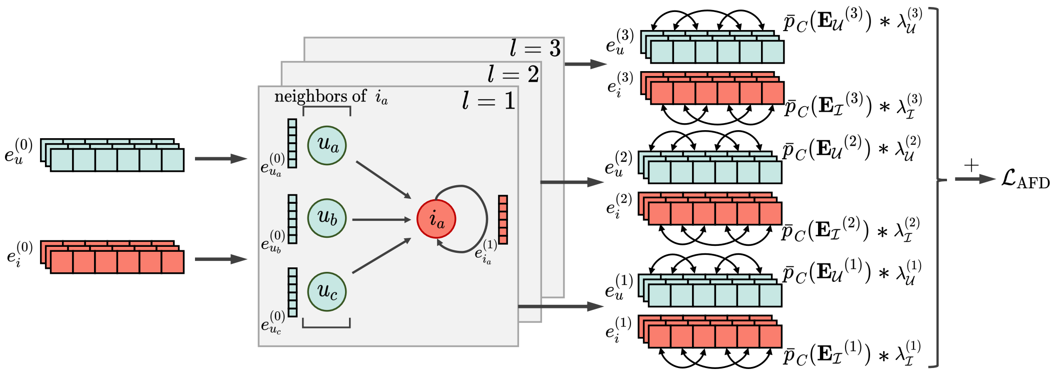
$e_u^{(0)}$
$e_i^{(0)}$
$l=3$
$l=2$
$l=1$
neighbors of i_a
$e_{u_a}^{(0)}$
u_a
$e_{u_b}^{(0)}$
u_b
$e_{u_c}^{(0)}$
u_c
i_a
$e_{i_a}^{(1)}$
$e_u^{(3)}$
$\bar{p}_C(\mathbf{E}_{\mathcal{U}}{}^{(3)}) * \lambda_{\mathcal{U}}^{(3)}$
$e_i^{(3)}$
$\bar{p}_C(\mathbf{E}_{\mathcal{I}}{}^{(3)}) * \lambda_{\mathcal{I}}^{(3)}$
$e_u^{(2)}$
$\bar{p}_C(\mathbf{E}_{\mathcal{U}}{}^{(2)}) * \lambda_{\mathcal{U}}^{(2)}$
$e_i^{(2)}$
$\bar{p}_C(\mathbf{E}_{\mathcal{I}}{}^{(2)}) * \lambda_{\mathcal{I}}^{(2)}$
$e_u^{(1)}$
$\bar{p}_C(\mathbf{E}_{\mathcal{U}}{}^{(1)}) * \lambda_{\mathcal{U}}^{(1)}$
$e_i^{(1)}$
$\bar{p}_C(\mathbf{E}_{\mathcal{I}}{}^{(1)}) * \lambda_{\mathcal{I}}^{(1)}$
$+$
$\mathcal{L}_{\text{AFD}}$

Figure 3: An overview of the AFDGCF framework, highlighting its model-agnostic nature and dynamic application of correlation penalties to user and item representations at each layer for enhanced performance.

In general, when calculating row or column correlation coefficients, it is common practice to first normalize the matrix to have a mean of 0 and a variance of 1 [12]. Here, to simplify the analysis process furthermore, we assume that the representation matrix $\mathbf{E}$ satisfies double standardization, *i.e.*, each row and column of $\mathbf{E}$ have a mean of 0 and a variance of 1. Under such assumptions, we can define the column correlation coefficient matrix of $\mathbf{E}$ as

$$\mathbf{P_C} = \frac{1}{m}\mathbf{E}^T\mathbf{E}\,, \quad \mathbf{P_C}_{ij} = \rho(\mathbf{E}_{*i}, \mathbf{E}_{*j}). \tag{10}$$

Similarly, the row correlation coefficient matrix of $\mathbf{E}$ is

$$\mathbf{P_R} = \frac{1}{n}\mathbf{E}\mathbf{E}^T, \quad \mathbf{P_R}_{ij} = \rho(\mathbf{E}_{i*}, \mathbf{E}_{j*}). \tag{11}$$

Based on this, we can use $\|\mathbf{P_C}\|_F$ and $\|\mathbf{P_R}\|_F$ to describe the column correlation and row correlation of matrix $\mathbf{E}$, respectively. In fact, the relationship between row correlation and column correlation of a matrix has been studied in previous research [13]. Next, we will demonstrate the association between them based on this.

Theorem 1. *If matrix $\mathbf{E}$ satisfies double standardization, then its corresponding row correlation matrix $\mathbf{P_R}$ and column correlation matrix $\mathbf{P_C}$ satisfy $\|\mathbf{P_R}\|_F \propto \|\mathbf{P_C}\|_F$.*

Proof. The singular value decomposition of matrix $\mathbf{E}$ is:

$$\underset{m\times n}{\mathbf{E}} = \underset{m\times d}{\mathbf{U}} \ \underset{d\times d}{\Sigma} \ \underset{d\times n}{\mathbf{V}^T}, \tag{12}$$

where Σ is the diagonal matrix of ordered singular values, $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices satisfying

$$\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}_d, \tag{13}$$

where $\mathbf{I}_d$ is the $d \times d$ identity. The squares of the diagonal elements:

$$e_1 \geq e_2 \geq \cdots \geq e_d > 0 \quad (e_i = \Sigma_i^2), \tag{14}$$

are the eigenvalues of

$$\mathbf{E}^T\mathbf{E} = \mathbf{V}^T\Sigma^2\mathbf{V}. \tag{15}$$

Then we have:

$$\begin{aligned} \frac{\sum_{i=1}^n \sum_{j=1}^n \mathbf{P_C}_{ij}^2}{n^2} &= \frac{\operatorname{tr}((\mathbf{E}^T\mathbf{E})^2)}{m^2n^2} = \frac{\operatorname{tr}(\mathbf{V}^T\Sigma^4\mathbf{V})}{m^2n^2} = c, \\ \frac{\sum_{i=1}^n \sum_{j=1}^n \mathbf{P_R}_{ij}^2}{m^2} &= \frac{\operatorname{tr}((\mathbf{E}\mathbf{E}^T)^2)}{m^2n^2} = \frac{\operatorname{tr}(\mathbf{U}\Sigma^4\mathbf{U}^T)}{m^2n^2} = c, \end{aligned} \tag{16}$$

where $tr(\cdot)$ represents the trace of a matrix and c can be computed to be a constant

$$c = \sum_{i=1}^{d} \frac{e_i^2}{(mn)^2}. \tag{17}$$

That is to say:

$$\frac{1}{m}\|\mathbf{P_R}\|_F = \frac{1}{n}\|\mathbf{P_C}\|_F. \tag{18}$$

□

The above theory reveals a proportional relationship between column correlation and row correlation of a matrix. In the GNN-based CF model, for the representation matrix $\mathbf{E}$, the column correlation represents the correlation between features, and the row correlation describes the similarity between user and item representations, which can be considered as a proxy for measuring smoothness. The larger the row correlation, the smaller the average difference between representations, indicating a higher level of smoothing. Therefore, we can conclude that over-correlation and over-smoothing positively correlate in degree. This is consistent with the observations from previous experiments. According to this conclusion, we can utilize the following proposed method to address the issue of over-correlation while mitigating the impact of over-smoothing.

4.3 Feature De-correlation

To alleviate the excessively high correlation in feature dimensions of user and item representations learned by GNN-based CF models, a straightforward approach is to design a loss that constrains the correlation between features. In terms of the types of loss, we have a variety of choices. If we consider each column of the representation matrix as a high-dimensional vector, we can impose constraints between any two columns using different distance metrics (e.g., Euclidean distance or cosine distance); if we view each dimension of the feature as a random variable, we can impose penalties based on their KL-divergence or correlation coefficient. Considering the substantial computational effort required to calculate the sum of KL-divergence between any two columns of a matrix, and the fact that Euclidean distance, cosine distance, and Pearson correlation coefficient become equivalent after data standardization, we naturally opt to design our loss function based on the correlation coefficient.

Our objective is to reduce the average correlation between columns of the representation matrix $\mathbf{E}$. Formally,

$$\bar{p}_C(\mathbf{E}) = \frac{1}{d(d-1)} \sum_{i \neq j} \rho(\mathbf{E}_{*i}, \mathbf{E}_{*j})^2, \quad i, j \in \{1, 2, \ldots, d\}. \tag{19}$$

Since the correlation coefficient of a vector with itself is always 1, we can also represent it using the previously defined column correlation coefficient matrix $\mathbf{P_R}$:

$$\bar{p}_C(\mathbf{E}) = \sqrt{\frac{1}{2}\|\mathbf{P}_\mathrm{C} - \mathbf{I}_d\|_F^2} = \frac{1}{\sqrt{2}}\|\mathbf{P}_\mathrm{C} - \mathbf{I}_d\|, \tag{20}$$

where $\mathbf{I}_d$ is the $d \times d$ identity and $\mathbf{P_R}$ can be calculated using the following formula:

$$\mathbf{P}_{\mathrm{C}ij} = \frac{\mathbf{Cov}_{\mathrm{C}ij}}{\sqrt{\mathbf{Cov}_{\mathrm{C}ii}\,\mathbf{Cov}_{\mathrm{C}jj}}}, \tag{21}$$

where $\mathbf{Cov_C}$ denotes the covariance matrix among the column vectors of the representation matrix $\mathbf{E}$:

$$\mathbf{Cov_C} = (\mathbf{E} - \bar{\mathbf{E}})^T(\mathbf{E} - \bar{\mathbf{E}}), \tag{22}$$

where $\bar{\mathbf{E}}$ represents a matrix composed of the mean values of each column of $\bar{\mathbf{E}}$. It is worth noting that here we utilize the entire representation matrix to compute the correlations between feature dimensions. In fact, to reduce the computational complexity, when dealing with large-scale user and item sets, it is possible to randomly sample a batch of users and items for calculating the correlations between feature dimensions.

4.4 Adaptive Feature De-correlation

Directly applying the function $\bar{p}_C(\cdot)$ to the learned representation matrix $\mathbf{E}$ of the GNN-based CF model is not the optimal choice. One critical consideration is that various GNN-based CF models employ distinct pooling strategies. For example, GCCF uses concatenation, while LightGCN uses mean pooling. This results in significant differences in the correlation of the representation matrix $\mathbf{E}$ for different models. To address this and ensure our approach remains model-agnostic, we propose penalizing the representations obtained after each message-passing operation as an alternative approach. Furthermore, since user and item feature distributions often exhibit substantial differences, applying penalties separately to the user and item representation matrix is necessary. This approach also helps in preserving the relationships between users and items learned by the model, thus avoiding potential disruptions. In this case, our feature de-correlation loss can be formulated in the following manner:

$$\mathcal{L}_{\mathrm{AFD}} = \sum_{l=1}^{L} \lambda_{\mathcal{U}}^{(l)} \bar{p}_C(\mathbf{E}_{\mathcal{U}}{}^{(l)}) + \lambda_{\mathcal{I}}^{(l)} \bar{p}_C(\mathbf{E}_{\mathcal{I}}{}^{(l)}). \tag{23}$$

It is worth noting that we do not impose a penalty on $\mathbf{E}_{\mathcal{U}}{}^{(0)}, \mathbf{E}_{\mathcal{I}}{}^{(0)}$ because the 0-th layer represents the initial user and item embeddings without undergoing graph convolution, usually subject to L_2 regularization constraints. $\lambda_{\mathcal{U}}{}^{(l)}, \lambda_{\mathcal{I}}{}^{(l)}$ represents the penalty coefficient for the l-th layer corresponding to users and items. If fixed at $1/L$, it indicates a fixed penalty of the same magnitude for each layer. However, this is not the most reasonable approach. The strength of penalty accepted between different layers should be dynamically adjusted during training based on the relative correlation magnitudes of each layer.

Table 1: Statistics of the utilized datasets.

Datasets	# User	# Item	# Interaction	Density
Movielens	6,040	3,629	836,478	$3.8e^{-2}$
Yelp	26,752	19,857	1,001,901	$1.8e^{-3}$
Gowalla	29,859	40,989	1,027,464	$8.4e^{-4}$
Amazon-book	58,145	58,052	2,517,437	$7.5e^{-4}$

For a GNN-based CF model with a specific number of layers, the representations of deeper layers often have greater embedding smoothness while having higher feature correlations. Considering the contribution of feature smoothness to the effectiveness of GNN-based CF models [14, 16, 44], it is crucial to maintain the smoothness of the representations of deeper layers while restricting the feature correlations of the model's final representations. Therefore, we propose the strategies which allocates lower penalty coefficients to the representations of deeper layers and maintains the total penalty amount, as follows:

$$\lambda_{\mathcal{U}}{}^{(l)} = \frac{1/\bar{p}_C(\mathbf{E}_{\mathcal{U}}{}^{(l)})}{\sum_{i=1}^{L} 1/\bar{p}_C(\mathbf{E}_{\mathcal{U}}{}^{(i)})}, \ \lambda_{\mathcal{I}}{}^{(l)} = \frac{1/\bar{p}_C(\mathbf{E}_{\mathcal{I}}{}^{(l)})}{\sum_{i=1}^{L} 1/\bar{p}_C(\mathbf{E}_{\mathcal{I}}{}^{(i)})}. \tag{24}$$

If the adaptive strategy is not employed, throughout the entire training process, the penalty coefficients for user and item correspondence at each layer remain fixed and unchanging. However, with the adoption of the adaptive strategy, the penalty coefficients for user and item correspondence at each layer will dynamically change in each training step and tend to stabilize as the model approaches convergence. The final loss function is shown below:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CF}} + \alpha \mathcal{L}_{\mathrm{AFD}}, \tag{25}$$

where $\mathcal{L}_{\mathrm{CF}}$ represents the original loss functions of the CF model like Bayesian Personalized Ranking (BPR) loss [31], Cross-Entropy (CE) loss, etc. α is a hyper-parameter used to adjust the relative magnitude of the adaptive feature de-correlation loss.

Overall, the proposed AFDGCF framework is a general solution specifically targeting the issue of feature over-correlation in GNN-based CF models. This framework is model-agnostic and can be effortlessly combined with a wide variety of GNN-based CF models, such as NGCF [41], GCCF [9], LightGCN [16], HMLET [19], etc. This integration is achieved by incorporating the loss $\mathcal{L}_{\mathrm{AFD}}$ into the existing CF loss function, as detailed in Equation 25. Furthermore, the AFDGCF framework is also helpful in addressing the over-smoothing issue. This dual capability is underpinned by our theoretical analysis, which examines the intricate dynamics between these two prevalent problems in GNN-based CF models.

5 EXPERIMENTS

To better understand the capabilities and effectiveness of our proposed AFDGCF framework, we instantiated it on four representative GNN-based CF models and evaluated its performance on four publicly available datasets. Specifically, we will answer the following research questions to unfold the experiments: **RQ1**: What degree of performance improvement can our proposed AFDGCF framework bring to the state-of-the-art methods in GNN-based CF? **RQ2**: Does adaptive de-correlation have a performance advantage compared to fixed one? **RQ3**: How do the hyper-parameters influence the effectiveness of the proposed AFDGCF framework? **RQ4**:

Table 2: Performance comparisons on four datasets. We have conducted the paired t-test to verify that the difference between each base model and our proposed method is statistically significant for $p < 0.05$. (Underlines represent the optimal performance)

Dataset	MovieLens			Yelp			Gowalla			Amazon-book		
Method	**Recall**	**NDCG**	**MAP**	**Recall**	**NDCG**	**MAP**	**Recall**	**NDCG**	**MAP**	**Recall**	**NDCG**	**MAP**
BPRMF	0.1697	0.2353	0.1347	0.0643	0.0437	0.0240	0.1110	0.0790	0.0509	0.0678	0.0470	0.0273
DMF	0.1739	0.2338	0.1320	0.0560	0.0375	0.0207	0.0828	0.0589	0.0371	0.0643	0.0447	0.0260
ENMF	0.1941	0.2613	0.1524	0.0742	0.0520	0.0298	0.1271	0.0914	0.0597	0.0839	0.0617	0.0374
MultiVAE	0.1889	0.2370	0.1318	0.0731	0.0494	0.0275	0.1188	0.0838	0.0550	0.0816	0.0574	0.0352
RecVAE	0.1852	0.2524	0.1475	0.0777	0.0536	0.0301	0.1466	0.1046	0.0695	0.1044	0.0772	0.0484
DGCF	0.1819	0.2477	0.1429	0.0712	0.0487	0.0272	0.1372	0.0974	0.0635	0.0951	0.0689	0.0424
NGCF	0.1743	0.2404	0.1376	0.0613	0.0419	0.0229	0.1129	0.0800	0.0519	0.0695	0.0484	0.0287
AFD-NGCF	**0.1764**	**0.2449**	**0.1416**	**0.0636**	**0.0432**	**0.0237**	**0.1233**	**0.0869**	**0.0566**	**0.0735**	**0.0511**	**0.0304**
Improv.	**1.20%**	**1.87%**	**2.91%**	**3.75%**	**3.10%**	**3.49%**	**9.21%**	**8.63%**	**9.05%**	**5.76%**	**5.58%**	**5.92%**
GCCF	0.1761	0.2460	0.1431	0.0658	0.0451	0.0247	0.1268	0.0907	0.0589	0.0898	0.0639	0.0387
AFD-GCCF	**0.1882**	**0.2566**	**0.1505**	**0.0739**	**0.0506**	**0.0284**	**0.1416**	**0.1015**	**0.0667**	**0.0977**	**0.0698**	**0.0427**
Improv.	**6.87%**	**4.31%**	**5.17%**	**12.31%**	**12.20%**	**14.98%**	**11.67%**	**11.91%**	**13.24%**	**8.80%**	**9.23%**	**10.34%**
HMLET	0.1799	0.2479	0.1433	0.0723	0.0503	0.0285	0.1427	0.1028	0.0679	0.0967	0.0699	0.0430
AFD-HMLET	**0.1922**	**0.2606**	**0.1523**	**0.0810**	**0.0562**	**0.0319**	**0.1541**	**0.1101**	**0.0727**	**0.1039**	**0.0760**	**0.0471**
Improv.	**6.84%**	**5.12%**	**6.28%**	**12.03%**	**11.73%**	**11.93%**	**7.99%**	**7.10%**	**7.07%**	**7.45%**	**8.73%**	**9.53%**
LightGCN	0.1886	0.2540	0.1470	0.0756	0.0525	0.0295	0.1433	0.1019	0.0664	0.1015	0.0744	0.0465
AFD-LightGCN	**0.1985**	**0.2689**	**0.1594**	**0.0831**	**0.0575**	**0.0327**	**0.1564**	**0.1117**	**0.0736**	**0.1078**	**0.0781**	**0.0486**
Improv.	**5.25%**	**5.87%**	**8.44%**	**9.92%**	**9.52%**	**10.85%**	**9.14%**	**9.62%**	**10.84%**	**6.21%**	**4.97%**	**4.52%**

Table 3: Comparison of training efficiency on four datasets.

Dataset	HMLET	AFD-H	LightGCN	AFD-L
MovieLens	167 eps 5.95s/ep	112 eps 6.87s/ep	309 eps 4.51s/ep	114 eps 5.12s/ep
Yelp	114 eps 9.54s/ep	109 eps 10.58s/ep	364 eps 6.76s/ep	151 eps 7.31s/ep
Gowalla	347 eps 10.25s/ep	205 eps 11.27s/ep	396 eps 6.98s/ep	302 eps 7.82s/ep
Amazon-book	458 eps 58.12s/ep	198 eps 61.23s/ep	742 eps 32.06s/ep	360 eps 35.73s/ep

Table 4: Ablation study of AFDGCF.

Dataset	Metric	AFD-N	AFD-f-N	AFD-G	AFD-f-G
Yelp	Recall	**0.0636**	0.0623	**0.0739**	0.0719
	NDCG	**0.0432**	0.0424	**0.0506**	0.0487
Gowalla	Recall	**0.1233**	0.1198	**0.1315**	0.1298
	NDCG	**0.0869**	0.0850	**0.0949**	0.0937

Dataset	Metric	AFD-H	AFD-f-H	AFD-L	AFD-f-L
Yelp	Recall	**0.0810**	0.0796	**0.0831**	0.0808
	NDCG	**0.0562**	0.0552	**0.0575**	0.0558
Gowalla	Recall	**0.1541**	0.1465	**0.1564**	0.1505
	NDCG	**0.1101**	0.1069	**0.1117**	0.1081

What impact does the AFDGCF framework have on the correlation of the learned representations?

5.1 Experimental Settings

Datasets and metrics. We selected four publicly available datasets to evaluate the performance of the proposed AFDGCF, including MovieLens [15], Yelp [1], Gowalla [11], and Amazon-book [26], which vary in domain, scale, and density. The statistical information for these four datasets is summarized in Table 1. For the Yelp and Amazon-book datasets, we excluded users and items with fewer than 15 interactions to ensure data quality. Similarly, for the Gowalla dataset, we removed those with fewer than 10 interactions. Moreover, we partitioned each dataset into training, validation, and testing sets using an 8:1:1 ratio. We chose Recall@K, Normalized Discounted Cumulative Gain (NDCG)@K, and Mean Average Precision (MAP)@K as the evaluation metrics (K=10). We also employed the all-ranking protocol [16] to avoid bias arising from sampling.

Methods for Comparison. To verify the effectiveness of our proposed method, we compared it with several state-of-the-art models: BPRMF [31], DMF [49], ENMF [3], MultiVAE [22], RecVAE [33], DGCF [43], NGCF [41], GCCF [9], LightGCN [16], HMLET [19]. AFD-NGCF/GCCF/LightGCN/HMLET are the instantiation of the AFDGCF framework we propose on those above four GNN-based CF models. Abbreviated as AFD-N/G/L/H.

Implementation Details. For a fair comparison, we employed the Adam optimizer [18] for all models with a learning rate of $1e^{-3}$ and set the training batch size to 4096. The embedding size or hidden dimension in all models were set to 128, and model parameters were initialized using the Xavier distribution. Furthermore, we used early stopping for the indicator NDCG@10, halting training when there was no further increase in NDCG@10 on valid set to prevent the model from over-fitting. For the four GNN-based CF models, we performed 3-layer propagation. As for the two VAE-based models, we set their encoder and decoder as a 1-hidden-layer MLP with $[n, 600, 128, 600, n]$ dimensions. The weight of L2 regularization is selected within the range $\{1e^{-2}, 1e^{-3}, 1e^{-4}, 1e^{-5}, 1e^{-6}\}$, and dropout ratios were selected from $\{0.0, 0.1, \cdots, 0.8, 0.9\}$. Other model-specific hyper-parameters were configured according to the settings prescribed in the original papers. For the proposed AFDGCF framework, we tuned the hyper-parameter α within the range $\{1e^{-5}, 5e^{-5}, \cdots, 5e^{-2}, 1e^{-1}\}$. The implementation of all models above and their evaluation was conducted using the open-source framework RecBole [53].

[1] https://www.yelp.com/dataset

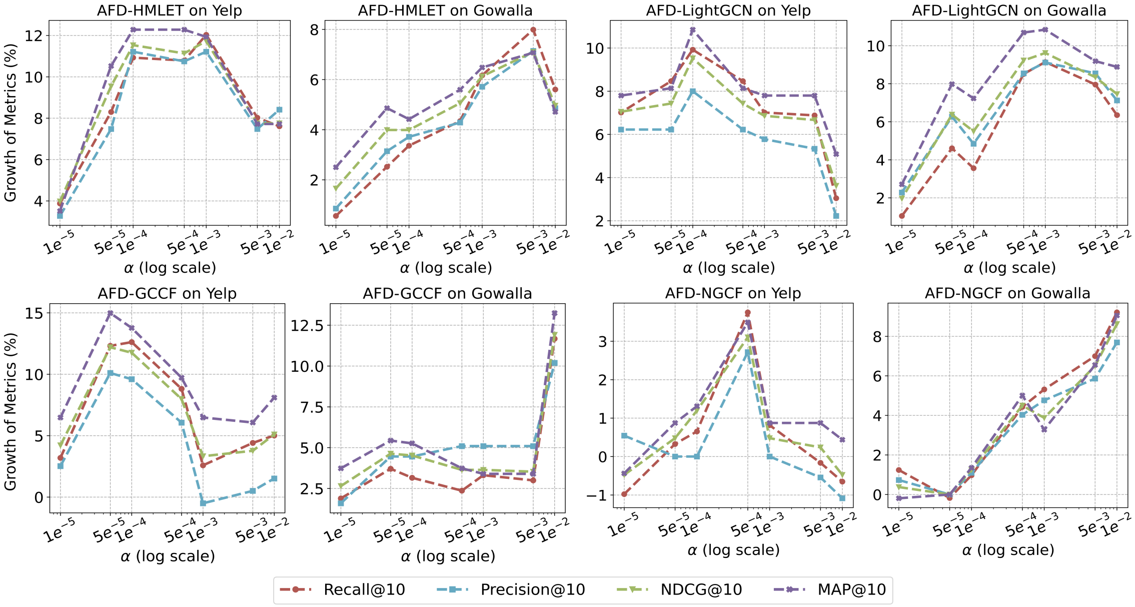

Figure 4: Performance comparison *w.r.t.* different α. Using the results of the original model (*i.e.*, $\alpha = 0$) as the reference.

5.2 Performance Comparisons (RQ1)

Table 2 demonstrates the effectiveness of our proposed AFDGCF framework as instantiated on four GNN-based CF models and provides a comparative analysis against various baseline methods across four distinct datasets. The table reveals that implementing AFDGCF in all the aforementioned GNN-based CF models is effective. This stems from notable performance enhancements achieved by mitigating the over-correlation issue, while successfully balancing mitigating over-smoothing and maintaining embedding smoothness. Furthermore, the AFDGCF framework demonstrates even more pronounced performance enhancements on large-scale datasets. On Yelp, for Recall@10, ADFGCF achieves performance improvements of 3.75%, 12.31%, 12.03%, and 9.92% over NGCF, GCCF, HMLET, and LightGCN, respectively. It is worth noting that in our experimental setup, we standardized the embedding dimension for all GNN-based CF methods to 128 and conducted three rounds of feature propagation. As a result, HMLET, contrary to its original design, does not outperform LightGCN on most datasets. Moreover, we can observe that most GNN-based CF models trail slightly behind VAE-based models in performance. This indicates that GNN-based CF models might be prone to over-fitting when dealing with extensive sparse interactions, a phenomenon linked to the two issues highlighted in this paper. Besides, although DGCF employs a formally similar correlation constraint, there are significant differences between DGCF and our proposed AFDGCF, both in terms of methods and results. Methodologically, DGCF combines de-correlation with intent-aware routing to learn disentangled embeddings, aiming to ensure relative independence among learned intents. However, within each intent, certain columns remain highly correlated. From an effectiveness perspective, the DGCF method suffers from over-fitting issues, leading to noticeable performance degradation on larger datasets. Meanwhile, as mentioned in their paper, finer-grained disentangling would lead to a performance decline in DGCF models. In these respects, our AFDGCF demonstrates a starkly different performance.

Recent studies have indicated that GNN-based CF models, like LightGCN, are much more difficult to train than traditional MF models [28, 37]. These models often require a larger number of epochs to reach the optimal state. We hypothesize that issues of over-correlation and over-smoothing diminish the distinction between positive and negative samples, thereby impacting learning efficiency. Fortunately, we found our AFDGCF framework can alleviate this phenomenon. As evidenced in Table 3 (where "ep" means "epoch"), within the AFDGCF framework, both LightGCN and HMLET attain optimal performance with fewer epochs on various datasets. In some cases, the required epochs are reduced by half.

5.3 Ablation study of AFDGCF (RQ2)

A key component of our proposed AFDGCF is the adaptive strategy. To validate this strategy, we conducted an ablation study to analyze its influence on performance. The experimental results are presented in Table 4, where AFD-f denotes applying a fixed penalty coefficient of $1/l$ to each layer within the AFDGCF framework. Clearly, we can observe that our adaptive approach consistently outperforms the fixed penalty coefficient across all GNN-based models, demonstrating the effectiveness of applying varying de-correlation penalties to distinct layers. Besides, we have conducted the paired t-test to verify that the difference between each AFD and AFD-f model is statistically significant for $p < 0.05$.

5.4 Hyper-parameter Analysis (RQ3)

In our AFDGCF framework, hyper-parameter α balances the CF task loss with the proposed adaptive feature de-correlation loss, acting as a regulator for the intensity of de-correlation. Figure 4

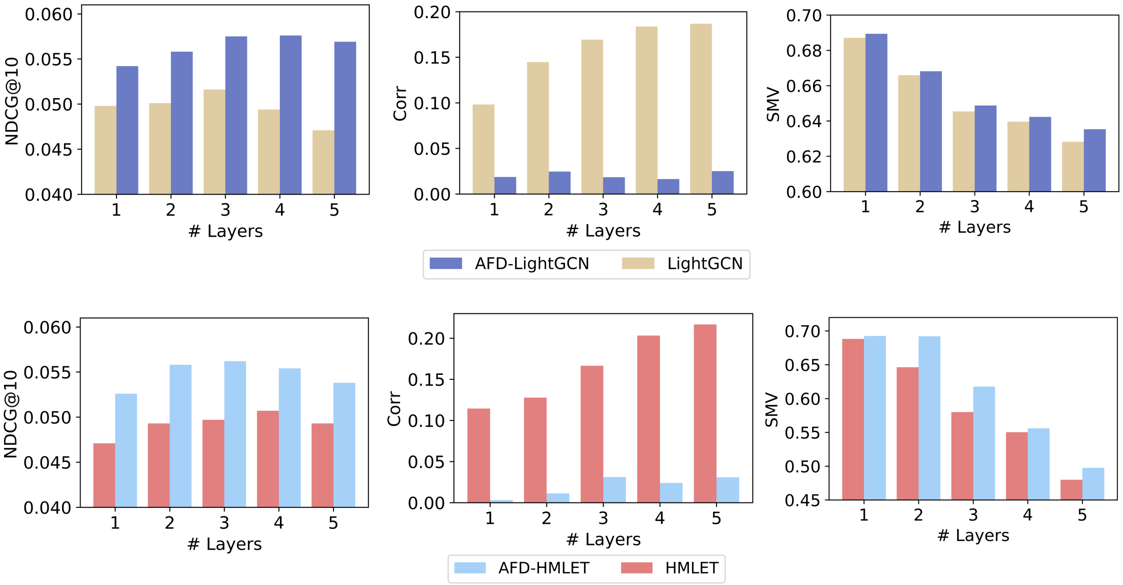

Figure 5: Comparison of GNN-based CF models with different layers before and after applying the AFDGCF framework in terms of NDCG@10, *Corr* and *SMV* on Yelp Dataset.

reports the performance of AFDGCF as α varies across the range $\{1e^{-5}, 5e^{-5}, \cdots, 5e^{-3}, 1e^{-2}\}$. We can observe that setting an appropriate α can markedly enhance AFDGCF's performance. Notably, the ideal α varies between datasets, but remains fairly consistent for different models within the same dataset. This is attributed to our design of applying de-correlation penalties to each layer rather than the final representation. Furthermore, for the two datasets mentioned above, we have not shown results for larger values of α (*e.g.*, $5e^{-2}$ or $1e^{-1}$), although some model-dataset combinations may trend toward better performance at higher α values. However, in practice, when we apply larger values of α, the training process of the model collapses in most cases, yielding unusable results.

5.5 Case Study (RQ4)

We investigate the effects of our proposed AFDGCF framework using LightGCN and HMLET as examples in the following two aspects: i) the de-correlation effect on GNN-based CF models with different layers, and ii) the resulting performance enhancements.

In Figure 5, we have reported a comparison of the performance and the feature correlations of LightGCN and HMLET (with layers ranging from 1 to 5) before and after applying the AFD penalty. Observations can be made as follows:

- For any number of layers, the AFDGCF framework improves performance, and larger layers correspond to higher performance gains. These results indicate that with the increase in the number of message-passing iterations, the performance degradation of GNN-based CF models due to over-correlation and over-smoothing issues becomes more severe. However, the proposed AFDGCF framework can effectively mitigate this problem.
- The *Corr* metric of the representations learned by AFD-LightGCN and AFD-HMLET from layers 1 to 5 is much lower than that of LightGCN and HMLET, indicating that the AFDGCF framework effectively achieves its goal of controlling the correlation between feature dimensions in the representation matrix.
- The *SMV* metric of the representations learned by AFD-LightGCN and AFD-HMLET from layers 1 to 5 is higher than that of LightGCN and HMLET. These results indicate that our proposed method can alleviate the over-correlation issue by constraining the column-wise correlation of the representation matrix, while simultaneously reducing the row-wise correlation of the representation matrix, thereby mitigating the over-smoothing problem.

6 CONCLUSION

This research undertook an in-depth exploration of the largely neglected issue of over-correlation in GNN-based CF models, substantiating its widespread presence and associated degradation in model performance. Our investigations established a direct and positive correlation between over-correlation and over-smoothing, highlighting their combined detrimental impact on the effectiveness of GNN-based CF models. Specifically, we introduced the AFDGCF framework, a novel approach designed to mitigate the influence of over-correlation by strategically managing feature correlations. This was achieved through a layer-wise adaptive strategy, ensuring an optimal balance between minimizing over-smoothing and maintaining essential representation smoothness, a critical aspect for the success of GNN-based CF models. Empirical validations underscored the efficacy of the AFDGCF framework, demonstrating notable performance enhancements across different layers of various GNN-based CF models.

ACKNOWLEDGMENTS

This work was partially supported by National Natural Science Foundation of China (Grant No.92370204), Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2023A03J0008), Education Bureau of Guangzhou Municipality, Guangdong Science and Technology Department and Project funded by China Postdoctoral Science Foundation (Grant No.2023M730785).

REFERENCES

[1] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[2] 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.* https://openreview.net/forum?id=FKXVK9dyMM

[3] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient neural matrix factorization without sampling for recommendation. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–28.

[4] 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 *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 3438–3445.

[5] Liyi Chen, Zhi Li, Weidong He, Gong Cheng, Tong Xu, Nicholas Jing Yuan, and Enhong Chen. 2023. Entity Summarization via Exploiting Description Complementarity and Salience. *IEEE transactions on neural networks and learning systems* 34, 11 (2023), 8297–8309.

[6] Liyi Chen, Zhi Li, Yijun Wang, Tong Xu, Zhefeng Wang, and Enhong Chen. 2020. MMEA: entity alignment for multi-modal knowledge graph. In *Knowledge Science, Engineering and Management: 13th International Conference, KSEM 2020, Hangzhou, China, August 28–30, 2020, Proceedings, Part I 13*. Springer, 134–147.

[7] Liyi Chen, Zhi Li, Tong Xu, Han Wu, Zhefeng Wang, Nicholas Jing Yuan, and Enhong Chen. 2022. Multi-modal siamese network for entity alignment. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining.* 118–126.

[8] Liyi Chen, Chuan Qin, Ying Sun, Xin Song, Tong Xu, Hengshu Zhu, and Hui Xiong. 2024. Collaboration-Aware Hybrid Learning for Knowledge Development Prediction. In *Proceedings of the ACM Web Conference 2024*.

[9] 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.

[10] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. 2020. Simple and Deep Graph Convolutional Networks. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 1725–1735. https://proceedings.mlr.press/v119/chen20v.html

[11] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining.* 1082–1090.

[12] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. *Noise reduction in speech processing* (2009), 1–4.

[13] Bradley Efron. 2008. *Row and column correlations:(are a set of microarrays independent of each other?).* Division of Biostatistics, Stanford University.

[14] Wenqi Fan, Xiaorui Liu, Wei Jin, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2022. Graph trend filtering networks for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 112–121.

[15] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[16] 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.

[17] Wei Jin, Xiaorui Liu, Yao Ma, Charu Aggarwal, and Jiliang Tang. 2022. Feature Overcorrelation in Deep Graph Neural Networks: A New Perspective. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 709–719. https://doi.org/10.1145/3534678.3539445

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *The Third International Conference on Learning Representations.*

[19] Taeyong Kong, Taeri Kim, Jinsung Jeon, Jeongwhan Choi, Yeon-Chang Lee, Noseong Park, and Sang-Wook Kim. 2022. Linear, or non-linear, that is the question!. In *Proceedings of the fifteenth ACM international conference on web search and data mining.* 517–525.

[20] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[21] Xin Li and Hsinchun Chen. 2013. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems* 54, 2 (2013), 880–890.

[22] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference.* 689–698.

[23] 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.

[24] Meng Liu, Hongyang Gao, and Shuiwang Ji. 2020. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining.* 338–348.

[25] 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.

[26] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval.* 43–52.

[27] Andriy Mnih and Russ R Salakhutdinov. 2007. Probabilistic matrix factorization. *Advances in neural information processing systems* 20 (2007).

[28] Shaowen Peng, Kazunari Sugiyama, and Tsunenori Mine. 2022. SVD-GCN: A simplified graph convolution paradigm for recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management.* 1625–1634.

[29] Chuan Qin, Le Zhang, Rui Zha, Dazhong Shen, Qi Zhang, Ying Sun, Chen Zhu, Hengshu Zhu, and Hui Xiong. 2023. A comprehensive survey of artificial intelligence techniques for talent analytics. *arXiv preprint arXiv:2307.03195* (2023).

[30] Chuan Qin, Hengshu Zhu, Dazhong Shen, Ying Sun, Kaichun Yao, Peng Wang, and Hui Xiong. 2023. Automatic skill-oriented question generation and recommendation for intelligent job interviews. *ACM Transactions on Information Systems* 42, 1 (2023), 1–32.

[31] 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.

[32] T Konstantin Rusch, Michael M Bronstein, and Siddhartha Mishra. 2023. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993* (2023).

[33] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. 2020. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In *Proceedings of the 13th international conference on web search and data mining.* 528–536.

[34] Zhihao Shi, Xize Liang, and Jie Wang. 2023. LMC: Fast Training of GNNs via Subgraph Sampling with Provable Convergence. In *The Eleventh International Conference on Learning Representations.* https://openreview.net/forum?id=5VBBA91N6n

[35] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009).

[36] Qian Sun, Le Zhang, Huan Yu, Weijia Zhang, Yu Mei, and Hui Xiong. 2023. Hierarchical reinforcement learning for dynamic autonomous vehicle navigation at intelligent intersections. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 4852–4861.

[37] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards representation alignment and uniformity in collaborative filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 1816–1825.

[38] Chao Wang, Hengshu Zhu, Qiming Hao, Keli Xiao, and Hui Xiong. 2021. Variable interval time sequence modeling for career trajectory prediction: Deep collaborative perspective. In *Proceedings of the Web Conference 2021.* 612–623.

[39] Chao Wang, Hengshu Zhu, Peng Wang, Chen Zhu, Xi Zhang, Enhong Chen, and Hui Xiong. 2021. Personalized and explainable employee training course recommendations: A bayesian variational approach. *ACM Transactions on Information Systems (TOIS)* 40, 4 (2021), 1–32.

[40] Chao Wang, Hengshu Zhu, Chen Zhu, Chuan Qin, Enhong Chen, and Hui Xiong. 2023. SetRank: A Setwise Bayesian Approach for Collaborative Ranking in Recommender System. *ACM Transactions on Information Systems* 42, 2 (2023), 1–32.

[41] 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.

[42] 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* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 165–174. https://doi.org/10.1145/3331184.3331267

[43] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval.* 1001–1010.

[44] Yu Wang, Yuying Zhao, Yi Zhang, and Tyler Derr. 2023. Collaboration-Aware Graph Convolutional Network for Recommender Systems. In *Proceedings of the ACM Web Conference 2023.* 91–101.

[45] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2022. A survey on accuracy-oriented neural recommendation: From collaborative filtering to

information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 5 (2022), 4425–4445.

[46] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *Comput. Surveys* 55, 5 (2022), 1–37.

[47] Lianghao Xia, Chao Huang, Jiao Shi, and Yong Xu. 2023. Graph-less collaborative filtering. In *Proceedings of the ACM Web Conference 2023*. 17–27.

[48] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*. 70–79.

[49] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep matrix factorization models for recommender systems.. In *IJCAI*, Vol. 17. Melbourne, Australia, 3203–3209.

[50] 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 *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.

[51] Shengzhe Zhang, Liyi Chen, Chao Wang, Shuangli Li, and Hui Xiong. 2024. Temporal Graph Contrastive Learning for Sequential Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[52] Lingxiao Zhao and Leman Akoglu. 2020. PairNorm: Tackling Oversmoothing in GNNs. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rkecl1rtwB

[53] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *CIKM*. ACM, 4653–4664.

[54] Lei Zheng, Chun-Ta Lu, Fei Jiang, Jiawei Zhang, and Philip S Yu. 2018. Spectral collaborative filtering. In *Proceedings of the 12th ACM conference on recommender systems*. 311–319.