



C-GDN: core features activated graph dual-attention network for personalized recommendation

Xiongtao Zhang¹ · Mingxin Gan¹

Received: 19 July 2023 / Revised: 30 August 2023 / Accepted: 30 August 2023 /

Published online: 25 September 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

As a popular graph learning technique, graph neural networks (GNN) show great advantages in the field of personalized recommendation. Existing GNN-based recommendation methods organized user-item interactions (e.g., click, purchase, review, etc.) as the bipartite graph and captured the higher-order collaborative signal with the aid of the GNN to achieve personalized recommendation. However, there exists two limitations in existing studies. First, core features activating user-item interactions were not be identified, which causes that user-item interactions fail to be accurately exploited at the feature level. Second, existing GNNs ignored the mutual association among neighbors in information propagation, which results in structural signal in the bipartite graph not being sufficiently captured. Towards this end, we developed the core features activated graph dual-attention network, namely C-GDN, for personalized recommendation. Specifically, C-GDN firstly identifies core user and item features activating user-item interactions and employs these core features to initialize the bipartite graph, which effectively optimizes the utilizing of user-item interactions at the feature level. Furthermore, C-GDN designs a novel graph dual-attention network to conduct information propagation, which captures more sufficient structural signal in the bipartite graph by considering information from neighbors as well as their mutual association. Extensive experiments on three benchmark datasets shows that C-GDN outperforms state-of-the-art baselines.

Keywords Core features · Dual-attention mechanism · Information propagation · Graph neural network · Personalized recommendation

1 Introduction

In the age of information explosion, various technologies have been developed to assist online users in information filtering and decision-making. As one of the most widely applied decision

✉ Mingxin Gan
ganmx@ustb.edu.cn

Xiongtao Zhang
B20190412@xs.ustb.edu.cn

¹ Department of Management Science and Engineering, School of Economics and Management, University of Science and Technology Beijing, Haidian, Beijing 100083, China

support systems, recommender systems conduct in-depth mining of interactive records to infer user preferences and recommend items that users interested in He et al. (2022). With the advantages of personalization, recommender systems have been applied to many online services such as e-commerce, social media, location services and so on.

In order to effectively achieve recommender systems, a large number of recommendation methods (e.g., collaborative filtering, content-based methods, hybrid methods, etc.) continually emerge (Wu et al., 2022a). Recently, with the ability of non-linear modeling of deep learning (Gan & Ma, 2022), deep recommendation methods gain wide attention. Deep recommendation methods adopt neural networks to model either representations of users and items or the matching between users and items (Wu et al., 2022b). As typical deep recommendation methods, graph neural networks (GNN)-based recommendation methods are getting popular with the advantage to learn graph data (Wu et al., 2022b). Existing GNN-based methods usually organized user-item interactions as the bipartite graph and captured higher-order collaborative signal with the aid of GNN to achieve personalized recommendation. For example, NGCF (Wang et al., 2019) employed the GNN considering node interactions to mine higher-order collaborative signal in the bipartite graph and effectively enriched representations of users and items in recommender systems. LightGCN (He et al., 2020) effectively improved the efficiency and performance of NGCF by simplifying the GNN in NGCF. For fine-grained decoupling of user preference, MA-GNNs (Zhang et al., 2023) proposed the multi-aspect enhanced GNN to capture multi-aspect collaborative signal in the bipartite graph.

Although some success, existing studies on GNN-based recommendation methods usually directly organized user-item interactions as the bipartite graph, but failed to explore core users and item features activating user-item interactions. Since users' interactive behaviors are driven by specific user and item features (Zhang et al., 2019), ignoring core features causes that user-item interactions fail to be accurately exploited at the feature level. In addition, existing GNNs only focused on integrating information from neighbors in conducting information propagation, but ignored the mutual association among neighbors. Since there is not only user-item interactions but also user-user associations (e.g., the user social relationship) or item-item associations (e.g., the item complementary relationship) (Liu et al., 2022) in real recommender systems, existing GNNs only integrating neighbor information are not enough to capture sufficient structural signal in the bipartite graph. Due to these limitations, there is a certain research space to improve existing GNN-based recommendation methods. Below, we employ Fig. 1 to comb the research motivation.

From Fig. 1, we observe that: **(1) User-item interactions are activated by core user and item features.** Specifically, Fig. 1(a) shows an example on the core features activated recommender system, where u_1 represents the target user, i_1 represents the candidate item and there exists the interaction between u_1 and i_1 . The reason why u_1 interacts i_1 is because the target user is a “Nike fan” and the brand of the candidate item is correspondingly “Nike”. In this case, “Nike fan” and “Nike” can be treated as core features to activate the interaction between u_1 and i_1 . Compared with core features, other user features (e.g., “Male”, “Student”, etc.) or item features (“Shoes”, “Black and white”, etc.) may interfere with the understanding of the user-item interaction, which leads to the utilization of the user-item interaction be redundant. **(2) The information propagation in the bipartite graph not only needs to integrate neighbor information into the center node but also needs to consider the mutual association among neighbors.** Specifically, the bipartite graph showed in Fig. 1(a) contains not only user-item interactions but also the mutual association among users or

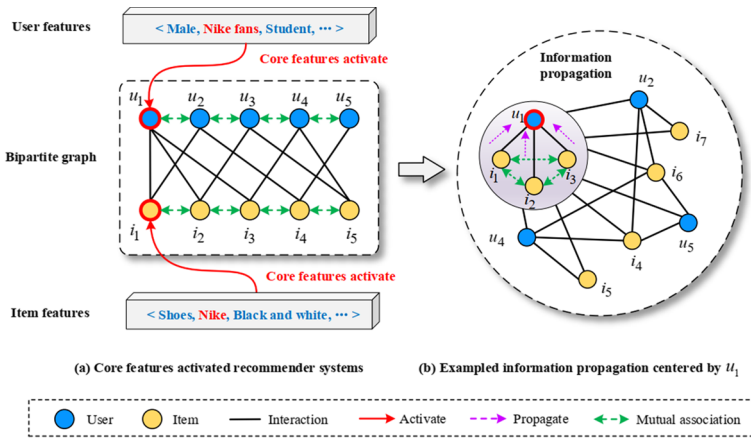


Fig. 1 The toy example to illustrate research motivations

items (e.g., the user social relationship, the item complementary relationship, etc.), which inspires us that capturing structural signal in the bipartite graph needs to comprehensively the association among neighbors. Taking the learning of the representation of u_1 as an example, Fig. 1(b) amplifies the information propagation process in the bipartite graph. Obviously, the information propagation not only needs to integrate propagated information from neighbors (i_1, i_2 and i_3) to u_1 but also needs to consider the mutual association (e.g., the complementary relationship, etc.) among these neighbor items. However, information propagation in existing GNNs (e.g., GraphSage (Wu et al., 2022b), GAT (Wang et al., 2019), etc.) only focuses on integrating information from neighbors to the center node, which fails to capture sufficient structural signal in the bipartite graph.

Inspired by the above observations, we aim to identify core features activating user-item interactions and optimize the information propagation of existing GNNs to improve performance of existing GNN-based recommendation methods. To achieve this goal, our research mainly makes the following contributions:

1. We develop the core features activated graph dual-attention network named C-GDN for personalized recommendation. On one hand, C-GDN explores user-item interactions with core user and item features. On the other hand, C-GDN more sufficiently captures structural signal in the bipartite graph.
2. In order to more accurately explores user-item interactions at the feature level, we design the core feature identifying layer to identify core user and item features activating user-item interactions, which discards the interference of irrelevant features to the utilization of user-item interactions.
3. In order to more sufficiently captured structural signal in the bipartite graph, we develop a novel GNN, graph dual-attention network (GDN), to conduct information propagation in the bipartite graph. Compared with existing GNNs, GDN adopts the dual-attention mechanism to consider not only different contributions from neighbors but also the mutual association among neighbors.

2 Related work

2.1 Feature-aware GNN-based recommendation methods

Feature information is important auxiliary information used in recommender systems (Deng et al., 2022). In relevant studies, common feature information in recommender systems includes context information (Forestiero, 2022), attribute information (Deng et al., 2022), unstructured text information (Hu et al., 2020) and so on. With the increasing popularity of GNN-based recommendation methods, feature-aware GNN-based recommendation method have also attracted wide attention in recent years. Using feature information to enrich representations of users and items is a common practice in feature-aware GNN-based recommendation methods. For example, GNewsRec (Hu et al., 2020) firstly employed convolutional neural network (CNN) to learn text of news, and then input learned text feature into the heterogenous GNN for personalized news recommendation. Some knowledge graph-based recommendation methods including Ke-LinUCB (Gan & Kwon, 2022) employed knowledge features to enrich item representations for achieve knowledge-aware recommendation. Besides, another way to utilize feature information to improve GNN-based recommendation is modeling higher-order feature interactions with GNN. For example, Fi-GNN (Li et al., 2019) represented the multi-field features in a graph structure and modeled feature interactions with GNN for the task on click-through rate (CTR) prediction. Similarly, GMCF (Su et al., 2021) exploited GNN to model two types of feature interactions, the inner interactions and the cross interactions, and then achieved personalized recommendation by graph matching. Although existing studies have tried to integrate feature information to enhance GNN-based recommendation methods, existing studies have not effectively identified the core user and item features activating user-item interactions. Since the intention of a user's interaction to an item is driven by specific user or item features (Zhang et al., 2019), ignoring core user and item features makes the modeling of user interest interfered by irrelevant user and item features. Different from the existing studies, we design the core feature identifying layer in our research to identify the core features of uses and items, which effectively improves feature-aware GNN-based recommendation.

2.2 GNN-based recommendation

Since GNN-based recommendation mainly relies on the information propagation mechanism of GNN to capture higher-order structural signal in the graph (Gan & Zhang, 2023), different information propagation mechanisms evolve into different GNN-based recommendation methods. Common information propagation of GNNs is to integrate information from neighboring nodes to the central node without difference. For example, GCMC (Berg et al., 2017) and PDA-GNN (Wu et al., 2023) adopted the pooling operation to aggregate neighbor information without difference. Furthermore, in order to avoid the central node representation be redundant, some studies conducted information propagation by considering different contributions of neighbor nodes. NGCF (Wang et al., 2019) differently integrated neighbors by improving graph convolutional network (GCN) with information interactions between nodes. MGAT (Tao et al., 2020) designed the gated attention network to learn embeddings of users and items in the multimodal information-aware bipartite graph, which performs well on multimodal recommendation. MAF-GNN (Song et al., 2022) adopted graph attention network (GAT) to respectively multiple user-related and item-related graphs for personalized recommendation. Despite some success, the information propagation of existing GNNs only

focus on integrating neighbor information into the center node. Since there exists mutual potential association among users or items (Liu et al., 2022), only integrating neighbor information into the center node fails to sufficiently capture structural signal in the bipartite graph. Different from existing works, we designed a novel GNN named GDN with dual-attention mechanism to capture structural information in a graph, which considers not only different contributions of neighbors but also mutual association between neighbors in the information propagation.

3 Problem statement

Before proposing our model, we define key concepts and present the task formulation. In addition, the key notations referred to the model are listed in Table 1.

3.1 Key concepts

Definition 1 (Core features) *Core features in our research refer to user and item-related features that drive the user interaction with the item. For example, suppose a user u with features {Teenager, Student, Nike fans} purchased an item i with features {Shoes, Black and white, Nike}, and the purchase motivation is that i is the product of the favorite brand of u . In this case, “Nike fans” is the core user feature to represent u , while “Nike” is the core item feature to represent i . Certainly, there may be also multiple core features for a user or an item when the user-item interaction activated by multiple user and item features.*

Definition 2 (Structural signal) *Structural signal in our research refers to the captured higher-order signal along with the structure of the bipartite graph by multiple layers of*

Table 1 The list of key notations

Notations	Meaning
$\mathbf{E}_u = \{\mathbf{e}_{u_1}, \dots, \mathbf{e}_{u_m}\}$	Initial embedding of users.
$\mathbf{E}_i = \{\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_n}\}$	Initial embedding of items.
$\mathbf{E}_f^u = \{\mathbf{e}_{f_1^u}, \dots, \mathbf{e}_{f_a^u}\}$	Initial embedding of user features.
$\mathbf{E}_f^i = \{\mathbf{e}_{f_1^i}, \dots, \mathbf{e}_{f_b^i}\}$	Initial embedding of item features.
$\gamma_{f_k^u}; \gamma_{f_k^i}$	Gate scores of features f_k^u and f_k^i .
$\mathbf{e}_u^{core}; \mathbf{e}_i^{core}$	Core feature-aware embeddings of u and i .
$\mathbf{e}_{t_i^{Bi}}; \mathbf{e}_{t_j^{Bi}}$	Initial embeddings of neighbors t_i^{Bi} and t_j^{Bi} .
$\mathbf{e}_{h^{Bi}}$	Initial embedding of the center node h .
$\left\{ \mathbf{e}'_{t_1^{Bi}}, \mathbf{e}'_{t_2^{Bi}}, \dots, \mathbf{e}'_{t_o^{Bi}} \right\}$	Association-aware embeddings of neighbors of the center node h .
$\beta_{ht_i}^{Bi}$	The attention score of t_i^{Bi} to h in the bipartite graph
$\mathbf{e}_{h^{Bi}}^{(l)}$	Embedding of center node h after performing l times of GDNs.
$\mathbf{e}_{h^{Bi}}^*$	The final embedding of the center node h .
\tilde{y}_{ui}	The predicted probability of the user-item interaction.

GNNs. In particular, we design a novel GNN, named GDN, to learn the structure of the bipartite graph. Compared with existing GNNs, structural signal captured by GDN considers not only higher-order neighbor information of the center node but also the mutual association among neighbors.

3.2 Task description

Our main objective is to achieve the personalized recommendation by identifying the core features of users and items as well as more sufficiently capturing structural signal in the bipartite graph. Suppose the user set is U , the item set is I , the user feature set is F^U and the item feature set is F^I , the task of our research is to predict users' interaction probability on items, which is mathematically expressed as:

$$\tilde{Y} = f(U, I, F^U, F^I) \tag{1}$$

Where \tilde{Y} is the predicted interaction probability of all users on items, $f(\cdot)$ denotes the predicting function to predict the interaction probability. With the predicted interaction probability, the Top-N item list is recommended for each user.

4 Methodology

4.1 Overview of C-GDN

The overview of C-GDN is illustrated in Fig. 2. According to in Fig. 2, we observe that C-GDN consists of four key parts: **a) Core feature identifying layer**, which is used to identify core features of users and items and initialize the bipartite graph. **b) Information propagation layer**, which is used to conduct information propagation in the bipartite graph with the designed GDN. **c) Embedding fusion layer**, which is used to fuse different order embeddings of users or items and generate final embeddings of users and items. **d) Prediction**

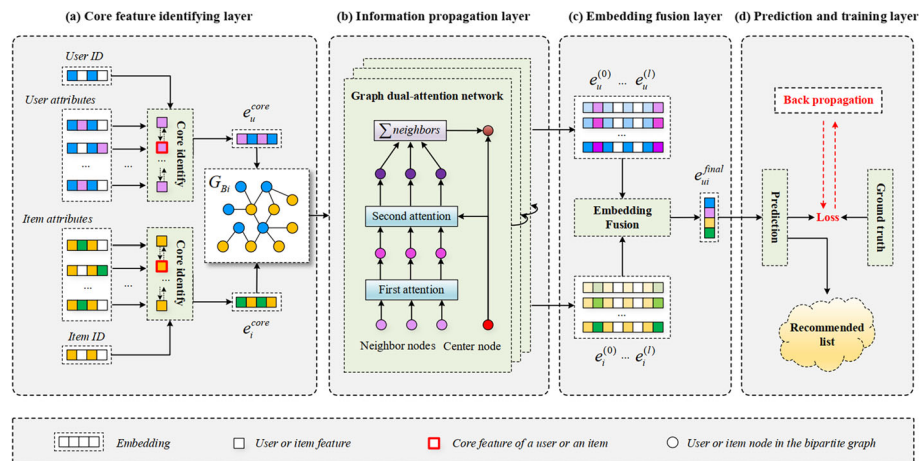


Fig. 2 Overview of C-GDN

and train layer, which is used to predict the probability of users interacting with items and train parameters of the whole model.

4.2 Core feature identifying layer

The core feature identifying layer is used to generate core feature-aware embeddings of users and items. For this goal, we firstly encode users, items, user attributes and item attributes with their ID information to generate their initial embeddings. And then, we adopt the gate mechanism to decide whether the feature would be the core feature to represent the user or the item. Finally, we use core features to represent the bipartite graph built on user-item interactions. Accordingly, this layer contains three steps, embedding initializing, core feature identifying and bipartite graph initializing. Next, we illustrate these steps in detail.

4.2.1 Embedding initializing

The embedding initializing essentially is to learn low-dimensional initial embeddings of all users, items, and their features. Based on ID information, we adopt widely used embedding layer (Kang et al., 2021) to generate initialized embeddings of users, items and their features. Suppose there are m users, n items, a user features and b item features, the initialized embeddings are expressed as:

$$\mathbf{E}_{initial} = \{\mathbf{E}_u; \mathbf{E}_i; \mathbf{E}_f^u, \mathbf{E}_f^i\} \quad (2)$$

Where, $\mathbf{E}_{initial}$ denotes the initialized embedding matrix including the initial user embedding $\mathbf{E}_u = \{\mathbf{e}_{u_1}, \dots, \mathbf{e}_{u_m}\}$, the initial item embedding $\mathbf{E}_i = \{\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_n}\}$, the initial feature embedding $\mathbf{E}_f^u = \{\mathbf{e}_{f_1^u}, \dots, \mathbf{e}_{f_a^u}\}$ and $\mathbf{E}_f^i = \{\mathbf{e}_{f_1^i}, \dots, \mathbf{e}_{f_b^i}\}$.

4.2.2 Core feature identifying

The process of core feature identifying is to control the introduction of features for representing users and items, which is essentially a process of information control to extracts core features and discards irrelevant features from multiple features of users and items (Zhang et al., 2019). Since the improved gated mechanism proposed by Tao et al. (2020) has the natural advantage of information control and more flexibly decides information representing the target object, we adopt the improved gating mechanism proposed by Tao et al. (2020) to realize the core feature identifying.

To be specific, suppose a user u includes x features, an item i includes y features, initial feature embeddings of u are written as $\{\mathbf{e}_{f_1^u}, \mathbf{e}_{f_2^u}, \dots, \mathbf{e}_{f_x^u}\}$, and initial feature embeddings of i are written as $\{\mathbf{e}_{f_1^i}, \mathbf{e}_{f_2^i}, \dots, \mathbf{e}_{f_y^i}\}$, the gate scores feature f_k^u and feature f_k^i based on the improved gated mechanism are calculated as:

$$\gamma_{f_k^u} = \text{softmax}\left(\frac{\mathbf{W}^u(\mathbf{e}_{f_k^u} \parallel \mathbf{e}_u) + \mathbf{e}_{f_k^u} \odot \mathbf{e}_u}{\sqrt{d}}\right) \quad (3)$$

$$\gamma_{f_k^i} = \text{softmax}\left(\frac{\mathbf{W}^i(\mathbf{e}_{f_k^i} \parallel \mathbf{e}_i) + \mathbf{e}_{f_k^i} \odot \mathbf{e}_i}{\sqrt{d}}\right) \quad (4)$$

Where $\gamma_{f_k^u}$ and $\gamma_{f_k^i}$ respectively represent gate scores of f_k^u and f_k^i . d is the embedding size, $(\cdot \parallel \cdot)$ is the concatenation operation, \odot denotes the element-wise product, $\mathbf{e}_{f_k^u} \odot \mathbf{e}_u$

represents the information interaction between f_k^u and u , while $\mathbf{e}_{f_k^i} \odot \mathbf{e}_i$ represents the information interaction between f_k^i and i . $(\mathbf{e}_{f_k^u} \parallel \mathbf{e}_u)$ and $(\mathbf{e}_{f_k^i} \parallel \mathbf{e}_i)$ are introduced to reserve the original features of the user and the item.

Furthermore, with gate scores of user features and item features, we generate core feature-aware embeddings of u and i as follows:

$$\mathbf{e}_u^{core} = \sum_{k=1}^x \gamma_{f_k^u} \mathbf{e}_{f_k^u} \tag{5}$$

$$\mathbf{e}_i^{core} = \sum_{k=1}^y \gamma_{f_k^i} \mathbf{e}_{f_k^i} \tag{6}$$

Where \mathbf{e}_u^{core} and \mathbf{e}_i^{core} are core feature-aware embeddings of u and i . Compared with initialized embeddings of users and items, core feature-aware embeddings emphasize core features activating user-item interactions.

4.2.3 Bipartite graph initializing

Initializing the bipartite graph is the basic to apply GNN (Wu et al., 2022b). Similar with existing studies (Wang et al., 2019; He et al., 2020), we construct the bipartite graph according to user-item interactions. To be specific, if a user interacted with an item, there exists the edge between the user node and the item node in the bipartite graph. We write the bipartite graph as $G_{Bi} = (V, E)$, where V represents the node set including users and items and E represents the edge set determined by user-item interactions. Since user or item nodes contained in G_{Bi} , the initialization of G_{Bi} essentially is to represent user nodes and item nodes in the graph. In our research, we initialize user nodes and item nodes with core feature-aware embeddings of users and items learned at Section 4.2.2, which aims to support the learning of G_{Bi} at the feature level.

4.3 Information propagation layer

The information propagation layer is used to sufficiently capture structural signal in the bipartite graph. For this goal, we designed the GDN to conduct information propagation in the bipartite graph. GDN contains three steps, neighbors’ mutual association modeling, neighbor information integrating and higher-order connectivity modeling. Next, we illustrate these steps in detail.

4.3.1 Neighbors’ mutual association modeling

Since the self-attention mechanism has the advantage on modeling associations of different embeddings (Zhang et al., 2019), we exploit the self-attention mechanism to model the mutual association among neighbors. Assuming that local neighbors of center node h in a bipartite graph is $N_h^{Bi} = \{t_1^{Bi}, t_2^{Bi}, \dots, t_o^{Bi}\}$. We model the mutual association by defining the attention weight α_{ij} between two neighbors, t_i^{Bi} and t_j^{Bi} , as follows:

$$\alpha_{i,j} = \frac{\mathbf{e}_{t_i^{Bi}} (\mathbf{e}_{t_j^{Bi}})^T}{\sum_{j=1}^o \mathbf{e}_{t_i^{Bi}} (\mathbf{e}_{t_j^{Bi}})^T} \tag{7}$$

Where $\mathbf{e}_{t_i^{Bi}}$ and $\mathbf{e}_{t_j^{Bi}}$ are initial embeddings of t_i^{Bi} and t_j^{Bi} . o represent the number of neighbors of h . Further, we obtain the association-aware embedding of t_i^{Bi} as follows:

$$\mathbf{e}'_{t_i^{Bi}} = \sum_{j=1}^o \alpha_{ij} \mathbf{e}_{t_j^{Bi}} \tag{8}$$

According to (8), embeddings of neighbors of center node h are comprehensively updated as $\{\mathbf{e}'_{t_1^{Bi}}, \mathbf{e}'_{t_2^{Bi}}, \dots, \mathbf{e}'_{t_o^{Bi}}\}$.

4.3.2 Neighbor information integrating

Inspired by GAT (Wang et al., 2019), we use attention mechanism to integrate neighbors with their different contributions. To be specific, based on obtained association-aware embeddings of neighbors at Section 4.3.1, we calculate attention score of neighbor t_i^{Bi} to the center node h with (9):

$$\beta_{ht_i}^{Bi} = \frac{\exp(\text{LeakyReLU}(\mathbf{W}_3[\mathbf{W}_1 \mathbf{e}_{h^{Bi}} \parallel \mathbf{W}_2 \mathbf{e}'_{t_i^{Bi}}]))}{\sum_{t=1}^n \exp(\text{LeakyReLU}(\mathbf{W}_3[\mathbf{W}_1 \mathbf{e}_{h^{Bi}} \parallel \mathbf{W}_2 \mathbf{e}'_{t_i^{Bi}}]))} \tag{9}$$

Where, $\beta_{ht_i}^{Bi}$ is the attention score of t_i^{Bi} , $\mathbf{e}_{h^{Bi}}$ is the initial embedding of h , $\text{LeakyReLU}(\cdot)$ represents the nonlinear transformation function, \mathbf{W}_1 , \mathbf{W}_2 and \mathbf{W}_3 are all the trainable weight matrices.

4.3.3 Higher-order connectivity modeling

Based on the dual-attention mechanism at Sections 4.3.1 and 4.3.2, we model the high-order connectivity to capture higher-order structural signal as following:

$$\mathbf{e}_{h^{Bi}}^{(l)} = \text{LeakyReLU}(\mathbf{T}_1^{(l-1)} \sum_{t_i \in N_h^{Bi}} \beta_{ht_i}^{Bi} \mathbf{e}'_{t_i^{Bi}}^{(l-1)} + \mathbf{T}_2^{(l-1)} \mathbf{e}_{h^{Bi}}^{(l-1)}) \tag{10}$$

Where, $\mathbf{e}_{h^{Bi}}^{(l)}$ represents the embedding of h after performing l times of GDNs, which captures structural signal for represent node h^{Bi} in the bipartite graph. $\text{LeakyRelu}(\cdot)$ represents the nonlinear activation function, $\mathbf{T}_1^{(l-1)}$ and $\mathbf{T}_2^{(l-1)}$ are both trainable parameter matrices.

4.4 Embedding fusion layer

The embedding fusion layer is used to fuse different order embeddings within the bipartite graph to generate the final user and item embeddings. Suppose the center node h gets l different embeddings, *i.e.*, $\mathbf{e}_{h^{Bi}}^{(0)}, \mathbf{e}_{h^{Bi}}^{(1)}, \dots, \mathbf{e}_{h^{Bi}}^{(l)}$, after L times of GDNs in the bipartite graph, then the final embedding of h is formalized as:

$$\mathbf{e}_{h^{Bi}}^* = \sum_{l=0}^L \tau^{(l)} \mathbf{e}_{h^{Bi}}^{(l)} \tag{11}$$

Where, $\mathbf{e}_{h^{Bi}}^*$ represents the final embedding of h . $\tau^{(l)}$ denotes the importance of the specific order embeddings. Inspired by LightGCN (He et al., 2020), we set $\tau^{(l)}$ uniformly as $\frac{1}{L+1}$.

When h represents the user u , we obtain the final embedding of u , e_u^* . While, when h represents the item i , we obtain the final embedding of i , e_i^* .

4.5 Prediction and training layer

Based on obtained final embeddings of users and items, we further exploit the Multiple Layer Perceptron (MLP) to predict the interaction of users on items. To be specific, based on the final user embedding e_u^* and the final item embedding e_i^* , the prediction process of the interaction is expressed as:

$$\tilde{y}_{ui} = f(\mathbf{e}_u^* \parallel \mathbf{e}_i^*) \quad (12)$$

Where, $(\cdot \parallel \cdot)$ represents the concatenation operation, $f(\cdot)$ is the MLP. It is worth mentioning that, the MLP is composed of multiple layers of full connect network. If the full connect network is the last layer of the MLP, the activation function is set Sigmoid, and if not, the activation function is set ReLU. \tilde{y}_{ui} represents the predicted probability of the user-item interaction.

To learn model parameters, we optimized the pairwise BPR loss (Wang et al., 2019), which has been intensively used in recommender systems. The objective function is as follows:

$$Loss = \sum_{(u,i,j) \in O} -\ln \sigma(\tilde{y}_{ui} - \tilde{y}_{uj}) + \lambda \|\Theta\| \quad (13)$$

Where, $O = \{(u, i, j) \mid (u, i) \in R^+, (u, j) \in R^-\}$ represents the pairwise training data, R^+ denotes the observed data, and R^- represents the unobserved data, $\sigma(\cdot)$ is the sigmoid function, Θ represents all parameters involved in our model, λ controls the L_2 regularization strength to prevent overfitting. In addition, we update model parameters with the gradients of the loss function.

5 Experiments

In this section, we first introduced datasets (see Section 5.1), and then described experiment settings including baselines, evaluation metrics and paramant settings (see Section 5.2). Furthermore, we conducted comparison experiments (see Section 5.3), ablation experiments (see Section 5.4), parameter sensitivity analysis(see Section 5.5).

5.1 Datasets

We used the three benchmark datasets, Movielens-100K¹, Movielens-1M² and Taobao³, commonly used in the recommender system to verify the validity of the proposed model. These three datasets are mainly used to verify the effectiveness of recommendation methods in different recommendation scenarios (the movie recommendation and the electronic commerce recommendation), so as to prove that research results can be generalizable to different domains. The statistical information of three datasets is shown in Table 2. Where, U-feature

¹ <https://www.kaggle.com/prajitdatta/movielens-100k-dataset>

² <https://www.kaggle.com/datasets/odedgolden/movielens-1m-dataset>

³ <https://www.kaggle.com/datasets/pavansanagapati/ad-displayclick-data-on-taobaocom>

Table 2 Statistics of the datasets

Dataset	Movielens-100k	Movielens-1M	Taobao
Users	943	6,040	85,518
Items	1,682	3,952	11,822
Interaction	100,000	1,000,209	178,314
UI-Sparsity	93.70%	95.81%	99.98%
U-features types	4	4	3
I-features types	3	3	3

types and I-features types respectively denotes the number of user features types and item features types in corresponding datasets.

The details of each dataset are as follows:

- (1) Movielens-100K and Movielens-1M are movie-oriented recommendation datasets (Harper & Konstan, 2015), which include not only the rating data of users to movies, but also rich fields on user and movie features. For our research, we determined the interaction relationship of a user to a movie according to the rating data. In addition, we selected three fields (“Age”, “Gender”, “Occupation” and “Zip code”) to determine user features and two fields (“Release date”, “Genres”, “Average rating”) to determine movie features.
- (2) Taobao is an advertising-oriented recommendation dataset (Zhou et al., 2018), which not only includes click data of users to advertisings, but also includes rich features of users and advertisings. For this dataset, we determined the interaction relationship of a user to an advertising according to the click data. In addition, we selected three fields (“Age level”, “Gender code” and “Shopping level”) to determine user features and two fields (“Campaign”, “Category” and “Brand”) to determine advertising features.

In order to ensure the quality of datasets, we use 20-core setting (Gan & Ma, 2022) to ensure that each user and item have at least 20 interactions during the experiment. For each processed dataset, we randomly selected 80% of the historical interaction items of each user as the training set and the remaining interaction items as the test set. For the training set, 10% of the interaction items are randomly selected as the validation set.

5.2 Experiment setting

5.2.1 Baselines

To demonstrate the effectiveness, we compare C-GDN with six mainstream baselines. These six baselines are mainly from three categories of recommendation methods, including **traditional deep recommendation methods** (NFM and GCMC), **GNN-based recommendation methods without feature information** (NGCF and LightGCN) and **GNN-based recommendation methods with feature information** (Fi-GNN and GMCF). Specifically, the reason to introduce NFM and GCMC to test whether GNN-based recommendation methods is better than traditional deep recommendation methods, so as to prove that GNN are superior than traditional neural networks to achieve recommendation. In addition, the reason to introduce NGCF and LightGCN to test whether GNN-based recommendation methods with feature information are superior than methods without feature information, so as to prove

that feature information is beneficial for GNN-based recommendation. Moreover, the reason to introduce Fi-GNN and GMCF to test whether C-GDN proposed by us is superior than existing GNN-based recommendation methods with feature information, so as to prove that our innovations are beneficial for improve existing research. The following introduces these baselines:

- **NFM** (He et al., 2017): This model is a mainstream collaborative filtering method based on deep learning. It captures the nonlinear feature interaction between the user and the item by using multiple hidden layers and concatenation operations.
- **GCMC** (Berg et al., 2017): This model is a classic graph-based recommendation method. It adopts the graph encoder to generate the representations for users and items, where only the first-order neighbors are considered.
- **NGCF** (Wang et al., 2019): This model is a mainstream GNN-based method. It uses the improved GCN to model the high-order connectivity between users and items.
- **LightGCN** (He et al., 2020): This model is a simplified version of NGCF by removing feature transformation and nonlinear activation modules. It makes GNN-based methods more concise and appropriate for personalized recommendation.
- **Fi-GNN** (Li et al., 2019): This model is a feature interaction-based CTR prediction method, which represents each data sample as a feature graph that each node in the graph is a feature field. Then, it models the interactions among features with GNN. In our research, we deployed this model in the Top-N recommendation scenario.
- **GMCF** (Su et al., 2021): This is also a feature interaction-based GNN recommendation method, which explicitly models the inner and cross feature interactions to represent users and items, and then performs personalized recommendation by the graph matching.

5.2.2 Evaluation metrics

We adopted four widely used evaluation metrics (Wang et al., 2019), Recall@N (Rec@N), Precision (Pre@N), Hit Ratio (HR@N) and Normalized Discounted Cumulative Gain (NDCG@N), to evaluate the Top-N recommendation performance of different models. Where, Rec@N measures the proportion of the number of user interactions in the recommended N items to the entire interaction set. Pre@N measures the fraction of the items the user will click among the recommended N items. HR@N is a recall-based metric, measuring the proportion of users who have at least one interact on the recommended items. NDCG@N differentiates the contributions of the accurately recommended items based on their ranking positions. The larger values of Rec@N, Pre@N, HR@N and NDCG@N, the better the recommendation performance. In our research, we report the value of N as 10 and 20.

5.2.3 Parameter settings

The experimental environments for our research were Linux, Python 3.7.4, TensorFlow 1.16.1 on Nvidia Geforce GTX 1080 Ti GPU. We conduct the grid searching for the embedding dimension in {32, 64, 128}, the sample number of local neighbors in {5, 10, 15, 20}, the dropout ratio in {0.1, 0.2, 0.3, 0.4}, the learning ratio in 0.0001, 0.0005, 0.001, 0.005, the decay rate in {0.97, 0.98, 0.99} and the regularization weight in {10e-1, 10e-2, 10e-3, 10e-4, 10e-5}. After searching, the embedding dimension is determined as 64, and the dropout ratio is determined as 0.1, the initial learning rate is determined as 0.001, the decay rate is determined as 0.98, and the regularization weight is determined as 10e-2. For baselines, we

adjusted the parameters to a better value suitable for the experimental data set by referring to the original paper. For parameters, we employ Xavier initializer to initialize parameters. We choose the Mini-Batch Adam to optimize parameters. The batch size is 256 for the Movielens-100K dataset and 1024 for the Movielens-1M dataset and the Taobao dataset. The epoch size is set 200 for convergence. The important hyper-parameters including the number of GDN layers and the number of MLP layers are reported in Section 5.5.

5.3 Comparison experiments

We compared the proposed C-GDN with baselines and obtained comparative results as shown in Tables 3, 4, and 5. According to these tables, we observe that C-GDN outperforms baselines on three datasets with the p-value of all metrics rejecting the null hypothesis with a level of significance of $\alpha=0.01$. These results prove the ability of C-GDN in improved all evaluation metrics significantly ($p<0.01$) compared to the baselines. The superior performance is due to: a) Compared with NMF and GCMC, C-GDN not only captures higher-order structural signals, but also employs feature information to enhance representations of users and items. b) Compared with NGCF and LightGCN, C-GDN employs core feature information to enhance representations of users and items. c) Compared with Fi-GNN and GMCF, C-GDN not only identifies core features activating user-item interactions, but also improves the information propagation in the bipartite graph with designed GDN. Moreover, we find that almost of all metrics of C-GDN have more significant improvement on Top-10 than Top-20. Since online users usually pay more attention to the top recommendations, C-GDN more matches actual demand of online users. There are also some findings by comparing all baselines:

- (1) Traditional deep recommendation methods (i.e., NMF and GCMC) show worse performance than other methods on three datasets. The reason for this result is that these two baselines fail to consider higher-order collaborative signal, while other baselines all employ GNN to consider higher-order collaborative signal to learn representations of users and items. It is also confirmed that GNN-based recommendation methods are superior to traditional deep recommendation methods.
- (2) GNN-based methods without features (i.e., NGCF, LighGCN) are better than traditional deep recommendation methods but worse than feature-aware GNN-based recommen-

Table 3 Overall performance comparison on Movielens-100K

Model	Rec@10	Rec@20	Pre@10	Pre@20	HR@10	HR@20	NDCG@10	NDCG@20
NMF	0.0194	0.0723	0.2539	0.2317	0.8490	0.8868	0.2287	0.3092
GCMC	0.0338	0.0817	0.3008	0.2988	0.8821	0.9293	0.3234	0.4140
NGCF	0.0358	0.1192	0.4440	0.3692	0.9071	0.9469	0.3571	0.4591
LightGCN	0.0394	0.1485	0.5300	0.4369	0.9306	0.9474	0.4261	0.569
Fi-GNN	0.0429	0.1584	<u>0.7741</u>	<u>0.6561</u>	<u>0.9138</u>	<u>0.9642</u>	0.4433	0.5683
GMCF	<u>0.0431</u>	<u>0.1626</u>	0.7329	0.6412	0.9116	0.9557	<u>0.4462</u>	<u>0.5693</u>
C-GDN	0.0460*	0.1696*	0.8211*	0.6946*	0.9422*	0.9722*	0.4860*	0.5764*
Imp	6.73%	4.31%	6.07%	5.87%	1.25%	0.83%	8.92%	1.25%
P-value	2.18e-4	4.11e-5	2.12e-6	2.35e-5	6.01e-5	2.04e-4	3.21e-5	1.09e-5

Note: Imp represents the percentage of performance improvement shown by our model compared to the best baseline. Bold scores are the best in each row, while underlined scores are the second best. * denotes statistically significant improvement by t-test with $p<0.001$. The same below

Table 4 Overall performance comparison on Movielens-1M

Model	Rec@10	Rec@20	Pre@10	Pre@20	HR@10	HR@20	NDCG@10	NDCG@20
NMF	0.1626	0.2483	0.6747	0.6220	0.6640	0.8167	0.2257	0.2355
GCMC	0.1658	0.2510	0.6838	0.7099	0.7424	0.8842	0.2308	0.2290
NGCF	0.1813	0.2690	0.7285	0.7105	0.7901	0.9209	0.2475	0.2550
LightGCN	0.1843	0.2744	0.7527	0.7139	0.8173	0.9280	0.2461	0.2556
Fi-GNN	0.1997	<u>0.2970</u>	0.7398	0.7121	<u>0.8215</u>	<u>0.9295</u>	0.2556	0.2579
GMCF	<u>0.2003</u>	0.2797	<u>0.7450</u>	<u>0.7212</u>	0.8115	0.9203	<u>0.2583</u>	<u>0.2751</u>
C-GDN	0.2069*	0.3062*	0.7709*	0.7295*	0.8441*	0.9494*	0.2677*	0.2792*
Imp	3.30%	3.10%	2.42%	1.33%	2.75%	2.14%	3.64%	1.49%
P-value	5.49e-4	1.13e-4	2.06e-4	2.06e-4	4.08e-5	1.13e-6	6.01e-4	1.14e-5

dation methods (i.e., Fi-GNN and GMCF) on three datasets. This result illustrates that considering feature information is beneficial to achieve more effective recommendation. Moreover, LighGCN outperforms NGCF on three datasets, which indicates that improving GNN is beneficial to improve recommendation performance.

- (3) Recommendation performance on two Movielens datasets is better than the Taobao dataset. This result illustrates that the same recommendation method may have different performance for different recommendation scenarios. The reason for this phenomenon may be that users' rating behaviors in the movie recommendation scenario more directly reflect users' interests, while users' click behaviors in the advertising recommendation scenario are easily disturbed by the platform operating mechanism. Therefore, the movie recommendation more accurately models users' interests than the advertising recommendation.

5.4 Ablation experiments

5.4.1 The effectiveness of identifying core features

In order to achieve the ablation analysis of the effectiveness of identifying core features, we generate three variants of C-GDN, which as followings:

Table 5 Overall performance comparison on Taobao

Model	Rec@10	Rec@20	Pre@10	Pre@20	HR@10	HR@20	NDCG@10	NDCG@20
NMF	0.0760	0.1097	0.1171	0.0943	0.4530	0.6602	0.0916	0.1157
GCMC	0.0752	0.1089	0.1177	0.0951	0.4589	0.6522	0.1000	0.1189
NGCF	0.0875	0.1262	0.1201	0.1032	0.5337	0.7341	0.1080	0.1249
LightGCN	0.0977	0.1415	0.1310	0.1102	0.5520	0.7727	0.1202	0.1305
Fi-GNN	0.1055	0.1445	0.1386	0.1132	0.5868	0.8018	0.1255	0.1339
GMCF	<u>0.1095</u>	<u>0.1502</u>	<u>0.1485</u>	<u>0.1216</u>	<u>0.6097</u>	<u>0.8637</u>	<u>0.1446</u>	<u>0.1504</u>
C-GDN	0.1154*	0.1533*	0.1542*	0.1289*	0.6227*	0.8731*	0.1503*	0.1553*
Imp	5.39%	2.06%	3.84%	6%	2.13%	1.09%	3.94%	3.26%
P-value	1.67e-3	3.21e-4	2.06e-5	2.95e-5	1.76e-3	1.13e-4	1.45e-3	2.75e-4

- **GDN**, this model doesn't consider the core features of users and items and only initializes the bipartite graph with embeddings of users and items based on ID information. For learning of the bipartite graph, we use GDN to conduct information propagation.
- **UC-GDN**, this model doesn't consider core item features and only considers core user features. For the initialization of the bipartite graph, we use embeddings based on core user features to initialize user nodes, while embeddings based on item ID information to initialize item nodes. For learning of the bipartite graph, we use GDN to conduct information propagation.
- **IC-GDN**, this model doesn't consider core user features and only considers core item features. For the initialization of the bipartite graph, we use embeddings based on user ID information to initialize user nodes, while embeddings based on core item features to initialize item nodes. For learning of the bipartite graph, we use GDN to conduct information propagation.

Figure 3 shows performance comparison of C-GDN with three variants and LightGCN (a typical GNN-based recommendation method) on three datasets. From Fig. 3, we observe that C-GDN outperforms GDN, UC-GDN, IC-GDN and LightGCN. This result illustrates that it is beneficial to identify core features both of users and items for personalized recommendation. The reason for this result is that C-GDN, compared with other models, not only identifies core features activating user-item interactions both of users and items. Therefore, C-GDN more accurately inferred user preference by overcoming the disturbing of irrelevant user and item features.

In addition, we notice that UC-GDN and IC-GDN have absolute advantages over GDN on different datasets. This result indicates that either core user features or core item features are beneficial to improve recommendation performance. Another interesting finding is that IC-GDN always performs better than UC-GDN, which illustrates that core item features are more significant to understand user-item interactions than core user features. The reason for

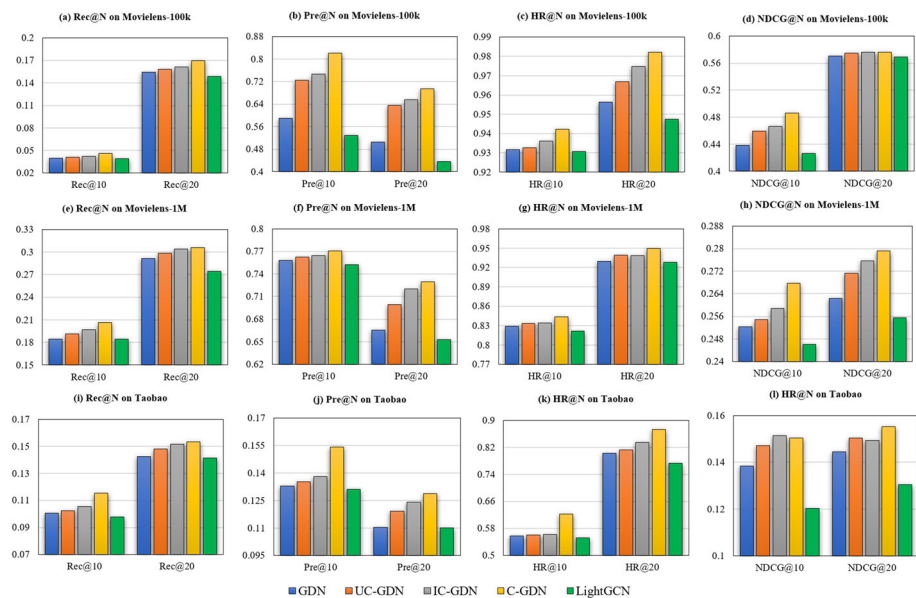


Fig. 3 Ablation analysis on identifying core features

this result may be due to that core item features is more related to users’ decision motivations than core user features. Moreover, compared LighGCN, GDN shows better recommendation performance. This result confirms that our designed GDN provides better recommendation support than mainstream improved GNN even without considering the feature information.

5.4.2 The effectiveness of the designed GDN

In order to achieve the ablation analysis of the designed GDN, we generate two variants of C-GDN, which as followings:

- **C-GFN**, this model removes the second attention mechanism (see Section 4.3.2) modeling different contributions of neighbors and only considers the mutual association among neighbors with the first attention mechanism (see Section 4.3.1) in the information propagation.
- **C-GSN**, this model removes the first attention mechanism (see Section 4.3.1) modeling the mutual association among neighbors and only considers the different contributions of neighbors with the first attention mechanism (see Section 4.3.2) in the information propagation.

Figure 4 shows performance comparison of C-GDN and its two variants on three datasets. From Fig. 4, we observe that C-GDN outperforms C-GFN and C-GSN. This result illustrates that identifying different contributions of neighbors and considering the mutual association between neighbors are both important. Interestingly, we also observe C-GFN significantly outperforms C-GSN on different datasets, which indicates that considering the mutual association of neighbors is more significant than considering the different contributions of neighbors in the information propagation.

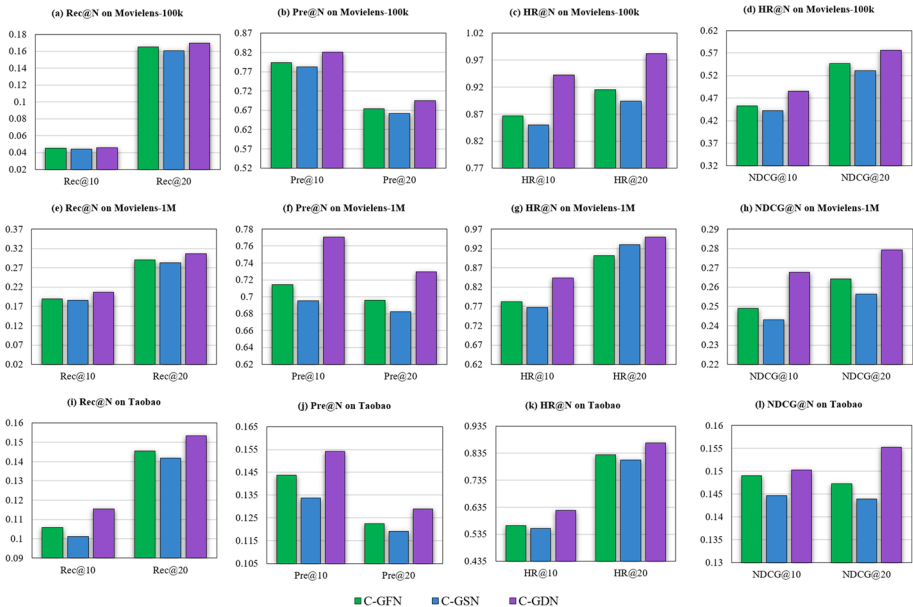


Fig. 4 Ablation analysis on f the designed GDN

5.5 Parameter sensitivity analysis

5.5.1 Effect of the number of GDN layers on recommendation performance

The GDN is used to capture structural signal in the bipartite graph. Since the number of GNN layers determines utilization of higher-order neighbor information (Wang et al., 2019; Liu et al., 2022), it is important to determine proper number of GDN layers in the bipartite graph. In order to test the effect of the number of GDN layers on recommendation performance, we search the number of layers in {1,2,3,4} respectively. Figure 5 shows performance of C-GDN corresponding to different layers of GDN on different datasets.

From Fig. 5, we have the following observations: a) When $L > 1$, C-GDN achieve improvement over $L=1$ on three datasets. The reason for this result is that higher-order structural signal in the bipartite graph is more beneficial for improving recommendation performance. b) When $L=3$ on the Movielens-100K dataset, $L=2$ on the Movielens-1M dataset and $L=2$ on the Taobao dataset, C-GDN achieve the better recommendation performance. While, when $L > 3$ on the Movielens-100K dataset, $L > 2$ on the Movielens-1M dataset and $L > 2$ on the Taobao dataset, the performance of C-GDN begins to decline. This result illustrates that the layers of GDN have effect on the performance of C-GDN and the optimal number of GDN layers on different datasets is different. According to experimental results, we respectively set L as 3, 2 and 2 for the Movielens-100K dataset, the Movielens-1M datasets and the Taobao dataset.

5.5.2 Effect of the number of fully connected layers in MLP on recommendation performance

The MLP is used to model the nonlinear relationship of the final representations of users or items. We search the number P of fully connected layers in {1,2,3,4,5} respectively and

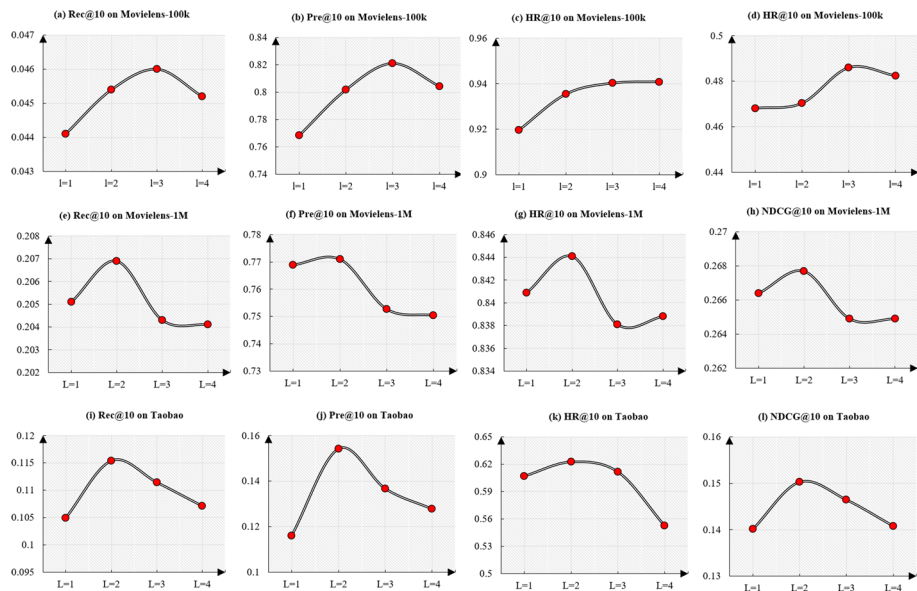


Fig. 5 Performance of C-GDN with different layers of GDN

evaluate the performance of our model on three datasets. Figure 6 shows performance of C-GDN corresponding to different P on three datasets. From Fig. 6, we have the following observations: As the number of layers increases, the performance of the model tends to be stable. This result illustrates MLP is beneficial to improve the stability of recommendation performance. In addition, for the Movielens-100K dataset and the Taobao dataset, C-GDN has better performance when the number of layers is 2 and begins to decline when the number is greater than 2. In addition, for the Movielens-1M dataset, C-GDN has better performance when the number of layers is 1 and begins to decline when the number is greater than 1. Therefore, we respectively set the number of MLP layers as 2 for the Movielens-100K dataset, 1 for the Movielens-1M dataset and 2 for the Taobao dataset.

6 Case study

In this section, we adopt the case study on Movielens-100K dataset to prove the effectiveness of identifying core user and item embeddings. The following is a detailed description of the case study.

6.1 Visual analysis of core features of users and items

In order to prove that C-GDN well identifies core features of users or items, we visualize gate scores of features of example users and items. Specially, we first randomly select five users in the test set and randomly select an interacted item for each user to generate five example users and items. And then, we present gate scores of features of example users and items, see Fig. 7. In each radar chart, each axis denotes a type of user or item features, each circle denotes the fixed scale value, data points on each axis represent gating scores

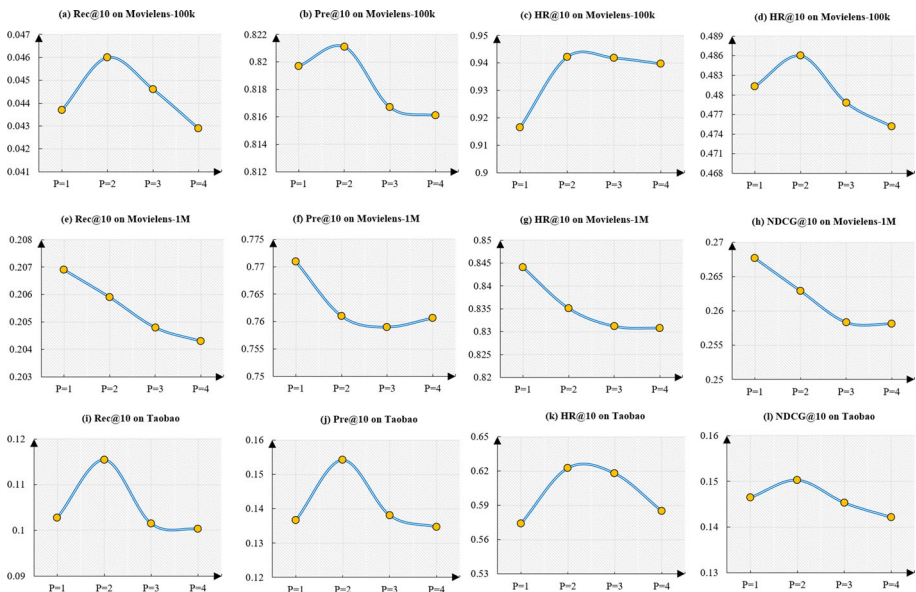


Fig. 6 Performance of C-GDN with different fully connected layers in MLP

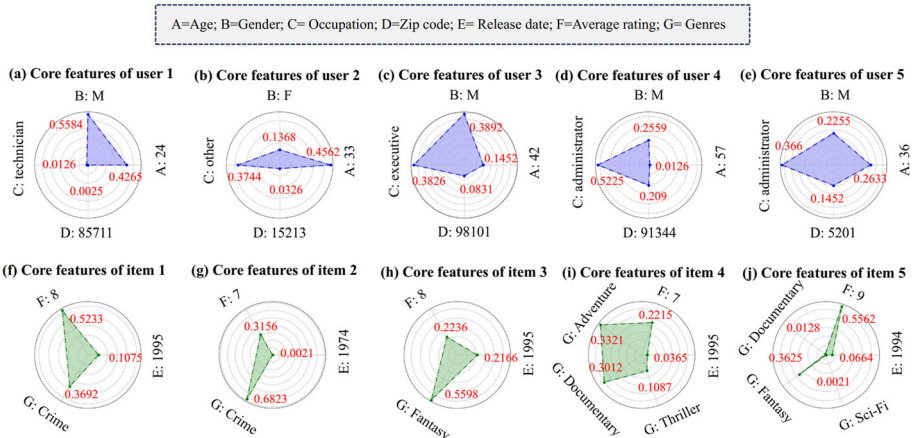


Fig. 7 Visualized gate scores of user and item features

(red font) for specific features, the shaded area with color denotes the combined results of multiple features to represent a user or an item. The reason why radar charts are chosen to visualize gate scores of features is that the radar chart has the advantage to visually represent and compare multidimensional data.

Obviously, in Fig. 7, different features have different contributions to represent the user or the item. Take user 1 and item 1 as the example, “Age: 24” and “Gender: M” are obviously more significant than “Occupation: technician” and “Zip code: 85711” to represent user 1, while “Average rating: 8” and “Genres: Crime” are obviously more significant than “Release date: 1995”. Therefore, in terms of the interaction between user 1 and item 1, “Age: 24” and “Gender: M” are core user features while “Genres: Crime” and “Date: 1995” are core item features. In addition, we also observe that different core feature types play different roles to activate user-item interactions. To be specific, core features of user 1 emphasize “Gender” and “Age” while core features of user 2 emphasize “Occupation” and “Age”. Correspondingly, core features of item 1 emphasize “Genres” and “Average Rating” while core feature of item 2 only emphasizes “Genres”. According to these findings from the gate scores, we clearly understand core features activate users’ interaction to items. Therefore, it is enough to prove that C-GDN well identify core features of users or items and identifying core features enhances the interpretability of recommendations to a certain extent.

6.2 Visual analysis of embeddings of users and items

In order to prove that identifying core features is beneficial to represent users and items. We visually analyze embeddings of users and items of GDN and C-GDN, where GDN removes the core feature identifying layer of C-GDN. Specially, we randomly selected five users with more than twenty interacted items and five items with more than twenty interacted users. Figure 8 visualizes embeddings of interacted items of five users and interacted users of five items.

According to the nature of collaborative filtering, item embeddings with different user labels or user embeddings with different item labels should have a higher degree of differentiation, and item embeddings with the same user label or item embeddings with the same user label should have a higher degree of aggregation. Obviously, item embeddings with

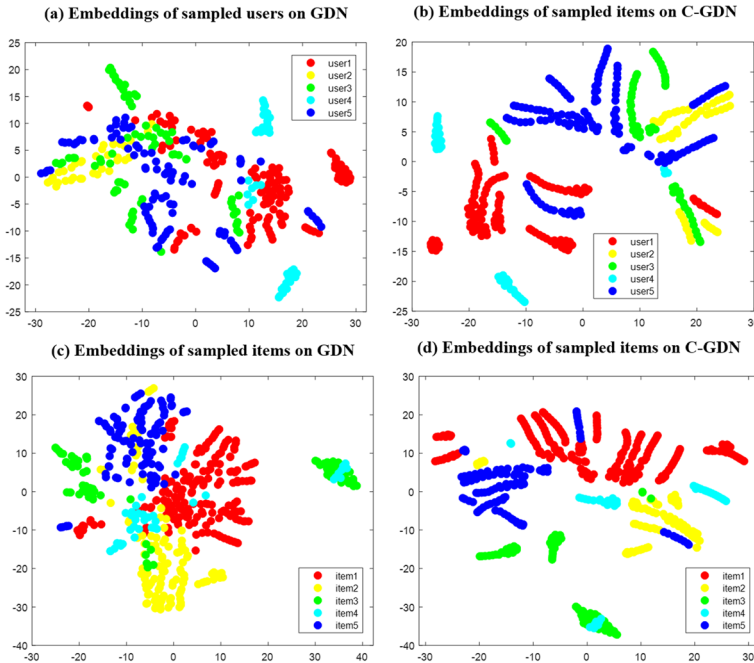


Fig. 8 Visualized embeddings of users and items on different models

different user labels in Fig. 8(a) are more difficult to distinguish than Fig. 8(b), while item embeddings with the same user label in Fig. 8(b) are more clustered than Fig. 8(a). Therefore, it is proved that identifying core item features is more conducive to learning accurate item representations. On the other hand, user embeddings with different item labels in Fig. 8(c) are more difficult to distinguish than Fig. 8(d), while user embeddings with the same item label in Fig. 8(d) are more clustered than Fig. 8(c). Therefore, identifying core user features is more conducive to learning accurate user representations.

7 Conclusion and discission

In this paper, we proposed the core features activated graph dual-attention network, namely C-GDN, for personalized recommendation. On one hand, C-GDN identified core features activating user-item interactions of users and items, which more accurately understands and explores user-item interactions at feature level. On the other hand, C-GDN designed a novel GDN to conduct information propagation in the bipartite graph, which considers not only the different contributions of neighbors but also the mutual association among neighbors in the bipartite graph learning. We conducted experiments on three benchmark datasets. Experimental results show that C-GDN outperforms mainstream personalized recommendation methods on multiple metrics.

C-GDN has certain theoretical and practical significance. In theory, C-GDN combines the advantages of feature-aware recommendation and GNN-based recommendation, which provides a novel and comprehensive view for personalized recommendation. In practice, C-GDN more effectively describes profiles of users and items by accurately identifying core

user and item features. The profiles of users and items are not only beneficial for users to more efficiently obtain the interested information, but also beneficial for service platforms to accurately deliver information. Moreover, C-GDN can be extended to other domains. For example, in the point of interest (POI) recommendation, C-GDN can effectively realize the personalized POI recommendation by inputting the check-in data, user features (e.g., Age, Occupation, etc.) and POI features (e.g., Business hours, Longitude and latitude, etc.).

Certainly, C-GDN can be further investigated from the following aspects. For an example, our research suppose that user preference is stable. However, user preference will change over time in the multiple specific scenarios. Therefore, our research is difficult to achieve dynamic recommendation based on users' dynamic preferences. In the future work, we will try to develop C-GDN for dynamic recommendation.

Acknowledgements This work was supported by the National Natural Science Foundation of China (nos. 72271024, 71871019).

Author Contributions Xiongtao Zhang: Conceptualization, Methodology, Data Curation, Software, Validation, Writing - Original Draft, Writing - review & editing. Mingxin Gan: Conceptualization, Writing - review & editing, Supervision, Funding acquisition.

Funding This work was supported by the National Natural Science Foundation of China (Nos. 72271024, 71871019).

Availability of supporting data Data will be made available on request.

Declarations

Ethics Approval Not Applicable.

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Berg, R. v. d., Kipf, T. N., & Welling, M. (2017). Graph convolutional matrix completion. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1–9. <https://doi.org/10.48550/arXiv.1706.02263>
- Deng, X., Liao, G., & Zeng, Y. (2022). Group event recommendation based on a heterogeneous attribute graph considering long-and short-term preferences. *Journal of Intelligent Information Systems*, pp. 1–27. Springer. <https://doi.org/10.1007/s10844-022-00758-w>
- Forestiero, A. (2022). Heuristic recommendation technique in internet of things featuring swarm intelligence approach. *Expert Systems with Applications*, vol. 187, p. 115904. Elsevier. <https://doi.org/10.1016/j.eswa.2021.115904>
- Gan, M., & Kwon, O. C. (2022). A knowledge-enhanced contextual bandit approach for personalized recommendation in dynamic domains. *Knowledge-Based Systems*, vol. 251, p. 109158. Elsevier. <https://doi.org/10.1016/j.knosys.2022.109158>
- Gan, M., & Ma, Y. (2022). Deepinteract: Multi-view features interactive learning for sequential recommendation. *Expert Systems with Applications*, vol. 204, p. 117305. Elsevier. <https://doi.org/10.1016/j.eswa.2022.117305>
- Gan, M., & Zhang, H. (2023). Viga: A variational graph autoencoder model to infer user interest representations for recommendation. *Information Sciences*, vol. 640, p. 119039. Elsevier. <https://doi.org/10.1016/j.ins.2023.119039>
- Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *Acm Transactions on Interactive Intelligent Systems*, vol. 5, pp. 1–19. Acm New York, NY, USA. <https://doi.org/10.1145/2827872>

- He, X., Deng, K., Wang, X., & et al. (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, pp. 639–648. <https://doi.org/10.1145/3397271.3401063>
- He, X., Liao, L., Zhang, H., & et al. (2017). Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182. <https://doi.org/10.1145/3038912.3052569>
- He, Y., Mao, Y., Xie, X., & et al. (2022). An improved recommendation based on graph convolutional network. *Journal of Intelligent Information Systems*, vol. 59, pp. 801–823. Springer. <https://doi.org/10.1007/s10844-022-00727-3>
- Hu, L., Li, C., Shi, C., & et al. (2020). Graph neural news recommendation with long-term and short-term interest modeling. *Information Processing & Management*, vol. 57, p. 102142. Elsevier. <https://doi.org/10.1016/j.ipm.2019.102142>
- Kang, W. C., Cheng, D. Z., Yao, T., & et al. (2021). Learning to embed categorical features without embedding tables for recommendation. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 840–850. <https://doi.org/10.1145/3447548.3467304>
- Li, Z., Cui, Z., Wu, S., & et al. (2019). Fi-gnn: Modeling feature interactions via graph neural networks for ctr prediction. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 539–548. <https://doi.org/10.1145/3357384.3357951>
- Liu, H., Zheng, C., Li, D., & et al. (2022). Multi-perspective social recommendation method with graph representation learning. *Neurocomputing*, vol. 468, pp. 469–481. Elsevier. <https://doi.org/10.1016/j.neucom.2021.10.050>
- Song, Y., Ye, H., Li, M., & et al. (2022). Deep multi-graph neural networks with attention fusion for recommendation. *Expert Systems with Applications*, vol. 191, p. 116240. Elsevier. <https://doi.org/10.1016/j.eswa.2021.116240>
- Su, Y., Zhang, R. M., Erfani, S., & et al. (2021). Neural graph matching based collaborative filtering. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 849–858. <https://doi.org/10.1145/3404835.3462833>
- Tao, Z., Wei, Y., Wang, X., & et al. (2020) Mgat: Multimodal graph attention network for recommendation. *Information Processing & Management*, vol. 57, p. 102277. Elsevier. <https://doi.org/10.1016/j.ipm.2020.102277>
- Wang, X., He, X., Wang, M., & et al. (2019) Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 165–174. <https://doi.org/10.1145/3331184.3331267>
- Wu, L., He, X., Wang, X., & et al. (2022a). A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, pp. 4425–4445. IEEE. <https://doi.org/10.1109/TKDE.2022.3145690>
- Wu, X., He, H., Yang, H., & et al. (2023). Pda-gnn: propagation-depth-aware graph neural networks for recommendation. *World Wide Web*, pp.1–22. Springer. <https://doi.org/10.1007/s11280-023-01200-z>
- Wu, S., Sun, F., Zhang, W., & et al. (2022b). Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, vol. 55, pp. 1–37. ACM New York, NY. <https://doi.org/10.1145/3535101>
- Zhang, C., Xue, S., Li, J., & et al. (2023). Multi-aspect enhanced graph neural networks for recommendation. *Neural Networks*, vol. 157, pp. 90–102. Elsevier. <https://doi.org/10.1016/j.neunet.2022.10.001>
- Zhang, T., Zhao, P., Liu, Y., & et al. (2019). Feature-level deeper self-attention network for sequential recommendation. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 4320–4326. <https://doi.org/10.5555/3367471.3367642>
- Zhou, G., Zhu, X., Song, C., & et al. (2018). Deep interest network for click-through rate prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1059–1068. <https://doi.org/10.1145/3219819.3219823>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.