CatGCN: Graph Convolutional Networks with Categorical Node Features

Weijian Chen, Fuli Feng, Qifan Wang, Xiangnan He, Chonggang Song, Guohui Ling, Yongdong Zhang

Abstract—Recent studies on Graph Convolutional Networks (GCNs) reveal that the initial node representations (i.e., the node representations before the first-time graph convolution) largely affect the final model performance. However, when learning the initial representation for a node, most existing work linearly combines the embeddings of node features, without considering the interactions among the features (or feature embeddings). We argue that when the node features are categorical, e.g., in many real-world applications like user profiling and recommender system, feature interactions usually carry important signals for predictive analytics. Ignoring them will result in suboptimal initial node representation and thus weaken the effectiveness of the follow-up graph convolution. In this paper, we propose a new GCN model named CatGCN, which is tailored for graph learning on categorical node features. Specifically, we integrate two ways of explicit interaction modeling into the learning of initial node representation, i.e., local interaction modeling on each pair of node features and global interaction modeling on an artificial feature graph. We then refine the enhanced initial node representations with the neighborhood aggregation-based graph convolution. We train CatGCN in an end-to-end fashion and demonstrate it on the task of node classification. Extensive experiments on three tasks of user profiling (the prediction of user age, city, and purchase level) from Tencent and Alibaba datasets validate the effectiveness of CatGCN, especially the positive effect of performing feature interaction modeling before graph convolution.

Index Terms—Representation Learning, Graph Neural Networks, Node Classification, User Profiling.

1 Introduction

GCNs have become a promising technique in various applications [1], such as recommender system [2], [3], [4], user profiling [5], [6] and text mining [7]. The main idea of graph convolution is to relate the representations of nodes based on the graph structure s.t. connected nodes should have similar representations, which can be seen as enforcing the smoothness constraint in the representation space. For example, the standard GCN [8] performs layer-wise representation relating as:

$$\mathbf{H}^{(l+1)} = \sigma(\widetilde{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \tag{1}$$

where $\mathbf{H}^{(l)}$ is the node representation matrix of the l-th layer, $\widetilde{\mathbf{A}}$ is the normalized graph adjacency matrix, and $\mathbf{W}^{(l)}$ is the weight matrix of the l-th layer (i.e., trainable model parameters of GCN). The $\mathbf{H}^{(0)}$ matrix stores the input features of nodes, e.g., the frequency of words of a document node [8]. We term $\mathbf{H}^{(0)}\mathbf{W}^{(0)}$ as the *initial node representation*, which performs linear transformation on the input features of each node and obtains a representation for the follow-up graph convolution operation.

Assuming the input node features are categorical, the feature matrix $\mathbf{H}^{(0)}$ is then high-dimensional yet sparse, in which each non-zero entry denotes the categorical feature value of a node. We can then understand the initial representation of a node (i.e., a row vector of $\mathbf{H}^{(0)}\mathbf{W}^{(0)}$) as linearly combining the embedding vectors of the node's categorical features (i.e., the row vectors of $\mathbf{W}^{(0)}$). With such a linear combination, the interactions among feature embeddings are not considered. Although the weight matrices of the following layers (e.g., $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$) may capture some interactions, the process is rather implicit and ineffective for learning cross feature effects [9], [10].

The tying of feature transformation and neighborhood aggregation in each graph convolution layer also limits the representation quality. Decoupled GCN such as APPNP first uses a conventional neural network on node features to obtain a representation vector (the same size as the label space) for each node; it then performs *pure neighborhood aggregation* — with no weight matrices and other trainable parameters — to refine the representation vector for prediction. Their strong performance inspires us to believe that the better the initial node representation is, the more benefits the follow-up graph convolution (or equivalently, neighborhood aggregation) can achieve. This is because that, the benefits brought by neighborhood aggregation and feature transformation are orthogonal — one exploits the signal from a node's neighbors whereas the other mostly depends on the features of a node itself. As such, if better (e.g., more discriminative) representation for a node can be obtained by leveraging its input features, the performance

- *Weijian Chen, Xiangnan He and Yongdong Zhang are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. E-mail: naure@mail.ustc.edu.cn, xiangnanhe@gmail.com, zhyd73@ustc.edu.cn.*
- *Fuli Feng is with the School of Computing, National University of Singapore, Computing 1, Computing Drive, 117417, Singapore. E-mail: fulifeng93@gmail.com.*
- *Qifan Wang is with Google Research. E-mail: wqfcr@google.com.*
- *Chonggang Song, Guohui Ling are with Tencent WeChat. E-mail: jerrycgsong@tencent.com, randyling@tencent.com.*
- *Fuli Feng and Yongdong Zhang are corresponding authors.*

after neighborhood aggregation should be better.

Although much effort has been devoted to inventing new GCN models, they mostly focus on graph convolution operations [11], [12]. To our knowledge, seldom research has considered improving the ability of GCN from the perspective of initial node representation, especially for categorical node features. In fact, many real-world applications have categorical features as raw data more commonly than continuous features, which are mostly restricted to multimedia content like images and videos. For example, in recommender systems, nodes are users and items that are normally described by user demographics (age, gender, interest tags) and item profiles (category, brand, etc.) [13]; in search engines, nodes are queries and documents that are described by bag-of-words or n-grams; on rating sites, content nodes such as music and videos are described with tags. For such categorical features, the interactions among features — e.g., the co-occurrence of multiple features — could contain important signal on the node's properties [9], [10], [14]. However, most GCNs apply a simple sum of feature embeddings as the initial node representation, which we believe is insufficient to model feature interactions and results in suboptimal node representation.

In this work, we explore how improved representation learning from categorical node features benefits GCN. We propose a new model named CatGCN, which integrates two kinds of explicit feature interactions into initial node representation learning: 1) local multiplication-based interaction on each pair of node features, and 2) global addition-based interaction on an artificial feature graph. We prove that in the artificial feature graph, performing one graph convolution layer with tunable self-connections can capture the interactions among all features. We then feed the enhanced initial node representations into a simplified/light GCN [3], [12] that performs neighborhood aggregation only to exploit the graph structure for node representation learning. The CatGCN is end-to-end trainable, such that all parameters in the initial node representation learning, and follow-up graph convolution and prediction layers can be optimized towards the final task.

The main contributions of the paper are summarized as follows:

- We emphasize the importance of tailoring GCNs for categorical node features, especially by modeling the interactions among features before graph convolution.
- We propose CatGCN, which performs two kinds of feature interaction modeling to enhance the initial node representations.
- We conduct experiments on user profiling tasks on large-scale datasets, verifying the positive effect of performing feature interaction modeling before graph convolution.

2 METHODOLOGY

We describe our method under the setting of node classification [8], whereas the idea is generally applicable to GCNs for other tasks like link prediction [3], [15] and community detection [16]. The graph structure is represented as an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ where N is the number of nodes. The main consideration of our work is that, each node u in the graph is described by **categorical features** $\mathbf{x}_u \in \mathbb{R}^d$ (d is the number of total features) where an entry $x_i^u = 0$ means the i-th feature value does not exist in the node (e.g., a female user cannot have "male" in her feature values). For a categorical feature vector $\mathbf{x}_u$, we denote the set of nonzero features as $\mathbb{S}_u = \{i|x_i^u \neq 0\}$. We summarize the symbols used in the paper in Table 1.

TABLE 1
Terms and notations used in the paper.

Symbol	Definition
$\mathbf{A} \in \mathbb{R}^{N\times N}$	the adjacency matrix of a graph
$\mathbf{D} \in \mathbb{R}^{N\times N}$	the degree matrix of a graph
$\mathbf{I} \in \mathbb{R}^{N\times N}$	the identity matrix of a graph
$\mathbf{H} \in \mathbb{R}^{N\times C}$	the initial node representations
$\mathbf{Y} \in \mathbb{R}^{N\times C}$	the final node representations
$\mathbf{x}_u \in \mathbb{R}^d$	the categorical features of node u
$\mathbb{S}_u = \{i\|x_i^u \neq 0\}$	the set of nonzero features of node u
$\mathbf{E} \in \mathbb{R}^{\|\mathbb{S}\|\times D}$	the embedding matrix of categorical features
$\mathbf{e} \in \mathbb{R}^D$	the embedding vector of one categorical feature
$\mathbf{P} \in \mathbb{R}^{\|\mathbb{S}\|\times\|\mathbb{S}\|}$	the adjacency matrix of one node's artificial feature graph
$\mathbf{Q} \in \mathbb{R}^{\|\mathbb{S}\|\times\|\mathbb{S}\|}$	the degree matrix of one node's artificial feature graph
$\mathbf{O} \in \mathbb{R}^{\|\mathbb{S}\|\times\|\mathbb{S}\|}$	the identity matrix of one node's artificial feature graph
$\mathbf{W}, \mathbf{W}_l, \mathbf{W}_g$	the weight matrices
$\mathbf{b}, \mathbf{b}_l, \mathbf{b}_g$	the bias vectors
$\mathbf{\Theta}$	all model parameters
$\odot$	element-wise product
σ	a non-linear activation function

2.1 Overall framework

The target of CatGCN is to improve initial node representations by explicitly incorporating the interactions of categorical features in a lightweight and efficient manner. As illustrated in Figure 1, CatGCN separately exploits the categorical features of a node itself and the signal from its neighbors. In particular, CatGCN first learns initial node representation $\mathbf{h}_u$ from its categorical features $\mathbf{x}_u$ with dedicated interaction modeling (detailed in Section 2.2). CatGCN then performs *pure neighborhood aggregation* (PNA) over the graph structure, which is formulated as:

$$\mathbf{Y} = PNA(\widetilde{\mathbf{A}}, \mathbf{H}, L), \quad \widetilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\hat{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}}, \quad \hat{\mathbf{A}} = \mathbf{A}+\mathbf{I}, \tag{2}$$

where $\mathbf{H}$ and $\mathbf{Y} \in \mathbb{R}^{N\times C}$ denote the initial node representations and final node representations with L-hop neighbors aggregated. Here, C is the number of prediction classes, $\hat{\mathbf{A}}$ is the adjacency matrix $\mathbf{A}$ with self-loops added (corresponding to the identity matrix $\mathbf{I}$). $\hat{\mathbf{A}}$ is normalized by node degrees which are organized into a diagonal degree matrix $\mathbf{D}$. Along the development of graph convolution operations, the L-hop neighborhood aggregation is either implemented in an iterative manner with L repeats of $\widetilde{\mathbf{A}}\mathbf{H}^{(l-1)}$ ($\mathbf{H}^0 = \mathbf{H}$) [8], or implemented in a simplified manner $\widetilde{\mathbf{A}}^L\mathbf{H}$ where $\widetilde{\mathbf{A}}^L$ is calculated as a pre-processing [12]. Following the principle of lightweight design, CatGCN adopts the simplified implementation to avoid the memory overhead of storing intermediate variables and the repeated computation during training.

Similar as standard GCNs, CatGCN is learned in an end-to-end manner by optimizing an objective function:

$$\mathcal{L} = \sum_{u\in\mathbb{U}} l(\mathbf{g}_u, \tilde{\mathbf{y}}_u) + \eta \parallel \mathbf{\Theta} \parallel_F^2, \quad \tilde{\mathbf{y}}_u = softmax(\mathbf{y}_u), \tag{3}$$

where $l(\cdot)$ is a classification loss such as cross-entropy [8] over the training set $\mathbb{U}$ of labeled nodes. The final node

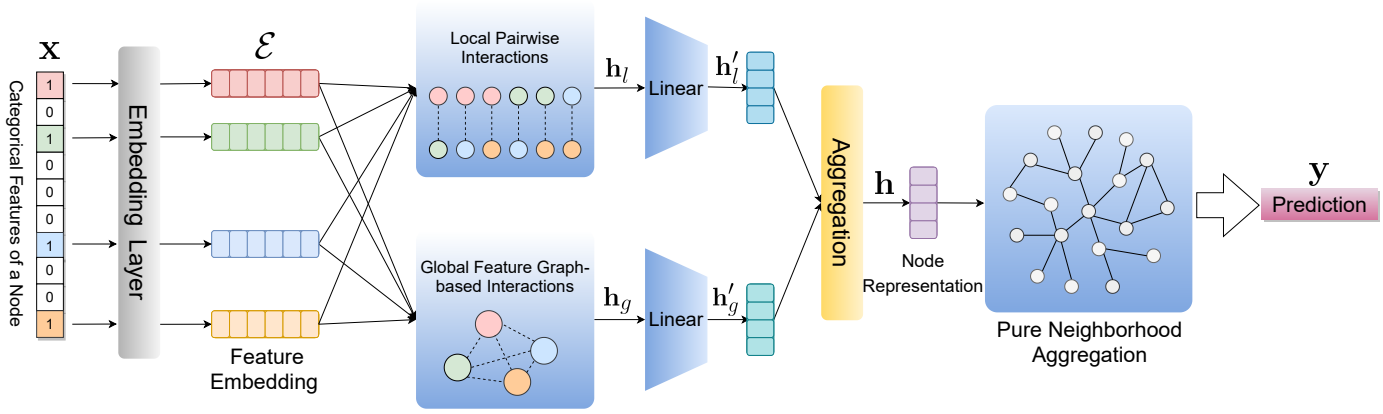


Fig. 1. The framework of CatGCN where an example node is taken to demonstrate the procedure of computing the initial node representation. Here, the feature number (d), embedding size (D), and prediction classes (C) are set as 10, 6, and 4, respectively.

representation $\mathbf{y}_u$ of node u is normalized to be a distribution over prediction labels $\tilde{\mathbf{y}}_u$. The one-hot vector $\mathbf{g}_u \in \mathbb{R}^C$ denotes the ground-truth of node u. $\mathbf{\Theta}$ represents all model parameters, and η is a hyper-parameter to balance the effect of loss and regularization. In the following, we introduce the learning of initial node representation $\mathbf{h}_u$ from its categorical features $\mathbf{x}_u$, the subscript u is thus omitted for the briefness of notations.

2.2 Interaction modeling of categorical features

Inspired by the effectiveness of explicit feature interaction modeling [9], [17], [18] in predictive analytics with categorical features, CatGCN focuses on improving the quality of initial node representations $\mathbf{h}$ via feature interaction modeling. To thoroughly capture feature interactions, our first belief is that separately modeling the feature interactions of different forms is essential since they convey different signals. Here, we consider the feature interactions of two forms: 1) local interaction between features, i.e., within a feature pair; and 2) global interaction amongst the whole feature set $\mathbb{S}$. Although much effort has been devoted to modeling local feature interactions [9], [17], [19], seldom research has considered the modeling of global interactions.

To bridge this gap, CatGCN integrates both local and global interactions into initial node representation learning. Specifically, as shown in Figure 1, CatGCN consists three main modules:

- **Feature embedding.** CatGCN first projects the categorical features into feature embeddings, *i.e.*, $\mathbf{x} \longrightarrow \mathcal{E} = \{\mathbf{e}_i | i \in \mathbb{S}\}$[1], so as to capture the relative relations among features in the embedding space. Note that $\mathbf{e}_i \in \mathbb{R}^D$ denotes the embedding of categorical feature i.
- **Interaction modeling.** Upon the feature embeddings, CatGCN explicitly models the local interaction and global interaction with multiplication-based operation and addition-based operation, respectively.

1. Note that each categorical feature in $\mathbb{S}$ is associated with the same node, and their weight can be regarded as 1, so the embedding weight in the following formulas is omitted. In cases where features come with non-binary weights, they can be used to multiply the corresponding embeddings as a preprocessing operation here.

- **Fusion.** Lastly, CatGCN is equipped with a fusion module to unify the benefits from both local and global interactions.

2.2.1 Local interaction modeling

For local feature interaction modeling, effective feature combinations can be mined to enrich input information. For example, people with pair-wise feature *gender_age*=$\{$*male, 20-25*$\}$ are more likely to be digital enthusiasts. This combination of features is more discriminating than either *gender*=$\{$*male*$\}$ or *age*=$\{$*20-25*$\}$ alone. The multiplication operation has been widely used to capture the correlation between entities in various tasks such as machine translation [20], recommendation system [17], and text classification [21]. In this way, local feature interactions are typically formulated as the element-wise product of feature embeddings. A representative operation is the bilinear interaction pooling [9],

$$\mathbf{h}_l = \sum_{i,j \in \mathbb{S} \ \& \ j>i} \mathbf{e}_i \odot \mathbf{e}_j = \frac{1}{2}\left[(\sum_{i \in \mathbb{S}} \mathbf{e}_i)^2 - \sum_{i \in \mathbb{S}} \mathbf{e}_i^2\right], \quad (4)$$

which aggregates the element-wise product on each pair of (different) feature embeddings. Here, $\mathbf{e}^2$ denotes $\mathbf{e} \odot \mathbf{e}$, and $\mathbf{h}_l$ denotes the initial node representation learned through local feature interaction modeling. Directly executing the operation has a quadratic time complexity w.r.t. the feature number (i.e., $O(|\mathbb{S}|^2)$), which can be reduced to linear complexity $O(|\mathbb{S}|)$ with an equivalent reformulation [9], [17]. This is an appealing property of bi-interaction pooling, which models pairwise interactions but with a linear complexity. After this operation, we can obtain more useful interactive features, specifically, $|\mathbb{S}|$ categorical features can be extended to $|\mathbb{S}|$ $(|\mathbb{S}|-1)/2$, which enriched the available information and was obviously of great value to the sparse features. Note that one can also perform high-order interaction modeling in a similar way [22], but the complexity increases polynomially and might be numerically unstable, so we do not further explore it here.

2.2.2 Global interaction modeling

Distinct from local interaction modeling, the purpose of global interaction modeling is to capture the node peculiarity information related to the predicted target. In real scenarios, the categorical features associated with a node are often

diverse, potentially reflecting the different peculiarities of the node. For instance, a user's purchase history includes laptops, cellphones, drones, running shoes, and sportswear, which indicates the peculiarity information of digital products and sports. Such peculiarities can be closely related to the prediction target of user interest, e.g., digital products indicates the user is a "digital enthusiast". Therefore, we need to filter out the latent peculiarities from the feature set $\mathbb{S}$ to facilitate the prediction. Inspired by the theory of spectral analysis [23], our belief is that the latent peculiarity lies in a certain frequency in the spectral domain. Our key consideration for the global interaction modeling is thus to uncover the signals along the spectrum and strengthen the signal closely associated with the prediction target at a particular frequency, to enhance the quality of the initial node representation. Considering the ability of GCN to filter frequency [12], we achieve the global interaction modeling with a carefully designed GCN.

We first use an artificial graph $(\mathbf{P}, \mathbf{E})$ to represent the nonzero features of the node $\mathbb{S}$ where $\mathbf{P} \in \mathbb{R}^{|\mathbb{S}|\times|\mathbb{S}|}$ is the adjacency matrix and $\mathbf{E} \in \mathbb{R}^{|\mathbb{S}|\times D}$ includes the embeddings of features in $\mathbb{S}$. $\mathbf{E}$ is initialized with Xaiver [24], which is learnable. As all features in $\mathbb{S}$ have inherent connections (e.g., co-occurrence), the artificial graph is thus a complete graph by natural, and the adjacency matrix $\mathbf{P}$ is an all-ones matrix (with self-loops). Aiming to capture the global interactions, graph convolution is conducted over the artificial graph. Formally,

$$\widetilde{\mathbf{P}} = \mathbf{Q}^{-\frac{1}{2}}(\mathbf{P} + \rho\mathbf{O})\mathbf{Q}^{-\frac{1}{2}} = \frac{\mathbf{P} + \rho\mathbf{O}}{|\mathbb{S}| + \rho},$$

$$\mathbf{h}_g = pool(\sigma(\widetilde{\mathbf{P}}\mathbf{E}\mathbf{W})). \tag{5}$$

$\widetilde{\mathbf{P}} \in \mathbb{R}^{|\mathbb{S}|\times|\mathbb{S}|}$ is the normalized adjacency matrix with probe coefficient ρ (see Section 2.2.2.2 for a detailed theoretical analysis), which adjusts the frequency to be strengthened. $\mathbf{Q} = (|\mathbb{S}| + \rho)\mathbf{O} \in \mathbb{R}^{|\mathbb{S}|\times|\mathbb{S}|}$ is the degree matrix of artificial graph where $\mathbf{O} \in \mathbb{R}^{|\mathbb{S}|\times|\mathbb{S}|}$ is an identity matrix. $\mathbf{h}_g$ denotes the initial node representation learned through global feature interaction modeling. $\mathbf{W}$ is the weight matrix; $\sigma(\cdot)$ is an activation function such as ReLU; $pool(\cdot)$ is a pooling function such as mean pooling to aggregate the global interactions across features. It should be noted that we model global interactions with only one graph convolution layer, which can largely reduce the memory and computation cost. This is because one graph convolution layer can achieve the equivalent effect of multiple layers.

Theorem. *On graph $\widetilde{\mathbf{P}}$, K-hop neighborhood aggregation equals to a 1-hop aggregation with smaller ρ. Formally,*

$$(\frac{\mathbf{P} + \rho_1\mathbf{O}}{|\mathbb{S}| + \rho_1})^K = \frac{\mathbf{P} + \rho_2\mathbf{O}}{|\mathbb{S}| + \rho_2}, \rho_1 \geq \rho_2 \geq 0, \tag{6}$$

$$\rho_2 = \frac{\rho_1^K}{\sum_{i=0}^{K-1} C_K^i \rho_1^i |\mathbb{S}|^{K-1-i}}. \tag{7}$$

2.2.2.1 Proof of the theorem: This theorem can be proved by using mathematical induction twice. We prove the upper half (i.e., Formula (6)) of this theorem, and the proof of the lower part (i.e., Formula (7)) is based on the first one.

1) Proof of Formula (6): The normalized adjacency matrix $\widetilde{\mathbf{P}}$ of the categorical feature artificial graph is a symmetric matrix, whose elements $(\{\widetilde{P}_{ij}|1 \leq i, j \leq |\mathbb{S}|\})$ have such forms:

$$\widetilde{P}_{ij} = \begin{cases} \frac{1+\rho_1}{|\mathbb{S}|+\rho_1}, & i = j, \\ \frac{1}{|\mathbb{S}|+\rho_1}, & i \neq j. \end{cases} \tag{8}$$

Now we prove that the K-power of this matrix satisfies Formula (6).

• $K = 2$. The quadratic power of $\widetilde{\mathbf{P}}$, i.e., $\widetilde{\mathbf{P}}^2$, has the entries of:

$$\widetilde{P}_{ij}^2 = \begin{cases} (\frac{1+\rho_1}{|\mathbb{S}|+\rho_1})^2 + \frac{|\mathbb{S}|-1}{(|\mathbb{S}|+\rho_1)^2} \\ = \frac{\rho_1^2 + 2\rho_1 + |\mathbb{S}|}{(|\mathbb{S}|+\rho_1)^2} = \frac{1+\rho_2}{|\mathbb{S}|+\rho_2}, & i = j, \\ 2(\frac{1+\rho_1}{|\mathbb{S}|+\rho_1})(\frac{1}{|\mathbb{S}|+\rho_1}) + \frac{|\mathbb{S}|-2}{(|\mathbb{S}|+\rho_1)^2} \\ = \frac{2\rho_1 + |\mathbb{S}|}{(|\mathbb{S}|+\rho_1)^2} = \frac{1}{|\mathbb{S}|+\rho_2}, & i \neq j, \end{cases} \tag{9}$$

where $\rho_2 = \frac{\rho_1^2}{|\mathbb{S}|+2\rho_1}$ and $\rho_2 \leq \rho_1$. That is to say, by setting the probe coefficient ρ with a small value ρ_2, performing 1-hop propagation is equivalent to a 2-hop propagation with probe coefficient of ρ_1.

• $K > 2$. We assume that the Formula (6) is correct for $K = k$, that is, we assume that $\widetilde{\mathbf{P}}^k$ has the diagonal elements $(1+\rho_k)/(|\mathbb{S}|+\rho_k)$ and the remaining values $1/(|\mathbb{S}|+\rho_k)$. Under this induction assumption, we must prove that the formula 6 is true for its successor, $K = k+1$. Based on $\widetilde{\mathbf{P}}^{k+1} = \widetilde{\mathbf{P}}^k\widetilde{\mathbf{P}}$ and the above induction assumption, the element $\widetilde{P}_{ij}^{k+1}$ of $(k+1)$-power of $\widetilde{\mathbf{P}}$ equals:

$$\begin{cases} (\frac{1+\rho_k}{|\mathbb{S}|+\rho_k})(\frac{1+\rho_1}{|\mathbb{S}|+\rho_1}) + \frac{|\mathbb{S}|-1}{(|\mathbb{S}|+\rho_k)(|\mathbb{S}|+\rho_1)} \\ = \frac{\rho_k\rho_1 + \rho_k + \rho_1 + |\mathbb{S}|}{(|\mathbb{S}|+\rho_k)(|\mathbb{S}|+\rho_1)}, & i = j, \\ (\frac{1+\rho_k}{|\mathbb{S}|+\rho_k})(\frac{1}{|\mathbb{S}|+\rho_1}) + (\frac{1}{|\mathbb{S}|+\rho_k})(\frac{1+\rho_1}{|\mathbb{S}|+\rho_1}) \\ + \frac{|\mathbb{S}|-2}{(|\mathbb{S}|+\rho_k)(|\mathbb{S}|+\rho_1)} \\ = \frac{\rho_k + \rho_1 + |\mathbb{S}|}{(|\mathbb{S}|+\rho_k)(|\mathbb{S}|+\rho_1)}, & i \neq j. \end{cases} \tag{10}$$

Now, we need to prove:

$$\widetilde{P}_{ij}^{k+1} = \begin{cases} \frac{1+\rho_{k+1}}{|\mathbb{S}|+\rho_{k+1}}, & i = j, \\ \frac{1}{|\mathbb{S}|+\rho_{k+1}}, & i \neq j. \end{cases} \tag{11}$$

Combine Formula (10) and Formula (11), we can calculate the probe coefficient ρ_{k+1} that satisfies the Formula (6), as follows:

$$\rho_{k+1} = \frac{\rho_k\rho_1}{|\mathbb{S}| + \rho_k + \rho_1} \leq \rho_k. \tag{12}$$

We have now fulfilled both conditions of the principle of mathematical induction. The Formula (6) is therefore true for every natural number K. In other words, performing an 1-hop propagation over the graph can achieve the same effect as performing a K-hop propagation.

2) Proof of Formula (7): We follow the same principle to prove the formula. First, according to the inferred value of ρ_2 above, it's easy to find that the Formula (7) is true when $K = 2$, as follows:

$$\rho_2 = \frac{\rho_1^2}{|\mathbb{S}| + 2\rho_1} = \frac{\rho_1^2}{\sum_{i=0}^{2-1} C_2^i \rho_1^i |\mathbb{S}|^{2-1-i}}.$$

Next, we assume that the Formula (7) is correct for $K = k$, formally:

$$\rho_k = \frac{\rho_1^k}{\sum_{i=0}^{k-1} C_k^i \rho_1^i |\mathbb{S}|^{k-1-i}}. \tag{13}$$

With this assumption, we must show that the rule is true for its successor, $K = k + 1$. Based on the conclusion of Formula (12) and Formula (13) above, we have

$$\begin{aligned}\rho_{k+1} &= \frac{\rho_k \rho_1}{|\mathbb{S}| + \rho_k + \rho_1} \\ &= \frac{\rho_1^{k+1}}{\sum_{i=0}^{k-1} C_k^i \rho_1^{i+1} |\mathbb{S}|^{k-1-i} + \sum_{i=0}^{k-1} C_k^i \rho_1^i |\mathbb{S}|^{k-i} + \rho_1^k}\end{aligned}$$

Now, we need to prove:

$$\rho_{k+1} = \frac{\rho_1^{k+1}}{\sum_{i=0}^{k} C_{k+1}^i \rho_1^i |\mathbb{S}|^{k-i}}$$

Based on the property of combination number, namely, $C_{k+1}^i = C_k^i + C_k^{i-1}$, we can derive the following:

$$\begin{aligned}\sum_{i=0}^{k} C_{k+1}^i \rho_1^i |\mathbb{S}|^{k-i} &= \sum_{i=0}^{k} (C_k^i + C_k^{i-1}) \rho_1^i |\mathbb{S}|^{k-i} \\ &= \sum_{i=0}^{k} C_k^i \rho_1^i |\mathbb{S}|^{k-i} + \sum_{t=1}^{k} C_k^{t-1} \rho_1^t |\mathbb{S}|^{k-t} \\ &= \sum_{i=0}^{k-1} C_k^i \rho_1^i |\mathbb{S}|^{k-i} + \rho_1^k + \sum_{i=0}^{k-1} C_k^i \rho_1^{i+1} |\mathbb{S}|^{k-i-1}\end{aligned}$$

Therefore, we can draw the conclusion that the Formula (7) is correct for $K = k+1$. The Formula (7) is therefore true for every natural number K.

2.2.2.2 Spectral analysis: In addition to heuristically understanding the global interactions as feature clusters in the embedding space, we present a more rigorous understanding from the spectral view. As to the artificial graph, the normalized graph Laplacian $\mathbf{L} = \mathbf{O} - \widetilde{\mathbf{P}} = \mathbf{O} - \mathbf{Q}^{-\frac{1}{2}}(\mathbf{P} + \rho\mathbf{O})\mathbf{Q}^{-\frac{1}{2}}$. $\mathbf{L}$ is a symmetric positive semi-definite matrix and can be decomposed into the form of $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{U} \in \mathbb{R}^{|\mathbb{S}| \times |\mathbb{S}|}$ is the matrix composed of orthogonal eigenvectors and $\mathbf{\Lambda} = diag(\lambda_i, 1 \le i \le |\mathbb{S}|) \in \mathbb{R}^{|\mathbb{S}| \times |\mathbb{S}|}$ is a diagonal matrix of its eigenvalues. The graph convolution is equal to:

$$\mathbf{g} * \mathbf{s} = \mathbf{U}((\mathbf{U}^\top \mathbf{g}) \odot (\mathbf{U}^\top \mathbf{s})) = \mathbf{U}\hat{\mathbf{G}}\mathbf{U}^\top \mathbf{s},$$

where $\mathbf{s} \in \mathbb{R}^{|\mathbb{S}|}$ denotes a signal to be transformed (each column of $\mathbf{E}$); $\mathbf{g}$ denotes a filter; and $\hat{\mathbf{G}} = diag(\hat{g}(\lambda_i), 1 \le i \le |\mathbb{S}|)$ represents the diagonal matrix consisting of the spectral filter coefficients $\hat{g}(\lambda_i)$. Functionally, the eigenvalues represent different frequencies; $\mathbf{U}^\top \mathbf{s}$ is the projection (decomposition) of signal $\mathbf{s}$ along the frequencies. Upon the decomposition, the graph convolution filters the signals according to the spectral filter coefficients.

As to our global interaction modeling, $\widetilde{\mathbf{P}}\mathbf{E} = (\mathbf{O} - \mathbf{L})\mathbf{E} = (\mathbf{O} - \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top)\mathbf{E} = \mathbf{U}(\mathbf{O} - \mathbf{\Lambda})\mathbf{U}^\top\mathbf{E}$. Thus, for a specific frequency λ_i, its spectral filter coefficient $\hat{g}(\lambda_i) = 1 - \lambda_i$. Note that the eigenvalues (filter frequencies) of $\mathbf{L}$ are $\lambda_1 = 0$ and $\lambda_2 = |\mathbb{S}|/(|\mathbb{S}| + \rho)$ ($|\mathbb{S}| - 1$ multiplicities), and the corresponding spectral filter coefficients are 1 and $\rho/(|\mathbb{S}| + \rho)$, respectively. Here, the filter frequency of $\lambda_1 = 0$ preserves the original input information, while $\lambda_2 = |\mathbb{S}|/(|\mathbb{S}| + \rho)$ is adjustable, which can filter out the global interaction signal[2]. That is, we can find the frequency λ_2 where the global interaction signal exists by adjusting probe coefficient ρ. Therefore, it is essential to set a proper probe coefficient ρ in the global interaction modeling (see results in Figure 5), which is a hyper-parameter of CatGCN.

2.2.3 Node representation fusion

Aiming to thoroughly exploit the benefit from both local feature interactions and global feature interactions, CatGCN fuses $\mathbf{h}_l$ and $\mathbf{h}_g$ into an overall node representation $\mathbf{h}$ through an aggregation layer. As aforementioned, $\mathbf{h} \in \mathbb{R}^C$ is the input of the pure neighborhood aggregation and required to be in the label space. As such, the aggregation layer is also responsible for projecting the representation into label space. Here we perform a late fusion strategy, which adds the two interacted representations after projecting them into the label space:

$$\begin{aligned}\mathbf{h}_l' &= \sigma(\mathbf{W}_l \mathbf{h}_l + \mathbf{b}_l), \\ \mathbf{h}_g' &= \sigma(\mathbf{W}_g \mathbf{h}_g + \mathbf{b}_g), \\ \mathbf{h} &= \alpha \mathbf{h}_g' + (1 - \alpha)\mathbf{h}_l',\end{aligned} \tag{14}$$

where $\alpha \in [0, 1]$ is a hyper-parameter to balance the influence of local and global interaction modeling, $\mathbf{W}_g$ and $\mathbf{W}_l$ are projection matrices. Note that we can take multiple fully connected layers here to enhance the expressiveness of the projection while ensuring the last one's output dimension is consistent with the predicted classes.

2.3 Discussion

Relation with Fi-GNN. Fi-GNN [25] is a click-through rate (CTR) prediction framework adopting GNN module, which also models the global addition-based interaction on an artificial feature graph. Fi-GNN adopts graph attention to model the structure of the feature graph, which dynamically calculates the strength of connections for each edge in the graph. However, as pointed out in [26], graph attention is not suitable for this situation which lacks supervised training on attention weights and is hard to find optimal initialization, leading to inferior performance. Moreover, Fi-GNN further stacks edge information transmission mechanism and recurrent embedding updating mechanism, which poses great challenge on model training, e.g., unaffordable computation and memory cost and severe overfitting. By contrast, CatGCN models global feature interactions in a very concise manner, which has the same complexity as a standard fully-connected layer (see Table 5 for an in-depth comparison).

2. In order to ensure the consistency of the global detection frequency, we need to fix the size of $|\mathbb{S}|$. This is also a reason for sampling a fixed number of features for each node.

TABLE 2
Statistics of the datasets.

Dataset	Attribute	Class	Node	Feature	Edge
Tencent	age	7	51,378	309	64,514
Alibaba	purchase	3	166,958	2,820	14,614,182
	city	4			

Relation with APPNP. To best of our knowledge, APPNP [11] is the first method that decouples the feature transformation and neighborhood aggregation in GCN layers. The target of APPNP is to alleviate the over-smoothing issue of deep GCN models which can lose focus at the upper layers. Instead of resolving over-smoothing, CatGCN focuses on enhancing the initial node representation which is a dual perspective. More specifically, CatGCN enhances the node representation through integrating two kinds of explicit interactions between categorical features, which has not been studied before. Further experimental results show that if APPNP unitizes our scheme to obtain the initial node representation, it will bring significant performance improvements (see Figure 3 for details).

3 EXPERIMENTS

In this section, we conduct extensive empirical studies to investigate the following research questions:

RQ1 How does our proposed feature interaction modeling strategies affecting the initial node representation?

RQ2 How does our CatGCN perform compared to the state-of-the-art methods?

RQ3 What are the factors that influence the effectiveness of CatGCN?

3.1 Experimental settings

3.1.1 Datasets

In order to investigate the actual performance of the model, we select three large-scale node classification datasets from real scenes: **Tencent-age**, **Alibaba-purchase**, and **Alibaba-city** [27]. **Tencent-age** is a social network graph with the target of predicting the user's age level. **Alibaba-purchase** and **Alibaba-city** [27] are also user profiling tasks on an e-commerce platform user graph, where the consumption level and city level are the prediction labels, respectively. These three datasets are collected from social platform Tencent and e-commerce platform Alibaba, and their construction process is as follows:

- **Tencent**[3]: This dataset is provided by the social networking platform Tencent Weibo, which includes users' preferences for a variety of items (*e.g.,* celebrities, organizations, and groups). We choose these items as the categorical features of user nodes. A preliminary cleanup of the data (e.g., filtering out users with inappropriate age settings) comes up with 1,238,563 users, which is termed as **Tencent-large**. In addition, considering that most GCNs are not designed for handling such large-scale graph, we select a sub-graph with 51,378 nodes of active users with at least 10 interaction on items. In our processing, if one user have followed the item i, we set $x_i = 1$, otherwise $x_i = 0$. In this way, we can obtain the multi-hot categorical features $\mathbf{x} \in \mathbb{R}^d$ of this user node, where $d = 309$ in this dataset. Although this dataset is provided for the recommendation task, it also provides information about the user age attribute, from which we selected over fifty thousands of users to perform the user profiling node classification task. Meanwhile, users of social platforms will interact with others in a series of ways, such as thumb up, comment, and forwarding, which leads to straightforward interconnections between users. In our experiment, we use the "follow" relationship to establish edges between user nodes. Note that the difference between the followed and following are ignored in our processing, that is, the edges we create are undirected.
- **Alibaba**[4]: This is a dataset of click-through rates for display ads on Alibaba's Taobao platform. In this scenario, we choose the categories of products as the categorical features affiliated to user nodes. Particularly, if user have clicked products belonging to the category i, we set $x_i = 1$, otherwise $x_i = 0$. Thus, we acquire the user categorical features $\mathbf{x} \in \mathbb{R}^d$ with dimensions $d = 2,820$ in Alibaba dataset. For our user profiling task, we screen two high-value user attributes, namely purchase, and city, corresponding to consumption level and city level where the user lives. Since there is no correlation like "follow" between users in the e-commerce platform, we establish the relationship between users based on co-click. In other words, if users jointly click the same product, we establish an edge between the two user nodes. Naturally, the edges between users established through this common behavior are undirected.

Statistics for above datasets are shown in Table 2. In addition, we construct a synthetic graph to investigate the effects of feature interaction modeling on the initial node representation. We first generate 1,000 nodes and randomly divide them into two classes with equal probability. We then sample edges between nodes of the same class from a Bernoulli distribution with a probability of 0.005, and decrease the probability to 0.001 for sampling edges across different classes. In this way, we acquire a synthetic graph with 1,000 nodes and 5,042 edges. Lastly, we generate a binary feature vector $\mathbf{x}$ with dimension of 100 for each node, where 10 entries are non-zero. In particular, we randomly assign non-zero entries within dimension 1-70 for node belongs to the first class, and dimension 31-100 for node in the second class. In this way, each class has 30 class-specific features and 40 intermingled features that are clues and barriers to distinguish the classes, respectively. Note that both the local interaction between a class-specific feature and an intermingle feature, and the global interaction can facilitate the classification. For each dataset, we randomly select 80%, 10%, and 10% of nodes to form the training, validation, and testing set, respectively.

3.1.2 Baseline models

We compare CatGCN with several recent GCN models, including the classical methods GCN [8], GAT [28], GraphSAGE [29] and the latest state-of-the-art models APPNP [11], SGC [12], CrossGCN [21] and GCNII [30].

3. https://www.kaggle.com/c/kddcup2012-track1

4. https://tianchi.aliyun.com/dataset/dataDetail?dataId=56

- **GCN** [8]: This is a semi-supervised classification model on graph-structured data, which can effectively aggregate information from the neighborhood by simultaneously encoding the graph structure and node features.
- **GAT** [28]: It adaptively allocates weight to neighborhood nodes through the masked self-attention layer, so as to distinguish the importance of different neighborhood nodes when aggregating neighborhood information.
- **GraphSAGE** [29]: This method implements representation learning on the large-scale graph by sampling local neighborhoods of nodes. In our experiment, we use the mean aggregator to complete the aggregation of neighborhood information.
- **APPNP** [11]: It connects GCN with Personalized PageRank [31] and expands the available neighborhood range without introducing additional parameters.
- **SGC** [12]: This method eliminates unnecessary nonlinearities and weight matrices in GCN and effectively reduces the complexity of the model. It not only improves performance but also significantly reduces computing costs.
- **CrossGCN** [21]: This method obtains cross features based on the traditional matrix factorization approach to enhance the feature learning ability. However, it can only model local feature interactions.
- **GCNII** [30]: It introduces initial residual connection and identity mapping to prevent the over-smoothing problem of GCN, which enables the model to be deeply stacked and brings performance gains.

For all aforementioned modes, we re-implements them using PyTorch Geometric [32], which have consistent or even better performance than the original paper. Our implementations are available at https://github.com/TachiChan/CatGCN.

3.1.3 Parameter settings

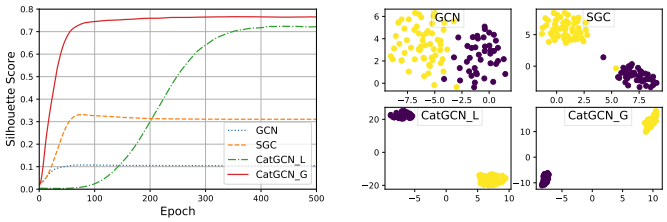
For all methods, the dimension of the categorical feature embedding layer and the size of all hidden layers are set to 64 for fair comparison. All trainable parameters are initialized with the Xavier method [24] and optimized with Adam [33]. We apply grid search strategy for hyper-parameters: the learning rate is tuned among $\{1e{-}1, 1e{-}2, 1e{-}3\}$, the L_2 regularization coefficient is searched in the range of $\{1e{-}5, 1e{-}4, ..., 1e{-}1, 0.0\}$, and dropout ratio is tuned in $\{0.0, 0.1, ..., 0.9\}$. For all baseline methods, their node representations are aggregated from the node's associated categorical features in the way of mean pooling, which is the normalized version of aforementioned $\mathbf{H}^{(0)}\mathbf{W}^{(0)}$. For CatGCN, we take ReLU as the activation function σ, and tune the aggregation parameter α within $\{0.0, 0.1, ..., 0.9, 1.0\}$. For each node, we sample a fixed number of categorical features from $\mathbb{S}$ to maintain a consistent global probe frequency λ_2, which is set to 10 in our experiment. The optimal hyper-parameters of CatGCN on different datasets are listed in Table 3. In all cases we adopt an early stopping strategy on the validation set with a patience of 10 epochs, and report the testing *Accuracy* and *Macro-F_1* [6], [34].

3.2 Effects on node representation quality (RQ1)

To investigate the influence of feature interaction modeling on the initial node representation, we test the following four models on the synthetic dataset: GCN [8], SGC [12], CatGCN_L and CatGCN_G. CatGCN_L and CatGCN_G are variants of CatGCN which calculate the initial node representation with local and global feature interaction modeling, respectively. As to GCN and SGC, they obtain the initial node representation from the normal feature transformation without explicit feature interaction modeling. Note that we set the same dimension for the initial node representation of the four models. In particular, all models are trained with 500 epochs where we extract the initial node representations of the testing nodes at each epoch.

TABLE 3
The optimal hyper-parameters of CatGCN on different datasets where lr and dr are short for learning rate and dropout rate, respectively.

Dataset	lr	L_2	dr	ρ	α	L
Tencent-age	0.1	$1e-4$	0.3	1	0.4	6
Alibaba-purchase	0.1	$1e-5$	0.3	39	0.9	8
Alibaba-city	0.1	$1e-5$	0.9	41	0.3	3



(a) Silhouette Score (b) t-SNE visualization

Fig. 2. (a) The Silhouette Score of initial node representation on the synthetic dataset during the training process. (b) The t-SNE visualization of the initial node representation at the end of training, i.e., Epoch 500.

Quantitatively, we evaluate the node representation quality through the Silhouette Score [35], which is defined over the set of testing nodes:

$$s = mean\left(\left\{\frac{b(u) - a(u)}{max(a(u), b(u))}\right\}\right), \tag{15}$$

where $a(u)$ is the mean intra-class distance of node u, i.e., the average distance between node u and the other nodes in the same class as u; and $b(u)$ is the mean inter-class distance of node u. Note that the value of Silhouette Score is in the range of [-1,1] where a larger value means the nodes in different classes are separated, i.e., the better node representation. Figure 2(a) shows the Silhouette Score of the four testing models along their training procedure. From the figure, we can see that both CatGCN_G and CatGCN_L can achieve higher Silhouette Score compared to GCN and SGC, which shows the benefit of feature interaction modeling. Moreover, we qualitatively evaluate the initial node representation by performing dimension reduction through t-SNE [36] and visualizing the nodes. Figure 2(b) illustrates the node representation of the testing models at epoch 500. From the figure, we can see that the initial node representation affiliated to different classes of CatGCN_L and CatGCN_G has a significantly higher distinction than GCN and SGC, which further reflects the strength of our proposed local and global feature interaction strategies.

3.3 Overall performance comparison (RQ2)

Table 4 shows the testing performance of all compared methods on the three datasets. From the table, we have the following observations:

TABLE 4
Node classification performance of all compared methods on the three real-world datasets.

Dataset	Tencent-age		Alibaba-purchase		Alibaba-city	
Method	Accuracy	Macro-F_1	Accuracy	Macro-F_1	Accuracy	Macro-F_1
GCN	0.2014(+24.6%)	0.1586(+20.2%)	0.4420(+25.9%)	0.3904(+14.9%)	0.2648(+30.6%)	0.2585(+9.2%)
GAT	0.2347(+ 6.9%)	0.1740(+ 9.5%)	0.4677(+19.0%)	0.4238(+ 5.8%)	0.3313(+ 4.4%)	0.2779(+1.5%)
GraphSAGE	0.2386(+ 5.2%)	0.1769(+ 7.7%)	0.4863(+14.4%)	0.4174(+ 7.4%)	0.2895(+19.4%)	0.2719(+3.8%)
APPNP	0.2472(+ 1.5%)	0.1822(+ 4.4%)	0.4860(+14.5%)	0.3939(+13.8%)	0.3066(+12.8%)	0.2692(+4.8%)
SGC	0.2411(+ 4.1%)	0.1777(+ 7.3%)	0.4832(+15.1%)	0.4167(+ 7.6%)	0.2880(+20.1%)	0.2717(+3.9%)
CrossGCN	0.2238(+12.1%)	0.1721(+10.7%)	0.3980(+39.8%)	0.3593(+24.8%)	0.3114(+11.0%)	0.2776(+1.7%)
GCNII	0.2310(+ 7.9%)	0.1777(+ 7.3%)	0.4275(+23.2%)	0.3778(+18.6%)	0.2925(+18.2%)	0.2669(+5.7%)
CatGCN(ours)	**0.2509**	**0.1906**	**0.5564**	**0.4484**	**0.3458**	**0.2822**

- In all cases, CatGCN outperforms all baselines with a significant gain of 12.41% on average, which is attributed to incorporating both the local and global feature interactions into the initial node representations. As such, this result validates the rationality of explicit interaction modeling of categorical features in GCN models.
- GAT performs better than the standard GCN, which shows the benefit of graph attention in these tasks. CatGCN may also achieve better performance if using attention in its PNA module, which is discarded purely for the consideration of computation cost.
- Many SOTA models fail to achieve ideal performance, and none of them can deliver consistently superior performance across all tasks. Note that in Alibaba-city task, GAT exceeds all the benchmark schemes, indicating that the adjacent nodes in the dataset may not meet the similarity, which is also a common phenomenon in real scenes. Existing models are designed from the perspective of neighborhood aggregation on the graph, so it is difficult to maintain stable performance in complex scenarios.
- Our feature interaction modeling is equivalent to adding two types of valuable input information to the initial node representation, one is the combination features, and the other is the global peculiarity information. This approach can increase the distinction of node representation and thus alleviate the interference caused by the noisy edge. The corresponding experimental results strongly support this claim (CatGCN consistently exceeds all baselines).

Large-scale graph. Recall that our target is to obtain a better initial node representation, and the feature interaction modeling strategies can also be applied to GCNs designed for handling large-scale graphs. In this light, we combine feature interaction modeling part into Cluster-GCN [37] and compare it with the vanilla Cluster-GCN on **Tencent-large**. After such operation, the *Accuracy* is improved from 0.2321 to 0.2419, and the *Macro-F_1* is improved from 0.176 to 0.177. This further verifies the applicability and effectiveness of our proposed framework in real massive data scenarios.

3.4 In-depth analysis (RQ3)

To further validate the rationality of our model design, we separately test the feature interaction modeling modules and the pure neighborhood aggregation module. To save space, we omit the results *w.r.t. Accuracy*, which have the similar trend as *Macro-F_1*.

3.4.1 Impacts of feature interaction modeling

We equip APPNP with the feature interaction modeling part of CatGCN (i.e., the local and global interaction modeling), which is named as Cat-APPNP. Figure 3 shows the performance of standard APPNP, Cat-APPNP, SGC, and CatGCN (i.e., Cat-SGC). Note that SGC is equivalent to CatGCN without feature interaction modeling. As can be seen, Cat-APPNP and Cat-SGC significantly outperform the corresponding APPNP and SGC, which further validates the effectiveness of enhancing initial node representation in GCN models and the advantages of modeling categorical features interactions. Moreover, the improvement of Cat-SGC over SGC is larger than that of Cat-APPNP over APPNP in all cases. We postulate the reason is that SGC and APPNP actually adopt different feature transformation strategies and the MLP adopted by APPNP can account for some feature interactions implicitly. Accordingly, applying the feature interaction modeling module can bring greater improvement to the initial node representation of SGC.

Furthermore, we develop two variants of CatGCN by removing the global and local interaction modeling mechanisms, which are named CatGCN_L and CatGCN_G, respectively. Figure 4 shows the performance of CatGCN_L, CatGCN_G, and CatGCN (see blue line), where the best result performance across all baselines is also depicted for better comparison (see grey line). It can be seen that removing any interaction modeling module from CatGCN will lead to performance degradation. At the same time, both variants outperform or rival all benchmark models. Therefore, both local and global interaction modeling mechanisms are effective for node representation learning, and their roles may be complementary. In addition, we can see that different variants competing on different tasks, which may be related to the different importance of local and global interaction information for different tasks. Note that if we look only at the feature interaction modeling part, our design can be understood as a plug-and-play framework that can be seamlessly integrated with the existing GNN models(e.g., Cat-APPNP).

3.4.2 Impacts of pure neighborhood aggregation

In order to analyze the role of neighborhood aggregation, we remove the PNA module of CatGCN and its two variants CatGCN_L and CatGCN_G. The corresponding result is shown in Figure 4. The dotted line represents the variation without using the pure neighborhood aggregation (i.e., without PNA). The comparison results show that neighborhood aggregation can effectively utilize the network structure to optimize the node representations even if there are no training parameters available, which illustrates that it can not be ignored in the graph convolution models.

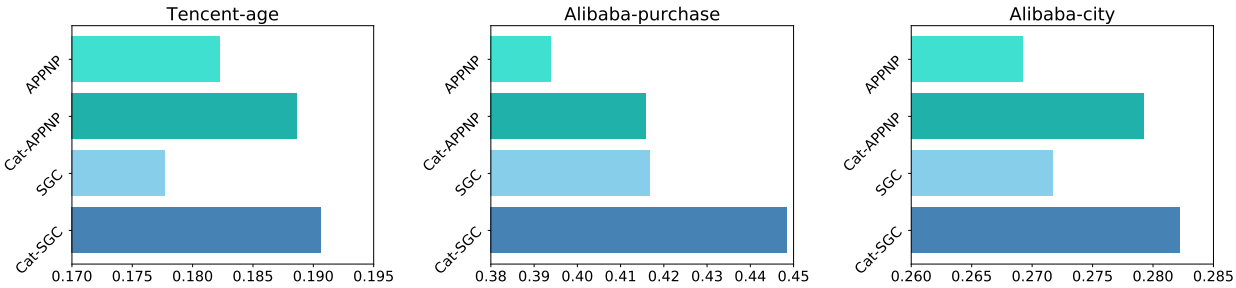


Fig. 3. Impacts of feature interaction modeling.

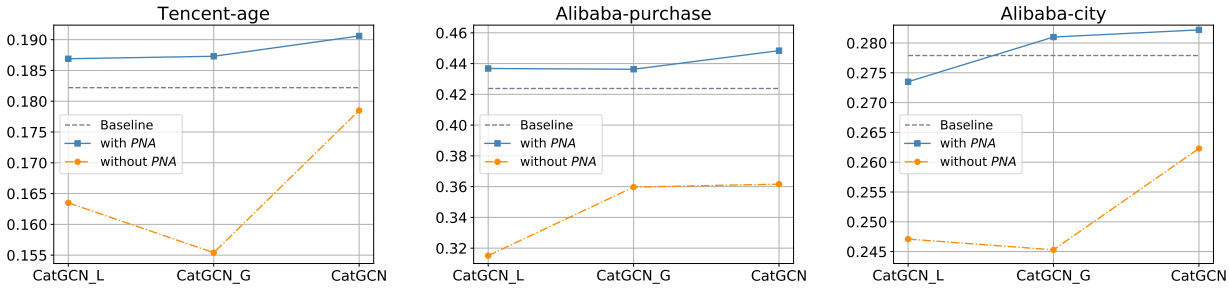


Fig. 4. Impacts of pure neighborhood aggregation.

3.4.3 Analysis on the global interaction modeling

To justify the advantages of the proposed global interaction modeling, we further study the impacts of probe coefficient (ρ) and perform in-depth comparison among CatGCN, Fi-GNN, and other global interaction modeling methods.

3.4.3.1 Impacts of probe coefficient (ρ): We first study how the probe coefficient influences the effectiveness of the proposed global feature interaction modeling. Figure 5 shows the performance of CatGCN_G as adjusting the probe coefficient ρ from 1 to 30. Note that we test CatGCN_G so as to avoid the interference of local interaction modeling. Moreover, CatGCN_G is set to L=1 without stacking multiple fully connected layers. From the figure, we can observe that: 1) the performance of CatGCN_G varies in a large range (0.246-0.268), which indicates the importance of integrating global peculiarity signal; 2) when ρ exceeds 21, the increase is relatively obvious, indicating that the frequency of global peculiarity signal λ_2 is around here under current settings; 3) the performance remains at a high level at $\rho \in [22, 30]$, possibly because the variation of λ_2 in this numerical interval is very limited ($\lambda_2 = |\mathbb{S}|/(|\mathbb{S}| + \rho)$).

3.4.3.2 Comparisons with Fi-GNN: To demonstrate the superiority of our design in modeling the global feature interactions, we further compare CatGCN with Fi-GNN w.r.t. performance, GPU memory usage and average time per epoch. For fair comparison, we adopt the same PNA module on the node representation outputted by the interaction modeling mechanism of Fi-GNN (named Fi-SGC). Due to the huge computational overhead of Fi-SGC's feature interaction modeling mechanism, it still cannot directly run on our dataset, which has a large number of categorical features. To tackle this issue, we let Fi-SGC share the field-specific weight matrix to reduce the memory requirements. Even so, the Fi-SGC can only be tested on Tencent dataset, while it will run out of memory on Alibaba dataset. The experiment results of Tencent dataset are shown in Table 5. From the table, we can find the hand-crafted design of Fi-GNN doesn't obtain higher performance. The complex design of the Fi-GNN not only consumes a lot of memory and increases computation time, but also results in performance degradation (compared to SGC). As a comparison, Cat-SGC (i.e., CatGCN) requires about half GPU memory usage and time cost, while significantly improving performance.

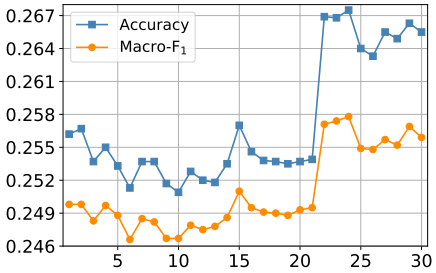


Fig. 5. Performance of CatGCN_G (L=1) as adjusting the probe coefficient ρ on Alibaba-city task.

Furthermore, we test two variants of CatGCN_G by replacing the global interaction modeling part with similarity matrix and graph attention mechanism [28], which are termed as SIMI-SGC and GAT-SGC, respectively. According to the experimental results in Table 5, SIMI-SGC, GAT-SGC and Fi-SGC have similar performance and are all inferior to SGC. Considering that they all use the relationship between feature embeddings to complete the optimization, such training strategy without supervision makes them hard to optimize and leads to poor performance [26]. This result thus indicates the advantage of performing global interaction modeling in the spectral domain.

4 Related Work

As this work explores the modeling of feature interactions in GCN, we review the recent researches on GCN and feature interaction modeling.

4.1 Graph convolutional networks

Graph convolutional networks have recently made remarkable achievements in a series of tasks such as node/graph classification [8], [30], [38], [39], [40], link prediction [15], [41], [42], [43], and community detection [16], [44]. Through coupling feature transformation and neighborhood aggregation, node features and graph structures are encoded simultaneously on each graph convolution layer, which ensures their ability to integrate information on the graph. To further improve the capability of graph convolutional networks, some strategies are proposed, such as introducing attention mechanisms to distinguish the node contribution [28], performing node sampling to increase the model scalability [29], [45], [46], and simplifying the model framework to reduce the computational cost [12]. Our work continues the idea of APPNP [11], which implies the separation of feature transformation and neighborhood aggregation is a better choice. We have made an in-depth exploration of the categorical node features, and the proposed framework CatGCN can well adapt to such graph data and obtain the most advanced performance.

4.2 Feature interaction modeling

Feature interactions are critical for revealing intrinsic peculiarity of the node that features affiliated, and they have been extensively explored, especially in real-world applications such as recommendation systems. The local feature interaction can help enrich valid feature information, and its effectiveness has been verified in several works [17], [19], [47]. On the other hand, plenty of researches [10], [48], [49], [50] have illustrated the importance of global feature interaction modeling. Further studies [51], [52] demonstrate that the combination of different levels of information can result in improved performance. Recently, with the rise of graph representation learning, the method of using graph neural network to model feature interactions has appeared, which achieves a good performance in the click-through rate prediction task [25]. In our work, we design two different mechanisms to learn the above different levels of information for the categorical node features. Specifically, for local interactions, we absorb the existing mature work bi-interaction pooling [9], while for global interactions, we design a specific graph convolutional network based on the nature of categorical feature interactions. The proposed model that combines these two mechanisms achieves optimal performance while remaining lightweight. Unlike the route described above, network representation methods with node attributes usually learn the node representation by jointly modeling the structure and attribute information of the network [53], [54], [55], [56], [57], [58]. Despite their encouraging success, little consideration has been given to using feature interaction modeling to optimize node representation. By contrast, CatGCN completes node representation optimization based on the node-specific feature set, which is orthogonal to the former design.

5 Conclusions

For the scenario of graph learning with categorical node features, we propose a novel GCN model named CatGCN.

TABLE 5
Comparison of different global interaction modeling strategies on Tencent-age task.

Methods	Accuracy	Macro-F_1	Memory usage	Time cost
SGC	0.2411	0.1777	989MB	0.03s
SIMI-SGC	0.2398	0.1764	1595MB	0.06s
GAT-SGC	0.2228	0.1737	3901MB	0.11s
Fi-SGC	0.2234	0.1719	4053MB	0.11s
Cat-SGC	0.2509	0.1906	2421MB	0.06s
CatGCN_G	0.2476	0.1873	2295MB	0.05s

*The testing platform is a Nvidia 2080Ti GPU with an Intel Core i9-9900X CPU (3.70GHz).

By designing local and global feature interaction modeling mechanisms explicitly, our proposed model can fully exploit the information of categorical features, and further integrate their advantages through differentiated aggregation, thus achieving significant improvement in multiple tasks on three large public datasets. Our proposed model has two cat-like strengths: lightweight (our design can achieve excellent performance with few parameters) and flexibility (the feature interaction modeling part can be seamlessly integrated with the existing GNN models to enhance their performance). Therefore, the proposed model has great potential in various real-world applications. In the future, we will incorporate more neighborhood aggregation techniques into CatGCN such as the graph attention and edge dropout [28]. At the same time, we will consider applying CatGCN to more practical applications, such as the recommender system [59], which might be an interesting direction.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2020AAA0106000) and the National Natural Science Foundation of China (62121002, U19A2079)

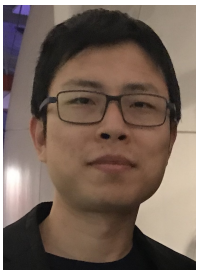
References

[1] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2020.
[2] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *KDD (Data Science track)*, 2018, pp. 974–983.
[3] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *SIGIR*, 2020.
[4] Y. Liu, C. Liang, X. He, J. Peng, Z. Zheng, and J. Tang, "Modelling high-order social relations for item recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2020.
[5] A. Rahimi, T. Cohn, and T. Baldwin, "Semi-supervised user geolocation via graph convolutional networks," in *ACL*, 2018, pp. 2009–2019.
[6] W. Chen, Y. Gu, Z. Ren, X. He, H. Xie, T. Guo, D. Yin, and Y. Zhang, "Semi-supervised user profiling with heterogeneous graph attention networks," in *IJCAI*, 2019, pp. 2116–2122.
[7] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *AAAI*, 2019, pp. 7370–7377.
[8] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
[9] X. He and T. Chua, "Neural factorization machines for sparse predictive analytics," in *SIGIR*, 2017, pp. 355–364.
[10] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *ADKDD*. ACM, 2017, pp. 12:1–12:7.
[11] J. Klicpera, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," in *ICLR*, 2019.

[12] F. Wu, A. H. S. Jr., T. Zhang, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *ICML*, 2019, pp. 6861–6871.
[13] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang, "A survey on neural recommendation: From collaborative filtering to content and context enriched recommendation," *CoRR*, 2021.
[14] H. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS@RecSys*, 2016, pp. 7–10.
[15] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *NeurIPS*, 2018, pp. 5171–5181.
[16] Z. Chen, L. Li, and J. Bruna, "Supervised community detection with line graph neural networks," in *ICLR*, 2019.
[17] S. Rendle, "Factorization machines," in *ICDM*, 2010, pp. 995–1000.
[18] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *KDD*, 2016, pp. 785–794.
[19] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *IJCAI*, 2017, pp. 3119–3125.
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.
[21] F. Feng, X. He, H. Zhang, and T. Chua, "Cross-gcn: Enhancing graph convolutional network with k-order feature interactions," *CoRR*, vol. abs/2003.02587, 2020.
[22] M. Blondel, A. Fujino, N. Ueda, and M. Ishihata, "Higher-order factorization machines," in *Advances in Neural Information Processing Systems*, 2016, pp. 3351–3359.
[23] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.
[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010, pp. 249–256.
[25] Z. Li, Z. Cui, S. Wu, X. Zhang, and L. Wang, "Fi-gnn: Modeling feature interactions via graph neural networks for CTR prediction," in *CIKM*. ACM, 2019, pp. 539–548.
[26] B. Knyazev, G. W. Taylor, and M. Amer, "Understanding attention and generalization in graph neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 4204–4214.
[27] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *KDD*, 2018, pp. 1059–1068.
[28] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.
[29] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NeurIPS*, 2017, pp. 1025–1035.
[30] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *ICML*, 2020, pp. 1725–1735.
[31] T. H. Haveliwala, "Topic-sensitive pagerank," in *WWW*, 2002, pp. 517–526.
[32] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.
[34] C. Wu, F. Wu, J. Liu, S. He, Y. Huang, and X. Xie, "Neural demographic prediction using search query," in *WSDM*, 2019, pp. 654–662.
[35] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
[36] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. 11, 2008.
[37] W. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C. Hsieh, "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks," in *KDD*, 2019, pp. 257–266.
[38] F. Feng, X. He, J. Tang, and T.-S. Chua, "Graph adversarial training: Dynamically regularizing based on graph structure," *IEEE Transactions on Knowledge and Data Engineering*, 2019.
[39] Y. Zhuang, Z. Liu, P. Qian, Q. Liu, X. Wang, and Q. He, "Smart contract vulnerability detection using graph neural network," in *IJCAI*, 2020, pp. 3283–3290.
[40] Z. Liu, P. Qian, X. Wang, Y. Zhuang, L. Qiu, and X. Wang, "Combining graph neural networks with expert knowledge for smart contract vulnerability detection," *IEEE Transactions on Knowledge and Data Engineering*, 2021.
[41] C. Huang, H. Xu, Y. Xu, P. Dai, L. Xia, M. Lu, L. Bo, H. Xing, X. Lai, and Y. Ye, "Knowledge-aware coupled graph neural network for social recommendation," in *AAAI*, 2021, pp. 4115–4122.
[42] L. Xia, Y. Xu, C. Huang, P. Dai, and L. Bo, "Graph meta network for multi-behavior recommendation," in *SIGIR*, 2021, pp. 757–766.
[43] X. Yang, X. Du, and M. Wang, "Learning to match on graph for fashion compatibility modeling," in *AAAI*, 2020, pp. 287–294.
[44] D. He, Y. Song, D. Jin, Z. Feng, B. Zhang, Z. Yu, and W. Zhang, "Community-centric graph convolutional network for unsupervised community detection," in *IJCAI*, 2020, pp. 3515–3521.
[45] J. Chen, T. Ma, and C. Xiao, "Fastgcn: Fast learning with graph convolutional networks via importance sampling," in *ICLR*, 2018.
[46] H. Gao, Z. Wang, and S. Ji, "Large-scale learnable graph convolutional networks," in *KDD*, 2018, pp. 1416–1424.
[47] Y. Juan, Y. Zhuang, W. Chin, and C. Lin, "Field-aware factorization machines for CTR prediction," in *RecSys*, 2016, pp. 43–50.
[48] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, "Product-based neural networks for user response prediction," in *ICDM*, 2016, pp. 1149–1154.
[49] X. Xin, B. Chen, X. He, D. Wang, Y. Ding, and J. Jose, "CFM: convolutional factorization machines for context-aware recommendation," in *IJCAI*, 2019, pp. 3926–3932.
[50] X. Yang, X. He, X. Wang, Y. Ma, F. Feng, M. Wang, and T.-S. Chua, "Interpretable fashion matching with rich attributes," in *SIGIR*, 2019, pp. 775–784.
[51] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: A factorization-machine based neural network for CTR prediction," in *IJCAI*, 2017, pp. 1725–1731.
[52] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "xdeepfm: Combining explicit and implicit feature interactions for recommender systems," in *KDD*, 2018, pp. 1754–1763.
[53] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *KDD*, 2019, pp. 793–803.
[54] Z. Zhang, H. Yang, J. Bu, S. Zhou, P. Yu, J. Zhang, M. Ester, and C. Wang, "ANRL: attributed network representation learning via deep neural networks," in *IJCAI*, 2018, pp. 3155–3161.
[55] H. Gao and H. Huang, "Deep attributed network embedding," in *IJCAI*, 2018, pp. 3364–3370.
[56] Z. Meng, S. Liang, H. Bao, and X. Zhang, "Co-embedding attributed networks," in *WSDM*, 2019, pp. 393–401.
[57] L. Liao, X. He, H. Zhang, and T. Chua, "Attributed social network embedding," *IEEE Transactions on Knowledge and Data Engineering*, pp. 2257–2270, 2018.
[58] J. Chen, M. Zhong, J. Li, D. Wang, T. Qian, and H. Tu, "Effective deep attributed network representation learning with topology adapted smoothing," *IEEE Transactions on Cybernetics*, 2021.
[59] X. Wang, X. He, M. Wang, F. Feng, and T. Chua, "Neural graph collaborative filtering," in *SIGIR*, 2019, pp. 165–174.

Weijian Chen is currently a Ph.D. student in the School of Information Science and Technology, University of Science and Technology of China (USTC). His research interests include user profiling, recommender system, graph neural networks, and knowledge graph.

Fuli Feng is a Research Fellow in the School of Computing, National University of Singapore (NUS). He received Ph.D. in Computer Science from NUS in 2019. His research interests include information retrieval, data mining, and multimedia processing. He has over 30 publications appeared in several top conferences such as SIGIR, WWW, and MM, and journals including TKDE and TOIS. His work on Bayesian Personalized Ranking has received the Best Poster Award of WWW 2018. Moreover, he has been served as the PC member for several top conferences including SIGIR, WWW, WSDM, NeurIPS, AAAI, ACL, MM, and invited reviewer for prestigious journals such as TOIS, TKDE, TNNLS, TPAMI, and TMM.



Qifan Wang is a research engineer in Google Research. He received the BS and MS degrees from Tsinghua University, and the PhD degree from Purdue University, all in computer science. His research interests include deep learning, natural language processing, information retrieval, data mining, and computer vision. He has over 40 publications in top-tier conferences and journals, including SIGIR, KDD, IJCAI, AAAI, NeurIPS, EMNLP, ECCV, CIKM, TPAMI and TOIS.

Xiangnan He is a professor at the University of Science and Technology of China (USTC). His research interests span information retrieval, data mining, and multi-media analytics. He has over 90 publications in top conferences such as SIGIR, WWW, and MM, KDD, and journals including TKDE, TOIS, and TMM. His work has received the Best Paper Award Honorable Mention in WWW 2018 and SIGIR 2016. He is in the Editorial Board of the AI Open journal, served as the PC chair of CCIS 2019, the area chair of MM 2019, ECML-PKDD 2020, and the (senior) PC member for top conferences including SIGIR, WWW, KDD, WSDM etc.

Chonggang Song is a senior researcher in Tencent WeChat. He received the Ph.D degree in Computer Science from the National Unicersity of Singapore in 2017. His research interests include data mining, graph analysis and recommender systems. His work focuses on applying cutting-edge techniques on practical applications in WeChat such as Story Recommendation and Video Channel Recommendation. His works has been published in top conferences such as ICDE and CIKM.

Guohui Ling is a senior researcher in Tencent WeChat. He received his BS in computer science from Sun Yat-sen University in 2007. His work focuses on mining billion-scale dynamic social network for real-world applications such as social advertising and video recommendation in WeChat. His works has been published in top conferences such as ICDE and DASFAA.

Yongdong Zhang (M'08–SM'13) received the Ph.D. degree in electronic engineering from Tianjin University, Tianjin, China, in 2002. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. He has authored more than 100 refereed journal and conference papers. His research interests include multimedia content analysis and understanding, multimedia content security, video encoding, and streaming media technology. Prof. Zhang was the recipient of the Best Paper Awards in PCM 2013, ICIMCS 2013, and ICME 2010, and the Best Paper Candidate in ICME 2011. He serves as an Editorial Board Member of the Multimedia Systems Journal and the IEEE TRANSACTIONS ON MULTIMEDIA.