Graph Reciprocal Neural Networks by Abstracting Node as Attribute

Liang Yang, Jiayi Wang
School of Artificial Intelligence
Hebei University of Technology
Tianjin, China
yangliang@vip.qq.com
jayeew@qq.com

Dongxiao He*
College of Intelligence and Computing
Tianjin University
Tianjin, China
hedongxiao@tju.edu.cn

Chuan Wang
State Key Laboratory of Information Security
Institute of Information Engineering, CAS
Beijing, China
wangchuan@iie.ac.cn

Xiaochun Cao
School of Cyber Science and Technology
Shenzhen Campus
Sun Yat-sen University
Shenzhen, China
caoxiaochun@mail.sysu.edu.cn

Bingxin Niu
School of Artificial Intelligence
Hebei University of Technology
Tianjin, China
niubingxin666@163.com

Zhen Wang
OPtics and ElectroNics (iOPEN)
School of Cybersecurity
Northwestern Polytechnical University
Xi'an, China
w-zhen@nwpu.edu.cn

***Abstract*—Graph neural network (GNN) can be formulated as the multiplication of the topology-related matrix (adjacency or Laplacian matrix) and node attribute matrix, i.e., operation in node-wise. Unfortunately, this unified formula reveals two inherent drawbacks. Firstly, the topology and node attribute are not reciprocal but biased. From employment, the topology information is repeatedly employed, while the node attribute is only used once. From parameterization perspective, the node attribute is parameterized with highly expressive MLPs, while topology is not. Secondly, the graph topology can not be fully explored. Only the local pairwise relation is explored, but the mesoscopic community structure, which is one of the most prominent characteristics of networks, is ignored. To alleviate these issues, this paper proposes the Graph Reciprocal Network (GRN) by treating node attribute and topology reciprocal. Firstly, it is illustrated that the node can be regarded and utilized as another kind of attribute. Secondly, a novel node representation scheme is proposed from the theory of Quadratic Networks, with a theoretical guarantee of the fine-grained element-wise product of the representations of the topology and attribute. Extensive experiments demonstrate the superior performance and robustness of the proposed GRN.**

***Index Terms*—graph neural networks, node attribute, community structure, quadratic networks, parameterization**

I. Introduction

Vanilla neural networks, which focus on regular data on grid, such as images, speech and video, possess the characteristic of universal approximation. Deep learning pursues semantic representations by stacking multiple neural networks and getting great performance breakthroughs [1]. Unfortunately, they often perform poorly on irregular data, such as manifold [2] and graph [3]. To overcome this issue, graph neural network (GNN) is present to tackle representation learning on graph by combing graph topology and node attribute [4], [5]. Most GNNs can from either spatial and spectral perspectives.

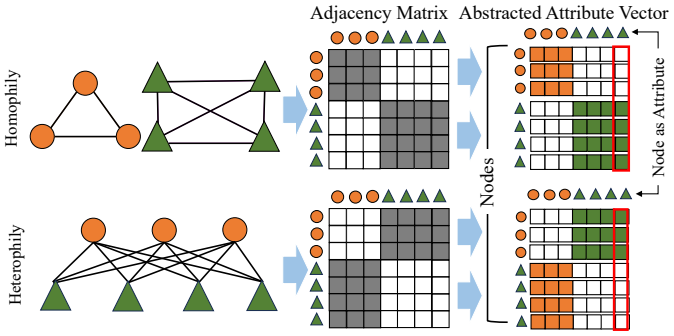

Fig. 1. Two intuitive examples of abstracting node as attribute on homophilic and heterophilic networks. Orange circles and green triangles stand for nodes from two classes. The gay boxes in adjacency matrices denote links between nodes, while green and orange boxes in abstracted attribute vectors represent attributes corresponding to nodes from different classes. As shown in red boxes, the attributes abstracted from topology possess high discriminability.

From spatial perspective, GNNs are designed following the principle of message passing [6]. Based on the homophily inductive bias, they often perform smoothing over the neighborhoods and can be formulated as the multiplication of the adjacency matrix and node attribute matrix. To alleviate the over-smoothing issue and extend to heterophilic networks, (initial) residual connection [7], [8] and equivalent multi-scale topology [9]–[11] with learnable weights are employed [12].

From spectral perspective [13], the node attributes are regarded as the signals on graph and processed in the spectral domain. The filter in graph spectral domain is the function of the eigenvalues of the Laplacian matrix [14] and can be approximated with polynomial basis functions, such as Cheby-

*Corresponding author.

shev polynomial [13] and Cayley polynomial [15]. Thus, the filter learning is transferred to the learning of polynomial coefficients. To avoid the expensive eigenvalue decomposition, these polynomials on the eigenvalues in spectral domain are equivalent to those on the Laplacian matrix in spatial domain. Therefore, they can be formulated as the multiplication of the polynomials of the Laplacian matrix and the node attribute matrix with learnable polynomial coefficients.

In summary, existing GNNs, no matter whether designed from spatial or spectral perspective, can be formulated as the multiplication of the topology-related matrix and node attribute matrix, which is the operation at the node-level. Unfortunately, this unified formula reveals that existing GNNs have two inherent drawbacks. Firstly, the topology and node attribute are not reciprocal but biased. On the one hand, the employment of two kinds of information is biased. The topology information is repeatedly employed, while the node attribute is only fed to the first layer. This causes the over-smoothing issue [16]. The initial residual connection, which is the widely-employed treatment to alleviate over-smoothing issue, is to correct this bias by feeding node attribute to all layers. On the other hand, the parameterization of two kinds of information is biased. The node attribute is parameterized with highly expressive MLPs, while topology is only parameterized with weighting coefficients for different scales. Although some attempts tend to refine the graph structure with an inference model, almost none of them employ expressive neural networks.

Secondly, the graph topology can not be fully explored. As revealed in the optimization perspective of GNNs (Section II-B), only the local pairwise relation is explored in existing GNNs. Besides, most parameterizations to refine graph structure also focus on the pairwise relation, such as the attention mechanism attention in GAT [17]. Unfortunately, the local pairwise relations are fragile to noise and hard to explore high-order and long-distance information. Actually, instead of the microscopic structure, the mesoscopic community structure, which is one of the most prominent characteristics of networks, is critical for network mining and embedding.

To alleviate these essential issues, this paper proposes the Graph Reciprocal Network (GRN) by treating node attribute and topology reciprocal from the perspectives of both employment and parameterization. Firstly, it is illustrated that the node, especially the hub node, can be regarded and utilized as another kind of attribute as shown in Fig. 1. This meets the cases in real world. For example, the famous people you follow on online social networks indicate your interest. This justifies the employment of MLP for representation learning on topology. Secondly, a novel node representation scheme is proposed from the theory of Quadratic Networks, which possess better universal approximation. This scheme can be interpreted as the fine-grained element-wise product of the representations of the topology and attribute. The main contributions of this paper are summarized as follows:

- We investigate the drawbacks of existing GNNs based on the multiplication of topology and attribute matrices, which perform a node-wise operation and cause over-smoothing.
- We propose a novel Graph Reciprocal Network (GRN) by abstracting nodes as attributes and assembling the network with the theory of Quadratic Networks.
- We conduct experiments to demonstrate the superior performance and robustness of the proposed GRN.

II. Preliminaries

A. Notations

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph with node set $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$ and edge set $\mathcal{E}$, where N is the number of nodes. The topology of graph $\mathcal{G}$ can be represented by its adjacency matrix $\mathbf{A} = [a_{ij}] \in \{0,1\}^{N\times N}$, where $a_{ij} = 1$ if and only if there exists an edge $e_{ij} = (v_i, v_j)$ between nodes v_i and v_j. The degree matrix $\mathbf{D}$ is a diagonal matrix with diagonal element $d_i = \sum_{i=1}^{N} a_{ij}$ as the degree of node v_i. $\mathcal{N}(v_i) = \{v_j | (v_i, v_j) \in \mathcal{E}\}$ stands for the neighbourhoods of node v_i. $\mathbf{X} \in \mathbf{R}^{N\times F}$ and $\mathbf{H} \in \mathbf{R}^{N\times F'}$ denote the collections of node attributes and representations with the i^{th} rows, i.e., $\mathbf{x}_i \in \mathbb{R}^F$ and $\mathbf{h}_i \in \mathbb{R}^{F'}$, corresponding to node v_i, where F and F' stand for the dimensions of attribute and representation.

B. Optimization Perspective of GNNs

Recent attempts tend to reveal the essence of graph convolutional networks. Among them, some works interpret and unify graph convolutional networks from the perspective of numeral optimization [18]–[20]. Specifically, they show that the graph convolution with residual connection is to minimize the following objective function via gradient descent

$$\begin{aligned} \mathcal{C} &= ||\mathbf{X}\mathbf{W} - \mathbf{H}||_F^2 + \lambda \mathrm{tr}\left(\mathbf{H}^T \tilde{\mathbf{L}} \mathbf{H}\right) \\ &= \sum_{i=1}^{N} ||\mathbf{x}_i \mathbf{W} - \mathbf{h}_i||_2^2 + \frac{\lambda}{2} \sum_{i=1}^{N} \sum_{j\in N(i)} \tilde{a}_{ij} ||\mathbf{h}_i - \mathbf{h}_j||_2^2, \end{aligned} \tag{1}$$

The first term stands for the distance between the node embeddings $\mathbf{h}_i$'s and attributes $\mathbf{x}_i$'s, while the second term which focuses on local pairwise relation stands for the distance between the representations of two connected nodes $\mathbf{h}_i$ and $\mathbf{h}_j$. Hence GNNs can be divided into two categories to model this pairwise relation. One treats topology information as noise free and smooth two nodes if they are connected, i.e., GCN [21] and SGC [22]. The other refine the graph structure to weaken the harmful noise. For example, GAT [17] models pairwise relation via normalized nodes feature similarity $\tilde{e_{ij}} = softmax(e_{ij}), e_{ij} = LeakyReLU(\mathbf{b}^t[\mathbf{W}\mathbf{h}_i^{(t-1)} || \mathbf{W}\mathbf{h}_j^{(t-1)}])$, where $\mathbf{b}^t$ is learnable parameters. However, these trapped in local pairwise relations methods are fragile to noise, i.e., topology and feature noises, and make the high-order topology information underutilized.

III. Methodology

A. Node as Attribute

Representation learning, e.g., deep learning, tends to embed i.i.d data, such as 2D images and 3D videos, in vector form

TABLE I
STATISTICS OF SMALL-SCALE NETWORKS. HR MEANS HOMPHILY RATE.

Dataset	Nodes	Edges	Features	Classes	HR
Cora	2,708	5,429	1,433	7	0.83
Citeseer	3,327	4,732	3,703	6	0.71
Pubmed	19,717	44,338	500	3	0.79
Chameleon	2,277	36,101	2,325	5	0.25
Squirrel	5,201	217,073	2,089	5	0.22
Actor	7,600	33,544	931	5	0.24
Texas	183	309	1,703	5	0.06
Cornell	183	295	1,703	5	0.11

TABLE II
STATISTICS OF LARGE-SCALE NETWORKS. HR MEANS HOMPHILY RATE.

Dataset	Nodes	Edges	Features	Classes	HR
Penn94	41,554	1,362,229	5	2	0.47
pokec	1,632,803	30,622,564	65	2	0.44
arXiv-year	169,343	1,166,243	128	5	0.22
snap-patents	2,923,922	13,975,788	269	5	0.07
genius	421,961	984,976	12	2	0.61
twitch-gammers	168,114	6,797,557	7	2	0.54

based on the fact that semantic information contained in the data itself is represented. On the contrary, the topological semantic information does NOT contain in node itself in the graph but is determined by other nodes it links to. Therefore, the algorithms for network embedding, such as DeepWalk [23] and node2vec [24], which tend to seek node representation, are very different from those for i.i.d. data. It causes two challenges. Firstly, it is difficult to employ existing neural networks for semi-supervised network embedding. Secondly, the topology component in GNNs can not be parameterized via existing neural networks.

However, nodes, especially hub ones, can actually be treated as another kind of attribute, which possesses high discriminability. Firstly, it intuitively meets the cases in the real world. In the online social network, the famous people you follow, e.g., Lionel Messi, indicate your interest, e.g., football. In the citation network, the seminal papers cited by a paper, e.g., the paper on DeepWalk or node2vec, shows its category, i.e., data mining/network embedding. Secondly, two representative examples of homophilic and heterophilic networks are provided in Fig. 1. A novel kind of attribute is abstracted from the topology (adjacency matrix). As shown in the red boxes, the abstracted attribute possesses high discriminability no matter whether the graphs are homophilic or heterophilic. Note that this discriminability is independent of the order of the nodes.

As long as nodes can be employed as another kind of attribute, existing neural networks can be applied to explore and model the topology. Similar to original attribute $\mathbf{H}_X = \mathbf{X}\mathbf{W}_X$ where $\mathbf{W}_X \in \mathbf{R}^{F\times F'}$ denotes the neural network for original attribute, MLP is employed to attributes abstracted from topology as $\mathbf{H}_A = \mathbf{A}\mathbf{W}_A$, where $\mathbf{W}_A \in \mathbf{R}^{N\times F'}$ denotes the neural network for abstracted attributes. Therefore, the node representation in one layer can be formulated as follows:

$$\mathbf{H} = [\mathbf{A}||\mathbf{X}]\left[\frac{\mathbf{W}_A}{\mathbf{W}_X}\right] = \mathbf{H}_A + \mathbf{H}_X = \mathbf{A}\mathbf{W}_A + \mathbf{X}\mathbf{W}_X. \quad (2)$$

Note that the final node representation is the summarization of those from topology and node attributes. Thus the topology and node attributes in Eq. (2) are reciprocal without bias from the perspectives of employment and parameterization. Thus, Eq. (2) can be regarded as the graph reciprocal operation.

B. Graph Reciprocal Networks

Previous section presents the graph reciprocal operation, i.e., one layer graph reciprocal network. This section tends to assemble multiple graph reciprocal operations to form the Graph Reciprocal Networks (GRN). Unfortunately, the assembly is not trivial to simultaneously keep the reciprocal characteristic and high expressive ability. Firstly, although it is natural to respectively embed topology and node attributes with two deep networks, the expressive ability is low since topology and node attributes are not deeply integrated. Secondly, the reciprocal characteristic is broken if GRN is constructed following existing GNNs, i.e., treating the node representation from previous layers as the node attribute in the next layer.

To simultaneously keep the reciprocal characteristic and high expressive ability, Quadratic Network is employed to construct the deep Graph Reciprocal Networks. Recent attempts demonstrate that Quadratic Networks possess better universal approximation compared to vanilla neural networks and have been successfully applied to many fields [25]–[27]. Different from vanilla neural network via $y = \sigma\left((\mathbf{w}^T\mathbf{x})\right)$, Quadratic Network embeds the vector-form input $\mathbf{x}$ as

$$y = \sigma\left((\mathbf{w}_1^T\mathbf{x}) \odot (\mathbf{w}_2^T\mathbf{x})\right),$$

where $\mathbf{w}_1$ and $\mathbf{w}_2$ are two different weight vectors to be learned, $\sigma()$ stand for the nonlinear mapping, and $\odot$ denotes the element-wise product. Following this scheme, the GRN can be iteratively constructed as

$$\mathbf{H}^{(t)} = \sigma\left(\mathbf{H}^{(t-1)} \odot \left(\mathbf{A}\mathbf{W}_A^{(t)} + \mathbf{X}\mathbf{W}_X^{(t)}\right)\right), \quad (3)$$

$$\mathbf{H}^{(0)} = \sigma\left(\mathbf{A}\mathbf{W}_A^{(0)} + \mathbf{X}\mathbf{W}_X^{(0)}\right). \quad (4)$$

where $\mathbf{W}_A^{(t)}$ and $\mathbf{W}_X^{(t)}$ are parameters for t^{th} layer. It is obvious that the element-wise product $\odot$ can preserve the reciprocal characteristic, thus the reciprocal characteristic of final representation can be iteratively guaranteed.

Expressive Ability of GRN: The expressive ability of GNNs is often illustrated by analyzing the interaction between topology and node attribute as the model depth increases. For simplicity, by removing the nonlinear mapping, the formula of the first layer of GRN is analyzed, i.e.

$$\begin{aligned}\mathbf{H} &= \left(\mathbf{H}_A + \mathbf{H}_X\right) \odot \left(\mathbf{H}_A + \mathbf{H}_X\right) \\ &= \mathbf{H}_A \odot \mathbf{H}_A + \mathbf{H}_X \odot \mathbf{H}_X + 2\mathbf{H}_A \odot \mathbf{H}_X\end{aligned}$$

where $\mathbf{H}_A \odot \mathbf{H}_A$ and $\mathbf{H}_X \odot \mathbf{H}_X$ are the Quadratic Network for topology and node attribute, respectively. The third term on the right-hand side, i.e.

$$\mathbf{H}_A \odot \mathbf{H}_X = \mathbf{A}\mathbf{W}_A \odot \mathbf{X}\mathbf{W}_X \quad (5)$$

TABLE III
CLASSIFICATION ACCURACY OF LARGE-SCALE NETWORKS.(BOLD INDICATES THE BEST, UNDERLINED INDICATES THE SECOND BEST). NOTE THAT THE EVALUATION METRIC FOR GENIUS IS ROC-AUC AS IN [28]. OOM DENOTES THE RUN-OUT-OF MEMORY ERROR.

Methods	Penn94	pokec	arXiv-year	snap-patents	genius	twitch-gamers
MLP	73.61±0.40	62.37±0.02	36.70±0.21	31.34±0.05	86.68±0.09	60.92±0.07
GCN	82.47±0.27	75.45±0.17	46.02±0.26	45.65±0.04	87.42±0.37	62.18±0.26
GAT	81.53±0.55	71.77±6.18	46.05±0.51	45.37±0.44	55.80±0.87	59.89±4.12
SGC	83.02±0.03	78.82±0.22	47.02±0.82	45.76±0.38	88.32±0.62	63.05±1.12
GraphSAGE	82.83±0.34	79.07±0.57	48.38±1.16	46.63±0.22	87.16±0.83	61.30±2.97
GCNII	82.92±0.59	78.94±0.11	47.21±0.28	37.88±0.69	90.24±0.09	63.39±0.61
APPNP	74.33±0.38	62.58±0.08	38.15±0.26	32.19±0.07	85.36±0.62	60.97±0.10
H2GCN	OOM	OOM	49.09±0.10	OOM	OOM	OOM
JKNet	79.38±0.54	77.23±0.19	48.02±1.29	44.84±0.28	87.91±0.39	62.33±2.14
FAGCN	83.41±0.72	OOM	40.63±0.40	OOM	88.18±0.67	63.29±0.41
GPRGNN	81.38±0.16	78.83±0.05	45.07±0.21	40.19±0.03	90.05±0.31	61.89±0.29
GloGNN	85.74±0.42	83.05±0.07	54.79±0.25	62.03±0.21	**90.91±0.13**	66.34±0.29
LINKX	84.71±0.52	82.04±0.07	56.00±1.34	61.95±0.12	90.77±0.27	66.06±0.19
GRN	**86.60±0.34**	**83.33±0.20**	**57.37±1.14**	**62.12±0.27**	90.85±0.07	**67.06±0.21**

performs the interaction between topology and node attribute. Note that the interaction in Eq. (5), which performs element-wise product, is very different from existing matrix multiplication based GNNs from both spectral domains and spatial domains. This kind of interaction essentially balances the topology and node attribute in *element-wise* instead of node-wise propagation in existing GNNs.

IV. EVALUATIONS

A. Datasets

We conduct the experiments for node classification on 14 benchmark datasets, which include 6 large-scale datasets and 9 small-scale datasets. Statistics of datasets are shown in I, II.

Citation networks: Cora, Citeseer, and Pubmed, which are widely used to evaluate GNNs, are the standard citation network benchmark datasets [29], [30]. arXiv-year is a directed subgraph of ogbn-arXiv. **WebKB webpage networks**: Cornell and Texas are the webpage networks that are captured from the computer science departments of these universities, respectively. **Co-occurrence network**: Actor network contains the co-occurrences of actors in films, which are extracted from the heterogeneous information networks. **Wikipedia networks**: Chameleon and Squirrel are the webpages extracted from different topics in Wikipedia [31]. **Social networks**: Penn94 is a subgraph extracted from Facebook whose nodes are students. Pokec is a friendship network from a Slovak online social network. genius is a subnetwork extracted from genius.com, which is a website for crowdsourced annotations of song lyrics. twitch-gamers is a subgraph from Twitch's platform.

B. Baselines

To verify the superiority of the proposed GRN, 10 baseline methods are employed for performance comparison. These methods are divided into two categories. The first category consists of classical methods for graph data, such as multiple layer perception (MLP), Graph Convolutional Network (GCN) [21], Graph Attention Network (GAT) [17], Simple Graph Convolutional (SGC) [22], Inductive Representation Learning on Large Graphs (GraphSAGE) [32]. The methods in the second category possess some attractive characteristics, including high accuracy on homophilic networks, heterophilic networks and massive graphs. This category contains GCN with Initial residual and Identity mapping (GCNII) [33], Personalized Propagation of Neural Predictions (APPNP) [7], Representation Learning on Graphs with Jumping Knowledge Networks (JKNet) [10], Beyond Homophily in Graph Neural Networks (H2GCN) [34], Beyond Low-frequency Information in Graph Convolutional Networks (FAGCN) [35], Generalized PageRank GNN (GPRGNN) [12], Finding Global Homophily in GNN (GloGNN) [36], Large Scale Learning on Non-Homophilous Graphs (LINKX) [28]. We collect the results from their original papers or employ the authors' implementation for specific datasets they are absent.

C. Parameter Setting

For the dataset in Table.I, we randomly split nodes for each class into 60%, 20% and 20% for training, validation and test , as suggested in [37]. For the dataset in Table.II, we randomly split nodes for each class into 50%, 25% and 25% for training, validation and test, as suggested in [28]. All results are obtained by computing over 10 random splits for fairness. The hyperparameters, including weight decay, dropout, and learning rate, are tuned on the validation set.

D. Results Analysis

The performance of the node classification task for each dataset across different methods is shown in Table III, IV. Hence experimental analysis is carried out from two aspects respectively.

1) Results on large-scale networks: Large-scale networks are universal in industrial applications yet hard to solve. Surprisingly, our proposed GRN achieves the new SOTA on

TABLE IV
CLASSIFICATION ACCURACY OF SMALL-SCALE NETWORKS.(BOLD INDICATES THE BEST, UNDERLINED INDICATES THE SECOND BEST).

Methods	Cora	Citeseer	Pubmed	Chameleon	Squirrel	Actor	Texas	Cornell
MLP	74.82±2.22	70.94±0.39	63.76±0.78	46.21±2.99	28.77±1.56	36.53±0.70	81.89±4.78	81.08±6.37
GCN	85.77±0.25	73.68±0.31	88.13±0.28	35.99±2.58	34.02±1.34	26.97±1.49	55.68±9.16	55.14±7.57
GAT	86.37±0.30	74.32±0.27	87.62±0.26	60.26±2.50	40.72±1.55	27.44±0.89	58.38±4.45	58.92±3.32
SGC	86.20±0.12	76.68±1.70	88.50±0.22	52.12±0.32	36.91±0.14	35.31±0.21	66.83±6.37	65.72±0.67
GraphSAGE	87.77±1.04	71.09±1.30	88.42±0.50	58.73±1.68	41.61±0.74	34.23±0.99	82.43±6.14	75.95±5.01
GCNII	88.65±2.78	77.08±1.21	**90.02±1.56**	60.61±2.00	37.85±2.76	36.18±0.61	69.46±1.86	74.86±2.73
APPNP	87.87±0.85	76.53±1.33	89.40±0.61	54.30±0.34	33.29±1.72	31.71±0.70	82.43±1.72	82.16±3.83
H2GCN	86.92±1.37	76.88±1.77	89.40±0.34	59.39±1.98	37.90±2.02	35.62±1.30	84.86±4.32	82.16±3.27
JKNet	**88.93±1.35**	74.37±1.53	87.68±0.30	62.31±2.76	44.24±2.11	36.47±0.51	65.35±4.86	56.49±3.22
FAGCN	87.77±1.69	74.66±2.27	88.60±0.64	61.12±1.95	40.88±2.02	36.81±0.26	61.82±8.71	67.95±10.02
GPRGNN	88.49±1.37	**77.99±1.64**	89.18±0.61	67.48±1.98	49.93±1.34	36.58±1.04	77.84±2.78	79.73±3.91
GloGNN	88.33±1.09	77.41±1.65	89.42±0.39	71.21±1.84	57.88±1.76	37.70±1.40	84.05±4.90	**85.05±5.10**
LINKX	84.64±1.13	73.19±0.99	87.86±0.77	68.42±1.38	61.81±1.80	36.10±1.55	74.60±8.37	77.84±5.81
GRN	86.28±1.15	76.35±1.81	89.82±0.37	**78.34±0.33**	**75.14±0.78**	**38.07±0.52**	**85.18±3.49**	84.29±5.72

5 networks in all 6. On the one hand, GRN significantly outperforms all basic GNNs. Except for the noticeable performance improvements compared to GCN and GAT, GRN makes remarkable progress compared to methods designed explicitly for homophilic and heterophilic networks, such as GCNII and GPRGNN. Some methods unsuitable for massive graphs occur run-out-of-memory errors, like H2GCN. On the other hand, GRN outperforms the methods designed for large-scale networks in most cases. For example, GRN is the runner-up, except for the genius. Fortunately, the difference between GloGNN and GRN on genius is very tiny. This observation demonstrates the superiority of the proposed GRN on the large-scale node classification task.

These improvements can be attributed to the novelty combination of topology and attribute information through Hadamard product. Traditional combination strategies such as concatenation and summation always introduce inescapable noise. On the contrary, GRN employs the topology information to reweight the attribute information, thus always keeping the valuable information and blocking the noises. Besides, run-out-of-memory error always stems from redundancy propagation which hinders the application of GNNs on massive graphs. Thanks to the simple yet effective Hadamard product, our proposed GRN is naturally efficient. In summary, the tradeoff between scalability and accuracy is well-balanced. These results inspire a broader industrial application for GRN.

2) Results on small-scale networks: The small real-world datasets can be divided into homophily and heterophily, respectively. On homophilic datasets, i.e., Cora, Citeseer, and Pubmed, GRN achieves comparable performances, which are slightly lower than SOTA, and the differences are tiny. It demonstrates the effectiveness of our proposed GRN on homophilic networks. These results correspond to peoples' perception that nodes belonging to the same class in homophilic networks always have similar topology structures, which benefits feature learning. On heterophilic datasets, GRN is compared with H2GCN and GPRGNN, which are all the GNNs designed for handling networks with heterophily. What is fascinating is that GRN significantly outperforms most of these baseline models. Specifically, GRN achieves new SOTA performance on Chameleon and Squirrel. These results redemonstrate the well-balanced tradeoff of GRN between scalability and accuracy. Since poor predictability attributes are widespread in networks with heterophily, GRN benefits a lot from the topology reweighted feature learning.

TABLE V
MODEL EFFICIENCY: AVERAGE TOTAL RUNNING TIME (S)

Model	GPRGNN	GloGNN	LINKX	GRN
Penn94	10.25	36.39	12.94	12.15
pokec	58.15	132.52	73.50	64.84
arXiv-year	8.35	5.36	3.97	3.20
snap-patents	79.11	86.17	15.39	11.06
genius	6.81	7.19	6.29	6.18
twitch-gamers	21.81	24.39	7.15	6.69

E. Efficiency Study

In this section, we study GRN's efficiency. We compare the average training time for effective methods on large-scale datasets for fairness. We use the same training set and hyperparameters for all these methods on each dataset and run the experiments for 500 epochs. All experiments are performed on the same device.

Table V clearly shows that GloGNN takes the most training time in most cases since its time-consuming topology denoising process and convolution operation. Compared with GloGNN, GPRGNN is more efficient since it performs convolution directly from its adjacency neighbors and has microscopic trainable parameters. However, GPRGNN is not good at handling large-scale networks as examined before II. On the whole, LINKX has balanced well between efficiency and effectiveness. However, our proposed GRN performs better than all those methods. GRN is simple enough and does not

employ convolution operation. These results show that GRN is highly efficient for tackling large-scale networks.

V. Conclusions

This paper investigates the multiplication of topology-related matrix and node attributes, which unifies graph neural networks from spatial and spectral domains. It observes that this formula does neither treat topology and attribute reciprocal from both employment and parameterization perspectives nor fully explore and capture topology structure. These drawbacks cause the over-smoothing issue and the performance reduction on networks beyond homophily. This paper proposes the Graph Reciprocal Network (GRN) by abstracting attributes from nodes and adopting the theory of Quadratic Networks to assemble the overall network. Theoretical analyses reveal two attractive characteristics of GRN. GRN performs fine-grained element-wise product of the representations of the topology and attribute instead of node-wise propagation in existing GNNs. Experiments demonstrate the superior performance and robustness of GRN.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No. U22B2036, 62376088, 61972442, 62102413, 62276187, U1936210, U1936208, 11931915), in part by the National Science Fund for Distinguished Young Scholarship of China (No. 62025602), in part by the Natural Science Foundation of Hebei Province of China under Grant F2020202040, in part by the Fok Ying-Tong Education Foundation China (No. 171105), in part by the Technological Innovation Team of Shaanxi Province (No. 2020TD-013), and in part by the Tencent Foundation and XPLORER PRIZE.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[2] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *CVPR*, pp. 5425–5434, 2017.

[3] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, 2017.

[4] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *TNNLS*, vol. 32, no. 1, pp. 4–24, 2021.

[5] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[6] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *ICML*, pp. 1263–1272, 2017.

[7] J. Klicpera, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," in *ICLR*, 2019.

[8] G. Li, M. Müller, A. K. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?," in *ICCV*, pp. 9266–9275, 2019.

[9] S. Luan, M. Zhao, X. Chang, and D. Precup, "Break the ceiling: Stronger multi-scale deep graph convolutional networks," in *NeurIPS*, pp. 10943–10953, 2019.

[10] K. Xu, C. Li, Y. Tian, T. Sonobe, K. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *ICML*, pp. 5449–5458, 2018.

[11] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. V. Steeg, and A. Galstyan, "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *ICML*, pp. 21–29, 2019.

[12] E. Chien, J. Peng, P. Li, and O. Milenkovic, "Adaptive universal generalized pagerank graph neural network," in *ICLR*, 2021.

[13] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NIPS*, pp. 3837–3845, 2016.

[14] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[15] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: Graph convolutional neural networks with complex rational spectral filters," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 97–109, 2019.

[16] Q. Li, Z. Han, and X. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *AAAI*, pp. 3538–3545, 2018.

[17] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[18] Y. Ma, X. Liu, T. Zhao, Y. Liu, J. Tang, and N. Shah, "A unified view on graph neural networks as graph signal denoising," 2020.

[19] M. Zhu, X. Wang, C. Shi, H. Ji, and P. Cui, "Interpreting and unifying graph neural networks with an optimization framework," in *WWW*, pp. 1215–1226, 2021.

[20] L. Yang, C. Wang, J. Gu, X. Cao, and B. Niu, "Why do attributes propagate in graph convolutional neural networks?," in *AAAI*, pp. 4590–4598, 2021.

[21] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[22] F. Wu, A. H. S. Jr., T. Zhang, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *ICML*, pp. 6861–6871, 2019.

[23] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *SIGKDD*, pp. 701–710, 2014.

[24] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *SIGKDD*, pp. 855–864, 2016.

[25] F. Fan, J. Xiong, and G. Wang, "Universal approximation with quadratic deep networks," *Neural Networks*, vol. 124, pp. 383–392, 2020.

[26] N. Tsapanos, A. Tefas, N. Nikolaidis, and I. Pitas, "Neurons with paraboloid decision boundaries for improved neural network classification performance," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 1, pp. 284–294, 2019.

[27] F. Fan, H. Shan, M. K. Kalra, R. Singh, G. Qian, M. Getzin, Y. Teng, J. Hahn, and G. Wang, "Quadratic autoencoder (Q-AE) for low-dose CT denoising," *IEEE Trans. Medical Imaging*, vol. 39, no. 6, pp. 2035–2050, 2020.

[28] X. Li, R. Zhu, Y. Cheng, C. Shan, S. Luo, D. Li, and W. Qian, "Finding global homophily in graph neural networks when meeting heterophily," in *ICML*, pp. 13242–13256, 2022.

[29] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[30] G. Namata, B. London, L. Getoor, and B. Huang, "Query-driven active surveying for collective classification," in *International Workshop on Mining and Learning with Graphs*, 2012.

[31] B. Rozemberczki, C. Allen, and R. Sarkar, "Multi-scale attributed node embedding," *arXiv:1909.130*, 2019.

[32] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NIPS*, pp. 1024–1034, 2017.

[33] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *ICML*, pp. 1725–1735, 2020.

[34] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra, "Beyond homophily in graph neural networks: Current limitations and effective designs," in *NeurIPS*, 2020.

[35] D. Bo, X. Wang, C. Shi, and H. Shen, "Beyond low-frequency information in graph convolutional networks," pp. 3950–3957, 2021.

[36] D. Lim, F. Hohne, X. Li, S. L. Huang, V. Gupta, O. Bhalerao, and S. Lim, "Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods," in *NeurIPS*, pp. 20887–20902, 2021.

[37] H. Pei, B. Wei, K. C. Chang, Y. Lei, and B. Yang, "Geom-gcn: Geometric graph convolutional networks," in *ICLR*, 2020.