Graph Neural Networks Beyond Compromise Between Attribute and Topology

Liang Yang*, Wenmiao Zhou*, Weihang Peng*
yangliang@vip.qq.com
School of Artificial Intelligence
Hebei University of Technology
Tianjin, China

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

Chuan Wang
wangchun@iie.ac.cn
State Key Laboratory of Information Security, IIE, CAS
Beijing, China

Xiaochun Cao
caoxiaochun@iie.ac.cn
School of Cyber Science and Technology
Sun Yat-sen University
Shenzhen, China

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

ABSTRACT

Although existing Graph Neural Networks (GNNs) based on message passing achieve state-of-the-art, the over-smoothing issue, node similarity distortion issue and dissatisfactory link prediction performance can't be ignored. This paper summarizes these issues as the interference between topology and attribute for the first time. By leveraging the recently proposed optimization perspective of GNNs, this interference is analyzed and ascribed to that *the learned representation in GNNs essentially compromises between the topology and node attribute*. To alleviate the interference, this paper attempts to break this compromise by proposing a novel objective function, which fits node attribute and topology with different representations and introduces mutual exclusion constraints to reduce the redundancy in both representations. The mutual exclusion employs the statistical dependence, which regards the representations from topology and attribute as the observations of two random variables, and is implemented with Hilbert-Schmidt Independence Criterion. Derived from the novel objective function, a novel GNN, i.e., Graph Neural Network Beyond Compromise (GNN-BC), is proposed to iteratively updates the representations of topology and attribute by simultaneously capturing semantic information and removing the common information, and the final representation is the concatenation of them. The performance improvements on node classification and link prediction demonstrate the superiority of GNN-BC on relieving the interference between topology and attribute.

*Three authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WWW '22, April 25–29, 2022, Lyon, France.
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

KEYWORDS

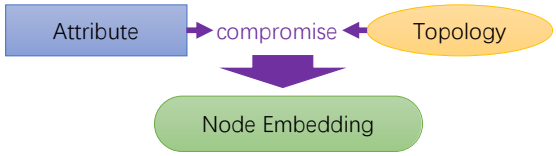
Graph neural networks, networks with heterophily, network topology, node attribute

ACM Reference Format:
Liang Yang*, Wenmiao Zhou*, Weihang Peng, Bingxin Niu, Junhua Gu, Chuan Wang, Xiaochun Cao, and Dongxiao He. 2022. Graph Neural Networks Beyond Compromise Between Attribute and Topology. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Lyon, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

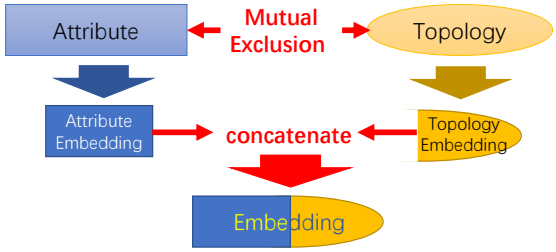
1 INTRODUCTION

Graph Neural Networks (GNNs) become the hot topic in deep learning for their superior performance on handling non-Euclidean data, and are applied to many fields ranging from computer vision to natural language processing [1, 2]. Motivated by the graph Fourier Transform in spectral graph theory [3, 4], some graph convolutional neural networks (GCNNs), including ChebyNet [5], GCN [6] and GWNN [7], are designed, while another kind of GNNs are proposed from the spatial perspective, i.e., message passing on graph [8], such as GraphSAGE [9], GAT [10] and GIN [11]. The success of GCNNs on combining topology and node content is attributed to low-pass filtering [12] and smoothing [13] from these perspectives, respectively.

Unfortunately, recent research realizes the interference between topology and node content. On one hand, topology tends to interfere the node attributes [14]. First, the well-known over-smoothing issue is actually the loss of expressive power of node attributes smoothened by the topology [13, 15]. Second, propagation also makes nodes, which possess similar attributes, obtain very different representations [16, 17]. On the other hand, attribute tends to distort the network embedding from topology. Specifically, for link prediction task, where topology information is dominant, GNNs based on both topology and attribute often perform worse than many topology-based network embedding methods [18], such as matrix factorization (MF), stochastic block model (SBM) [19] and node2vec [20], etc. Thus, the interference between topology and



(a) Existing GNNs following compromise strategy



(b) The proposed GNNs beyond compromise strategy

Figure 1: Comparison between GNNs following the compromise strategy and the proposed GNNs beyond the compromise strategy. Compromise strategy makes the network embedding balance between network topology and node attribute, and leads to the information loss of both topology and node attribute in the embedding. The proposed GNN encodse node attribute and topology with different representations and introduces mutual exclusion to reduce the redundancy and information loss in both representations.

node content is the primary cause of many issues in GNNs, and significantly degrades the performance of GNNs in many tasks.

To understand the interference between topology and node content, this paper investigates the underlying philosophy of GNNs by leveraging the recently proposed optimization perspective [21, 22]. The optimization perspective shows that the GNNs actually minimize a group of objective functions, which balance the node representation between node attribute and topology constraint, with gradient descent [23]. Therefore, GNNs seek the representation by compromising between the topology and node content as shown in Figure 1(a). From the viewpoint of node content, this compromise leads to the node content be smoothened. From the viewpoint of topology, this compromise distorts the topology toward node attribute. Therefore, the compromise between the topology and attribute is the reason for interference between them.

To alleviate the interference between topology and node content, this paper tends to break the compromise between them. A natural way to break the compromise is to fit attribute and topology with different representations in the objective function. Furthermore, to reduce the redundancy and enhance the ability on exploiting sufficient semantic information in embedding space with limited dimensionality, the representations of attribute and topology need to be mutual exclusive. The framework is shown in Figure 1(b). To implement the mutual exclusion, statistical dependence, which regards the representations from topology and attribute as the observations of two random variables, is employed, and Hilbert-Schmidt Independence Criterion (HSIC) [24] is added to the objective function to measure the dependence between topology and attribute. The derived GNN from the novel objective function, i.e., GNN Beyond Compromise (GNN-BC), iteratively updates the representations of topology and attribute by simultaneously capturing semantic information and removing the common information. The final representation is the concatenation of the representations of attribute and topology. This GNN is trained by feeding the final representation into the specific objective functions, such as the cross-entropy for semi-supervised node classification task.

The main contributions of this paper are summarized as follows:

- We raise and analyze the interference between topology and attribute in GNNs, which causes the over-smoothing issue, node similarity distortion issue and dissatisfactory link prediction performance, and ascribe this issue to the compromise between topology and attribute.
- We propose a novel GNN, i.e., GNN Beyond Compromise (GNN-BC), by deriving from the objective function, which fits attribute and topology with different representations and introduces mutual exclusion based on Hilbert-Schmidt Independence Criterion, to alleviate the interference between topology and node content.
- We experimentally evaluate the superiority of the proposed GNNs beyond the compromise strategy on both attribute and topology related tasks.

2 RELATED WORK

Although the success of the Graph Convolutional Layer (GCL) in Graph Neural Network (GCN) [6] is attributed to the Laplacian smoothing of the node feature among neigbourhoods [13] or low-passing filtering [12], the original node features will be over-smoothed by stacking too many GCLs, and the obtained node representations are only determined by the degree of nodes and graph connectivity but independent of their original feature [13, 25]. Thus, each GCL also induces the loss of expressive power and this loss is exponential as the number of layers increases [15]. There are two kinds of strategies to alleviate the over-smoothing issue. The first strategy modifies the learned representations. APPNP and GCNII balance the output of GCL and original attribute [26, 27], while PairNorm constrains the total distance of node representations constant before and after GCL [28]. The second strategy modifies the network topology to control the propagation. DropEdge randomly drops edges in each training epoch to constrain the labelled node be correctly predicted [29], while GRAND randomly drops nodes and constrains all node representations not be significantly impacted. Other methods combine multi-scale topology information to alleviate over-smoothing issue, such as PPNP, MixHop, GDN and JKNet [26, 30–33]. The equivalence between PPNP and APPNP in [26] bridges the gap between these two strategies.

2.1 Homophily vs. Heterophily

Network homophily states that based on node attributes, similar node pairs may be more likely to attach to each other than dissimilar pairs, while network heterophily depicts the opposite, i.e., dissimilar pairs are more likely to be attached. Thus, smoothing-based GCN and its variants only meet the property of network with high homophily. Unfortunately, networks with heterophily are ubiquitous. Thus, it is important to make GNNs can be adaptively applied to networks with both homophily and heterophily. To design GNNs

for network with heterophily, Geom-GCN augments the propagation by considering the similarity in embedding space as well as topology space [34]. H2GCN and GraphSage concatenate the aggregated feature from different hops instead of summation by finding that second-order neighbourhods should be more similar than first-order ones in networks with heterophily [9, 35]. To facilitate the filtering beyond low-frequency, GPRGNN and FAGCN allow the propagation with negative weights by generalizing APPNP [26] and GAT [10], respectively [36, 37]. Recent works jointly investigate the oversmoothing issue and heterophily and consider them as the two sides of the same coin [38].

3 PRELIMINARIES

This section provides the notations used in the paper and some classic GNNs for the analysis in the next section.

3.1 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.

3.2 Graph Neural Networks

Although existing graph neural networks are proposed from the perspectives of spectral and spatial, respectively, most of them follow the message passing scheme [8] based on the connection between these two perspectives [39], such as GCN [6], SGC [12], APPNP [26] and GCNII [27]. The graph convolutional layers of GCN, SGC and APPNP are as follows.

$$\text{GCN} \qquad \mathbf{H}^{(t+1)} = \sigma(\tilde{\mathbf{A}}\mathbf{H}^{(t)}\mathbf{W}), \quad \mathbf{H}^{(0)} = \mathbf{X} \tag{1}$$

$$\text{SGC} \qquad \mathbf{H}^{(t+1)} = \tilde{\mathbf{A}}\mathbf{H}^{(t)}, \quad \mathbf{H}^{(0)} = \mathbf{X} \tag{2}$$

$$\text{APPNP} \qquad \mathbf{H}^{(t+1)} = (1-\alpha)\tilde{\mathbf{P}}\mathbf{H}^{(t)} + \alpha\mathbf{X} \tag{3}$$

where $\tilde{\mathbf{A}} = \bar{\mathbf{D}}^{-\frac{1}{2}}\bar{\mathbf{A}}\bar{\mathbf{D}}^{-\frac{1}{2}}$ with $\bar{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\tilde{\mathbf{P}} = \bar{\mathbf{D}}^{-1}\bar{\mathbf{A}}$ are the symmetric and asymmetric normalized adjacency matrix, respectively. Besides, many GNNs tend to overcome over-smoothing issue by incorporating multi-scale information [32], such as JKNet [33] and DAGNN [25], as follows

$$\mathbf{H} = \sum_{t=1}^{T} \alpha_t \tilde{\mathbf{A}}^t \mathbf{X}, \tag{4}$$

Note that the mapping function in intermediate layer in SGC, APPNP, JKNet and DAGNN are omitted, since they all follow the Decoupled GCN scheme [40], which adopts only one mapping function instead of one for each layer.

Table 1: Attribute distortion induced by topology in terms of node classification performance. The performances of some GNNs are worse than those of attribute-based MLP.

Dataset	Texas	Wisconsin	Actor	Cornell
GCN	59.46±5.25	59.80±6.99	30.26±0.79	57.03±4.67
GAT	58.38±4.45	55.29±8.71	26.28±1.73	58.92±3.32
GraphSAGE	**82.43±6.14**	81.18±5.56	34.23±0.99	75.95±5.01
GCN-Cheby	77.30±4.07	79.41±4.46	34.11±1.09	74.32±7.46
MLP	81.89±4.78	**85.29±3.61**	**35.76±0.98**	**81.08±6.37**

4 ANALYSIS

In this section, the optimization perspective of GNNs is given. Then, based on this perspective, a novel analysis is introduced to reveal the limits of existing GNNs.

4.1 Optimization Perspective of GNNs

Recently, some attempts unify GNNs from the perspective of numerical optimization [21–23]. They show that the message passing in GNNs, such as GCN [6], actually is the gradient descent of the graph Laplacian regularization with node attribute as the initial, i.e.,

$$tr(\mathbf{H}^T\tilde{\mathbf{L}}\mathbf{H}) = \frac{1}{2}\sum_{uv} a_{uv}\left\|\frac{\mathbf{h}_u}{\sqrt{d_u+1}} - \frac{\mathbf{h}_v}{\sqrt{d_v+1}}\right\|^2. \tag{5}$$

That is, each graph convolutional layer is equivalent to one gradient descent of Eq. (5). Since the solution to Eq. (5) is $\mathbf{h}_u = \sqrt{d_u+1}\mathbf{1}$, where $\mathbf{1}$ is the vector of ones, the obtained node representation from many graph convolutional layers is also $\mathbf{h}_u = \sqrt{d_u+1}\mathbf{1}$, which is only determined by the degree of node. Thus, GCN tends to be over-smoothing by stacking multiple layers. To prevent over smoothing issue, many GNNs, such as APPNP [26], GCNII [27] and JKNet [33], balance the Eq. (5) with node attributes $\mathbf{X}$ as

$$\|\mathbf{H} - \mathbf{X}\|_F^2 + \lambda tr(\mathbf{H}^T\tilde{\mathbf{L}}\mathbf{H}), \tag{6}$$

where λ is the weighting parameter. Although the trick of balancing can alleviate the over-smoothing issue, it doesn't essentially solve the problem caused by the graph Laplacian regularization in Eq. (5). In the following sections, the common philosophy of GNNs is analyzed under this optimization perspective.

4.2 Compromise between Topology and Attribute

As shown in Eq. (6), the unified optimization perspective of GNNs consists of two terms. The first term constrains the learned representations $\mathbf{H}$ be similar with the original node attributes. Since the second term, i.e, graph Laplace regularization, possesses the formula in Eq. (5), it makes the learned representation $\mathbf{H}$ meet the topology structure, i.e., connected nodes with $a_{ij} = 1$ own similar representations. Therefore, it can be concluded that: *The learned representations from GNNs tend to compromise (balance) between original node attribute and graph topology.* Although this compromise strategy seems effective on integrating node attribute and graph topology, it possesses one remarkable defect that interference between topology and attributes weakens the expressive power of representation.

Table 2: Topology distortion induced by attribute in terms of link prediction performance. The performance of VAGE is worse than that of many topology-based methods.

Dataset	MF	SBM	node2vec	VGAE
USAir	94.08±0.80	**94.85±1.14**	91.44±1.78	89.28±1.99
PB	**94.30±0.53**	93.90±0.42	85.79±0.78	90.70±0.53
C.ele	85.90±1.74	**86.48±2.60**	84.11±1.27	81.80±2.18
Router	78.03±1.63	**85.65±1.93**	65.46±0.86	61.51±1.22
E.coli	93.76±0.56	**93.82±0.41**	90.82±1.49	90.81±0.63

On one hand, topology tends to interfere the node attributes. First, the well-known over-smoothing issue is actually the loss of expressive power of node attributes smoothened by the topology [13, 15]. To make connected nodes possess similar representation, some representative attributes are smoothened by propagation over the neighbourhoods, and thus the representation power of node attribute degrades after smoothing. This is demonstrated by the fact that multi-layered perception (MLP) based on node attributes outperforms GNNs, which combine attributes and topology, on some datasets [35]. Table 1 from [35] shows some representative results. Second, propagation also makes nodes, which possess similar attributes, obtain very different representations [16, 17]. These two aspects show the attribute distortions induced by topology.

On the other hand, attribute tends to distort the network embedding from topology. Recent studies reveal that most exiting network embedding methods, such as DeepWalk [41], LINE [42], node2vec [20], are equivalent to factorizing the multi-scale normalized adjacency matrix [43], and proposed computationally efficient matrix factorization methods, such as AROPE [44] and NetSMF [45]. Specifically, the Singular Value Decomposition (SVD) is applied on the multi-scale normalized adjacency matrix as follow

$$\mathbf{U}\Sigma\mathbf{V}^T = \sum_{t=1}^{T} \alpha_t \hat{\mathbf{A}}^t, \tag{7}$$

where $\hat{\mathbf{A}}$ denotes a general normalized adjacency matrix, and α_t represents the weights of different scales. The orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ contain the singular vectors and diagonal matrix Σ consists of singular values. If $\hat{\mathbf{A}}$ is symmetric, $\mathbf{U}$ and $\mathbf{V}$ are similar except for the signal, according to the relationship between the Singular Values Decomposition and Eigenvalue Decomposition [44]. Then

$$\bar{\mathbf{U}} = \mathbf{U}\sqrt{\Sigma}, \quad \bar{\mathbf{V}} = \mathbf{V}\sqrt{\Sigma}, \tag{8}$$

are employed as the network embedding for node classification and link prediction tasks.

The relationship between representations from GNNs and topology based network embedding can be revealed by reformulating the multi-scale GNNs in Eq. (4) as

$$\mathbf{H} = \left(\sum_{t=1}^{T} \alpha_t \tilde{\mathbf{A}}^t\right)\mathbf{X} = \bar{\mathbf{U}}\left(\bar{\mathbf{V}}^T\mathbf{X}\right). \tag{9}$$

The representation learned from GNNs, i.e., $\mathbf{H}$, is equivalent to converting topology-based embedding $\bar{\mathbf{U}}$ via the transformation $\bar{\mathbf{V}}^T\mathbf{X}$. Note that the transformation matrix $\bar{\mathbf{V}}^T\mathbf{X}$ models the correlation between topology information $\bar{\mathbf{V}}$ and node attribute information $\mathbf{X}$. If the attribute information is the same as topology information, i.e., $\bar{\mathbf{V}} = \mathbf{X}$, then the representation of GNNs is the same as topology-based embedding, i.e., $\mathbf{H} = \bar{\mathbf{U}}$, due to the orthogonality of the singular vector matrix, i.e., $\bar{\mathbf{V}}^T\bar{\mathbf{V}}^{=}\mathbf{I}$. Unfortunately, since the attribute information is very different from the topology information in practice, Eq. (9) shows topology-based embedding is distorted toward the attribute information space. Thus, some nodes with similar topology information may obtain different representations after the transformation, and some representative information contained in the topology-based embeddings tend to lose after distortion. Table 2 from [18] shows the performance on link prediction task, which demonstrates the impact of attribute on exploiting topology information. Variational graph auto-encoder (VAGE) [46] based on both topology and attribute performs worse than many topology-based network embedding methods, such as matrix factorization (MF), stochastic block model (SBM) [19] and node2vec [20].

Therefore, the compromise strategy between topology and attribute, which is the underlying philosophy of most existing GNNs based on message passing, tends to cause the interference between topology and node attributes. This interference leads to the loss of important information in representations learned from both topology and node attribute, and thus degrades the expressive power of the combination between topology and attribute.

5 METHODS

This section aims to propose a novel strategy to combine topology and attribute information to alleviate the inherent drawback in the widely-adopted compromise strategy. Firstly, the framework is introduced. Then, the key component of the framework, i.e., mutual exclusion, is given. Finally, an effective implementation along with the model analysis is provided.

5.1 Motivations and Framework

Recall that the unified optimization perspective in Eq. (6) shows the representation $\mathbf{H}$ of GNNs is essentially a compromise between the topology $\mathbf{A}$ and attribute $\mathbf{X}$ by balancing the losses of fitting to attribute and topology, i.e., $\|\mathbf{H}-\mathbf{X}\|_F^2$ and $tr(\mathbf{H}^T\tilde{\mathbf{L}}\mathbf{H})$. Thus, a natural way to break the compromise is to fit attribute and topology with different representations. For generalization, the framework can be formulated as

$$\mathcal{A}\big(\mathbf{Y}, f(\mathbf{X})\big) + \mathcal{T}\big(\mathbf{Z}, g(\mathbf{A})\big), \tag{10}$$

where $\mathbf{Y} \in \mathbf{R}^{N\times F'}$ and $\mathbf{Z} \in \mathbf{R}^{N\times F'}$ stand for the representations for attribute and topology, respectively. $f(\mathbf{X})$ and $g(\mathbf{A})$ represent information extractions from attribute and topology, respectively, such as multi-layered perception on attribute $f(\mathbf{X}) = \mathbf{X}\mathbf{W}$ and multi-scale topology information $g(\mathbf{A}) = \sum_t \mathbf{A}^t$. $\mathcal{A}(\cdot,\cdot)$ and $\mathcal{T}(\cdot,\cdot)$ denote the losses of representation on fitting attribute and topology, respectively. Then, the final node embedding can be obtained by concatenation as

$$\mathbf{H} = [\mathbf{Y}\|\mathbf{Z}], \tag{11}$$

where $\|$ stands for the concatenation operation. Since $\mathbf{Y}$ and $\mathbf{Z}$ respectively exploit the attribute and topology, concatenation can overcome the interference between attribute and topology.

Unfortunately, $\mathbf{Y}$ and $\mathbf{Z}$ may contain some common patterns in both attribute and topology. Thus, the final representation $\mathbf{H}$ may

be redundant, and result in the incapability of exploiting sufficient semantic information in embedding space with limited dimensionality. To reduce the redundancy and enhance the expressive power, the common patterns in attribute and topology should be only exploited by either attribute representation $\mathbf{Y}$ or topology representation $\mathbf{Z}$. To this end, the representations of attribute and topology need to be mutual exclusive, and the framework Eq. (10) can be enhanced to

$$\mathcal{A}\big(\mathbf{Y}, f(\mathbf{X})\big) + \mathcal{T}\big(\mathbf{Z}, g(\mathbf{A})\big) + \mathcal{M}(\mathbf{Y}, \mathbf{Z}), \tag{12}$$

where the third term $\mathcal{M}(\mathbf{Y}, \mathbf{Z})$ constrains the mutual exclusion between the two representations. Therefore, the framework in Eq. (12) possesses the ability to reduce both interference and redundancy, and thus may facilitate the effective combination of topology and attribute.

5.2 Mutual Exclusion

The main challenge in the implementation of Eq. (12) is how to define the mutual exclusion term $\mathcal{M}(\mathbf{Y}, \mathbf{Z})$, since the representations from topology and attribute may possess different scales and distributions. Thus, instead of implementing mutual exclusion by directly comparing the representations, statistical dependence is employed [24]. Specifically, statistical dependence measures the dependence between two random variables $\mathbf{y}$ and $\mathbf{z}$ with rows in $\mathbf{Y}$ and $\mathbf{Z}$ as their observations, respectively. Here, the Hilbert-Schmidt Independence Criterion (HSIC) [24] is adopted to measure the dependence for its nonlinearity, flexibility and computational efficiency.

Since HSIC is an quantity defined based on the cross-covariance, we first review its definition. Let's define two mapping functions from $\mathbf{y}$ and $\mathbf{z}$ to kernel space as $\phi(\mathbf{y}) : \mathcal{Y} \mapsto \mathcal{F}$ and $\psi(\mathbf{z}) : \mathcal{Z} \mapsto \mathcal{J}$, such that the inner product between vectors are given by kernel functions $k_1(\mathbf{y}_i, \mathbf{y}_j) =< \phi(\mathbf{y}_i), \phi(\mathbf{y}_j) >$ and $k_2(\mathbf{z}_i, \mathbf{z}_j) =< \psi(\mathbf{z}_i), \psi(\mathbf{z}_j) >$. The cross-covariance is the covariance of two random variables as:

$$C_{\mathbf{yz}} = \mathbb{E}_{\mathbf{yz}}\Big[(\phi(\mathbf{y}) - \mu_{\mathbf{y}}) \otimes (\psi(\mathbf{z}) - \mu_{\mathbf{z}})\Big], \tag{13}$$

where $\mu_{\mathbf{y}} = \mathbb{E}_{\mathbf{y}}[\phi(\mathbf{y})]$ and $\mu_{\mathbf{z}} = \mathbb{E}_{\mathbf{z}}[\psi(\mathbf{z})]$ stand for the means and $\otimes$ represents the outer product. Then, the HSIC is defined as the Frobenius norm of the associated cross-covariance $C_{\mathbf{yz}}$, i.e.

$$\text{HSIC}(\mathbf{p}_{\mathbf{yz}}, \mathcal{F}, \mathcal{J}) = \|C_{\mathbf{yz}}\|_F^2, \tag{14}$$

where $\|\mathbf{A}\|_F = \sqrt{\sum_i \sum_j a_{ij}}$. In practice, the expectation $\mathbb{E}_{\mathbf{yz}}[\cdot]$ can't be obtained, since the joint distribution $\mathbf{p}_{\mathbf{yz}}$ is unknown. Therefore, HSIC in Eq. (14) can be approximated by employing the observations of random variables $\mathbf{y}$ and $\mathbf{z}$, i.e., the rows of $\mathbf{Y}$ and $\mathbf{Z}$, respectively. This approximation is given in [24] as

$$\text{HSIC}(\mathbf{Y}, \mathbf{Z}, \mathcal{F}, \mathcal{J}) = \frac{1}{(N-1)^2} tr\left(\mathbf{K_1 Q K_2 Q}\right), \tag{15}$$

where $\mathbf{K}_1$ and $\mathbf{K}_2$ are the Gram matrices with elements as $k_1(\mathbf{y}_i, \mathbf{y}_j)$ and $k_2(\mathbf{z}_i, \mathbf{z}_j)$, respectively. $\mathbf{Q} = \mathbf{I} - \frac{1}{N}\mathbf{1}$ is to center the Gram matrix to have zero mean. To make the computation efficient, the representations $\mathbf{Y}$ and $\mathbf{Z}$ are normalized with sigmoid nonlinear function instead of multiplying with $\mathbf{Q}$, and the inner produt kernel is adopted. Thus, Eq. (15) can be reformulated as computationally efficient forms by omitting the constant coefficient

$$\text{HSIC}(\mathbf{Y}, \mathbf{Z}) = tr(\mathbf{Y^T Z Z^T Y}) = \|\mathbf{Y^T Z}\|_F^2. \tag{16}$$

Minimizing $\text{HSIC}(\mathbf{Y}, \mathbf{Z})$ tends to make representations from topology and attribute diverse to reduce the redundancy.

5.3 Implementations

By employing the HSIC as the mutual exclusion term, the framework defined in Eq. (12) can be implemented as

$$O = \|\mathbf{Y} - \mathbf{X}\|_F^2 + \lambda_1 tr(\mathbf{Z^T \tilde{L} Z}) + \lambda_2 \|\mathbf{Y^T Z}\|_F^2, \tag{17}$$

where λ_1 and λ_2 are the hyper-parameters to balance the impacts from different terms. As discussed in previous section that exiting GNNs can be derived by minimizing the objective function defined in Eq. (6), a novel GNN can be derived by minimizing the novel objection function defined in Eq. (17) with respect to both $\mathbf{Y}$ and $\mathbf{Z}$. Since Eq. (17) breaks the comprise between topology and attribute, the derived GNN is named as GNN Beyond Compromise, i.e. GNN-BC. To this end, the partial derivatives of O with respect to both $\mathbf{Y}$ and $\mathbf{Z}$ are set to zero, respectively, i.e.

$$\frac{\partial O}{\partial \mathbf{Y}} = 2(\mathbf{Y} - \mathbf{X}) + 2\lambda_2 \mathbf{Z Z^T Y} = 0, \tag{18}$$

$$\frac{\partial O}{\partial \mathbf{Z}} = 2\lambda_1 \tilde{\mathbf{L}} \mathbf{Z} + 2\lambda_2 \mathbf{Y Y^T Z} = 0. \tag{19}$$

Since the two variables $\mathbf{Y}$ and $\mathbf{Z}$ are correlated, the following two iterative updating rules can be obtained

$$\mathbf{Y}^{(k+1)} = \mathbf{X} - \lambda_2 \mathbf{Z}^{(k)} \mathbf{Z^{(k)}}^T \mathbf{Y}^{(k)}, \tag{20}$$

$$\mathbf{Z}^{(k+1)} = \left(\tilde{\mathbf{A}} - \frac{\lambda_2}{\lambda_1} \mathbf{Y}^{(k)} \mathbf{Y^{(k)}}^T\right) \mathbf{Z}^{(k)}. \tag{21}$$

By iteratively updating them, the objective in Eq. (17) can be minimized. As shown in previous section, to make the computation efficient, $\mathbf{Z}^{(k)} \mathbf{Z^{(k)}}^T$ and $\mathbf{Y}^{(k)} \mathbf{Y^{(k)}}^T$ should be normalized. Thus, Eqs. (20) and (21) are improved to

$$\mathbf{Y}^{(k+1)} = \mathbf{X} - \lambda_2 \sigma\left(\mathbf{Z}^{(k)} \mathbf{W} \mathbf{Z^{(k)}}^T\right) \mathbf{Y}^{(k)}, \tag{22}$$

$$\mathbf{Z}^{(k+1)} = \left(\tilde{\mathbf{A}} - \frac{\lambda_2}{\lambda_1} \sigma\left(\mathbf{Y}^{(k)} \mathbf{Y^{(k)}}^T\right)\right) \mathbf{Z}^{(k)}, \tag{23}$$

where $\sigma(\cdot)$ stands for the sigmoid function, and $\mathbf{W}$ is an additional learnable mapping function to enhance the expressive ability. The attribute and topology representations after K iterations, i.e., $\mathbf{Y}^{(K)}$ and $\mathbf{Z}^{(K)}$ are concatenated to form the final representation $\mathbf{H}$ as in Eq. (11). To train the proposed GNNs, i.e., the parameter $\mathbf{W}$, the final representation is fed into the specific objective functions, such as the cross-entropy for semi-supervised node classification task.

Complexity: If we assume the dimensions of $\mathbf{Y}^{(k)}$ and $\mathbf{Z}^{(k)}$ as same as the node attribute $\mathbf{X} \in \mathbb{R}^{N \times F}$ and the number of edges as M, the complexity of Eqs. (22) and (23) is $O(NF^2 + MF)$, which is linear with the size of the network. Therefore, the proposed novel GNN is efficient.

5.4 Model Analysis

Let's analyze the iterative updating rules in Eqs. (20) and (21). To deeply understand them, the key components are $\mathbf{Z}^{(k)} \mathbf{Z^{(k)}}^T \mathbf{Y}^{(k)}$ and $\mathbf{Y}^{(k)} \mathbf{Y^{(k)}}^T \mathbf{Z}^{(k)}$ where $\mathbf{Y}^{(k)}$ and $\mathbf{Z}^{(k)}$ stand for attribute and topology representations, respectively.

Table 3: Benchmark dataset statistics for node classification.

Dataset	Cora	Pubmed	Citeseer	Chameleon	Squirrel	Computer	Photo
# Nodes	2,708	19,717	3,327	2,277	5,201	13,752	7,650
# Edges	5,429	44,338	4,732	36,101	217,073	245,861	119,081
# Features	1,433	500	3,703	2,325	2,089	767	745
# Classes	7	3	6	5	5	10	8
Homphily Rate	0.656	0.644	0.578	0.024	0.055	0.272	0.459

Table 4: Mean Classification Accuracy (Bold indicates the best, underlined indicates the second best).

Dataset	Cora	Pubmed	Citeseer	Chameleon	Squirrel	Computer	Photo
GCN	85.77±0.25	88.13±0.28	73.68±0.31	28.18±0.23	23.96±0.26	82.52±0.32	90.54±0.21
GAT	86.37±0.30	87.62±0.26	74.32±0.27	42.93±0.28	30.03±0.25	81.95±0.38	90.09±0.27
GraphSAGE	87.77±1.04	88.42±0.50	71.09±1.30	49.24±1.68	36.28±1.73	83.11±0.23	90.51±0.25
MLP	74.82±2.22	63.76±0.78	70.94±0.39	49.67±0.78	37.04±0.46	70.48±0.28	78.69±0.30
GCNII	88.49±2.78	89.57±1.56	77.08±1.21	60.61±2.00	37.85±2.76	86.13±0.51	90.98±0.93
APPNP	87.87±0.85	89.40±0.61	76.53±1.33	54.30±0.34	33.29±1.72	81.99±0.26	91.11±0.26
JKNet	**88.93±1.35**	87.68±0.30	74.37±1.53	62.31±2.76	44.24±2.11	77.80±0.97	87.70±0.70
Geom-GCN-I	85.19±1.13	**90.05±0.90**	**77.99±1.23**	60.31±1.77	33.32±1.59	NA	NA
Geom-GCN-P	84.93±0.51	88.09±1.37	75.14±1.50	60.90±1.13	38.14±1.23	NA	NA
Geom-GCN-S	85.27±1.48	84.75±1.39	74.71±1.17	59.96±2.03	36.24±1.05	NA	NA
GPRGNN	88.65±1.37	89.18±0.61	**77.99±1.64**	67.48±1.98	49.93±1.34	82.90±0.37	91.93±0.26
FAGCN	87.77±1.69	88.60±0.64	74.66±2.27	61.12±1.95	40.88±2.02	86.09±0.40	91.96±0.71
H2GCN-1	86.92±1.37	89.40±0.34	77.07±1.64	57.11±1.58	36.42±1.89	OOM	OOM
H2GCN-2	87.81±1.35	89.59±0.33	76.88±1.77	59.39±1.98	37.90±2.02	OOM	OOM
GNN-BC	88.75±1.21	88.13±2.15	76.70±0.77	**74.63±0.93**	**61.41±1.55**	**89.60±0.89**	**93.17±0.67**

(1) Analysis to Eq. (20). $\mathbf{Z}^{(\mathbf{k})^T}\mathbf{Y}^{(k)}$ can be seen as the correlation matrix between topology and attribute, and thus $\mathbf{Z}^{(k)}\left(\mathbf{Z}^{(\mathbf{k})^T}\mathbf{Y}^{(k)}\right)$ can be regarded as converting the topology embedding $\mathbf{Z}^{(k)}$ into the space of attribute embedding $\mathbf{Y}^{(k)}$ via domain adaption. Therefore, Eq. (20) is to remove the common information captured by topology representation from attribute.

(2) Analysis to Eq. (21). It can be reformulated as $\mathbf{Z}^{(k+1)} = \tilde{\mathbf{A}}\mathbf{Z}^{(k)} - \frac{\lambda_2}{\lambda_1}\mathbf{Y}^{(k)}\mathbf{Y}^{(\mathbf{k})^T}\mathbf{Z}^{(k)}$. The first term $\tilde{\mathbf{A}}\mathbf{Z}^{(k)}$ is the Label Propagation Algorithm (LPA) in community detection [47]. Similar to (20), the second term $\mathbf{Y}^{(k)}\mathbf{Y}^{(\mathbf{k})^T}\mathbf{Z}^{(k)}$ can be regarded as converting the attribute embedding $\mathbf{Y}^{(k)}$ into the space of attribute embedding $\mathbf{Z}^{(k)}$ via domain adaption. Thus, Eq. (21) is to remove the common information captured by attribute representation from topology embedding.

Therefore, these two iterative updating rules meet the requirements of mutual exclusion.

6 EVALUATIONS

6.1 Node Classification Task

6.1.1 Datasets and splitting. For node classification task, three kinds of networks are adopted. Cora, Citeseer, and Pubmed, which are widely used to verify GNNs, are standard citation network benchmark datasets [48, 49]. In these networks, nodes and edge represent papers and citations between them, respectively. Words in the paper are employed to represent the node feature in bag-of-word form. The academic topic of paper is taken as the label of node. Chameleon and Squirrel are two page-page networks on specific topics in Wikipedia, where nodes represent webpages and edges are mutual links between pages [50]. Node features correspond to some informative nouns in the Wikipedia pages, and nodes are classified into four categories via quartiles in term of the number of the average monthly traffic of the page. Computers and Photo are two networks of Amazon co-purchase relationships [51]. In these networks nodes represent goods and edges stand for the connected two goods being frequently bought together. Each node owns a bag-of-words feature extracted from product reviews. The categories of the goods are employed as the label of node. Dataset statistics are summarized in Table 3.

For all datasets, we randomly split nodes of each class in to 60%, 20% and 20% for training validation and testing, and run on test sets over 10 random splits, as suggested in [34].

6.1.2 Baselines. To verify the effectiveness of the proposed GNN-BC on node classification task, 14 methods are employed as the baselines with default hyper-parameters. They are divided into 3 categories:

- Classic GNN models for node classification task including vanilla GCN [6], GAT [10] and GraphSAGE [9]. GCN simplifies spectral graph convolution as propagation with fixed-weight, while GAT extends GCN by learning the propagation

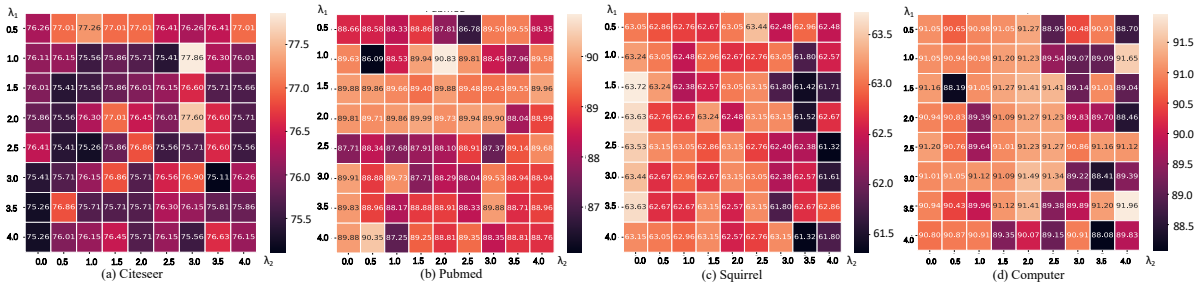


Figure 2: Node classification accuracies as the two hyper-paraemters λ_1 and λ_2 varying from 0 to 4.

Table 5: Link Prediction Task (Bold indicates the best, underlined indicates the second best).

Dataset	nodes	edges	average node degree	node2vec	LINE	VGAE	GPR-GNN	Geom-GCN	GNN-BC
USAir	332	2,126	12.81	**91.44±1.78**	81.47±10.71	89.28±1.99	86.87±1.54	84.35±3.24	91.07±0.21
NS	1,589	2,742	3.45	91.52±1.28	89.63±1.90	**94.04±1.64**	90.01±3.11	86.47±1.03	93.98±1.47
PB	1,222	16,714	27.36	85.79±0.78	76.95±2.76	90.70±0.53	88.74±1.34	80.08±1.19	**91.01±0.53**
Yeast	2,375	11,693	9.85	93.67±0.46	87.45±3.33	93.88±0.21	86.90±6.85	87.72±1.93	**94.47±1.10**
C.ele	297	2,148	14.46	84.11±1.27	69.21±3.14	81.80±2.18	82.69±1.89	76.88±2.95	**86.64±1.63**
Power	4,941	6,594	2.67	**76.22±0.92**	55.63±1.47	71.20±1.65	73.45±3.61	71.74±3.05	74.71±0.61
Router	5,022	6,258	2.49	65.46±0.86	67.15±2.10	61.51±1.22	**72.17±2.55**	68.40±2.66	65.02±0.86
E.coli	1,805	14,660	12.55	90.82±1.49	82.38±2.19	90.81±0.63	88.54±3.31	86.92±1.87	**95.17±0.56**

with attention mechanism. GraphSAGE replaces the summation operation in GCN with concatenation operation over neighbourhoods with different ranges.

- Deep GNNs designed to tackle over-smoothing issue including GCNII [27], APPNP [26] and JKNet [33]. JKNet adds residual connection between intermediate layer and output layer, while APPNP adds initial residual connection between intermediate layer and input layer. GCNII extends APPNP by augmenting with decaying weighting parameter to the mapping function in each layer.
- Models designed for networks with heterophily including 2-layer MLP, Geom-GCN [34], GPRGNN [36], FAGCN [37] and H2GCN [35]. For Geom-GCN, three variants, which employs different network embedding strategies, are used. Geom-GCN-I, Geom-GCN-P, and Geom-GCN-S employ Isomap [52], Poincare embedding [53], and struc2vec [54], respectively. For H2GCN, H2GCN-1 uses one embedding round (K = 1) and H2GCN-2 uses two rounds (K = 2).

6.1.3 Results analysis. Results are summarized in Table 4. We observe that our proposed model achieves new remarkable state-of-the-art results on Chameleon, Squirrel, Computer and Photo, which demonstrates the superiority of it. Among them, Chameleon and Squirrel are representative networks with heterophily, while Computer and Photo are large networks with homophily. Note that the performances of GNN-BC are 7.15% and 11.47% higher that those of GPRGNN, which is the SOTA method on networks with heterophily, on Chameleon and Squirrel, respectively. Note that recently proposed H2GCN tends to be out-of-memory on large networks, since its concatenation over neighbourhood with different ranges, while GNN-BC overcome this issue by concatenating representations of topology and attribute. This demonstrates that the proposed GNN-BC breaks the smoothing effect of existing GNNs and is more effective to incorporate network topology and node attribute than existing ones on network with heterophily.

Besides, although GNN-BC's performance on classic small networks with homophily, i.e., Cora, Citeseer and Pubmed, are lower than those of SOTA, their differences is very slight. Besides, none of the existing methods can consistently achieve the SOTA on all networks, their performance on networks with heterophily are significantly lower than the proposed GNN-BC. For example, most of them including methods designed for networks with heterophily, such as GCNII, Geom-GCN and H2GCN, only can achieve about 40% on Squirrel.

These results suggest that by breaking the compromise between attribute and topology, our proposed GNN-BC is more effective and universal than the previous models on incorporating topology and attribute for node classification.

6.2 Link Prediction Task

6.2.1 Datasets and baselines. In this section, we evaluate the performance of our proposed model for link prediction task on eight datasets: USAir, NS, PB, Yeast, C.ele, Power, Router and E.coli as shown in Table 5. We randomly remove 10% existing links from each dataset as positive testing data and sample the same number of nonexistent links as negative testing data. We use the remaining 90% existing links as well as the same number of additionally sampled nonexistent links to construct the training data. The baselines fall into two categories. Node2vec [20] and LINE [42] are advanced methods of network embedding. Besides, variational graph auto-encoder (VGAE) [46] uses a node-level GNN to learn node embeddings to best reconstruct the network.

6.2.2 Results analysis. We report *area under the ROC curve* (AUC) score for each model on the test set in Table 5. Numbers show mean results and standard error for 10 runs. The results show

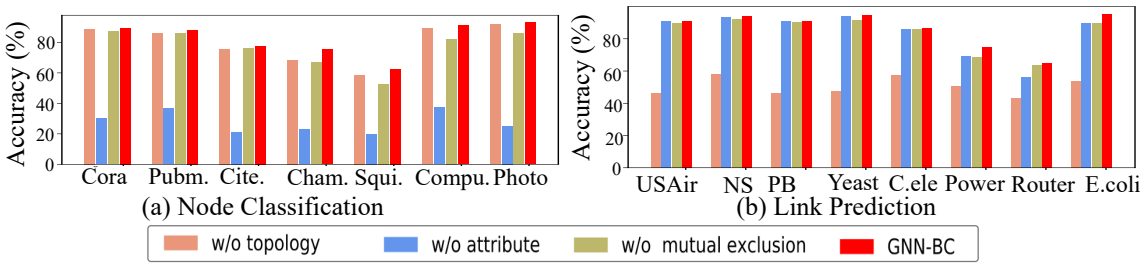


Figure 3: The performances of GNN-BC with different components on node classification and link prediction tasks.

that our model outperforms VGAE on 7 out of 8 datasets, which demonstrates the superiority of concatenating representations from attribute and topology under mutual exclusion constrain. Moreover, our model obtains competitive performance compared with node2vec, which shows our model can avoid topology distortion induced by attribute to some extent.

6.3 Parameter Analysis

To perform robustness analysis and provide practical implementation suggestion, the parameter analysis is investigated in this section. There are two hyper-parameters λ_1 and λ_2 in both objective function Eq. (17) and the equivalent GNN-BC, i.e., Eqs. (22) and (23), which balance the impact of topology, attribute and mutual exclusion constrain. To demonstrate the robustness of GNN-BC, these two hyper-parameters vary from 0 to 4, and the corresponding node classification performances in term of accuracy on four representative networks are shown in Figure 2.

It can be observed that GNN-BC achieves superior performances as hyper-parameters varying in a large range on all the networks. This demonstrates the robustness of GNN-BC on networks with different ratio of homopily. Besides, higher accuracy can be achieved on heterophilic networks with smaller λ_1 ($\lambda_1 \leq 1.5$), since the original attributes play more important roles. Furthermore, better performance can be obtained on homophilic networks with small λ_2 ($\lambda_2 \leq 2$), since it tends to be more redundant to combine both topology and attribute via smoothing effort. This phenomenon meets the design of objective function and GNN-BC and demonstrates the effectiveness of different components in the objective function. Thus, these two hyper-parameters can be tuned in practice if some prior information about the network is given.

6.4 Ablation Study

To provide intuitive understanding to the model's components, this section performs the ablation study on node classification and link prediction tasks. To this end, GNN-BC with either topology and attribute representation and the GNN-BC without the mutual exclusion constraint are compared with, respectively. The results on node classification and link prediction are shown in Figures 3. Figure 3 provides an overall comparison. However, it is not obvious to effect the proposed mutual exclusion constraint, since some components possess very low performance, i.e., the component without attribute in node classification and the the component without topology in link prediction.

It can be observed that GNN-BC achieves the best performance in all the tasks. In node classification task, embedding from node attribute play more critical role, while embedding from topology is more important in link prediction task. Besides, compared with the variant without mutual exclusion term, the mutual exclusion constraint can significantly and consistently improve the performance on almost all the networks, since it effectively reduces the redundancy and information loss to enhance the expressive power.

Note that, without the mutual exclusion, the performances of the combination of topology and attribute may lower than those of baselines with one kinds of information. For example, the performance of combination without mutual exclusion is lower than that of baseline based only on topology in the link prediction task on Yeast. This may be caused by that the redundancy weakens the expressive power of individual topology and attribute encoders. This also demonstrates the importance of mutual exclusion term.

7 CONCLUSIONS

This paper unifies some serious issues in existing graph neural networks, i.e., the over-smoothing issue, node similarity distortion issue and dissatisfactory link prediction performance issue, as the interference between topology and attribute. Then, this interference is analyzed and ascribed to that the learned representation in GNNs essentially compromises between the topology and node attribute. To alleviate the interference, this paper breaks this compromise by proposing a novel objective function, which fits node attribute and topology with different representations and introduces mutual exclusion constraints to reduce the redundancy in both representations. The proposed novel objective function induces a novel GNN, i.e., Graph Neural Network Beyond Compromise (GNN-BC), by iteratively updating the representations of topology and attribute. The performance improvements on node classification and link prediction demonstrate the effectiveness of GNN-BC on relieving the interference between topology and attribute.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China under Grant 2020YFB1406704, the National Natural Science Foundation of China under Grant 61972442, Grant 62102413, Grant U2001202, Grant U1936208, Grant 61876128, in part by the Key Research and Development Project of Hebei Province of China under Grant 20350802D and 20310802D; in part by the Natural Science Foundation of Hebei Province of China under Grant F2020202040, in part by the Natural Science Foundation of Tianjin of China under Grant 20JCYBJC00650, and in part by the China Postdoctoral Science Foundation under Grant 2021M703472.

REFERENCES

[1] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *TNNLS*, 32(1):4–24, 2021.

[2] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[3] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[4] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.*, 30(3):83–98, 2013.

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, pages 3837–3845, 2016.

[6] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[7] Bingbing Xu, Huawei Shen, Qi Cao, Yunqi Qiu, and Xueqi Cheng. Graph wavelet neural network. In *ICLR*, 2019.

[8] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *ICML*, pages 1263–1272, 2017.

[9] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, pages 1024–1034, 2017.

[10] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[11] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[12] Felix Wu, Amauri H. Souza Jr., Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks. In *ICML*, pages 6861–6871, 2019.

[13] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, pages 3538–3545, 2018.

[14] Liang Yang, Zhiyang Chen, Junhua Gu, and Yuanfang Guo. Dual self-paced graph convolutional network: Towards reducing attribute distortions induced by topology. In *IJCAI*, pages 4062–4069, 2019.

[15] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *ICLR*, 2020.

[16] Wei Jin, Tyler Derr, Yiqi Wang, Yao Ma, Zitao Liu, and Jiliang Tang. Node similarity preserving graph convolutional networks. In *WSDM*, pages 148–156, 2021.

[17] Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, and Jian Pei. AM-GCN: adaptive multi-channel graph convolutional networks. In *KDD*, pages 1243–1253, 2020.

[18] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *NeurIPS*, pages 5171–5181, 2018.

[19] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *JMLR*, 9:1981–2014, 2008.

[20] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, pages 855–864, 2016.

[21] Yao Ma, Xiaorui Liu, Tong Zhao, Yozen Liu, Jiliang Tang, and Neil Shah. A unified view on graph neural networks as graph signal denoising, 2020.

[22] Liang Yang, Chuan Wang, Junhua Gu, Xiaochun Cao, and Bingxin Niu. Why do attributes propagate in graph convolutional neural networks? In *AAAI*, pages 4590–4598, 2021.

[23] Meiqi Zhu, Xiao Wang, Chuan Shi, Houye Ji, and Peng Cui. Interpreting and unifying graph neural networks with an optimization framework. In *WWW*, pages 1215–1226, 2021.

[24] Arthur Gretton, Olivier Bousquet, Alexander J. Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *ALT*, pages 63–77, 2005.

[25] Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *SIGKDD*, pages 338–348, 2020.

[26] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *ICLR*, 2019.

[27] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *ICML*, pages 1725–1735, 2020.

[28] Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. In *ICLR*, 2020.

[29] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *ICLR*, 2020.

[30] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *ICML*, pages 21–29, 2019.

[31] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. In *NeurIPS*, pages 13333–13345, 2019.

[32] Sitao Luan, Mingde Zhao, Xiao-Wen Chang, and Doina Precup. Break the ceiling: Stronger multi-scale deep graph convolutional networks. In *NeurIPS*, pages 10943–10953, 2019.

[33] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *ICML*, pages 5449–5458, 2018.

[34] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *ICLR*, 2020.

[35] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *NeurIPS*, 2020.

[36] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *ICLR*, 2021.

[37] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. pages 3950–3957, 2021.

[38] Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks, 2021.

[39] Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. Analyzing the expressive power of graph neural networks in a spectral perspective. In *ICLR*, 2021.

[40] Hande Dong, Jiawei Chen, Fuli Feng, Xiangnan He, Shuxian Bi, Zhaolin Ding, and Peng Cui. On the equivalence of decoupled graph convolution network and label propagation. In *WWW*, pages 3651–3662, 2021.

[41] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In *SIGKDD*, pages 701–710, 2014.

[42] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.

[43] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *WSDM*, pages 459–467, 2018.

[44] Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. Arbitrary-order proximity preserved network embedding. In *SIGKDD*, pages 2778–2786, 2018.

[45] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Chi Wang, Kuansan Wang, and Jie Tang. Netsmf: Large-scale network embedding as sparse matrix factorization. In *WWW*, pages 1509–1520, 2019.

[46] Thomas N. Kipf and Max Welling. Variational graph auto-encoders. *CoRR*, abs/1611.07308, 2016.

[47] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 76:036106, Sep 2007.

[48] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[49] Galileo Namata, Ben London, Lise Getoor, and Bert Huang. Query-driven active surveying for collective classification. In *International Workshop on Mining and Learning with Graphs*, 2012.

[50] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *arXiv:1909.13021*, 2019.

[51] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation, 2019.

[52] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[53] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NIPS*, pages 6338–6347, 2017.

[54] Leonardo Filipe Rodrigues Ribeiro, Pedro H. P. Saverese, and Daniel R. Figueiredo. *struc2vec*: Learning node representations from structural identity. In *SIGKDD*, pages 385–394, 2017.