Diverse Message Passing for Attribute with Heterophily

Liang Yang [1,2], **Mengzhe Li** [1], **Liyang Liu** [1]
Bingxin Niu [1], **Chuan Wang** [2], **Xiaochun Cao** [2], **Yuanfang Guo** [3]
[1]School of Artificial Intelligence, Hebei University of Technology, Tianjin, China
[2]State Key Laboratory of Information Security, IIE, CAS, Beijing, China
[3]State Key Laboratory of Software Development Environment, Beihang University, China
yangliang@vip.qq.com

Abstract

Most GNNs are restricted in the *uniform* message passing framework, where all attributes of one node are considered as a whole and share the uniform propagation weights along one edge, and focus on the uniform weights learning. This is induced by the simplification of homophily and heterophily as node-level property and the ignorance of attributes difference. Actually, different attributes possess diverse characteristics. In this paper network homophily rate defined with node label is extended to attribute homophily rate by taking attribute as weak label. Statistics show attributes possess diverse homophily. To meet this observation, a diverse message passing (DMP) framework, which specifies each attribute propagation weight along each edge, is proposed. Besides, two strategies are proposed to significantly reduce the model complexity of DMP to prevent overfitting issue. By investigating the spectral characteristics, existing spectral GNNs are just equivalent to a degeneration of diverse message passing. Theoretical analysis from numerical optimization perspective reveals that Diverse Message Passing generates multiple groups of graph partition candidates for classifier to determine how to combine them to form the final partition, thus possesses more powerful expressive ability compared to classic uniform message passing. And this powerful expressive ability also indicates the ability to prevent over-smoothing issue. Evaluations on real networks demonstrate the superiority of diverse message passing on preventing over-smoothing and handling network with heterophily.

1 Introduction

Originating from spectral graph theory, Graph Neural Networks (GNNs) show superior performance on modeling ubiquitous irregular data in many fields, such as computer vision, natural language processing and information retrieval, etc [1, 2]. By reducing the computational complexity of graph Fourier transformation via approximation, most GNNs, especially Graph Convolutional Networks (GCN) [3] and its variants including SGC [4] actually perform Laplacian smoothing [5] and low-passing filtering [4]. Although this intuitive strategy achieves state-of-the-art on networks with homophily, such as Cora, CiteSeer and Pubmed, it possesses two fatal drawbacks. First, it only performs well on shallow networks, but leads to serious over-smoothing issue by stacking many layers [5, 6, 7]. Second, it is ineffective to handle network with heterophily, even worse than the simple multiple layer perceptron (MLP) [8].

There are many attempts to overcome above two drawbacks. Their common philosophy behind these methods is weakening the smoothing effect. One kind of methods directly modifies the smoothed node representations with local information. For example, APPNP [9] and GCNII [10] average

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

the smoothed node representation with original node attribute, JKNet [7] and PPNP [9] combine multi-scale information by summation, and MixHop [11], Stronger Multi-scale [12] and H2GCN [8] combine multi-scale information by concatenation. Another kind of methods modifies the network topology, which indirectly induces the over-smoothing. For example, DropEdge [13] and GRAND [14] randomly remove edges and nodes, respectively, while FAGCN [15] and [16] relax the positive edge, which induces smoothing effect, to be real number (both positive and negative). Recent research tries to bridge and unify these two issues [17].

No matter designed for networks with homophily or heterophily, most GNNs can be unified under the message passing framework. They consider all attributes of one node as a whole, and propagate them using the same (positive or negative) propagation weight along one edge. That is, most Graph Neural Networks (GNNs) are restricted in the *uniform* message passing framework and focus on the uniform weights learning as GCN [3], GAT [18] and FAGCN [15]. This is induced by the simplification of homophily and heterophily as node-level property [19]. In two representative methods to handle networks with heterophily, Geom-GCN [20] and H2GCN [8], network homophily rate is defined by just utilizing network topology and node labels.

Actually, different attributes possess diverse characteristics. To investigate this, the concept of network homophily rate defined in [20] is extended to attribute level. Specifically, attribute homophily rate for each attribute is defined by replacing the real node label with each attribute as weak label. Statistics in Figure 1 show that attribute homophily rates vary in a large range, no matter on networks with homophily and heterophily. Thus, GNNs should meet the diversity of attribute homophily rate. To this end, a diverse message passing (DMP) framework, which specifies each attribute propagation weight along each edge, is proposed to adaptively handle diverse attribute homophily. Besides, two strategies are proposed to significantly reduce the model complexity of DMP to prevent overfitting issue. By investigating the spectral characteristics, existing spectral GNNs are just equivalent to a degeneration of diverse message passing. Theoretical analysis from numerical optimization perspective reveals that different form Uniform Message Passing, which directly generates a graph partition, Diverse Message Passing generates multiple groups of graph partition candidates and then the classifier in semi-supervised task determines how to combine them to form the final partition. This shows that Diverse Message Passing possesses more powerful expressive ability, compared to classic uniform message passing. And this powerful expressive ability also indicates the ability to prevent over-smoothing issue.

2 Notations and Preliminary

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 $\mathbf{H} \in \mathbf{R}^{N\times F'}$ denotes the collection of node attribute and representation with the i^{th} row, 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' stands for the ddimensions of attribute and representation.

Most Graph Neural Networks (GNNs) follow an aggregation-combination strategy [21], where each node representation is iteratively updated by aggregating node representations of neighbourhoods and combining the aggregated representation with the node representation itself as follows

$$\bar{\mathbf{h}}_v^k = \mathrm{AGGREGATE}^k\left(\left\{\mathbf{h}_u^{k-1} | u \in \mathcal{N}(v)\right\}\right), \quad \mathbf{h}_v^k = \mathrm{COMBINATE}^k\left(\mathbf{h}_v^{k-1}, \bar{\mathbf{h}}_v^k\right), \tag{1}$$

where $\bar{\mathbf{h}}_v^k$ stands for the aggregated representation from neighbourhoods. Except for the concatenation, such as GraphSAGE [22] and H2GCN [8], average (or summation) is widely adopted as the implementation of $\mathrm{COMBINATE}^k(\cdot,\cdot)$ function, such as GCN [3], GAT [18] and GIN [23] etc. Excepted for the MAX and LSTM implementations in GraphSAGE [22], most GNNs utilize average function as the implementation of $\mathrm{AGGREGATE}^k$. Therefore, most GNNs can be unified under the following formula

$$\mathbf{h}_v^k = \sigma\left(\left(c_{vv}^k \mathbf{h}_v^{k-1} + \sum_{u\in\mathcal{N}(v)} c_{uv}^k \mathbf{h}_u^{k-1}\right)\mathbf{W}^k\right), \tag{2}$$

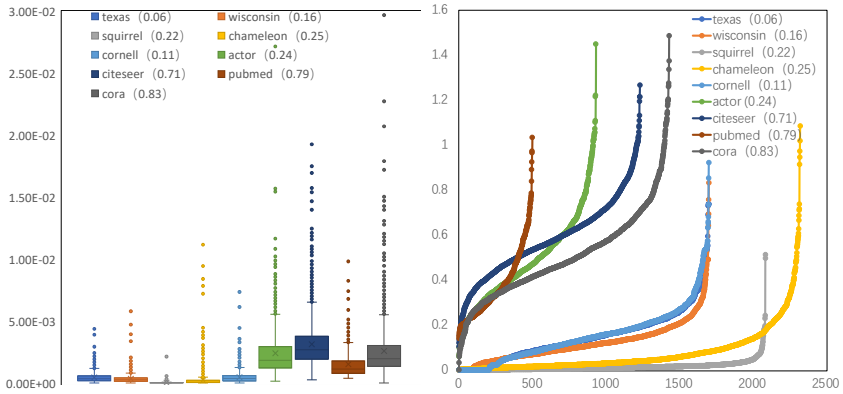

Figure 1: Attribute homophily on networks with different homophily. Each number under network name denotes the network homophily computed as in [20]. (a) Distributions of attribute homophily scores, i.e., $\{\beta_f\}_{f=1}^F$ on different networks. Each column represents one network. (b) Attribute homophily score, i.e., $\{\beta_f\}_{f=1}^F$, on different networks. Each curve denotes one network. To better visualization, attribute homophily scores in each network are sorted in ascending order.

where $\mathbf{W}^k$ is the learnable parameters and the $\sigma(\cdot)$ is the nonlinear mapping function. The scalar c_{uv} is the averaging weight. For example, GCN [3] sets $c_{uv}^k = 1/(\sqrt{(d_u+1)(d_v+1)}$, GIN [23] sets $c_{uv}^k = 1$ for $u \neq v$ and $c_{vv}^k = 1+\epsilon^k$, and GAT [18] learns non-negative c_{uv}^k using attention mechanism. Recently, to handle the network with heterophily via high-passing filtering, GPRGNN [16] sets $c_{uv}^k = \gamma^k/(\sqrt{(d_u+1)(d_v+1)}$ with γ^k as learnable real number while FAGCN [15] directly relaxes the learnable c_{uv} in GAT to real number.

3 Diverse Message Passing

In this section, the motivation of diverse message passing is first provided. Then, the diverse aggregation framework is given followed by two implementations to reduce model complexity.

3.1 Motivations

The unified formula in Eq. (2) can be regarded as the *uniform* message passing, where all the attributes propagated from node v to node u, i.e., all elements of $\mathbf{h}_u^{k-1}$, share the common weight c_{uv}^k. Thus, all the attributes are aggregated with the same weighting scheme in the local region $\mathcal{N}(v) \cup \{v\}$ of graph. Since aggregation can be interpreted from both Laplacian smoothing [5] and low-passing filtering [4] perspectives, the same weighting scheme indicates that all attributes share the similar distribution over graph.

To investigate the distributions of attributes and measure their similarities, the attribute homophily rate is proposed. The network homopily rates, including both the node version [20] and edge version [8], quantify the degree of similar node being connected. The node similarity in network homopily rate is measured according to node label. Here, the attribute homophily rate is defined by measuring the node similarity according to each attribute. Specifically, the attribute homophily rate of attribute f is defined as follows

$$\beta_f = \frac{1}{\sum_{v\in\mathcal{V}} x_{vf}} \sum_{v\in\mathcal{V}} \beta_{vf} = \frac{1}{\sum_{v\in\mathcal{V}} x_{vf}} \sum_{v\in\mathcal{V}} \left(x_{vf} \frac{\sum_{u\in\mathcal{N}(v)} x_{uf}}{d_v} \right), \tag{3}$$

where $\frac{\sum_{u\in\mathcal{N}(v)} x_{uf}}{d_v}$ is the rate of neighbourhoods with attribute f, $\beta_{vf} = x_{vf}\frac{\sum_{u\in\mathcal{N}(v)} x_{uf}}{d_v}$ is the rate of neighbourhoods which share the same attribute f as node v, and the attribute homophily rate of attribute f, i.e., β_f, is the average of β_{vf} over all the nodes possessing attribute f.

Figure 1 shows the distribution of $\{\beta_f\}_{f=1}^F$ on networks with different homophily computed as in [20]. For better visualization, both the macroscopic and microscopic perspectives are provided.

Figure 1(a) shows the distribution of attribute homophily (macroscopic perspective) on each network (each column) in box plot. Figure 1(b) describes the attribute homophily of different attributes, i.e., $\{\beta_f\}_{f=1}^F$ in one curve for one network as microscopic perspective.

It can be observed that networks with higher homophily also possess higher averaged attribute homophily, and vice versa. However, attribute homophily $\{\beta_f\}_{f=1}^F$ distributed in a large range, no matter on networks with high or low attribute homophily. It shows that different attributes possess remarkably different attribute homophily. This indicates that the distributions of attribute over graph are not uniform, but diverse. Therefore, diverse aggregation scheme, i.e. averaging sheme, should be design to fully explore the diverse distributions of attribute over graph for effective node representation learning.

3.2 Diverse Aggregation Framework

To implement the diverse aggregation, the uniform message passing in Eq. (2) is improved to make different attributes possess different propagation weights. To this end, the scaler c_{vu}^k in Eq. (2) is augmented to vector $\mathbf{c}_{vu}^k$, which has the same length as the node representation $\mathbf{h}_u^{k-1}$. Each element in c_{vu}^k is the weight of the corresponding attribute in $\mathbf{h}_u^{k-1}$ propagated from node u to node v. Thus, Eq. (2) is enhanced to

$$\mathbf{h}_v^k = \sigma\left(\left(\mathbf{c}_{vv}^k \odot \mathbf{h}_v^{k-1} + \sum_{u\in\mathcal{N}(v)} \mathbf{c}_{uv}^k \odot \mathbf{h}_u^{k-1}\right)\mathbf{W}^k\right), \tag{4}$$

where $\odot$ denotes the element-wise product of vectors. The remaining issue is how to learn $\mathbf{c}_{uv}^k$'s. If all $\mathbf{c}_{uv}^k$'s are set as free parameters to be directly learned, the model complexity, i.e., the number of parameters, of directly learning $\mathbf{c}_{uv}^k$'s is $\mathcal{O}(|\mathcal{E}|F)$, which may induce overfitting. In the following, two strategies are designed to reduce the number of learnable parameters.

The first strategy is to constrain all the parameters $\mathbf{c}_{uv}^k$'s be determined by a learnable function. Since $\mathbf{c}_{uv}^k$ can be seen as the averaging weight between nodes v and u, thus we adopt the attention scheme,

$$\mathbf{c}_{uv}^k = tanh\left([\mathbf{h}_v^{k-1}||\mathbf{h}_u^{k-1}]\mathbf{W}_c^k\right), \tag{5}$$

where $[\mathbf{h}_v^{k-1}||\mathbf{h}_u^{k-1}]$ represents the concatenation of representations $\mathbf{h}_v^{k-1}$ and $\mathbf{h}_u^{k-1}$, and the learnable mapping function $\mathbf{W}_c^k \in \mathbb{R}^{F\times F}$ is shared across all edges, where F is the dimension of both $\mathbf{h}_v^{k-1}$ and $\mathbf{c}_{uv}^k$. Thus, the model complexity is significantly reduced to $\mathcal{O}(F^2)$. Note that, different from attention in GAT [18], which constrains the learned propagation weights c_{uv}^k's be non-negative via softmax nonlinear function, the learned weight vectors $\mathbf{c}_{uv}^k$'s are relaxed to be real number by adopting tanh nonlinear function $tanh(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)} = sigmoid(2x) - 1$, whose output is zero-centered and ranges in $(-1, 1)$. As the positive weights are equivalent to low-passing filtering, negative weights facilitate the filtering beyond low-frequency.

The second strategy is to simplify Eq. (4) by assuming that all the neighbourhods share the same propagation vector, i.e. $\mathbf{c}_{uv}^k = \bar{\mathbf{c}}_v^k$ for all $u \in \mathcal{N}(v)$. This simplification takes the assumption that the neighbourhoods of fixed-hop equally contribute to the node representation, which has also been adopted in GraphSAGE [22], H2GCN [8] and GIN [23]. Thus, Eq. (4) can be simplified to

$$\mathbf{h}_v^k = \sigma\left(\left(\mathbf{c}_v^k \odot \mathbf{h}_v^{k-1} + \bar{\mathbf{c}}_v^k \odot \sum_{u\in\mathcal{N}(v)} \mathbf{h}_u^{k-1}\right)\mathbf{W}^k\right). \tag{6}$$

It can be seen as the weighted averaging of the node representation $\mathbf{h}_v^{k-1}$ and the neighbourhood representation $\bar{\mathbf{h}}_v^{k-1} = \sum_{u\in\mathcal{N}(v)} \mathbf{h}_u^{k-1}$, where both $\mathbf{c}_v^k$ and $\bar{\mathbf{c}}_v^k$ can be regarded as element-wise averaging weight. Thus, the learning of both $\mathbf{c}_v^k$ and $\bar{\mathbf{c}}_v^k$ can be determined by node and neighbourhood representations. Similar to Eq. (5), the attention mechanism is adopted to learn them as

$$\mathbf{c}_v^k = tanh\left([\mathbf{h}_v^{k-1}||\bar{\mathbf{h}}_v^{k-1}]\mathbf{W}_c^k\right). \tag{7}$$

The learning of $\bar{\mathbf{c}}_v^k$ is similar, but with different parameter $\mathbf{W}_{\bar{c}}^k$. This strategy shares the same model complexity as the first strategy.

3.3 Interpretation from Spectral Perspective

In previous subsection, diverse message passing is motivated and proposed form the spatial perspective of GNN. Actually, GNNs originate from spectral graph theory, and ChebNet [24] and GCN [3] are two representative methods. ChebNet use Chebyshev polynomials of Laplacian matrix to define graph convolution as $\mathbf{H} = \sum_{r=0}^{R} T_r(\tilde{\mathbf{L}})\mathbf{X}\mathbf{W}_r$, where $\tilde{\mathbf{L}}$ is scaled Laplacian matrix and $T_r(x)$ is Chebyshev polynomials. GCN simplifies ChebNet by first constraining it to 1-order model, i.e., $R = 1$, as

$$\mathbf{h}_v^k = \sigma\left(\mathbf{h}_v^{k-1}\mathbf{W}_0^k + \left(\sum_{u\in\mathcal{N}(v)} \mathbf{h}_u^{k-1}\right)\mathbf{W}_1^k\right), \tag{8}$$

and then making two terms share the same mapping function, i.e., $\mathbf{W}_0^k = \mathbf{W}_1^k = \mathbf{W}^k$. It can be observed that if all nodes share and same $\mathbf{c}_v^k$ and $\bar{\mathbf{c}}_v^k$ as $\mathbf{c}^k$ and $\bar{\mathbf{c}}^k$, the second strategy in Eq. (6) degrades to 1-order ChebNet in Eq. (8) with $\mathbf{W}_0^k = (\mathbf{1}\cdot\mathbf{c}^k)\odot\mathbf{W}^k$ and $\mathbf{W}_1^k = (\mathbf{1}\cdot\bar{\mathbf{c}}^k)\odot\mathbf{W}^k$ where $\mathbf{1}$ stands for the vector of ones, which possesses the same length as vectors $\mathbf{c}^k$ and $\bar{\mathbf{c}}^k$. That is, the mapping function $\mathbf{W}$ in both ChebNet and GCN have contained the attribute-wise propagation weights, $\mathbf{c}^k$ and $\bar{\mathbf{c}}^k$, shared by all edges. Therefore, most spectral GNNs fail in uniform message passing framework from the spatial perspective, which is also indicated in [25].

In fact, the attribute-wise weights $\mathbf{c}_{uv}^k$ in Eq. (4) and $\mathbf{c}_v^k$ in Eq. (6) can't be integrated into the mapping parameter $\mathbf{W}^k$, since these weights are different on different edges in Eq. (4) and on different nodes in Eq. (6) to model attribute heterophily. Thus, diverse message passing may break the ceiling of spectral GNNs.

3.4 Theoretical Analysis

Recently, some efforts have been paid to interpreted GNNs from the perspective of numerical optimization [26, 27, 28]. Each graph convolutional layer in GCN [3] can be seen as the gradient descent to minimize the graph Laplacian regularization $tr(\mathbf{H}^T\tilde{\mathbf{L}}\mathbf{H}) = \frac{1}{2}\sum_{u,v}\|\mathbf{h}_u - \mathbf{h}_v\|_2^2$ with node attribute $\mathbf{X}$ as start point, where $tr(\cdot)$ stands for the trace of the matrix. However, most of them only unify and interpret GNNs based on the fixed propagation weight in Eq. (4), such as APPNP [9], JKNet [7] and DAGNN [29].

In this subsection, the theoretical difference between the uniform message passing in Eq. (2) and diverse message passing in Eq. (4) is analyzed. Since the propagation weights on all attributes should be learnable to facilitate the diverse message passing, the previous numerical optimization framework for GNNs with fixed propagation should be relaxed to learnable. Thus, propagation weights c_{uv}^k in Eq. (2) and $\mathbf{c}_{uv}^k$ in Eq. (4) are assumed to be learnable. Here, the analysis focuses on the underlying philosophy of learning propagation weights by edge-wise or channel-wise. Thus, the specific learning function is ignored.

Uniform Message Passing in Eq. (2) with learnable weighs c_{uv}^k can be seen as the combination of two steps: 1) learning propagation weight to form new graph topology, and 2) propagating attribute on the new graph. Thus, the following theorem holds.

Theorem 1. *The Uniform Message Passing in Eq. (2) with learnable weighs c_{uv} is the gradient descent of the following objective function with node attribute $\mathbf{X}$ as the initial of representation $\mathbf{H}$*

$$\min_{\mathbf{C},\mathbf{H}} \sum_{u,v}\left(b_{uv}c_{uv} + \gamma c_{uv}^2\right) + 2tr(\mathbf{H}^T\mathbf{L}_C\mathbf{H}), \tag{9}$$

$$s.t.\ \forall u \sum c_{uv} = 1,\ 0 \le c_{uv} \le 1,\ \mathbf{H} \in \mathbf{R}^{N\times F}, \tag{10}$$

where $b_{uv} = g\left(a_{uv}, dis(\mathbf{x}_i, \mathbf{x}_j)\right)$ denotes the similarity between nodes u and v according to both topology a_{uv} and the distance between attributes $dis(\mathbf{x}_i, \mathbf{x}_j)$. $\mathbf{A} = [a_{uv}]$ is the adjacency matrix of $\mathcal{G}$, $\mathbf{C}$ is the collection of c_{uv}, i.e., the adjacency matrix of the learned graph, and $\mathbf{L}_C$ is the Laplacian matrix of the adjacency matrix $\mathbf{C}$.

The proof is provided in Appendix. The sketch of proof is that minimizing Eq. (9) is perfomed by alternately updating c_{uv} and $\mathbf{H}$, which correspond to the two step of Uniform Message Passing. To

further understand the insight of the objective function of Uniform Message Passing in Eq. (9), two important theorems are provided. They bridge the graph Laplacian regularization, eigenvalue of the Laplacian matrix and the number of connected components in the graph.

Theorem 2. *[Ky Fan's Theorem [30]] It holds*

$$\min_{\mathbf{H}\in\mathbf{R}^{N\times F},\mathbf{H}^T\mathbf{H}=\mathbf{I}} tr(\mathbf{H}^T\mathbf{L}_C\mathbf{H}) = \sum_{f=1}^{F}\sigma_f(\mathbf{L}_C), \tag{11}$$

where $\sigma_f(\mathbf{L}_C)$ denotes the f^{th} smallest eigenvalue of the Laplacian matrix $\mathbf{L}_C$.

Theorem 3. *[[31, 32]] The multiplicity F of the eigenvalue 0 of the Laplacian matrix $\mathbf{L}_C$ is equal to the number of connected components in the graph with the similarity matrix $\mathbf{C}$.*

Theorem 3 shows if $rank(\mathbf{L}_C) = N - F$, then the graph is partitioned into F connected components. Thus, as shown in the following theorem, Uniform Message Passing actually performs graph partition based on the similarity between nodes according to both topology and the attribute distance.

Theorem 4. *The Uniform Message Passing in Eq. (2) actually partitions graph into F connected components based on the similarity $b_{uv} = g\left(a_{uv}, dis(\mathbf{x}_i, \mathbf{x}_j)\right)$ with the following objective function*

$$\min_{\mathbf{C}} \sum_{u,v}\left(b_{uv}c_{uv} + \gamma c_{uv}^2\right) \tag{12}$$

$$s.t.\ \forall u \sum c_{uv} = 1,\ 0 \le c_{uv} \le 1,\ rank(\mathbf{L}_C) = N - F, \tag{13}$$

Diverse Message Passing in Eq. (4) with learnable weighs $\mathbf{c}_{uv}^k$ extends uniform message passing by passing different attributes with different weights. Thus, following the derivation of Theorem 4, we obtain the following theorem for Diverse Message Passing

Theorem 5. *The Diverse Message Passing in Eq. (4) actually partitions graph into 2 connected components (F groups) based on the each similarity $b_{uv}^{(f)} = g\left(a_{uv}, dis(x_{if}, x_{jf})\right)$ with the following objective function*

$$\min_{\mathbf{C}^{(f)}} \sum_{u,v}\left(b_{uv}^{(f)}c_{uv}^{(f)} + \gamma(c_{uv}^{(f)})^2\right) \tag{14}$$

$$s.t.\ \forall u \sum c_{uv}^{(f)} = 1,\ 0 \le c_{uv}^{(f)} \le 1,\ rank(\mathbf{L}_C^{(f)}) = N - 2, \tag{15}$$

Thus, different form Uniform Message Passing, which directly generates a F-components partition, Diverse Message Passing generates F groups 2-components partitions as candidates and then the classifier in semi-supervised task determines how to combine them to form the final F-components partition. This shows that Diverse Message Passing possesses more powerful expressive ability, compared to classic uniform message passing. Besides, this powerful expressive ability also indicates the ability to prevent over-smoothing issue. If the graph is partitioned into F connected components via uniform message propagation, the nodes in same components will converge to the same representation [5]. Thus, the representations of all nodes must be in one of F values, and thus become over-smoothed. In contrast, by generating F group 2-component partitions, different partition will converge to different node representations. Thus, the representation of all nodes may be in one of 2^F values. Thus, the over-smoothing can be prevented.

4 Evaluations

In this section, the proposed Diverse Message Passing (DMP) is evaluated on real data from two perspectives: 1) handling networks with heterophily and 2) preventing over-smoothing issue.

4.1 Datasets

The DMP is validated on 9 networks as shown in Table 1. These 9 networks are from 4 graph datasets. **Citation networks**. Cora, Citeseer, and Pubmed, which are widely used to verify GNNs, are standard citation network benchmark datasets [33, 34]. In these networks, nodes and edge represent papers

Table 1: Datasets statistics

Dataset	Cora	Citeseer	Pubmed	Cham.	Squirrel	Actor	Cornell	Texas	Wisconsin
# Nodes	2,708	3,327	19,717	2,277	5,201	7,600	183	183	251
# Edges	5,429	4,732	44,338	36,101	217,073	33,544	295	309	499
# Features	1,433	3,703	500	2,325	2,089	931	1,703	1,703	1,703
# Classes	7	6	3	5	5	5	5	5	5
Homphily	0.83	0.71	0.79	0.25	0.22	0.24	0.11	0.06	0.16

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. **WebKB webpage networks** . Cornell, Texas, and Wisconsin are the webpage networks from computer science departments of various universities, respectively. In these network, nodes and edges denote web pages and hyperlinks between them. Similar to Citations networks, words in the web page are employed to represent the node feature in bag-of-word form. The web pages are manually classified into the five categories, student, project, course, staff, and faculty. **Co-occurrence network**. Actor network represents the actor co-occurrence in film extracted from the heterogeneous information networks, which describe the complex relation between film, director, actor and writer network [35]. In this network, nodes and edges stand for actors and their co-occurrence in film. The actor's Wikipedia page is used to extract feature and the label of node. **Wikipedia network**. Chameleon and Squirrel are webpages networks in Wikipedia [36]. They are extracted from different topics. Similar to WebKB, nodes and edges denote web pages and hyperlinks between them, and informative nouns in the web page are employed to represent the node feature in bag-of-word form. Webpages are classified in term of the average number of the monthly traffic. In Table 1, the network homphily rates, as computed in [20], are provided. It shows citation networks often possess high homphily rate, while others are often with low homopily, i.e., network with heterophily.

Besides, three heterogenous information networks (HINs), i.e., DBLP, ACM and IMDB, are also employed [37]. Different form homogeneous networks shown in Table 1, HINs possess multiple types of nodes or edges, and multiple meta-path can be utilized. **DBLP** contains 14,328 papers, 4,057 authors,, 20 conferences and 8, 789 terms. The authors are divided into four areas. **ACM** consists of 3,025 papers, 5,835 authors and 56 subjects. Papers are divided into three classes (Database, Wireless Communication, Data Mining). **IMDB** contains 4,780 movies, 5,841 actors and 2,269 directors. The movies are divided into three classes (Action, Comedy, Drama) according to their genre. Node features of them are in bag-of-word form by extracting words from either papers or plots.

4.2 Baselines

To demonstrate the superiority of DMP, 12 state-of-the-art methods are adopted as baselines. Three of them, i.e., GCN [3], GAT [18] and GraphSAGE [22] are the standard graph neural networks, and achieve good performance on network with high homophily. Cheby [24] is a classic spectral GNN. GCN, Cheby and GraphSAGE with JKNet, which combines multi-scale information by summation via residual connections. MixHop [11] combines multi-scale information by concatenation, and H2GCN [8] shows concatenation is the key to deal with network with heterophily. Geom-GCN [20] attempts to leverage network embedding for GNNs on network with homophily and heterophily.

4.3 Implementation Details

For all datasets, nodes in each class are randomly split into three groups, 48% for training, 32% for validation, and 20% for testing as in [8]. The performances of all models are the means on the test sets over 10 random splits. The hyper-parameters, including weight decay, dropout, initial learning rate and patience for learning rate decay, are tuned by searching on validation set. Adam [38] is adopted as the optimizer for all models. As GCN and GAT, standard DMP adopts two layer model.

There are many different implementations of our proposed DMP. The two strategies introduced in Section 3.2 are named as DMP-1 (Eq. (4) with weight learning as Eq. (5)) and DMP-2 (Eq. (6) with weight learning as Eq. (7)), respectively. DMP-Deg stands for the degraded DMP introduced in Eq. (8) of Section 3.3. Besides, two implementations consider how to combine the information form first-order and second-order topology information, i.e., Concatenation as H2GCN [8] and GraphSage [22] (denoted as "-Con") and Summation as GCN [3] and GAT [18] (denoted as "-Sum"). Finally,

Table 2: Mean Classification Accuracy (Bold indicates the best, italics indicates the second best).

Methods	Texas	Wisconsin	Actor	Squirrel	Cham.	Cornell	Citeseer	Pubmed	Cora
GraphSAGE	82.43	81.18	34.23	41.61	58.73	75.95	76.04	88.45	86.90
GCN	64.86	56.86	31.12	32.28	53.51	54.05	75.53	84.71	85.51
GAT	58.38	55.29	26.28	30.62	54.69	58.92	75.46	84.68	82.68
SAGE+JK	83.78	81.96	34.28	40.85	58.11	75.68	76.05	88.34	85.96
Cheby+JK	78.38	82.55	35.14	*45.03*	**63.79**	74.59	74.98	89.07	85.49
GCN+JK	66.49	74.31	34.18	40.45	*63.42*	64.59	74.51	88.41	85.79
GCN-Cheby	77.30	79.41	34.11	43.86	55.24	74.32	75.82	88.72	86.76
MixHop	77.84	75.88	32.22	43.80	60.50	73.51	76.26	85.31	**87.61**
GEOM-GCN	67.57	64.12	31.63	38.14	60.90	60.81	**77.99**	**90.05**	85.27
H2GCN	84.86	*86.67*	**35.86**	36.42	57.11	82.16	*77.04*	*89.40*	*86.92*
DMP-Deg	78.38	80.39	33.09	32.46	54.38	*83.78*	76.87	88.10	86.31
DMP-2-Sum	78.37	84.31	34.93	32.18	55.92	*83.78*	76.27	88.15	85.31
DMP-2-Con	83.78	84.31	34.67	44.28	60.53	*83.78*	75.97	85.31	85.31
DMP-1-Posi	*86.48*	84.31	*35.72*	34.96	51.53	70.27	75.67	88.10	86.11
DMP-1-Sum	*86.48*	86.27	34.21	43.42	50.21	70.27	76.13	88.13	82.28
DMP-1-Con	**89.19**	**92.16**	35.06	**47.26**	62.28	**89.19**	76.43	89.27	86.52

to show the effect of negative propagation weights in above DMP implementations, positive weight learning by replacing $tanh(\cdot)$ in Eq. (5) as $softmax(\cdot)$ is represented as "-Posi".

4.4 Supervised Node Classification

The results of supervised node classification are shown in Table 2. Overall, DMP achieves the comparable performance on networks with high homophily, such as Cora, CiteSeer and Pubmed, compared to all the state-of-the-art baselines, while it significantly outperforms others, especially baselines designed for network with heterophily H2GCN and Geom-GCN, on networks with heterophily, such as Texas, Wisconsin and Squirrel. Specifically, compared to H2GCN, which is the SOTA on network with heterophily, the improvements on Texas, Wisconsin and Squirrel are 5.10%, 6.33% and 29.76%, respectively. This demonstrates the superiority of DMP in handling network with heterophily.

Expressive Ability. 1) As shown in Section 3.2, DMP-2 is the simplification of DMP-1 and DMP-Deg is the degradation of DMP-1. The results in Table 2 verify this characteristic. The performances of DMP-2 are higher than that DMP-1, and both of them are much higher than performance of DMP-Deg. This phenomenon is more remarkable on network with heterophily, since these networks require GNNs with powerful expressive ability to explore informative attributes. 2) GraphSAGE, MixHop and H2GCN show the concatenation possesses high expressive ability than summation used in GCN and GAT. This strategy is also employed in the implementation of DMP. In Table 2, the accuracies of "-Con"'s are higher than those of "-Sum"'s. 3) The superiority of allowing negative propagation weights is also validated by the improvements of DMP-1-Con over DMP-1-Posi, especially on networks with heterophily. Thus, the final DMP implementation, i.e., DMP-1-Con, combines above three tricks.

4.5 Semi-supervised Classification on Heterogenous Information Networks

Since only one type of nodes are labelled in HINs, the effectiveness of DMP is verified on the nodes with given labels. For semi-supervised node task, 20% of the nodes in each class are used for training. To demonstrate the universality, DMP ignores the node types and meta-paths used in HAN [37] and treats them as homogenous networks. Two implementation of DMP, i.e., DMP-1-Con and DMP-1-Sum are compared with GAT [18] and HAN [37]. The classification accuracies in terms of Macro-F1 Score are reported in Table 3. HAN outperforms GAT by manually selecting the semantic meta-paths and combining the results from multiple meta-paths. Note that although the proposed DMP ignores

Table 3: Accuracy on HINs in terms of Macro-F1 Score.

Methods	DBLP	ACM	IMDB
GAT	90.97	86.23	49.44
HAN	**92.83**	90.96	56.77
DPM-1-Sum	84.56	92.62	**58.58**
DPM-1-Con	92.49	**92.64**	57.73

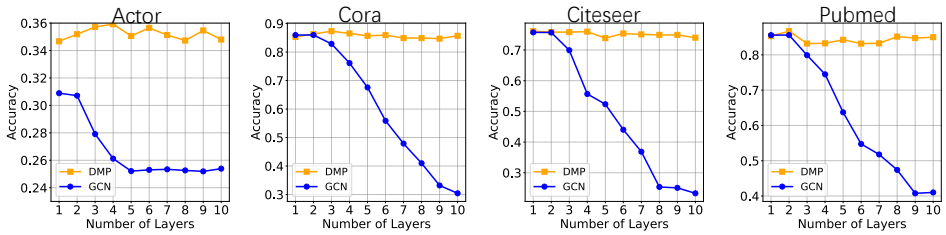

Figure 2: Classification accuracy results with various depths.

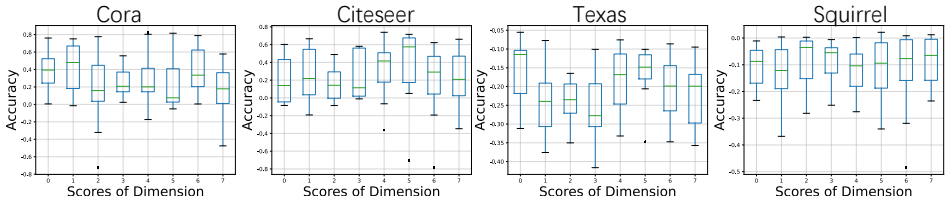

Figure 3: Distributions of learned weights of sampled attribute dimensions.

the semantic meta-path, it achieves comparable or better performance compared to HAN. The main reason of this surprising results is that the by adopting diverse message passing, the semantic information can be explored and learning instead of manually selecting. Specifically, since the HINs consist of multiple nodes or edges, the mutual impacts between different kinds of nodes must be different. This difference not only can be manually utilized by selecting semantic meta-paths as in HAN, but also can be automatically employed in DMP, which allows diverse attribute propagations. This demonstrates the powerful expressive ability of DMP.

4.6 Preventing Over-smoothing Issue

In Section 3.4, the expressive ability and potential on preventing over-smoothing issue are theoretically analyzed. In this section, the potential on preventing over-smoothing issue is experimentally verified on network with homophily. The performance comparisons between DMP and GCN with various depths are shown in Figure 2. It can be observed that as the number of layers increases, the performances of GCN significantly degrade, while those of DMP remain stable. This meets the above theory analysis of DMP on preventing over-smoothing issue.

4.7 Visualization of Learned Weights

To provide an intuitive understanding, this section provides the visualization of the distributions of learned weights of each attribute dimension. Due to all networks possess high attribute dimension, only 8 attribute dimensions are sampled for each network. For each attribute dimension, different edges obtain different propagation weights via learning scheme. Figure 3 visualizes the distributions of learned weights of sampled attribute dimensions. Most learned weights in networks with homophily, such as Cora and Citeseer, are positive, while those in networks with heterophily, such as Texas and Squirrel, are positive. This also meets the definitions of networks with homophily and heterophily.

5 Conclusions

This paper investigates the attribute homophily rate and its impact on the design of GNNs. Most existing GNNs perform *uniform* message passing by ignoring this factor. However, statistics on networks show attributes possess diverse homophily. To meet this observation, a diverse message passing (DMP) framework, which specifies each attribute propagation weight along each edge, is proposed followed by two strategies to reduce model complexity. Theoretical analysis not only shows diverse message passing may break the expressive ability ceiling of spectral GNNs but also demonstrates diverse message passing generate more graph partition candidates for classifier to determines how to combine them to form the final partition, thus possesses more powerful expressive ability from the perspective of numerical optimization. And this powerful expressive ability also indicates the ability to prevent over-smoothing. Evaluations on real networks validates the superiority of diverse message passing on preventing over-smoothing and handling network with heterophily.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 61972442, Grant 61802391, Grant U2001202, Grant U1936208 and Grant 61802282, 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 Hebei Province Innovation Capacity Enhancement Project under Grant 199676146H, in part by the Natural Science Foundation of Tianjin of China under Grant 20JCYBJC00650, in part by the Key Program of the Chinese Academy of Sciences under Grant QYZDB-SSW-JSC003, and in part by State Key Laboratory of Software Development Environment under Grant SKLSDE-2020ZX-18.

References

[1] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, 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] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[4] Felix Wu, Amauri H. Souza Jr., Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 6861–6871, 2019.

[5] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3538–3545, 2018.

[6] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[7] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 5449–5458, 2018.

[8] 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 *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[9] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[10] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pages 1725–1735, 2020.

[11] 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 *Proceedings of the 36th International*

Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, pages 21–29, 2019.

[12] Sitao Luan, Mingde Zhao, Xiao-Wen Chang, and Doina Precup. Break the ceiling: Stronger multi-scale deep graph convolutional networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 10943–10953, 2019.

[13] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[14] Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. Graph random neural networks for semi-supervised learning on graphs. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[15] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. 2021.

[16] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*, 2021.

[17] 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.

[18] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[19] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[20] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[21] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1263–1272, 2017.

[22] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034, 2017.

[23] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[24] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3837–3845, 2016.

[25] 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 *International Conference on Learning Representations*, 2021.

[26] 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.

[27] Meiqi Zhu, Xiao Wang, Chuan Shi, Houye Ji, and Peng Cui. Interpreting and unifying graph neural networks with an optimization framework, 2021.

[28] Liang Yang, C. Wang, J. Gu, Xiaochun Cao, and Bingxin Niu. Why do attributes propagate in graph convolutional neural networks? In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, (AAAI-21), February 2-9, 2021*, 2021.

[29] Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 338–348, 2020.

[30] Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

[31] Bojan Mohar, Y Alavi, G Chartrand, and OR Oellermann. The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2(871-898):12, 1991.

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

[33] 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.

[34] 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.

[35] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816. ACM, 2009.

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

[37] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. Heterogeneous graph attention network. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 2022–2032, 2019.

[38] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

A Appendix

The appendix provides the description of the Figure and the proofs of the theorems.

A.1 Description of Figure 1

As we show in Figure 1: Different from the previous methods of measuring network homophily through edges or nodes, we propose a formula Eq.(3) to measure homophily by attributes in this paper. The data generation process is as follows: multiply the adjacency matrix $\mathrm{A} \in \{0,1\}^{n\times n}$ and the node feature matrix $\mathrm{X} \in \mathbb{R}^{n\times F}$ generate a new matrix $\mathbb{R}^{n\times F}$, which indicates the number of different attribute that the neighbors $u \in N(v)$ of node $v \in V$, the sum of each column represents the number of $f \in F$ in the network. Compared with the number of nodes in the network that contain f, it is the homophily of the attribute in the network. After visualizing the data, we get Figure 1. In graph (a),we find that the attribute distribution in the network with high homophily is higher than with low homophily. In graph (b), the similarity of network attributes with low homophily increases slowly in early stage. On the contrary, high homophily network rises faster. So, we come to a conclusion the network homophily computed as in [20] is proportional to the different attributes homophily in one network. Therefore, diverse message passing for attribute is necessary.

A.2 Proof of Theorem 1

There are two groups of variables in Eq. (9), i.e., node representations $\mathbf{H}$ and learned topology $\mathbf{C}$. Thus, Eq. (9) can be minimized with respective to one group of variables by fixing another group of variables. When graph topology $\mathbf{C}$ is fixed, minimizing Eq. (9) is equivalent to minimizing the following objective w.r.t. node representations $\mathbf{H}$

$$\min_{\mathbf{H}} tr(\mathbf{H}^T\mathbf{L}_C\mathbf{H}). \tag{16}$$

According to [26, 27, 28], the gradient descent methods to minimize Eq. 16 is the same as the graph convolutional operation of GCN, i.e., Eq. (2) with fixed c_{uv} by ignoring mapping function $\mathbf{W}$ and nonlinear function $\sigma(\cdot)$.

When node representations $\mathbf{H}$ are fixed, minimizing Eq. (9) is equivalent to minimizing the following objective w.r.t. node representations $\mathbf{C}$

$$\min_{\mathbf{C},\mathbf{H}} \sum_{u,v} \left(b_{uv}c_{uv} + \gamma c_{uv}^2\right) + 2tr(\mathbf{H}^T\mathbf{L}_C\mathbf{H}), \tag{17}$$

$$s.t.\ \forall u \sum c_{uv} = 1,\ 0 \le c_{uv} \le 1. \tag{18}$$

Eq. (17) can be rewritten as

$$\min_{\mathbf{C}} \sum_{u,v} \left(b_{uv}c_{uv} + \gamma c_{uv}^2\right) + \sum_{u,v} c_{uv}||\mathbf{h}_u - \mathbf{h}_v||^2. \tag{19}$$

If we let

$$o_{uv} = b_{uv} + ||\mathbf{h}_u - \mathbf{h}_v||, \tag{20}$$

Eq. (17) is equivalent to

$$\min_{\mathbf{c}_u^T\mathbf{1}=1,\ 0\le\mathbf{c}_u\le 1} \left\| \mathbf{c}_u + \frac{1}{2\gamma}\mathbf{o}_u \right\|^2, \tag{21}$$

where $\mathbf{c}_u$ and $\mathbf{o}_u$ are the vectors containing c_{uv} and o_{uv}, respectively. Eq. (21) can be minimized by applying Lagrangian multipliers methods and KKT condition. Thus, the solution to Eq. (17) is

$$c_{uv} = \left(-\frac{o_{uv}}{2\gamma} + \eta\right)_+ = ReLU\left(-\frac{o_{uv}}{2\gamma} + \eta\right) \tag{22}$$

where η is one of the Lagrangian multipliers. The tuning of η makes $\mathbf{c}_u^T\mathbf{1} = 1$. If b_{uv} in Eq. (9) is set as the combination of topology and node attributes as

$$b_{uv} = -\zeta a_{uv}\left(\mathbf{w}^T[\mathbf{x}_u||\mathbf{x}_v]\right), \tag{23}$$

where $[\mathbf{x}_u||\mathbf{x}_v]$ denotes the concatenation of $\mathbf{x}_u$ and $\mathbf{x}_v$, and $\mathbf{w}$ is the learnable parameters, which share the same length as $[\mathbf{x}_u||\mathbf{x}_v]$. a_{uv} stands for the corresponding element in the adjacency matrix. ζ denotes the importance of this term. Then Eq. (22) can be written as

$$c_{uv} = ReLU\left(\frac{\zeta}{2\gamma}a_{uv}\left(\mathbf{w}^T[\mathbf{x}_u||\mathbf{x}_v]\right) - \frac{||\mathbf{h}_u - \mathbf{h}_v||}{2\gamma}\right). \tag{24}$$

If the importance parameter ζ is large, Eq. (24) can be simplified as

$$c_{uv} = ReLU\left(\frac{\zeta}{2\gamma}a_{uv}\left(\mathbf{w}^T[\mathbf{x}_u||\mathbf{x}_v]\right)\right). \tag{25}$$

It can be observed that $c_{uv} \neq 0$ only if $a_{uv} \neq 0$, i.e., nodes v and u are connected. Thus, the learned topology in Eq. (25) is similar to that in GAT [18].

Therefore, the Uniform Message Passing in Eq. (2) with learnable weighs is the gradient descent of the following objective function in Eq. (9).

A.3 Proof of Theorem 4

Note that the rank of a matrix $\mathbf{L}_C \in \mathbf{R}^{N\times N}$ is the difference between N and its multiplicity of the eigenvalue 0, i.e., $rank(\mathbf{L}_C) = N - F$. As shown in Theorem 3, the multiplicity F of the eigenvalue 0 of the Laplacian matrix $\mathbf{L}_C$ is equal to the number of connected components in the graph with the similarity matrix $\mathbf{C}$. Thus, the constraint $rank(\mathbf{L}_C) = N - F$ is equivalent to partitioning graph into F connected components. Besides, to constrain the multiplicity of the eigenvalue 0 is F is equivalent to minimizing the fist F smallest eigenvalues, i.e, $\sum_{f=1}^{F}\sigma_f(\mathbf{L}_C)$. According to Theorem 2, $\sum_{f=1}^{F}\sigma_f(\mathbf{L}_C)$ is the minima of $tr(\mathbf{H}^T\mathbf{L}_C\mathbf{H})$. Therefore, Eq. (9) in Theorem 1 is equivalent to Eq. (12) in Theorem 4, and the Uniform Message Passing in Eq. (2) actually partitions graph into F connected components by learning topology $\mathbf{C}$.

A.4 Proof of Theorem 5

Proof. Most of the proof is similar to the proof to Theorem 4. The remaining part is to prove each learned graph actually partition graph into two connected components, i.e., the constraint $rank(\mathbf{L}_C^{(f)}) = N - 2$.

According to Theorem 3, to partition graph into two connected components, multiplicity of the eigenvalue 0 should be 2, and thus $\sigma_1(\mathbf{L}_C) + \sigma_2(\mathbf{L}_C)$ should be minimized. In fact, $\sigma_1(\mathbf{L}_C) = 0$ for all Laplacian matrix $\mathbf{L}_C$. Thus, to partition graph into two connected components, $\sigma_2(\mathbf{L}_C)$ should be minimized. According to [32], it holds that

$$\sigma_2(\mathbf{L}_C) = \inf_{\mathbf{h}\perp\mathbf{1}}\frac{\mathbf{h}^T\mathbf{L}_C\mathbf{h}}{\mathbf{h}^T\mathbf{h}}. \tag{26}$$

Here, we prove that the following approximation

$$\inf_{\mathbf{h}\neq\mathbf{1},||\mathbf{h}||_2=1}\mathbf{h}^T\mathbf{L}_C\mathbf{h}, \tag{27}$$

is bounded by $\sigma_2(\mathbf{L}_C)$. To this end, each $\mathbf{h} \neq \mathbf{1}, ||\mathbf{h}||_2 = 1$ can be decomposed into two mutually perpendicular vectors, i.e.

$$\mathbf{h} = \alpha\mathbf{1} + \mathbf{g} \tag{28}$$

where $\mathbf{g}^T\mathbf{1} = 0$. That is $\alpha = \frac{\mathbf{h}^T\mathbf{1}}{N}$ is the average of all elements in $\mathbf{h}$. Besides, the norm of $\mathbf{g}$ is bounded, i.e., $t_1 \leq ||\mathbf{g}||^2 \leq t_2$. Thus, we obtain

$$\begin{aligned}\mathbf{h}^T\mathbf{L}_C\mathbf{h} &= (\alpha\mathbf{1}+\mathbf{g})^T\mathbf{L}_C(\alpha\mathbf{1}+\mathbf{g}) &(29)\\ &= \alpha^2\mathbf{1}^T\mathbf{L}_C\mathbf{1} + \mathbf{g}^T\mathbf{L}_C\mathbf{g} + 2\alpha\mathbf{1}^T\mathbf{L}_C\mathbf{g} &(30)\\ &= \mathbf{g}^T\mathbf{L}_C\mathbf{g} &(31)\end{aligned}$$

and

$$t_1\sigma_2(\mathbf{L}_C) = t_1\inf_{\mathbf{g}\perp\mathbf{1}}\frac{\mathbf{g}^T\mathbf{L}_C\mathbf{h}}{\mathbf{g}^T\mathbf{g}} \leq \inf\mathbf{h}^T\mathbf{L}_C\mathbf{h} \leq t_2\inf_{\mathbf{g}\perp\mathbf{1}}\frac{\mathbf{g}^T\mathbf{L}_C\mathbf{h}}{\mathbf{g}^T\mathbf{g}} = t_2\sigma_2(\mathbf{L}_C). \tag{32}$$

Thus, $\inf_{\mathbf{h}\neq\mathbf{1},||\mathbf{h}||_2=1} \mathbf{h}^T\mathbf{L}_C\mathbf{h}$ is bounded by $\sigma_2(\mathbf{L}_C)$. Therefore, if the $\mathbf{H} = \mathbf{h} \in \mathbf{R}^N$ in Eq. (9) is a vector and all elements are not equal, then minimizing $tr(\mathbf{H}^T\mathbf{L}_C\mathbf{H})$ is actually minimizing $\sigma_2(\mathbf{L}_C)$ according to Eq. (32). Thus, it is equivalent to the constraint $rank(\mathbf{L}_C) = N - 2$.

Since the diverse message passing in Eq. (4) is equivalent to learning different graph $\mathbf{C}^{(f)}$ for different attributes f, thus each learned graph $\mathbf{C}^{(f)}$ is to partition graph into two components, i.e., the constraint $rank(\mathbf{L}_C^{(f)}) = N - 2$. □